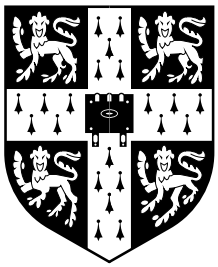
Temporal network metrics and their application to real world networks

John Kit Tang



Robinson College
University of Cambridge

2011

This dissertation is submitted for
the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

This dissertation does not exceed the regulation length of 60 000 words, including tables and footnotes.

Summary

The analysis of real social, biological and technological networks has attracted a lot of attention as technological advances have given us a wealth of empirical data. Classic studies looked at analysing static or aggregated networks, i.e., networks that do not change over time or built as the results of aggregation of information over a certain period of time. Given the soaring collections of measurements related to very large, real network traces, researchers are quickly starting to realise that connections are inherently varying over time and exhibit more dimensionality than static analysis can capture. This motivates the work in this dissertation: new tools for temporal complex network analysis are required when analysing real networks that inherently *change over time*.

Firstly, we introduce the temporal graph model and formalise the notion of *shortest temporal paths*, used extensively in graph theory, and show that as static graphs ignore the time order of contacts, the available links are overestimated and the true shortest paths are underestimated. In addition, contrary to intuition, we find that slowly evolving graphs can be efficient for information dissemination due to *small-world behaviour in temporal graphs*. Secondly, we then turn our attention to the identification of important or *central* nodes in a network. Since two key measures for node centrality, namely closeness and betweenness, are based on shortest paths in a static graph, we define *temporal centrality* based on temporal shortest paths. We demonstrate that the ranking achieved by temporal centrality is superior to static analysis by demonstrating how temporal centrality can be exploited to improve mobile malware containment. Thirdly, we study the *predictability of centrality ranking* in temporal networks utilising correlogram plots between top-k node rankings. We show that in real human contact networks, temporal centrality can be predicted and demonstrate that these predictions are useful for mobile malware containment, compared to static centrality prediction. Finally, we investigate the concepts of *temporally connected components* and show that temporal analysis gives us a precise understanding of the diffusion properties of real contact networks that is missed by static analysis. The conclusions of this thesis are that the use of time aware metrics for the analysis of real networks opens the doors to more precise and effective exploitation of complex network science: while we have given a number of application examples, the future directions of this research are still many.

Acknowledgments

Firstly, I am indebted to my supervisor and teacher, Cecilia Mascolo, for her continual support, guidance and nurture to become an independent researcher; looking back through the years of electronic correspondence I now realise how much I have learnt from her and the multitude of collaborations that she has opened up for me. Secondly, I thank Mirco Musolesi who played a huge part in mentoring me through the precise process of conducting research and scientific writing. Thirdly, I thank Vito Latora and Murtaza Zafer who both taught me incredible lessons which shaped my own research philosophy; Hyoungshick Kim, Vincenzo Nicosia and Salvatore Scellato for invigorating collaboration; and Jon Crowcroft and Ross Anderson for their wisdom on guiding my thesis. Also, I thank the friends I have made in the lab, especially Salvo, Kiran, Liam, Bence, Ilias, Christos and Tassos for making my life at Cambridge more enjoyable.

I thank the EPSRC and Lise Gough for providing me the financial support which enabled me to complete this PhD; Robinson College and the Cambridge Philosophical Society for generous travel grants; Piete Brooks for rearing the Condor Compute Grid; IBM Research for the stimulating work experience; and Philip Treleaven for giving me a taste for research as an undergraduate and for the opportunities he has opened up for me.

I thank my parents and grandparents for their sacrifices in supporting me through my education and their wisdom which has shaped the person I am today. I thank my brother, Eric, and sister, Katie, whose successes have inspired me. I thank my friends (Lovejoy, Oscar, Åsbjørn, Jenny, Janet, Mohsen, Attard and many others) for keeping me sane; and I thank Carole, Charlotte, Sylvia and Graham for their loving support and Sunday roasts.

Finally, this dissertation is dedicated to my loving Wife, Annabel; without her endless love, patience and motivation I would never have been able to complete this journey: *Amor Vincit Omnia*.

In loving memory of Grandma.

Contents

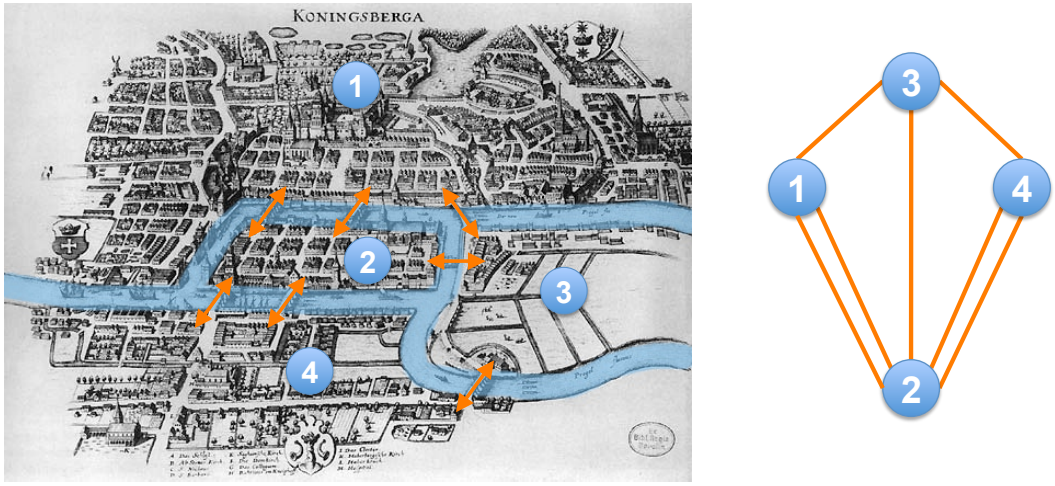
1

Introduction

Networks are all around us: from the cities and roads that we live in to the physical telecommunication cables that connect our computers forming the Internet, and from the intricate layout of neurons and synapses that drive our brains to the relationships between friends; the term "networks", whether road, computer, online social or otherwise, has now become common in our everyday vocabulary. Though the term has been integrated into our culture, the analysis of such a topological abstraction is in itself still a growing science where everyday networks can be modelled as a set of nodes (e.g., cities, computers, neurons or people) which are connected by edges (e.g., roads, cables, synapses or relationships) and the analysis of the non-trivial features of such networks has opened a branch of study known as *complex network analysis* [AB02]. At its roots, complex network analysis is founded on *graph theory* [BLM+06] and hence networks are commonly referred to as *graphs*.

Indeed the publication regarded as the beginnings of graph theory was that of Leonhard Euler's study on the seven bridges of Königsberg, published in 1736, which posed the question of whether a walk existed through the city of Königsbergs, which

is divided into four landmasses by its river, that would cross its seven bridges once and only once. By mapping the city into a topological graph representation with landmasses as nodes and bridges as edges (depicted visually in Figure 1.1), Euler was able to reason on this graph and prove that there was in fact *no* solution.



(a) Map (b) Topological

Figure 1.1: Example of topological mapping of the seven bridges of Königsbergs to a graph. Königsbergs was split by its river into four land masses. The graph in panel (b) is visually depicted as nodes (circles) and edges (lines connecting circles).

Clearly, even this first study was motivated by empirically observed data, albeit on a small scale, but this is still relevant to the study of modern day networks; to fully understand such systems we need to collect data on the actual networks themselves. However, partly due to the lack of technology to collect large scale network data, until the last decade it was believed that such networks possessed simple and trivial structure and hence either small scale (i.e. $<$ 100 node) networks were studied which did not represent a representative sample of the network or *random* networks were generated for such analysis [Bol01]. However, advancements in technology in terms of measurement and computerisation of systems such as transport, power and online social networks, has presented us with a wealth of empirically collected data on real networks. Consequently, seminal works have uncovered non-random features of these networks such as *small world* behaviour where long distance shortcuts can

help reduce the average number of exchanges required to deliver a message even when there exists locally dense clusters of nodes [WS98]; and possible explanations of how such structure forms using *preferential attachment* in scale free networks. Indeed, it is the availability of real, empirically observed datasets that have driven the research over the last decade and motivates this thesis by extending empirical observations to include *time information* of such complex networks. When attempting to apply existing complex network analysis techniques to the growing number of empirically collected network data with rich temporal information, my personal experiences found that existing tools could not capture the full dynamism of networks data which inherently changed over time.

Subsequently, the **subject of this thesis** is the development of temporal metrics and their effectiveness in analysing information dissemination processes in real time-varying networks compared to static analysis.

This statement is purposely general since we want to emphasise the validity of the contributions of this thesis. Before we can proceed, firstly, the term "time-varying" needs to be understood by exploring the types of time information available in real networks (this shall be covered next, in Section 1.1). Secondly, "information dissemination" can refer to the textbook analysis of paths and shortest path lengths in graph theory or to the more practical opportunistic message passing in technological networks. This thesis presents metrics for such a spectrum of analysis. With this in mind, we now explore the range of real networks, which exhibit such temporal information, so that we can understand the fundamental types of time information which motivate this thesis.

1.1 Real Networks Change Over Time

Many excellent surveys [BLM+06, AB02] and books [New10, EK10] exist on the study of networks and cover the range of empirical networks that have been employed in past studies. Expanding on these discussions of real networks, we focus on highlighting the temporal information available in these well-studied empirical networks and also introduce a range of networks collected more recently, which inherently possess temporal information. This taxonomy is by no means exhaustive but serves to demonstrate the types of inherent temporal information available in

these datasets, the range of empirical data collection techniques and the applicability of the techniques presented in this dissertation to a wide range of disciplines. The available and collectable temporal network data is summarised in Table 1.1.

1.1.1 Social Networks

As human beings, we are not only fascinated in understanding how our own bodies and minds work but also the collective behaviours of relationships and interactions between people. This fascination has inspired sociologists and social psychologists to conduct seminal studies to understand the extents of which individuals will conform to preconceived roles during a mock prison scenario [HBZ73]; to demonstrate that people can be influenced to take orders from authority figures [Mil63]; how people can conform to social influence from their peers [AG51]; and to understand the actual number of contacts which separate any two individuals in real social networks [Mil67].

In particular, the latter study incubated the idea that the network between acquaintances possessed properties which allowed a median of six exchanges between friends and friends-of-friends to deliver a letter to a distant acquaintance and popularised the term "six-degrees of separation". Even though this study was performed in the late sixties it has inspired more recent research into the inhomogeneous structure of real social networks through the use of empirically observed social networks; we shall discuss one such study further in Section 2.2 but for now we maintain our focus on real networks which enabled these studies.

1.1.1.1 Social Relationship Networks

We start with some network datasets that will be recognised by many readers familiar in complex network and social network analysis. Many early studies have employed social network data extracted from online websites, for example, the study of small-world networks [WS98] utilised a network of film actors connected by film appearances and which was generated from the Internet Movie Database website[1] and the study of community structure [DA05, NG04] employed the network of research publication co-authorship [New01b] which was constructed from a number of

[1] `http://us.imdb.com`

databases including MEDLINE (a biomedical research database), the Los Alamos physics e-print archive website. In addition to websites, national surveys have been a useful source of social network data, for example, in studying the spreading of sexually transmitted diseases upon a network of sexual encounters [GLMP08, LEA+01].

Temporal information: All these network datasets, whether actors, researchers or sexual partners, all possess the same temporal information which, although was not collected as part of the original dataset, is available from the original source. More specifically, this temporal information is the *timestamp* of links, for example, the date that a set of actors performed in the same movie, the date of publication of co-authors and the time of a sexual encounter. These timestamps might seem trivial at first, but the timestamp of these collaborations determines the *order* of relationships between subsequent nodes over time. From a practical view, this is important if we wish to trace the passage of a sexually transmitted disease through the sexual contact network; including time order in the study of networks shall be studied further in Section 3.4.2.

1.1.1.2 Online Social Networks

The popularity of websites that allow us to keep in touch with friends has exploded over the last decade. The convenience of maintaining friendship networks online, sending messages to friends, arranging events, uploading photos and sharing locations and thoughts has produced household brands such as Facebook[2] and Twitter[3]. Such are their popularity that they have shaped our common vernacular with words being added to the Oxford English Dictionary such as *social graph* ("noun: a representation of the interconnection of relationships in an online social network"), *retweet* ("verb: (on the social networking service Twitter) repost or forward (a message posted by another user)"), *unfollow* ("verb: stop tracking (a person, group, or organisation) on a social networking site"), *cyberbullying* ("noun: the use of electronic communication to bully a person, typically by sending messages of an intimidating or threatening nature"); no doubt inspired by the usage of these online social networks (OSN). Clearly, since these services are online the mass of data needs to be stored and maintained by the service providers and this data becomes

[2] `http://www.facebook.com`
[3] `http://www.twitter.com`

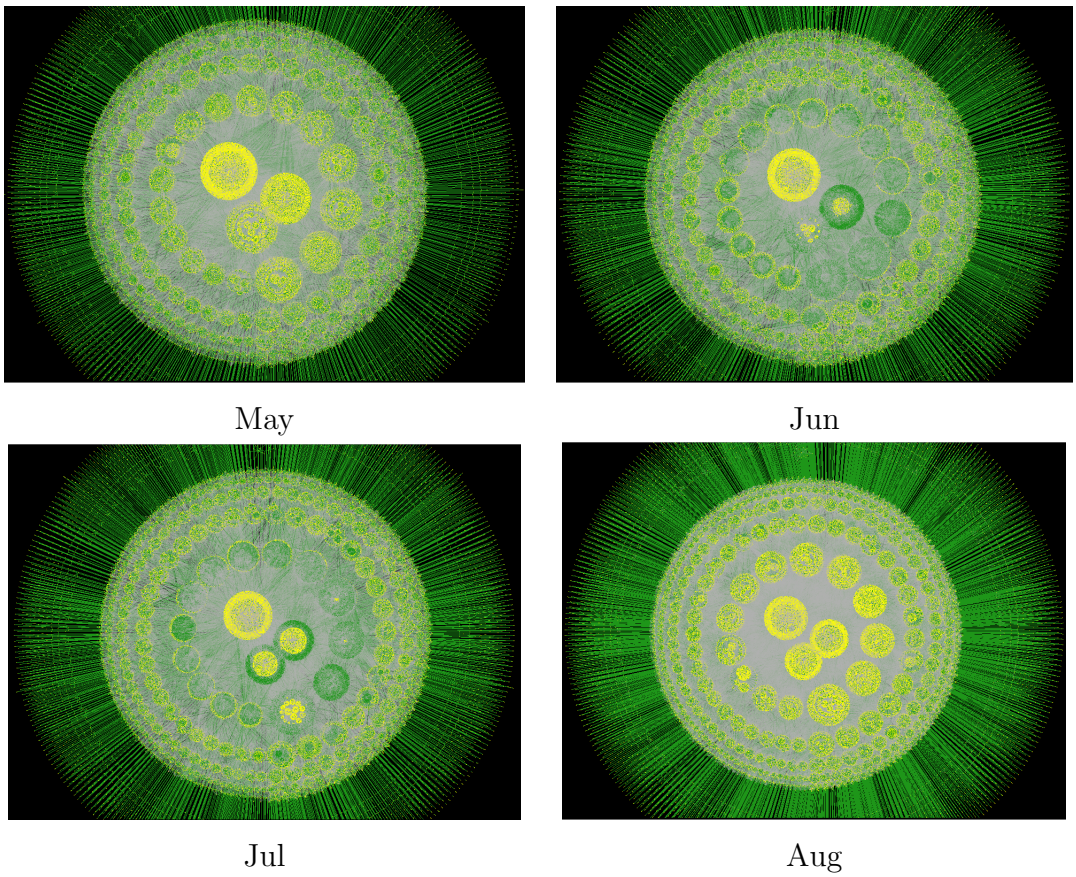


Figure 1.2: Macroscopic changes in each connected component of the Gowalla friendship networks over four months. The number of nodes increases from 110k to 160k during this time [SNM11].

a valuable asset for marketers and researchers alike. Typically, researchers are not able to request a copy of the data from the service providers, but are allowed to indirectly access publicly accessible data by writing a program to connect to the remote website for academic purposes. This practise, also known as *crawling*, has enabled a number of key studies to be published on the analysis of large OSN datasets:

- **Facebook**: being a very popular OSN there exists several different crawled datasets [TMP11, BAAS09, fac]. However, the most comprehensive is that of the University of Santa Barbara [WBS+09] which includes both the social network and interactions between users. Firstly, an individual user profile provides information on their friends and the profiles of these friends provide

information on their friends. This means we can construct the social network of people (nodes) and their relationships (edges). Secondly, interaction networks: individuals can post messages on one anothers profile pages; this again creates an *interaction network* (nodes represent people and a directed edge represents a message being sent).

Temporal information: As users add new friends and delete ex-partners, enemies etc. over time and, hence, we have information on the social network topology as it changes over time. In the case of interactions, messages are time-stamped and so information about interactions at different times is available.

- **Twitter**: Twitter is a service where users can share a short 140 character message, known as "tweets", with friends. Users subscribe to (or "follow") any other users profile to access their tweets but, unlike other OSN such as Facebook, friendship do not need to be reciprocated. Recent datasets have crawled the entire corpus of tweets over a one-month period [KLPM10] and a subset of tweets that included tweet location information over a twelve-day period [SMMC11]. The former dataset contained 41.7 million user profiles, 1.47 billion directed links and 106 millions tweets; whereas the latter dataset which filtered out users with geographic information contained 400,000 user profiles, 183 million directed links and 334.5 million tweets. Both these datasets allow use to construct two different graphs: a graph of followers and a graph of tweets.

  Temporal information: Firstly, both datasets contain timestamps of each tweet and hence we can trace the dynamic spread of tweets and retweets as it cascades through the user network. Secondly, since users can constantly follow and unfollow users, the topology of followers changes over time.

- **Location Based Social Networks**: In addition to maintaining friendships online, the latest feature is the ability for a user to update their current location either manually or using GPS, built into many devices, which allows people to know where their friends are currently located. Such services are known as *location based online social networks* or *LBN's*, for short. Several datasets exist from several popular LBN services such as Foursquare[4] [SMML10a],

[4] `http://www.foursquare.com`

Gowalla[5] [SNM11] and Brightkite[6] [SMML10a]. With this user location information, we can construct a graph of user co-locations i.e. users who report that they are at the same location at the same time. Such graphs have been used in link prediction problems [SNM11].

Temporal information: User checkins are timestamped and so we have spatio-temporal information. This naturally means the topology of co-located users changed over time. Again, users can maintain friendships online as can be seen for Gowalla in Figure 1.2.

1.1.1.3 Human Contact Networks

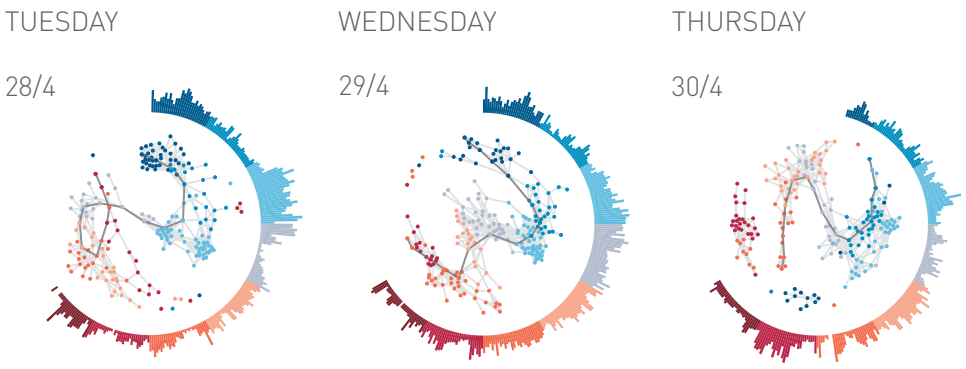


Figure 1.3: Daily contact graph of visitors at the Science Gallery, Dublin. Circumference represents number of contacts at the time corresponding to 12-hour clock. (Source: `http://www.sociopatterns.org`)

The study of close-range human contacts has received attention from epidemiologists [ISB+11] who wish to study the spread of viruses and technologists who are interested in opportunistic routing in pocket switched networks [HCY08] and mining the daily routine of users [EPL09]. This has resulted in several experiments that aim to record participants' meetings with other people. In the Haggle study [HCY08], participants were asked to carry a Bluetooth enabled devices on their person that would scan and record other Bluetooth devices, which were in proximity (within

[5] `http://www.gowalla.com`
[6] `http://www.brightkite.com`

a 30 metre range). Different environment and number of participants were used, ranging from an office with 12 users to a conference with 78 users, with the intention of investigating decentralised routing of messages between mobile devices. In the Reality Mining study [EPL09], 100 participants were given a Bluetooth enabled smartphone to carry on campus over the course of 9 months with the goal of data mining human social behaviour, such as predictability. Again, the devices would record other Bluetooth devices that were in proximity to itself. In the EmotionSense study [RMM+10], social psychologists and computer scientists asked 18 participants to carry Bluetooth and other sensor enabled devices to record their interactions with other people and their emotions, sensed through the device microphone. The SocioPatterns project[7] used RFID tags on necklaces to record face-to-face proximity (1 to 1.5 metres) co-locations, to study the spread of airborne viruses.

All four studies allow us to infer when a pair or even a group of people are in proximity (for either radio communication or to transmit a biological virus). From this, a graph of people (nodes) and their contacts (edges) can be generated.

Temporal information: Since we have timestamps when a device comes into and out of range of another device, we have information on the duration of a meeting; conversely, there is information on the time between successive meetings between the same pair of devices and potentially, periodic patterns between user co-locations. The topology of the graph also changes over time as people move into and out of the range of eachother (see Figure 1.3).

1.1.1.4 Human Influence Networks

Within social science research, it is not just the social graph that is of interest but the semantic information regarding the participants and their relationships. For example, in understanding how smoking habits are influenced by friendships, Mercken et. al. [MSS+10] used anonymous questionnaires to ask 1326 adolescents at 11 Finnish high schools their best friends, the number of cigarettes smoked during a week and alcohol consumption; the questionnaire was repeated 12, 24 and 30 months after the initial questionnaire. From this data, a social network of best friends can be generated along with semantic attributes of each person regarding their cigarette and alcohol consumption.

[7] `http://www.sociopatterns.org`

Temporal information: since the questionnaire was repeated over 4 years, the social network and attributes of each person changes year-by-year. This allowed the study to examine the influence of friends and peers to smoking and drinking habits.

1.1.2 Biological Networks

1.1.2.1 Neural Networks

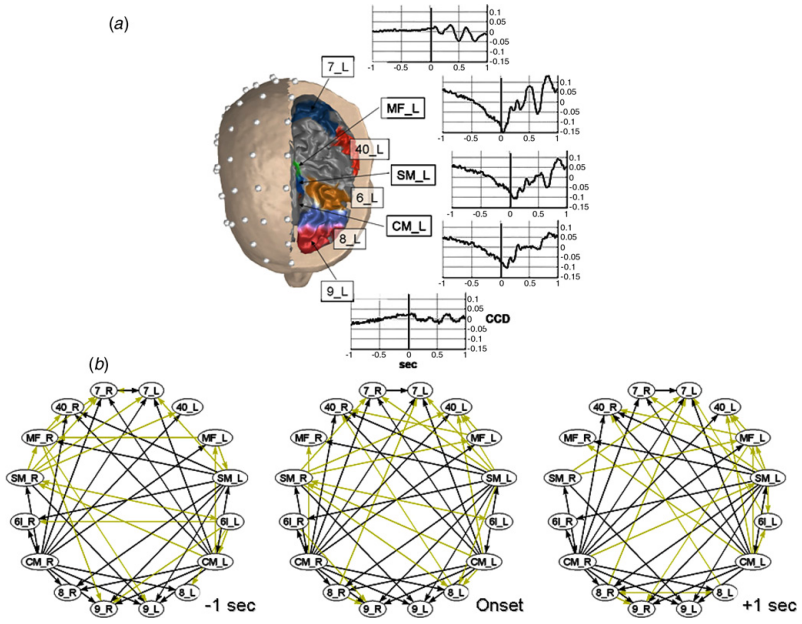


Figure 1.4: Mapping the human cortical network [FLA+08] (a) Electrode placement on human scalp for neural network representation. Correlations between cortical regions change over time. (b) Cortical correlations can be represented as a graph during three time instants (-1 sec, onset, +1 second). (*Reproduced with kind permission of IOP publishing and F. De Vico Fallani.*)

The brain can be represented as a network of neurons (nodes) and synapses (edges) which propagate signals between neurons [AB09][8]. It is generally accepted that the size of an animals brain (number of neurons) determines the computational power

[8]In fact, recently it has been proposed that two additional networks can be derived from the brain: astrocytes and microvascular [DW10]; we concentrate on the well studied neuronal network.

or intelligence of an animal; as humans we have a much larger expected number of neurons relative to our body size of any mammal, estimated to be around 86 billion neuronal cells and 85 non-neuronal cells [ACG+09]. However, mapping the human brain (both neurons and synapses) is currently an intractable task and at present the only completely mapped neural network is of the nematode worm *C. Elegans*, with only 282 neurons. Small world behaviour has been reported in the neural networks of the C. Elegans neural network [WS98] and this combination of applying network analysis to neural networks has opened new possibilities to analyse approximations of the human neural network, whilst the technology to fully map the human brain improves.

The main method of constructing a neural network is by monitoring the electrical or magnetic activity of the brain using external sensors. For example, when participants are asked to perform a simple physical task, their brain activity can be monitored via electroencephalography (EEG) data [FLA+08] and magnetoencephalographic data (MEG) [KHS+11]. Nodes represent different areas of the brain and activity represents an edge between two nods.

Temporal information: the human brain is dynamic in two ways: firstly, even when we remain still, the brains electrical activity is constantly changing as it controls our stationary respiratory and cardiovascular functions; this results in an evolving topology of engaged synapses or correlation between cortical regions of interest over time (see Figure 1.4). Secondly, it has been shown that the neural network can rewire itself over time as the brain learns [DGB+04].

1.1.2.2 Ecological Networks

Ecologists have been using diagrammatic abstractions to visualise and classify the evolution of animals into species, for example humans falling under the class of mammals along with many other types of animals who share the same warm-blooded ancestry. Such a structure is known as a *tree* since it branches at certain evolutionary junctions. More recently, networks have been employed to describe the complex relationship between predator and prey, known as *food webs*. Such networks are collected through painstaking observation on-location and networks represent different species (nodes) and a directed predator-prey relationship (edge) [BU89].

Temporal information: many of the studies are interested in the seasonal effects on food webs and hence the network topology is observed at different time points. For example, the Chesapeake Bay data [BU89] is collected over 4 years, once for each season with a total of 16 observations.

1.1.3 Technological Networks

Since technological networks generally store data in digital form, this has facilitated the availability of datasets generated from these sources.

1.1.3.1 World Wide Web

The World Wide Web (WWW) or "web" is made up of billions of webpages which can hyperlink to (and be hyperlinked from) other webpages. This naturally produces a network of webpages (nodes) and directed hyperlinks (edges) and has been used in the study of scale-free networks [BA99]. Such data is regularly crawled by search engines such as Google[9], Yahoo![10] and Bing[11] etc. by following and recording hyperlinks from one webpage to another, so that search results are informed by the popularity of certain webpages. In the same manner, such datasets are available for collection by the researchers.

Temporal information: the WWW changes every day as webpages and hyperlinks are both added and deleted, therefore, there is rich information on the changing topology of the web.

1.1.3.2 Internet

The Internet is comprised of routers that carry the traffic from any computer or device connected to the Internet. The Internet can be decomposed into connected sub-networks that are under separate administrative authorities [FFF99] known as *domains* or *Autonomous System* (AS). It is possible to study the Internet at two

[9] `http://www.google.com`
[10] `http://www.yahoo.com`
[11] `http://www.bing.com`

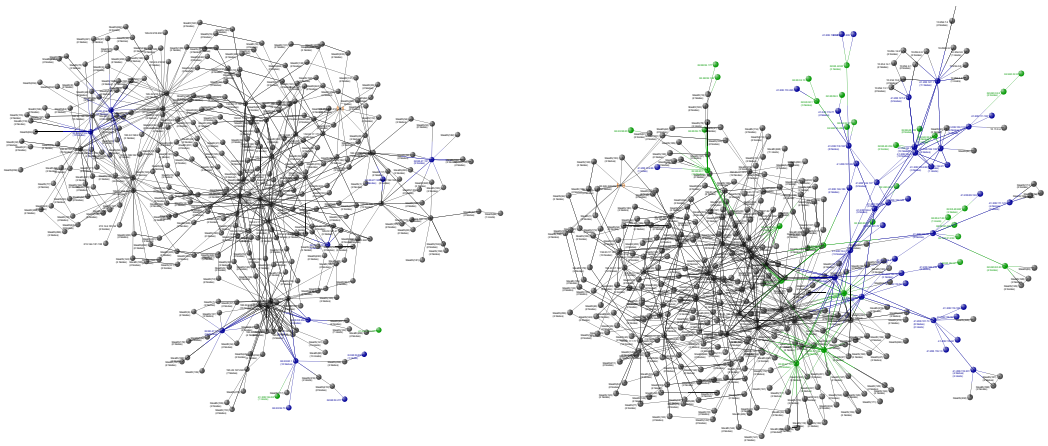
(a) 1999 (100,000 nodes) (b) 2011 (270,000 nodes)

Figure 1.5: Topological Map of the Internet developed by Lumeta Corporation: Lumeta continues a long-term research project, started at Bell Labs, to collect routing data on the Internet using their IPsonar technology. The project consists of frequent path probes, one to each registered Internet entity. From this, trees are built mapping the paths to most of the networks on the Internet. The specific endpoints or network services on those endpoints are not the goal of this map, but rather the subject being mapped here is the topology of the "center" of the Internet. These paths change over time as the routes are reconfigured and as the number of routers across the Internet increase over time. (*Reproduced with kind permission of Lumeta Corporation: Patent(s) Pending & Copyright ©Lumeta Corporation 2000-2011. All Rights Reserved.*)

granularities: firstly, at the router level where nodes are routers and physical connections between routers are edges; and secondly, at the domain level, where an AS is a node and connections between ASes are edges. The domain level can be broken down further into physical connections between AS and logical connections, which are derived from business policies which dictate the flow of traffic between neighbouring ASes.

The network topology themselves are collected by institutes such as the Advanced Network Technology Center[12] at University of Oregon, the National Laboratory for Applied Networking Research (NLANR)[13] and the Lumeta Internet Mapping

[12] `http://www.routeviews.org`
[13] `http://www.psc.edu/networking/nlanr`



(a) 11 Aug 2011 (b) 22 Aug 2011

Figure 1.6: Libya on the Internet: (a) map depicts the presence of Libya on the Internet as seen on the 11th Aug, only 13 backbone routers can be seen from Lumeta's Somerset, NJ headquarters. (b) On 22 Aug 2011, after rebels were report to have seized several major cities, 68 backbone routers can be seen. (*Reproduced with kind permission of Lumeta Corporation: Patent(s) Pending & Copyright ©Lumeta Corporation 2000-2011. All Rights Reserved.*)

Project [14], which have permission to access participating AS routers to monitor connections and traffic. Several studies of the network constructed from the Internet at these two levels have uncovered power-law structure in the degree distribution of the network[FFF99] and how the evolution of the topology affects the density of the network[LKF05].

Temporal information: The Internet is highly dynamic with the addition and removal of new routers and Internet service providers (ISPs); snapshots of this changing topology can be captured (Figure 1.5). In addition, geo-political events might dictate the access to some parts of the network such as seen in the recent government restrictions in Egypt and Libya (Figure 1.6). As part of the data collected, there is information available on the traffic demands between nodes which can be used to construct a timestamped communication network and also a complex set of rules between which dictate the routes and traffic between ISPs and routers.

[14] `http://www.lumeta.com/Internet-map`

1.1.3.3 WiFi hotspot

Wireless technology has given us the freedom to connect to the Internet and browse the web away from the desk and fixed connections. WiFi hotspots are ubiquitous in the home, at the office, on campus, airports and high streets. With the devices regularly connecting to WiFi routers the logs of device access can be used to identify device locations and infer device co-locations to generate a graph of devices (nodes) and co-locations (edges). Numerous studies have collected data on different time scales, spatial properties and environments including campus environment over 5 years across 450 WiFi access points[KHAY09]; office environment over a week across 151 access points[BC03]; and city environment over 3 years across numerous free access points across Montréal[LGP07].

Temporal information: Timestamps are associated to a device connecting to (and disconnecting from) an access point. This gives us connection duration, co-location with other devices other time and possible periodic behaviour of connections (e.g., if a user connects at the same time and location every weekday in the office).

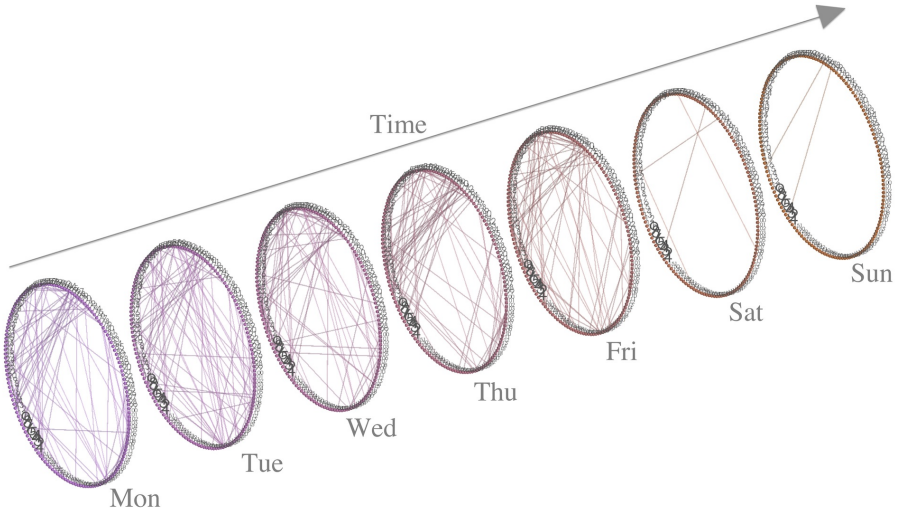


Figure 1.7: Daily map of email exchanges between corporate employees.

1.1.3.4 Digital Communication Networks

The Internet has given us the ability to communicate globally through email and instant messenger. Generally, such messages are routed through a centralised server

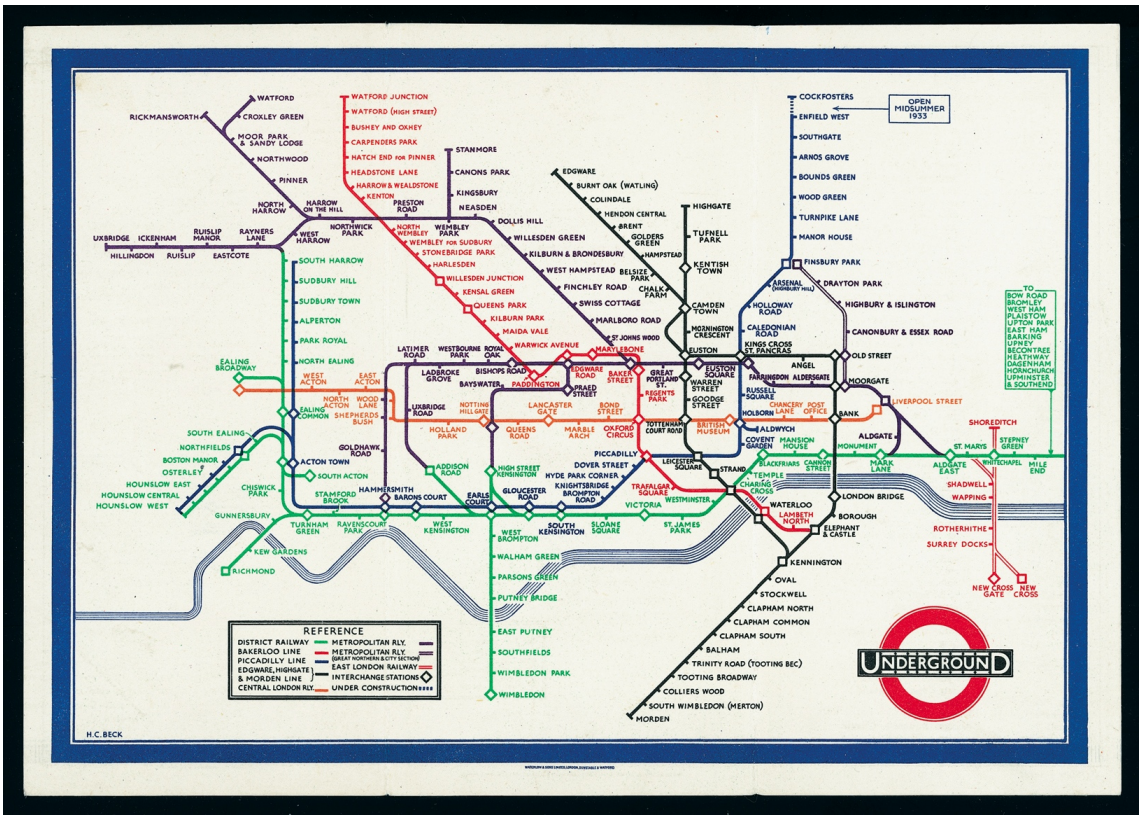


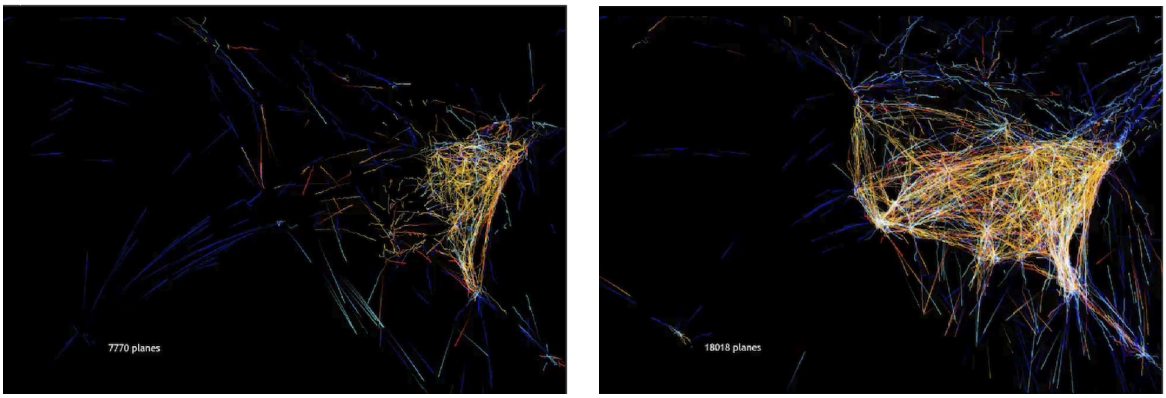
Figure 1.8: Beck's topological London Underground Tube Map from 1933. (©*Transport of London. Reproduced with kind permission of Transport of London.*)

and access to this server provides us with data on interactions between users. From this data, a network of users (nodes) and the messages sent (edges) can be constructed. Studies have included scale-free properties of 57,000 University email users [EMB02], corporate emails between 151 colleagues during a 3 year period before a corporate bankruptcy filing [SA05] (see Figure 1.7) and six-degrees of separation between users on a global scale instant messenger service with 30 billion conversations between 240 million users [LH08].

Temporal information: Messages between users are timestamped and an instant messenger session (i.e. the time two users are engaged in conversation) is engaged over some duration.

1.1.4 Urban Networks

Urban planners strive to improve the economic and social environment of communities and this encompasses a wide range of research that is suitable for the network abstraction of node and edges. For analysis, the *spatial networks* abstraction is employed which captures the topological or schematic (as opposed to exact geographical) connections between points of interest such cities, junctions or stations; an iconic example is that of the Beck's London Underground map (Figure 1.8).



(a) 11 AM (EST) (b) 5 PM (EST)

Figure 1.9: Airline route map over the United States during the morning and afternoon. (*Reproduced with kind permission of Aaron Koblin.*)

The network of cities and the roads [SFF+10] give rise to traffic flow analysis [Lie03] and quickest route algorithms for satellite navigation systems which can avoid traffic [Gol99]; the layout of junctions and roads within cities dictates the accessibility and popularity of certain areas [PLW+09]; the network of stations, lines and interchanges of public transport systems; analogously, the network of airports and routes in airline maps [BBPV04] (Figure 1.9); the network of power grids and power lines [WS98]; and even the space and layout of office building can be analysed as networks, where rooms are nodes and passages between rooms are edges, to understand efficient architectural layout [Hil96]. Such networks are inherent in our everyday lives with city-level, national-level and public transport maps easily accessible in online digital forms.

Temporal information: Firstly, traffic demands on roads, power lines and even public transport is dictated by human periodic behaviour and hence, when repre-

senting a network to measure traffic, the edge weights inherently change over time. Secondly, the topology of such urban networks are likely to change over time as new roads, cities, stations and lines are built. Finally, public transport is dictated by *timetables* and serves as an example of the importance of time ordering when travelling on a route that requires multiple changes.

		Temporal Information	
Dataset	**Nodes**	**Description**	**Available in dataset?**
Actor co-stars [WS98]	225226	time of movie	no (available from source)
Co-authorship [New01b]	70975	time of publication	no (available from source)
Sexual Partners [LEA+01]	2810	time of contact	no (available from source)
*Facebook, friendship [WBS+09]	6m	addition & deletion of friends	no (available from source)
*Facebook, interactions [WBS+09]	6m	time of interaction	yes
Twitter, complete [KLPM10]	41.7m	time of tweet	yes
Twitter, geo [SMML10a]	409093	time of tweet	yes
Foursquare [SMML10a]	58424	time of checkin	yes
Gowalla [SNM11]	159391	time of checkin	yes
Brightkite [SMML10a]	54190	time of checkin	yes
*INFOCOM'06 [SGC+09]	78	time of contact start & end	yes
*Reality Mining [EPL09]	100	time of contact start & end	yes
*EmotionSense [RMM+10]	18	time of contact start & end	yes
SocioPatterns [ISB+11]	140000	time of contact start & end	yes
Adolescent Smokers [MSS+10]	1326	4 snapshots	yes
C. Elegans [WS98]	282	topological changes over time	no (hard to collect)
*Brain Network, EEG [FLA+08]	16	brain activity over time	yes
Chesapeak Bay [BU89]	33	4 year seasonal changes	yes
WWW [BA99]	153127	topological changes over time	no (available from source)
Internet [FFF99]	3888	Traffic, topological changes	no (available from source)
Dartmouth WiFi [KHAY09]	5338	time of connection	yes
*Kiel University Email [EMB02]	56969	time of emails	yes
*Enron Email [SA05]	151	time of emails	yes
MSN Instant Messenger [LH08]	180m	time of messages	yes
Power Grid [WS98]	4941	Traffic demands on lines	no (available from source)
Airline Routes [BBPV04]	3880	Route timetable	no (available from source)

Table 1.1: Summary of empirical datasets with temporal information. * indicate datasets used in this dissertation.

1.1.5 Summary and Discussion

1.1.5.1 Classifying Temporal Information

We have covered a range of empirical datasets used in the literature, some of which have been used for static complex network analysis but where the temporal information is accessible from the original source (e.g., timestamps in the actor, co-author

datasets), others where the temporal information is inherent but would be non-trivial to collect (e.g. mapping the neural network of the C. Elegans over time), and some of which were collected with timestamps present. We have summarised the datasets with these distinctions in Table 1.1.

Through these example datasets, we can isolate four distinct sources of time information, which will inform our temporal graph model in Chapter 3, namely:

1. **Timestamps** can be associated to both nodes (new users or users leaving an OSN) and edges (a friendship being added or removed, a message being sent, a meeting between two people etc.).

2. **Duration** is implicit in these timestamps are some form of duration, for example the length of time a friendship lasts, the time it takes for a message to be sent and delivered, how long two people meet.

3. **Frequency** can be analysed once we have a list of timestamps for an edge or node; this can uncover patterns in edge or node occurrences. Furthermore, periodicity is present in certain datasets such as transport traffic (e.g., during the morning and evening, before and after work) and human contact networks (e.g., daily meetings with colleagues or family members).

4. **Time-order** was highlighted in several datasets where, for example, the timetable in public transport systems and a message or virus passing through a network. As we shall demonstrate in Chapter 3, this is an important piece of information which is missed in static graph analysis. More generally time-order can be described as time *dependency* between events, for example, changing the order of timestamped events would have an effect on metrics defined upon it.

On top of this, we can also categorise two types of dynamic graph behaviour. Firstly, **topological changes** over time occur with fluctuations of the edges between nodes as meetings between people begin and end or traffic moves along to congest and free up roads. Secondly, **process driven changes** are driven by some form exchange of information i.e. a message or a virus.

1.1.5.2 Static versus Temporal Analysis

We should stress that the aim of this dissertation is not to reject the use of static network analysis, but merely offer an alternative view whereby the incorporation of temporal information can potentially lead to more accurate analysis of networks where temporal information is inherent. Static graph analysis simplifies the analysis of real network by ignoring time information but is still useful for many types of analysis where, for example, time information is not required or only a single snapshot is required for analysis. In other cases, temporal analysis does not make sense since the changes would be minute, for example, the topology of the power grid does not change very frequently. However, the analysis of the traffic demands on the cables carrying power would fluctuate frequently and in this case could potentially benefit from temporal analysis.

1.1.5.3 Evolving versus Temporal Networks

We should also distinguish between the concepts of the well studied *evolving* network and the proposed *temporal* network analysis. Evolving networks are *generative* models such as *preferential attachment* [BA99], which describes the *accumulation* of nodes and edges over time. The preferential attachment model was devised to understand how scale-free (where the degree distribution can be described by a power-law function) network topologies are formed over time as new nodes join a network. Simply, the recipe captures a snowball effect, where new nodes have a higher probability to form a link to popular nodes. This is used to explain the scale-free structure of the WWW as new webpages are added they hyperlink to existing well-known webpages; new researchers are more likely to co-author a paper with well-known, respected peers etc. Such evolving networks lends well to static analysis since the most current cumulative topology is of interest and has given rise to insightful results such as shrinking diameters (maximal shortest path length) and densification over time as nodes and edges are added to the graph [LKF05].

This is different from our proposed temporal model of *fluctuations* in edges and nodes.

1.1.5.4 Parallels with opportunistic networking

This dissertation straddles applied network science, particularly within computer science, with complex network analysis. In particular, it is pertinent to note that within computer science, *opportunistic networking* and *delay tolerant networking* (DTN) has studied message dissemination between mobile devices via intermediate hops; such studies *inherently take into account time*. For this reason, is important to note that this thesis is distinct from opportunistic networking and DTN research in that, the aim is to formalise metrics that measure the message dissemination properties of the *whole* network (similar to the characteristic path length from a global perspective, and characteristic clustering coefficient from a local perspective). By taking this higher level view of the complete network, we aim to firstly, uncover universal rules which can describe all types of time-varying complex networks (in addition to ad hoc mobile networks) and secondly, describe the relationship between message dissemination metrics and structural properties of time-varying networks.

1.2 Contributions

The major contributions of this thesis are twofold: firstly, the definition of temporal metrics and secondly, the demonstration of the utility of temporal analysis on real networks. These contributions are summarised as follows:

- Firstly, we define the notions of **temporal shortest paths** and **temporal shortest path lengths** which are fundamental to the study of information dissemination in real networks. These are defined upon a temporal graph model which extends the traditional static network (or graph) representation to take into account time information; Intuitively, this model is a series of snapshots of the network topology as it changes over time. We also define metrics to measure the **temporal local efficiency** to capture information dissemination between neighbouring nodes and **temporal correlation coefficient** to characterise the evolution speed of a temporal graph. Utilising these metrics to study real network datasets, we find that, firstly, since static aggregated graphs ignore time order of links, this overestimate the available links to facilitate a shortest path and therefore, underestimates the true shortest path

length between nodes; and secondly, contrary to intuition, slowly evolving graphs can still be configured for efficient information dissemination between nodes, exhibiting small-world behaviour in time-varying networks (Chapter 3).

- Secondly, we redefine well established metrics from social network analysis pertaining to the identification of important nodes in a network for quick information dissemination and mediation, namely **temporal closeness** and **temporal betweenness** centrality. We apply these temporal centrality metrics to a corporate email dataset during the year previous to a bankruptcy filing and find that temporal centrality identifies more intuitively important people in the corporation compared to those identified by static analysis. We also exploit temporal centrality in short range mobile malware containment and devise two schemes based on patching key mediating nodes (using temporal betweenness) and opportunistically spreading the patch from key nodes (using temporal closeness). We find that the former scheme is not efficient to due many alternative temporal paths which a mobile worm can utilise, however, the latter scheme can spread quicker than the mobile worm (Chapter 4).

- Thirdly, we present a technique for finding temporal correlation in the rankings of node centrality for **top-k node centrality prediction**. We find that there is legacy correlation of top-k nodes, such that if a node is important now, then it is likely to be important at the same time tomorrow. We find that a simple ageing function can help predict future important nodes and we evaluate this accuracy again on mobile malware containment (Chapter 5).

- Fourthly, we define **temporally connected components** in the study of reachability of nodes in real networks. We show that the problem of finding strongly connected components in a time-varying graph can be mapped into the problem of discovering the maximal-cliques in an opportunely constructed static graph, which we name the affine graph, and is therefore a NP-complete problem. Despite this, we demonstrate that temporal component analysis can better capture the connectedness of a time-varying network compared to static analysis which overestimates the reachability between nodes (Chapter 6).

1.3 Chapter Outline

The remainder of this dissertation is organised as follows. As we have seen, real networks change over time and we argue that existing static network analysis cannot fully capture the dynamic nature of these networks. In Chapter 2, we start with defining measures used in static network analysis to gain insight into why this may be the case and to aid in the derivation of temporal measures. In Chapter 3), we present the temporal graph model and define temporal distance metrics. In Chapter 4, we define temporal centrality metrics and study a real corporate email dataset and short range mobile malware containment. In Chapter 5, we study temporal centrality prediction. In Chapter 6, we investigate temporal reachability in real networks. Finally, in Chapter 7 insights and consequences, which can be drawn from this dissertation, are presented and we discuss directions for future research.

1.4 List of Publications

During the course of my PhD, I have had the following five papers published and currently have two papers under review. Chapter 3 is based on [TMML09, TSMML10]. Chapter 4 is based on [TMMLN10, TMML11]. Chapter 5 is based on [TKMM11]. Chapter 6 is based on [NTMRML11]. [TMML10] is an extended version of [TMML09].

Published Works

[TMML09] John Tang, Mirco Musolesi, Cecilia Mascolo and Vito Latora. Temporal Distance Metrics for Social Network Analysis. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Online Social Networks (WOSN '09)*, pages 31–36, Aug 2009, Barcelona, Spain.

[TMML10] John Tang, Mirco Musolesi, Cecilia Mascolo and Vito Latora. Characterising Temporal Distance and Reachability in Mobile and Online Social Networks. In *ACM SIGCOMM Computer Communication Review (CCR)*. Vol. 40 (1), pages 118-124. Jan 2010. ACM Press.

[**TMMLN10**] John Tang, Mirco Musolesi, Cecilia Mascolo, Vito Latora and Vincenzo Nicosia. Analysing Information Flows and Key Mediators through Temporal Centrality Metrics. In *Proceedings of the 3rd ACM SIGOPS Workshop on Social Networks Systems (SNS '10)*, pages 1–6, Apr 2010, Paris, France.

[**TSMML10**] John Tang, Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo and Vito Latora. Small World Behavior in Time-Varying Graphs. In *Physical Review E*, Vol. 81 (5), 055101 (R), May 2010.

[**TMML11**] John Tang, Cecilia Mascolo, Mirco Musolesi and Vito Latora. Exploiting Temporal Complex Network Metrics in Mobile Malware Containment. In *Proceedings of the 12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '11)*, pages 1–9, June 2011, Lucca, Italy.

Under Review

[**TKMM11**] John Tang, Hyoungshick Kim, Cecilia Mascolo, Mirco Musolesi. SPOT: Socio-Temporal Opportunistic Patching of Short Range Mobile Malware. *Submitted to the 13th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '12).*

[**NTMRML11**] Vincenzo Nicosia, John Tang, Mirco Musolesi, Gianni Russo, Cecilia Mascolo and Vito Latora. Components in time-varying graphs. *Submitted to Physical Review E.*

2

Static Complex Network Theory

Introduction

In the previous chapter, several examples of real networks were presented along with their representation as a graph. More formally, we define a *graph* G as a 2-tuple (V, E) where V is the set of *nodes* (or *vertices*) and E is the set of *edges* (or *links*) connecting a pair of nodes. A graph can be directed or undirected– where an edge between two nodes is either non-mutual or symmetric, respectively. A graph can also be weighted or unweighted– where a weighted graph can have values assigned to an edge, for example the current traffic on or time it takes to drive across a road. Unless explicitly stated, this chapter shall define metrics upon an undirected, unweighted graph. A graph can be represented as an N-by-N adjacency matrix A, where $N = |V|$, and the value a_{ij} at row i and column j is non-zero if an edge exists from node i to j. In the case of an unweighted graph $a_{ij} = 1$ if there is an edge, 0 otherwise; in the case of a weighted graph a_{ij} can be any real number.

This chapter provides a background to the tools employed in the analysis of static networks relevant to this thesis, with the aim to understand how temporal metrics can refine the static definition.

Chapter Outline

Section 2.1 discusses simplifications of static graphs when constructed from real network datasets with time information. In Section 2.2, key metrics pertinent to the study of information dissemination in temporal networks are presented in the context of small-world studies, identifying important nodes in networks and connectedness of a graph. Conclusions are presented in Section 2.3.

2.1 Static Model

As discussed in the previous chapter, a graph can be used to model a wide range of real work networks, where a node could be a person, city, neuron or webpage etc. and an edge could represent a relationship, road, synapse or hyperlink etc., respectively. However, we highlight the fact that this is a simplification of the real network characteristics since *time information is ignored*. More specifically, there are two types of simplification that can be observed in existing literature:

- Firstly, many studies collect the current topology of the graph, for example the graph of movie co-stars [WS98], the network of webpages [BA99] or the network of power grids [WS98]. However, the network data collected was only the **current snapshot** of the complete time-line of the network: actors star in new movies over time and the graph of co-stars grows; new webpages hyperlinked to existing pages and the WWW graph changes over time; friendships are created and removed from OSNs and so the friendship graph is different etc.

- Secondly, where temporal information is available, many studies explicitly ignore time information and construct a static graph from the **union of edges** across all temporal occurrences. For example, in the study of scale-free properties of email networks [EMB02] the authors explicitly state that "the nodes

of the e-mail network correspond to e-mail addresses which are connected by a link if an e-mail has been exchanged between them", which means that if several emails were sent between a pair of nodes, that edge is counted once (ignoring the frequency). In the study of the worldwide airport network [BBPV04] two nodes (airports) are linked by an edge if there is a direct flight at any time over the course of a year; again, this takes the union of an edge across time.

In fact, both of these simplification can be regarded as a form of edge *aggregation* and hence we refer to static graphs as *static aggregated graphs* in this dissertation. Again, in its defence, static network analysis is very powerful in aiding the study of real networks as demonstrated by seminal results produced over the last decade.

2.2 Static Analysis

2.2.1 Small-world metrics

The *small-world* phenomenon was first studied by Stanley Milgram [Mil67] in 1967 who performed an experiment asking 160 random selected participants in the US town of Omaha, Nebraska, to deliver a package to a specified target, an acquaintance of Milgram's, who worked in Boston, Massachusetts. The information supplied to participants were the targets name, occupation and address, however, participants were prohibited from mailing the letter directly to the target, instead were requested to send the package onto their own friends or acquaintances whom they felt could get the letter "closer" to the target. These friends or acquaintances were then given the same instructions and for each transition, a postcard was sent back to Milgram with details of the receiving party. Through this experiment, Milgram showed that the package could be delivered through a "chain" of acquaintances forming a path; the average path length of the 44 completed chains[1] was 6 which led to the phrase "six degrees of separation"[2]. Milgram made two interesting observations [Mil67]: firstly, these chains provide an upper bound to the shortest paths; and secondly, the penultimate person in 48 percent of the chains were mediated by only 3 people,

[1]The other 126 chains terminated at acquaintances who failed to participate.

[2]This term was not used in Milgram's paper but was later popularised by the play of the same name [Gua90]

which suggests that highly popular or "clustered" nodes are important for funnelling message to a destination.

Watts & Strogatz [WS98] later formalised these observation through two metrics: the characteristic path length and the clustering coefficient; we now present these metrics followed by their findings on real world networks.

2.2.1.1 Paths and Shortest Path Length

Before we can define the characteristic path length, we first need to define the concepts of paths and path lengths. A path P_{ij} is defined as a list of nodes starting from node i and finishing at j, where an edge exists between each intermediate pair of nodes and the *length* of a path is measured by the number of intermediate hops from source to destination. There may be many different paths of different lengths from the which we refer to as the set $\mathcal{P}_{ij}$. Also, all paths are *acyclic*, in that there are no cycles or repeated nodes in a path.

The shortest (or geodesic) path length, d_{ij} from i to j is defined as the minimum path length over all paths $P_{ij} \in \mathcal{P}_{ij}$. From this, the characteristic (or average) path length, L is defined as:

$$L = \frac{1}{N(N-1)} \sum_{i \neq j \in V} d_{ij} \tag{2.1}$$

This captures the *global* characteristics of a graph since transitive paths can connect every pair of nodes.

2.2.1.2 Clustering Coefficient

Clustering coefficient measures the number of nodes that are also neighbours with one another. More formally, for a node i, its clustering coefficient C_i is calculated as the fraction of links that exists between the neighbours of a node k_i of node i, over the total possible number of edges $k_i(k_i - 1)/2$. From this, the average clustering coefficient of a graph, C, is defined as:

$$C = \frac{1}{N} \sum_{i \in V} C_i \tag{2.2}$$

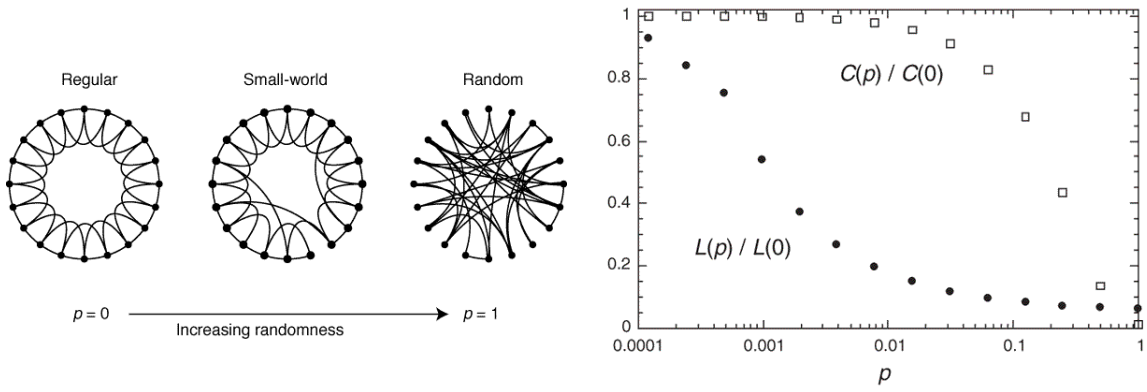
This captures the *local* interactions between nodes since it only considers neighbours or close relationships of each node.

2.2.1.3 Small-world behaviour

The intuition is that small-world networks exhibit strong clusters of nodes such as groups of friends which are more likely to be linked from certain nodes to distant clusters, providing a "shortcut". It is the combination of these close-knit clusters of nodes (which can interact *locally*) and these shortcut links (which aid in *global* interactions) that help in reducing the number of transitive hops between any two nodes in a large network.

To demonstrate this intuition, Watts & Strogatz calculated these two global and local metrics by extrapolating between a totally ordered graph and a random graph (see Figure 2.1(a)). In the ordered graph, nodes are arranged around a ring as a regular lattice where each node is connected to K nearest neighbours; by definition this gives a strong local cohesion as demonstrate by high values of C but poor distant connections resulting in high values of the characteristic path length L ($p = 0.0001$ in Figure 2.1(b)). On the other side of the spectrum, to create a random graph from the lattice, for every node take its links and with probability p rewire this edge with a random node. If p is high then local connections to neighbours will be lost and only links to distant nodes remain which gives low values of characteristic path length L and low values of clustering C ($p = 1$ in Figure 2.1(b)). The key insight is that when we interpolate between the regular lattice to the random graph by changing the value of p (Figure 2.1(b)), we obtain a graph which can be both highly clustered and exhibit a low characteristic path length; which has been become known as a "small-world" graph.

Small-world behaviour was also found in empirically observed networks, where it is measured relative to random graph with the same number of nodes N and average node degree $\langle k \rangle$. For a network to exhibit small-world properties, it is expected that L be similar to that of a random graph and exhibits much higher clustering C compared to the same random graph. In the original Watts & Strogatz paper, it is demonstrated that this property held in social (movie co-stars), neural (C. Elegans) and power grid networks. Proceeding studies have shown that this seemingly universal property exists in many other networks including email [EMB02],



(a) (b)

Figure 2.1: **(a)** Watts & Strogatz network model: extrapolating from a pure lattice to a random graph with probability p of rewiring an edge. **(b)** Log-normal plot of the characteristic path length and clustering coefficient (y-axis) as the rewiring probability p is increased (x-axis). (*Reprinted by permission from Macmillan Publishers Ltd: Nature ([WS98]), copyright (1998).*)

co-authorship [New01a] and airport [BBPV04]. In Chapter 3 we shall present a temporal analogy to the small-world property when taking into account time information.

2.2.2 Efficiency

The small-world measures, proposed by Watts & Strogatz, assumed that the graph was unweighted and connected. The latter needs to be true since if two nodes, i, j cannot communicate via any transitive hops, then $d_{i,j}$ is infinite and cannot be used as part of the average. We shall see later that dense and connected static graphs may actually be very sparse and disconnected when broken down to their temporal equivalent since not all contacts occur at the same time. This assumption is a problem in static graphs where many real networks are disconnected. To overcome this, Latora & Marchiori [LM01] described an Efficiency function which calculated the inverse of the shortest path length d; disconnected node pairs naturally had an efficiency of $1/\infty = 0$, which can be used as part of a mean summation. More formally, the average efficiency E of a graph G can be defined as the harmonic mean:

$$E = \frac{1}{N(N-1)} \sum_{i \neq j \in V} \frac{1}{d_{ij}} \tag{2.3}$$

and, similar to characteristic path length L which captures the global properties of the network, is also referred to as global efficiency E_{glob}. In the same vain, to parallel the local dynamics that C captures, a local efficiency E_{loc} metric is defined as:

$$E_{loc} = \frac{1}{N} \sum_{i \in V} E(G_i) \tag{2.4}$$

where G_i is the neighbour subgraph of a node i. In a small-world network both the global and local efficiency are much higher compared to a random graph.

2.2.3 Centrality

In complex network and social network analysis, centrality refers to the identification of the most "important" nodes in a network. Clearly, node importance is an ambiguous term and could be interpreted in many different ways depending on the application. For example, one could interpret importance as being equal to *popularity* e.g. a person with the most friends; one might argue that a person who can deliver a message quickly to the most people in a network is important; or perhaps, one might give precedence to a person that bridges the most communication channels and therefore is key to mediating between different parties.

In fact, all three interpretations have been well studied in social network analysis are more commonly known, respectively, as *degree*, *closeness* and *betweenness* centrality [WF94].

2.2.3.1 Degree Centrality

Indeed, one of the simplest measures in network analysis is node *degree* N_i, which measures the number of neighbours of a node where $N_i = \sum_{j \in V} a_{ij}$. Since the degree is defined for each node, it is straight forward to derive a measure of centrality based on popularity. The degree centrality of a node i is defined as the number

of neighbours N_i of i normalised by the maximum number of distinct connections, more formally:

$$C_i^{\texttt{deg}} = \frac{N_i}{N-1}. \tag{2.5}$$

2.2.3.2 Closeness Centrality

From a practical perspective, closeness centrality measures how quickly a node can communicate with all other nodes in a network. This is calculated for a node i as the average shortest path length, d, to all other nodes in the network. Formally, this can be defined in terms of shortest path lengths:

$$C_i^{\texttt{clo}} = \frac{1}{N-1} \sum_{j \neq i \in V} d_{ij}. \tag{2.6}$$

or in terms of efficiency to handle disconnected nodes:

$$C_i^{\texttt{eff}} = \frac{1}{N-1} \sum_{j \neq i \in V} \frac{1}{d_{ij}}; \tag{2.7}$$

2.2.3.3 Betweenness Centrality

Betweenness centrality measures the shortest paths that pass through a node and can be thought of as the proportional flow of data through each node. The betweenness of node i is calculated as the proportional number of shortest paths between all node pairs in the network, that pass through i. More formally, this is defined as:

$$C_i^{\texttt{bet}} = \sum_{j \neq i, k \neq i \in V} \frac{p_{jk}(i)}{p_{jk}} \tag{2.8}$$

where $p_{j,k}$ is the number of shortest paths starting from source node i and destination node j, and $p_{j,k}(i)$ are those paths which pass through node i [WF94]. A key point is that betweenness also takes into account *alternative shortest paths* which is meaningful in measuring the robustness of a node to attack; if a node i is the *only* bridging node on a path then its removal would be highly detrimental, whereas, if there were another path that did not include i then its role would be less critical.

2.2.4 Reachability

The reachability of nodes in a graph is important for many reason, for example, one might be interested in finding out if a route exists via some telecommunication channel to deliver an email; whether or not a journey using public transport exists that can be taken from home to work; or if it is possible to drive to Fiji? We can reason upon these questions by mapping them to a graph and calculating the paths from source to destination. Intuitively, the answer to these simple questions depends on the given input source and destination. If, say, we live in a major city, such as London, and our friend whom we wish to deliver an email to also lives in London, then it is highly likely that we are both connected by (and to) the Internet. If, however, the friend lives in some remote mountain range in the Himalayas, such a channel most probably does not exist; in other words, the destination is *unreachable* and there is no path between these two nodes. Many land masses are well interconnected by roads, such mainland Europe, China, the United States and Australia etc., however this does not mean they are connected to one another. Within graph theory, the idea that there are independently connected networks is more formally known as *components*. Defining the concepts of connected components depends on whether we are reasoning on a directed or undirected graph, since directed graphs reduce the number of channels available for any pair of nodes to be connected via; for this reason we now define these concepts explicitly for directed and undirected graphs.

2.2.4.1 Connected Components

In order to define graph components, we need to introduce the concept of connectedness, first for pairs of nodes, and then for the whole graph. We will consider the case of undirected and directed static graphs separately. Two nodes i and j of an undirected graph G are said to be *connected* if there exists a path between i and j. G is said to be *connected* if all pairs of nodes in G are connected, otherwise it is said unconnected or disconnected. A *connected component* of G associated to node i is the maximal connected subgraph containing i, i.e., it is the subgraph of all nodes connected to node i. If an undirected graph is not connected, it is always possible to find a partition of the graph into a set of disjoint connected components, and it is simple to prove that this partition is unique.

Defining connectedness for pairs of nodes in a directed graph is more complex than in an undirected graph, because a directed path may exist through the network from node i to node j, but this does not guarantee that a path from j to i also exists. Consequently, two different definitions of connectedness between two nodes exists, namely *weak* and *strong* connectedness [DMS01]. Two nodes i and j of a directed graph G are said *strongly connected* if there exists a path from i to j and a path from j to i. A directed graph G is said *strongly connected* if all pairs of nodes (i, j) are strongly connected. A *strongly connected component* of G associated to node i is the maximal strongly connected subgraph containing node i, i.e., it is the subgraph which is induced by all nodes which are strongly connected to node i. A *weakly connected component* of G is a component of its *underlying undirected graph* G^u, which is obtained by removing all directions in the edges of G. Two nodes i and j of G are *weakly connected* if they are connected in G^u, and a directed graph G is said to be *weakly connected* if the underlying undirected graph G^u is connected. Hence, the components of a directed graph can be of two different types, namely weakly and strongly connected. It is useful to review also the definitions of components *associated to a node* of a directed graph. We have four different definitions:

1. The *out-component of node* i, denoted as $\mathrm{OUT}(i)$, is the set of nodes j such that there exists a directed path from i to $j, \forall j$.

2. The *in-component of a node* i, denoted as $\mathrm{IN}(i)$, is the set of nodes j such that there exists a directed path from j to $i, \forall j$.

3. The *weakly connected component of a node* i, denoted as $\mathrm{WCC}(i)$, is the set of nodes j such that there exists a path from i to $j, \forall j$ in the underlying undirected graph G^u.

4. The *strongly connected component of a node* i, denoted as $\mathrm{SCC}(i)$, is the set of nodes j such that there exists a directed path from i to j and also a directed path from j to $i, \forall j$.

We have already used the last two concepts for the definitions of weakly and strongly connected components of a directed graph given above. In fact, the property of weakly and strongly connectedness between two nodes is reflexive, symmetric and transitive, i.e., in mathematical terms, it is an *equivalence relation*. Therefore, it is

possible to define weakly and strongly connected components of a graph by means of the weakly and strongly connected components associated to the nodes of the graph: a strongly (weakly) connected component of node is also a strongly (weakly) connected component of the whole graph.

Conversely, the definitions of out-component and in-component of a node are not based on *equivalence relations*. In fact, the symmetry property does not yield: $i \in \mathrm{OUT}(j)$ does not imply $j \in \mathrm{OUT}(i)$. This means that out- and in-components can be associated only to nodes, and cannot be directly extended to the entire graph. In practice, we cannot partition a graph into a disjoint set of in- or out-components, while it is possible to identify a partition of a static graph into a disjoint set of weakly or strongly connected components. However, the in- and out-components of the nodes of a graph can be used to define the strongly connected components of the graph. From the above definitions, we observe that $i \in \mathrm{OUT}(j)$ if and only if $j \in \mathrm{IN}(i)$. Furthermore, we notice that i and j are strongly connected if and only if $j \in \mathrm{OUT}(i)$, and at the same time $i \in \mathrm{OUT}(j)$. Or equivalently, if and only if $j \in \mathrm{OUT}(i)$ and $j \in \mathrm{IN}(i)$. Therefore the strongly connected component of node i is the intersection of $\mathrm{IN}(i)$ and $\mathrm{OUT}(i)$.

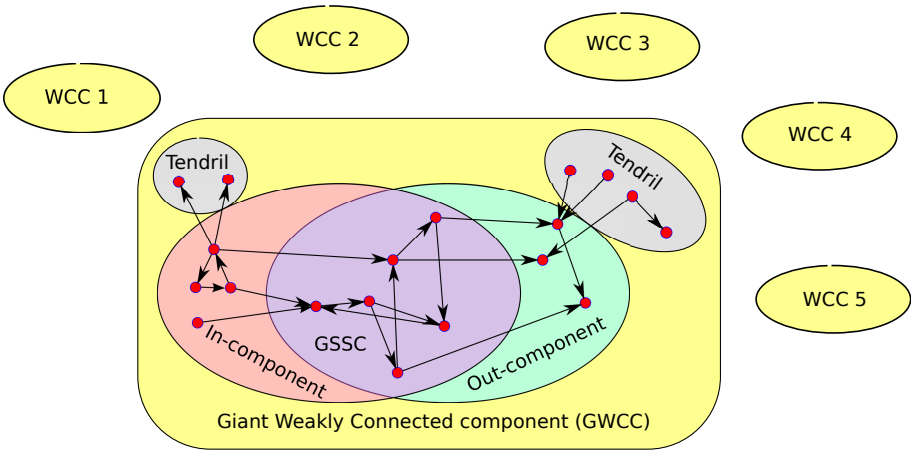


Figure 2.2: A directed graph can be partitioned into a set of disjoint weakly connected components (in yellow). Furthermore, each of these components has a rich internal structure, as shown for the GWCC.

We are now ready to describe in detail the rich interplay between the various concepts of connectedness in a directed static graph. In the most general case, as shown in

Figure 2.2, a directed graph can be decomposed in a set of disjoint weakly connected components. In a large graph, one component will be larger than all the others and it will be called *giant weakly connected component* GWCC. If we treat each link in the GWCC as bidirectional, then every node in the GWCC is reachable from every other node in the GWCC. As shown in Figure 2.2, the GWCC contains the *giant strongly connected component* GSCC, consisting of all sites reachable from each other following directed links. All the sites reachable from the GSCC are referred to as the *giant OUT component*, and the sites from which the GSCC is reachable are referred to as the *giant IN component*. The GSCC is the intersection of the giant IN- and OUT-components. All sites in the GWCC, but not in the IN- and OUT-components, are referred to as "tendrils".

2.3 Conclusions

We have presented a range of analysis related to information dissemination in networks from the textbook definition of shortest paths to the study of small world phenomena. An important point which should be highlighted, is that the studies presented here are all derived from this simple concept of shortest paths: small world measures the relationship between shortest paths lengths and clustering; closeness and betweenness are based on shortest paths; connected components are defined in terms of shortest path lengths. Indeed this observation leads us into the next chapter where we take the logical step of defining *temporal* shortest paths, which form the foundation of subsequent temporal metrics.

3

Temporal Graphs and Distance Metrics

Introduction

In this chapter, we tie together key observations that were concluded from the previous two chapters. Firstly, in Section 1.1 we categorised four important pieces of temporal information that should be captured in a temporal graph model and metrics, namely timestamps, time-order, frequency and duration. Secondly, as we have seen in Section 2.1, real networks exhibit temporal information but many studies have simplified the analysis of these networks by ignoring temporal information through the aggregation of edges over time. Thirdly, through our discussion of existing static network analysis (Section 2.3), we have seen that important graph metrics are founded on the simple concept of shortest paths.

Chapter Outline

In Section 3.1 we present a model that captures these temporal properties of real networks, which we refer to as *temporal graphs*; such temporal graphs can be thought

of as a series of snapshots of the network topology over time. This discretisation fits with many of the empirical datasets discussed in Section 1.1, such as annual friendship questionnaires, monthly snapshots of an OSN, constant scanning rate of Bluetooth sighting, however this does not preclude modelling continuous time, which can be approximated by decreasing the window size to an appropriately fine granularity. This also raises the question of selecting an appropriate window size for a given dataset; although a given dataset may have been collected at coarse time interval, other datasets will contain timestamps on a finer time scale; we cover this in Section 3.4.2.5. Another noteworthy point is that there are several equivalent temporal graph representations that could be used, for example time-stamped edges [KKK02] and multi-slice graphs [MRM$^+$10]; though these are equivalent, in our studies we find that the temporal graph model is the most intuitive and better suited for visually comprehending topological changes; this shall be considered further in Section 3.3.3.1. In Section 3.2, we define fundamental concepts of temporal paths and temporal shortest path lengths which will form the foundation for measures based on the concept of shortest paths in later chapters.

Following these definition we provide two studies using empirical networks using this model and metrics. In Section 3.4.2 we compute these temporal distance metrics in the analysis of shortest paths in online social networks compared to the static counterpart. In Section 3.4.3 we investigate the relationship between communication efficiency versus the evolution speed of a time-varying network. Finally, we draw conclusion in Section 3.5.

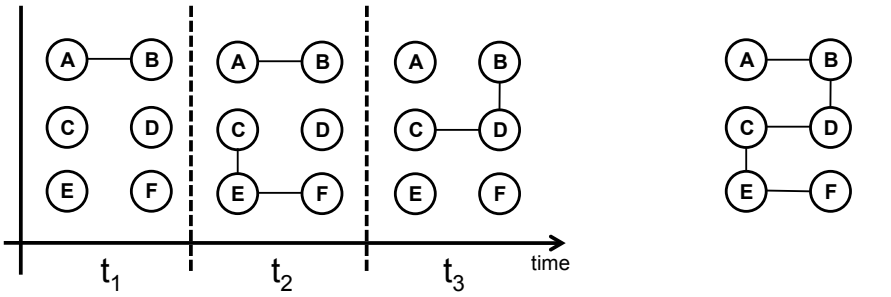
3.1 Temporal Graphs

Consider the sequence of interaction in Table 3.1; these interactions could represent meetings between friends, activity between two cortical regions of the brain, or traffic flow in a computer network. From this we can construct the example temporal graph (Figure 3.1(a)) and corresponding static aggregated graph (Figure 3.1(b)), where interactions between a pair of nodes defines an edge or, equivalently, generated from the union of all edges in the temporal graph.

To give an intuition as to the benefits of using a temporal graph over the static counterpart, consider the path from node A to node F; using the static graph there

N1,N2	Timestamp	Duration
A,B	1	2
C,E	2	1
E,F	2	1
B,D	3	1
C,D	3	1

Table 3.1: Example interaction sequence between 6 nodes. The first column defines a pair of nodes interacting, the second column defines the time of their interaction and the third column defines the duration of the interaction.



(a) Temporal Graph (b) Static Graph

Figure 3.1: Example directed Temporal Graph with three time windows and six nodes, generated from interactions in Table 3.1

seems to be a path from A to F via (A,B,D,C,E,F), however, when we take into account the time information in the temporal graph, there is in fact no path that satisfies his route. This is due to the *time-order*; the interaction between the sub-path (B,D,C) and (C,E,F) occur in the wrong time order to facilitate the path. We shall quantify this using empirical traces later in Section 3.4.2, but first let us first formally define these concepts of temporal graphs and temporal path metrics.

Definition 1 (Temporal Graph) *Given a real-world network interaction dataset starting at t_{min} and ending at t_{max}, the (undirected) temporal graph $\mathcal{G}^w(t_{min}, t_{max})$ is defined as an ordered sequence of undirected graphs (G_0, $G_2, \ldots, G_{\tau-1}$) where:*

- *$G_t = (V_t, E_t)$ is a 2-tuple consisting of a set of nodes V_t and edges E_t in the window t;*

- *there exists a link between node i and node j in E_t if there is some link in the real network between i and j during the time interval $[(t_{min} + (w \times t)), (t_{min} + (w \times (t + 1)))]$;*
- $\tau - 1 = ((t_{max} - t_{min})/w) = |\mathcal{G}^w(t_{min}, t_{max})|$ *is the number of graphs in the sequence;*
- w *is the duration of each time window expressed in some time units (e.g.,seconds or hours); and*
- $|E| = \sum_{t=0}^{\tau-1} |E_t|$ *as the total number of edges across all windows in the temporal graph.*

This definition can be trivially extended to the case of a *directed* temporal graph by means of a sequence of *directed* graphs, where there exists a link *from i to j* in E_t if there is a contact *from i to j* during the time interval $[(t_{min} + (w \times t)), (t_{min} + (w \times (t + 1)))]$.

3.1.1 Simplifying Assumption

Firstly, we shall only consider unweighted graphs since the datasets employed in this thesis contain only binary contact information and although weighted graphs can capture some sense of duration along a path in a static graph, the dependencies (i.e. time order) in fluctuations of links (either binary of continuous) is still not captured in static weighted graphs. However, weighted temporal graphs would be a good candidate for future work. Secondly, we shall concentrate on systems where the number of nodes remains constant (i.e., there are no birth or deaths of nodes) but where there is fluctuation of the edges between nodes (which represent some contact, message being sent, traffic etc.). This is reasonable since the networks discussed in Chapter 1 all exhibit a stable value of N over a short time-scale; however, this assumption does not prevent the analysis of networks where the number of nodes grows. For example, we can model the temporal graph with the maximum number of expected nodes N across all time windows or the temporal metrics described in the next section could be normalised by the number of nodes in each time windows. The effects of a non-constant N on temporal metrics is not explored in this thesis. For

simplicity we shall refer to the set of nodes in a temporal graph as $V = V_t, \forall t \in [0, \tau)$ and $N = |V|$.

3.2 Temporal Metrics

We now turn our attention to the definition of metrics to measure temporal distance, clustering and evolution speed.

3.2.1 Temporal paths and shortest path length

As we have highlighted in the previous chapter (Section 2.3), fundamental to the study of information dissemination in networks is the concepts of paths and path lengths. We have also seen that shortest paths have been applied to many different applications for example measuring the indirect number of links between friends-of-friends [Mil67]; finding the fastest route to send an electronic message through the Internet; or planning the quickest route to drive to work. However, shortest path length on static graphs returns the *number of hops* from a source node to destination node; this does not retain temporal information and hence cannot capture the true duration or speed of dissemination. Instead, we now formalise a metric fundamental to this thesis that we call the *shortest temporal path length* which gives an indication of the speed of message delivery from a source to destination. Before we can formalise this metric we first define the concepts of temporal paths. Following this, we then run through an example calculation of the temporal shortest path length and then define the algorithm that is used to compute the temporal shortest path length and temporal shortest paths.

Definition 2 (Temporal Path) *A temporal path,* $p_{ij}^{h} = (n_0^{W_0}, \ldots, n_\eta^{W_\eta})$*, starting at node* $i = n_0$ *and finishing at node* $j = n_\eta$ *can be defined over* $\mathcal{G}^w(t_{min}, t_{max})$ *as a sequence of* η *hops via a distinct node* $n_a^{W_a}$ *at time window* W_a*, where node* n_a *is passed a message if and only if there is an edge between* n_{a-1} *and* n_a *at time window* $W_{a-1} \leq W_a$*; and* $0 \leq W_a < \tau$*.*

To allow generality in the temporal graph model and distance metrics we also introduce the *horizon* parameter h, which is the maximum number of hops through

which a message is replicated within the same window. For example, returning to Figure 3.1(a), calculating the temporal shortest path from node A to C with horizon $h = 2$ there is a temporal sub-path (A, B) at window 1 and (B, D, C) at window 3. If the horizon $h = 1$, then this temporal shortest path does not exist since in window 3, only one hop is allowed and only node D can be reached. The horizon parameter can be interpreted as the *speed* that a message travels through the network (or the speed of transfer over a link) and is directly related to the window w size used to model the network. Throughout this thesis we assume that the typical time for a message to pass from a node to one of its neighbours is of the same order as the typical time at which the graph changes (i.e. h=1). The relationship between these two parameters shall be investigated in Section 3.4.2.5 and 3.4.3.4. For clarity, subsequent definitions will implicitly include the horizon parameter h.

We call Q_{ij} the set of all temporal paths between nodes i and j. If a temporal path from i to j does not exist, i.e., $Q_{ij} = \emptyset$, we say that (i, j) is a *temporally disconnected node pair*, and we set the distance $l_{ij} = \infty$.

Using the function $D(p_{ij}) = (w \times W_\eta)$ which returns the delivery time (at window W_η) for the given path relative to t_{min}, the *shortest temporal path length* is defined as:

$$d_{ij} = min(D(p_{ij})), \forall p_{ij} \in Q_{ij}. \tag{3.1}$$

Since the shortest temporal path may not be unique, we define the set S_{ij} of all *shortest temporal paths* from node i to j as:

$$S_{ij} = \{p_{ij} \in Q_{ij} : (D(p_{ij}) = d_{ij})\}. \tag{3.2}$$

We can also define temporal shortest path length in terms of *efficiency* [LM01]; the *temporal efficiency* E_{ij} between nodes i and j as:

$$E_{ij} = \frac{1}{d_{ij} + 1} \tag{3.3}$$

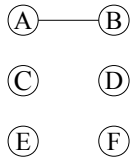
3.2.2 Example calculation of d_{ij}

To give an intuition as to how the temporal shortest path length, d_{ij}, is calculated, we first give an example calculation between nodes in the network in the temporal

graph from Figure 3.1(a). In the next section we describe the algorithm for both d_{ij} and S_{ij}.

We assume global knowledge of the temporal graph and two global lists, D and R, indexed by node identifier are maintained. D keeps track of the number of temporal hops to reach a node and R keeps track of nodes that are reached. We initialise the value of every nodes of D to 1 and R to *False*. Starting with the first time window, we check that the source node i has been sighted. If so, we perform a depth first search (DFS) to see if any unreached nodes have a path to a node that was reached in a previous window. The maximum depth of DFS is dictated by the horizon h and if there is more than one path, we choose the shortest. If a node j is reachable then we set $R[j] = True$ otherwise we increment the distance $D[j]$. If the source node i is not reachable then we increment all $D[j]$ since we cannot establish a transitively connected path from the source. We then repeat this for the next window.

Time Window 1: Starting with the first window we focus on the reachability from a source node A. Figure 3.2 shows the snapshot of the graph topology at $t = 1$ and the upper table shows the state of lists R and D after the initialisation phase. We first check if we can see the source node A. Since node A appears in this first window, $R[A]$ is set to *True*. We then iterate through every other node in the window to check for reachability. Since there is a path between A and B and also since A was reached already we update $R[B]$ to *True*. However for node C, there are no edges to any other nodes so we increment the distance $D[C]$. The same applies to nodes D, E and F and the lower table shows the state of D and R after processing the first window.



Initialization

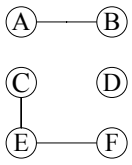
	A	B	C	D	E	F
D	1	1	1	1	1	1
R	F	F	F	F	F	F

Variables

	A	B	C	D	E	F
D	1	1	2	2	2	2
R	T	T	F	F	F	F

Figure 3.2: Distance and Reachability of Window 1.

Time Window 2: The second window is shown in Figure 3.3. We iterate through all unreached nodes C, D, E and F and perform DFS to see if they can be reached via already reached nodes i.e. A or B. As we can see, there are edges amongst the unreached nodes, however, none are with A or B so again the distance D for nodes C, D, E and F are incremented. The state of D and R are shown in Figure 3.3 after processing the second window.

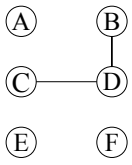


Variables

	A	B	C	D	E	F
D	1	1	2	2	2	2
R	T	T	F	F	F	F

Figure 3.3: Distance and Reachability of Window 2.

Time Window 3: In the third and final window starting from node C, we check if there is a path to a previously reached node. In this case performing DFS gives us two nodes we can reach D and B in the current window, but only node B has been reached in a previous window. We only care that there is a valid path not the number of hops within the current window, so we set $R[C] = True$. Since the value of $D[C]$ is 3 and $R[C]$ is *True*, we now know that a message from node A can reach node C in 3 time windows. Therefore the temporal distance $d_{AC} = 3$. For node D there is a path to node C and node B, but since only node B was reached in a previous window we use this path and set $R[D]$ to $True$. For nodes E and F, a message from node A has still not arrived and so the final state shown in Figure 3.4 reflects this. For all values of R that are *False* we can treat the distance D as ∞.



Final

	A	B	C	D	E	F
D	1	1	3	3	4	4
R	T	T	T	T	F	F

Figure 3.4: Distance and Reachability of Window 3.

Result: Table 3.2 shows the temporal path length calculated for every node pair, where the diagonal describes when a node was first seen by another node. As we

mentioned earlier paths in static undirected graphs are assumed symmetric, for example in Figure 3.1(b) there is a path between nodes A to C and vice versa. However, in Table 3.2 this is not the case due to the ordering of the edges and this can be verified visually in Figure 3.1(a).

	A	B	C	D	E	F
A	1	1	3	3	∞	∞
B	1	1	3	3	∞	∞
C	∞	3	2	3	2	2
D	∞	3	3	3	∞	∞
E	∞	3	2	3	2	2
F	∞	3	2	3	2	2

Table 3.2: Temporal path length for all nodes.

3.2.3 Algorithm & Complexity

We now describe the algorithms used in computing the temporal shortest path length, d_{ij}, and temporal shortest paths, S_{ij}. As illustrated in the previous example for the temporal shortest path length, d_{ij}, we essentially compute the reachable nodes within each time window using a standard static shortest path length algorithm with maximum depth h before moving on to the next window. For this reason, we first list standard algorithms for calculating static shortest path lengths and paths on a static model will be listed based on breadth first search (BFS) to control the search depth, followed by algorithms to calculate d_{ij} and S_{ij} on the temporal model are listed.

The listings are in terms of *single-source* to all destination nodes, since there is negligible additional complexity between single destination and multiple destinations from a single source node. This is due to the nature of BFS that has to maintain all reached nodes. In addition, for brevity we shall concentrate on undirected graphs.

3.2.3.1 Shortest Path Length on Static Models

On a static graph model, to calculate the shortest path lengths from a source node i and all destination nodes $j \in V$, we can use a modified breadth first search (BFS)

[Sed88]. Since a breadth first search visits the nearest neighbours first and then for each of these neighbours visits their direct neighbours, we are in effect incrementing the search perimeter (or number of hops). It is straightforward then, to see that once we have reached the destination node then the shortest path from i has been found since we could not have reached it in a smaller perimeter. More formally, the pseudo code for `BFSStaticShortestPathLengths()` is presented in Algorithm 1.

Algorithm 1: `BFSStaticShortestPathLengths()`

Input: Graph G, Source node i, Maximum Depth m
Output: List of shortest path lengths to all other nodes n in V

```
1  begin
2      Initialise list of node distances, d(n) ← ∞, ∀n ∈ V ;
3      Initialise empty queue Q ;
4      Add source node i to Q ;
5      d(i) ← 0 ;
6      while Q is not empty do
7          k ← Head(Q) ;
8          Remove k from Q ;
9          if d(k) = m then
10             break ;                          /* Maximum depth reached */
11         Find all neighbour nodes F_k where d(n) > d(k), ∀n ∈ F_k ;
12         Add all neighbour nodes n ∈ F_k to the end of Q iff n is not already in Q ;
13         Set all neighbour nodes n ∈ F_k distance to d(n) ← d(k) + 1 ;
14     return d;
```

The runtime complexity is $O(|V| + |E|)$ in the worst case since all nodes could potentially be added to the queue (Line 12) and potentially all edges could also be traversed (Line 11).

3.2.3.2 Shortest Paths on Static Models

Calculating shortest paths is more complicated and requires the maintenance of predecessor nodes which track neighbouring nodes that were used to reach a node n. The full set of shortest paths from the source to destination can be reconstructed using this data structure. It is important to note that many textbook

definitions [Sed88, CLRS01] of the Dijkstra or Floyd-Warshall algorithm for finding shortest paths, maintain only a *single* predecessor for each node and hence only a *single* shortest path can be reconstructed; we are interested in *all* shortest paths which is important, for example, in the calculation of betweenness centrality which relies on this knowledge. However, this comes with a caveat of increased time complexity in path reconstruction. More formally, the pseudo code for `BFSStaticShortestPaths()` is presented in Algorithm 2; this is similar to the calculation of path lengths but with the addition of Lines 3 and 15 to maintain the set of predecessors of a node. To reconstruct the full path we can use `ReconstructPaths()` (Algorithm 4).

Algorithm 2: `BFSStaticShortestPaths()`

Input: Graph G, Source node i, Maximum Depth m
Output: Predecessors of nodes on shortest path to source

```
1  begin
2      Initialise list of node distances, d(n) ← ∞, ∀n ∈ V ;
3      Initialise empty list of sets Pred(i) ← ∅, ∀n ∈ V;
4      Initialise empty queue Q ;
5      Add source node i to Q ;
6      d(i) ← 0 ;
7      while Q is not empty do
8          k ← Head(Q) ;
9          Remove k from Q ;
10         if d(k) = m then
               //Maximum depth reached
11             break ;
12         Find all neighbour nodes F_k where d(n) > d(k), ∀n ∈ F_k ;
13         Add all neighbour nodes n ∈ F_k to the end of Q iff n is not already in Q ;
14         Set all neighbour nodes n ∈ F_k distance to d(n) ← d(k) + 1 ;
15         Update predecessor of neighbours Pred(n) ← Pred(n) ∪ {k}, ∀n ∈ F_k ;
16     return Pred;
```

The complexity of calculating shortest paths is the same as calculating shortest path lengths, $O(|V| + |E|)$, since the maintenance of predecessors can be performed in linear time.

Algorithm 3: `ReconstructPaths()`

Input: Source i, List of sets of predecessors $Pred$

Output: List of paths to given all destination nodes

```
1  begin
2      AllPaths = ∅ ;
3      foreach Destination node n ∈ V do
4          AllPaths(n) ←ReconstructPathSingleDest(i, n, Pred, ∅) ;
5      return AllPaths ;
```

Algorithm 4: `ReconstructPathSingleDest()`

Input: Source i, Destination j, List of sets of predecessors $Pred$, *Optional path* (where default= $\emptyset$)

Output: List of paths to given destination node

```
1  begin
2      path= ConcatenateLists([j], path) ;
       //Terminating conditions
3      if i = j then
4          return [path] ;
5      if Pred(j) = ∅ then
6          return ∅ ;
7      paths = ∅ ;                                  //Collect up different paths
8      foreach node in Pred(j) do
9          if node not in path then
10             newpaths = RecontructPaths(i,node,Pred,path)
               //Each pred might have multiple pred
11             foreach newpath in newpaths do
12                 paths ← paths ∪ newpath ;              //Seperate each path
13     return Paths ;
```

The predecessor data structure can be thought of as a directed, acyclic graph. Enumerating all possible paths on this structure takes exponential time $O(2^N)$. To convince ourselves of this, consider the simple case: a graph with N nodes, where node i is connected with every node $k > i$; enumerating all paths from node 1 to N, there are exactly 2^{N-2} paths.

3.2.3.3 Shortest Temporal Path Length

To compute $d_{ij}^h(t_{min}, t_{max})$ our algorithm uses the modified breadth first search (`BFSStaticShortestPathLengths()` in Algorithm 2), giving us the temporal distance from a source node i to all other nodes. The idea is that starting from node i and the first window, we find all reachable nodes with path lengths not exceeding the horizon variable h. We then mark all reached nodes with their temporal distance set to the current window. If the destination node has not been reached, then we repeat the same procedure in the next window but *for all nodes reached in the previous window* as the source node. We list the pseudo code (Algorithm 5) to find distances from a single source node to all destination nodes.

For all nodes j where $R[j] = False$ then the temporal distance $d_{ij}^h = \infty$, otherwise if $R[j] = True$ then $d_{ij}^h = D[j]$. To find all the temporal distance for all node pairs, we repeat for all source nodes. The runtime complexity is $O(\tau.(N + |E|))$ in the worst case when there is at least one destination node unreachable and we need to check all windows τ.

3.2.3.4 Shortest Temporal Path

The algorithm to find the single source shortest temporal paths extends the algorithm to find shortest temporal path *lengths*, by maintaining a list of predecessor for each node, records the neighbour(s) that a node was reached from. A reached node k can only have multiple predecessors if each of the predecessors reached j in the same window i.e. the first window that j was reached. Again, by keeping track of pointers to predecessors we are effectively maintaining a tree structure and to recall paths, we traverse the predecessor pointers from any reached node back to the source node i.

Algorithm 5: `TemporalShortestPathLength()`

Input: TemporalGraph $\mathcal{G}$, Source node i, Horizon h

Output: Shortest temporal path length to all other nodes n in $\mathcal{G}$

```
1  begin
2      Reset all node distances D(n) ← ∞, ∀n ∈ V ;
3      Reset all node reachability R(n) ← False, ∀n ∈ V ;
4      Set source node as reached, R(i) ← True ;
5      Set current window index w ← 0 ;
6      foreach window G ∈ 𝒢 do
7          foreach node n that was reached in a previous window (n ∈ V,
           where R(n) = True and D(n) < w) do
               //Find previously unreachable nodes
               //which can now be reached
8              K ← BFSStaticShortestPathLengths(G,n,h) ;
9              foreach k ∈ K do
10                 if previously unreached node, R(k) = False then
11                     Set R(k) ← True ;
12                     Set D(k) ← w ;
13         w ← w + 1 ;
14     return D;
```

We should also note that since we are interested in the window that a node is reached, the multitude of temporal paths may have different hop lengths; for this reason the predecessor data structure needs to also maintain the sub-path within the current window with a maximum hop count equal to the horizon. A hop also needs to be recorded as a 2-tuple (node,window). More formally, the pseudo code for `TemporalShortestPaths()` is presented in Algorithm 6.

The full set of temporal paths can be reconstructed using a slightly modified version of `ReconstructPaths()` (Algorithm 4) used for static shortest paths. Since the predecessor data structure maintains a 2-tuple (node,window), instead of a single node for each hop of the path, each hop is now a 2-tuple (node,window). We can simply modify `ReconstructPaths()` so that access to each element of $Pred(n)$ for a node n returns *pred* from the 2-tuple (*pred*,*window*).

Algorithm 6: `TemporalShortestPaths()`

Input: TemporalGraph $\mathcal{G}$, Source node i, Horizon h

Output: Predecessors of nodes on shortest temporal path to source

```
1  begin
2      Reset all node distances D(n) ← ∞, ∀n ∈ V ;
3      Reset all node reachability R(n) ← False, ∀n ∈ V ;
4      Reset all node predecessors P(n) ← ∅, ∀n ∈ V ;
5      Set source node as reached, R(i) ← True ;
6      Set current window index w ← 0 ;
7      foreach window G ∈ 𝒢 do
8          foreach node n that was reached in a previous window (n ∈ R,
           where R(n) = True and D(n) < w) do
               //Find previously unreachable nodes
               //which can now be reached
9              K ← BFSStaticShortestPathLengths(G,n,h) ;
10             foreach k ∈ K do
11                 if previously unreached node (R(k) = False) then
12                     Set R(k) ← True ;
13                     Set D(k) ← w ;
14                     Add predecessor P(r) ← P(r) ∪ {(k, w)} ;
15         w ← w + 1 ;
16     return Pred ;
```

The time complexity of constructing predecessors is the same as temporal path lengths ($O((\tau.(N + |E|))$, however, the reconstruction of temporal paths is dominant since we effectively have τN nodes as input to `ReconstructPaths()` and hence the complexity is $O(2^{\tau N})$. In practise, we have not found that the computation time to be prohibitively slow, especially since temporal paths between each pair of nodes can be computed in parallel. Future work could investigate more efficient implementations when applied to specific applications, for example calculating betweenness centrality requires the count of all shortest paths from a source to destination node (rather than enumerating all paths), which can be performed in polynomial time [Bra01].

3.2.4 Temporal distance is a quasi-metric

We should note that temporal distance is not a *metric*, in a strict mathematical sense, since it does not satisfy the symmetry condition unless the temporal graph is a *temporally strongly connected component* (this shall be discussed in more detail in Chapter 6); this is due to the *temporal* direction of a path. When the symmetry condition is broken, such metrics are more accurately referred to as *quasi-metrics* [Ste95]. This also applies to paths in directed static graphs since paths between pairs of nodes are not guaranteed to be symmetric. Further, for static undirected graphs distance is only embedded in metric space if the graph is connected, however, in the research literature, the terminology of *metrics* is still commonly applied to paths and distance on graphs, regardless. To avoid confusion we shall also refer to temporal distance as a metric throughout this dissertation.

3.2.5 Characteristic Temporal Path Length

From these temporal distance measures, we can define the *characteristic* or *average* temporal path length L, similar to that defined by Watts & Strogatz [WS98]:

$$L = \frac{1}{N(N-1)} \sum_{ij} d_{ij} \tag{3.4}$$

We assume that information expires after a certain time period so that if two nodes i and j are temporally disconnected then we shall set $d_{ij} = w\tau$ i.e., the maximum time for delivery in the temporal graph.

Alternatively, in order to avoid the potential divergence , we can define the *temporal global efficiency* of $\mathcal{G}$ as [LM01]:

$$E = \frac{1}{N(N-1)} \sum_{ij} \frac{1}{d_{ij}} \tag{3.5}$$

Low values of L (high values of E) indicate that the nodes of the graphs can communicate efficiently.

3.2.6 Local Temporal Efficiency

Local temporal metrics capture the dynamics of each node and its neighbours across the whole time space. The generalisation of the local efficiency E_{loc} for temporal graphs we propose is as follows.

We first define $\mathcal{N}_i(t_{min}, t_{max})$ as the set of all first-hop neighbours seen by node i at least once in the time interval $[t_{min}, t_{max}]$. We indicate as $k_i(t_{min}, t_{max})$ the number of nodes in the set $\mathcal{N}_i(t_{min}, t_{max})$. We then consider the sequence of subgraphs $G_t^{\mathcal{N}_i(t_{min},t_{max})}$, $t = t_{min}, t_{min+w}, \ldots, t_{max}$ where each $G^{\mathcal{N}_i(t_{min},t_{max})}$ is the neighbour subgraph of node i, considering only the nodes in $\mathcal{N}_i(t_{min}, t_{max})$ and retaining the edges from $G_{t_{min}}$.

We can define the local efficiency of node i in the time window $[t_{min}, t_{max}]$ as:

$$E_{loc_i}(t_{min}, t_{max}) = E_T\{G_t^{\mathcal{N}_i(t_{min},t_{max})} \;\; t \in [t_{min}, t_{max}]\} \tag{3.6}$$

that is the efficiency of the time varying graph of the first neighbours of i in the time window $[t_{min}, t_{max}]$, i.e. the shortest-path for time-varying graphs are computed for $G_t^{\mathcal{N}_i(t_{min},t_{max})}$, $t \in [t_{min}, t_{max}]$. Note that by definition, for E_{loc} the horizon is always 1 since we are only considering the direct neighbours of node i.

3.2.7 Temporal Correlation Coefficient

In a temporal graph $\mathcal{G}$, what matters is not only the probability distribution $P(G)$ over the graphs in $\mathcal{G}$, but also how the graphs are ordered in time. By counting the number of times a given graph G appears in the time sequence, we can construct $P(G)$. To fully describe time-varying graphs we also need to know how graphs are correlated in time. For instance we need to know the conditional probabilities $P(G_t|G_{t-1})$ of observing graph G_t after graph G_{t-1} (more in general, the probabilities $P(G_t|G_1, G_2, \ldots, G_{t-1})$ of observing graph G_t after the sequence $G_1, G_2, \ldots, G_{t-1}$). In most cases, the contacts between the same node pair in time-varying systems tend to be clustered in time, i.e. they show persistence over time [Hol05]. For instance, people tend to engage in relations for continuous intervals of time. Hence, a given link has a higher probability to appear in graph G_t if it was already present in graph G_{t-1}. To quantify this effect, Clauset & Eagle defined a measure to compare

two given graphs which are adjacent in time which they named the *adjacency correlation* [CE07]; by averaging over all possible adjacent time windows in $\mathcal{G}$ we can define the average topological overlap of the neighbour set of a node between two successive graphs in the sequence C:

$$C = \frac{\sum_i C_i}{N} \quad C_i = \frac{1}{T-1} \sum_{t=1}^{T-1} \frac{\sum_j a_{ij}(t) a_{ij}(t+1)}{\sqrt{[\sum_j a_{ij}(t)][\sum_j a_{ij}(t+1)]}} \tag{3.7}$$

We name this metric the *temporal-correlation coefficient* of $\mathcal{G}$. The value of C is in the range [0,1]. In particular, if all graphs in the sequence are equal, we have $C = 1$.

3.3 Literature Review

3.3.1 Introduction

Several surveys have recently appeared in computer science circles on time-varying graphs [HS11, CFQS10, Wu10]; we extend our review to include the multitude of other disciplines which have also considered time information in networks. Indeed, within different research circles, temporal graphs may also be known as *longitudinal* (social sciences), *time-varying* (physics) or *dynamic* (computer science) graphs (or networks).

3.3.2 Related Work

In 1958, Ford and Fulkerson [FF58] first considered maximal flows in a network where edges are labelled with traversal time. Cooke et. al. [CH66] then provided an optimal algorithm to solve the problem using a modified Bellman shortest path algorithm. However, this representation is different from our model in that they still fundamentally assume that edges are available across time but that their capacity may be indirectly full based on the flow of traffic from competing paths, whereas our model explicitly represents the fluctuations of edge availability based on some external factor such as people moving away or failure of a node. It does, however, introduce the notion of time into graph theory with congestion analysis indirectly

cause time ordering. This analysis was then naturally later applied to transport and logistical planning by Halpern [HP74, Hal77], however, was not analysed on real networks. Lacking insight into real time-varying networks, assumptions on complete connectivity in graphs were made, which do not stand in real networks.

More recently, Kempe & Kleinberg [KKK02] introduced such analysis to the field of computer science. The Kempe temporal network model labels edges with a time order as opposed to a duration of traversal and hence can be more accurately named as *time-stamped graphs*. This can also be interpreted as the state of the network at a certain time. The goal was to prove that the maximum flow, minimum cut theorem would still holds in time respecting paths and they showed the cases when Menger's theorem would hold when considering such time order. The focus of this study was not on empirical network datasets, but rather from a graph theoretic perspective.

Moody [Moo02] again employs a time-stamped graph to investigate the difference in reachability compared to static aggregated graphs. To compare the differences, the author proposes a *reachability graph* where a static graph A is generated from the time-stamped graph B and there is an edge between a pair of nodes in A if there is a time-respecting path in time-stamped graph B. Using a single empirical dataset of sexual activity at a high school, the main result was that a static graph overestimate reachability which mirrors our findings in this thesis. However, Moody uses a measure of available paths as opposed to shortest paths, which is the main focus of this thesis. We also extend this observation by measuring the actual duration to deliver messages, not just connectedness of time-respecting graphs. The author also alludes to possible future work on temporal extensions to centrality but does not formalise this concept in his study; in the next chapter, we shall provide the definition and analysis of such techniques. In addition, there are a couple of problems with the definition of the reachability graph. Firstly, for a real process, where there may be transitive, multi-hop messages, the reachability graph does not capture the time order of transitive hops. Secondly, there is no notion of reciprocation between pairs of nodes and hence does not capture the true reachability of nodes in a network. In Chapter 6 we address these issues by introducing the concept of an *affine graph*.

Within computer science, time information is inherently part of analysing the delay and data delivery in delay tolerant networks (DTN) [JFP04]. The field also proposed routing algorithm for delay tolerant networks such as Epidemic routing [VB00] which

uses an opportunistic approach to pass messages on via every possible contact at a future time, however, the goal of data dissemination is different from the structural analysis using complex network techniques that we propose in this thesis.

Hui et. al [HCY08] propose a message delivery scheme in pocket switched networks called *Bubble* which uses the most important nodes both globally and within communities to decide on the next hop. They propose an algorithm to identify the most central nodes ($RANK$) using the number of shortest delay paths that pass through a node, however this does not take into account the *fraction* of alternative paths and also they present a strong correlation between such central nodes with degree and so favour this since it i suited for a decentralised algorithm. We are interested in extending the analytical evaluation of different types of centrality, namely betweenness taking into account alternative paths and closeness to find nodes that can propagate messages quickest, as these are suited for different processes. This is studied in the next chapter.

An analysis of different interpretations of temporal shortest paths was performed by Ferreira et. al. [XFJ03, Fer04, FGM07]. Using the same temporal graph model that we utilise in this dissertation, they introduce three variations of the shortest path: The *shortest* path has the minimum number of time ordered hops or transitive exchanges between two nodes. The *fastest* path has a subset of the set of shortest paths that also arrives at the destination the earliest. The *foremost* path is the latest or most up to date path to reach a destination node. The goal of this research group is on the communication between satellites that exhibit known periodic orbits, though earlier work concentrated on random waypoint models. Fundamentally, the focus of their study differs from this thesis in that we are interested in more general properties of a range of real life complex network which exhibit time information. Their work culminated in a routing protocol, which took into account availability of future links [FGM07], however there is a big assumption that every node knows the future state of the network. In Chapter 5 we present techniques to predict future important nodes in mobile phone networks based on regularity of human interactions.

Following from this, the first real attempt at handling temporally disconnected node pairs and analysing real time-ordered networks was by Holme [Hol05]. Holme uses the same time-stamped graph model of Kempe & Kleinberg and analyses the equiv-

alent variations of shortest path as introduced by Ferreira et. al. They were also the first attempt to analyse real networks traces however due to the available computational hardware only random samples from the traces were used. Also although the authors recognise that disconnected node pairs are important they represent the average reachability time using two separate metrics: one for the average time over connected node pairs and a second ratio of disconnected node pairs. Alongside each network analysed they also present other metrics such as number of nodes, average degree and time span of network. The metrics defined in this thesis extends this by incorporating disconnected node pairs and normalising by time span and number of nodes to produce a single, succinct value for a given network. Also, the focus of this study was the reachability of real time ordered networks, but was not compared with static network representations; in this thesis we seek to quantify the difference between temporal and static analysis.

Kossinets et. al. [KKW08], analyse information dissemination processes focussing on identifying the diffusion of the most recent piece of information about a certain topic in a social network. We instead are interested in measuring the smallest delay path of generic information spreading processes starting from the beginning of a process.

Similarly, Kostakos [Kos09] presented the concept of temporal graphs and an equivalent measure of delivery time between nodes of a temporal graph. However again this provides a skewed indication of the global delay of the information diffusion process since it does not take into account pairs of nodes for which a transitive path does not exist. Also the lack of normalisation over nodes or time do not lend for easy comparison between networks. Again, the author analyses two networks: one email and one Bluetooth. However, the Bluetooth trace is based on a limited number of access point scanning for passing devices as opposed to actual proximity contacts between devices. Based on the proposed application of message delivery, it is hard to make any clear claims on how efficient social networks are for message delivery. This is coupled again with the problem of multiple metrics to represent message delivery of a network.

More recently, there has been some work on incorporating time into social network analysis. The first piece of work that attempted to analyse social networks with temporal information was by Clauset & Eagle [CE07] where the authors used a

temporal model to calculate the average degree and clustering coefficient. However, the metrics are still *static* in the sense that they calculate the metric on each window independently. Our work creates temporal metrics for temporal models that capture the dynamics and dependencies across all windows.

Mathematicians have employed the temporal graph model represented as a series of adjacency matrices in the study of random graph models so that spectral analysis can be utilised [GH09, GH11]. Grindrod & Higham propose a random temporal graph models based on a markovian edge process which captures features of empirically observed networks. For example they propose a *range dependent* probability of edge birth and death between successive windows of temporal graph and, using spectral analysis techniques, find that a short-range dependence is present in neurological networks, between spatially nearby areas of the brain. We take a similar approach in Section 3.4.3 to capture the relationship between evolution speed and dissemination through a combination of a random temporal graph model and empirical data.

Social scientists have looked in the dynamics of friendship networks and their influence on smoking behaviour in 1326 adolescents at 11 Finnish high schools [MSS+10]. Unique to this discipline is the emphasis on semantic information on the participants in addition to the network topology information, gathered by means of questionnaires at several time intervals. However, due to the manual collection techniques only 4 annual topological snapshots are available. This study differs in that we concentrate on automated collection of finer grained temporal interactions but trade off the semantic information. One such dataset, which we use in this thesis, is that of the Enron email that contains both fine grained timestamps and semantic information on the role of each user.

Recent studies have applied dynamic networks to community detection. Mucha et. al. [MRM+10] proposed a *multislice* network, where the extra dimension can be temporal. The idea is to link the same node between time slices and hence can be collapsed into a single static representation. From this, the authors generalise the formalisation of the well studied modularity [NG04] utilised in static network analysis. Williams et. al. [WWA11] demonstrate that a snapshot representation of the network topology over time can be exploited in periodic community detection. Although community detection is an important tool in network analysis and for understanding communication between nodes, this thesis shall focus on metrics

for measuring information dissemination, identifying key nodes and reachability in temporal graphs.

3.3.3 Discussion

Although there has been analysis on time respecting paths and parallel works in community detection, there has been little work on extending such concepts to other fundamental complex and social network metrics such as clustering coefficient, centrality or connected components etc. which also make fundamental assumption on constant time nor is there any thorough analysis on real social networks. There has been little understanding on any real differences between static and temporal analysis on real networks due to the problems associated with temporally disconnected graphs therefore normalisation between traces.

Past work on time-respecting paths has not investigated different starting time points in a time-varying network (for example, temporal distance measurements taken at daily start points) instead only taking a single measurement from the start of the network dataset. This misses important aspect of time information, namely the time dependencies and any periodic behaviour that is apparent in human behaviour [SMML10b, WWA11]. This is also informative in the derivation of centrality prediction techniques in Chapter 5.

Another point we highlight is that past work on discrete models for time-varying networks ignores the issue of window sizes, partly due to the reliance of artificial models that are generated through discrete time steps [FGM07] and partly due to the complexity of handling an additional parameter. In this thesis we have generalised the temporal graph model to take as parameters the window size w and horizon h. We highlight the importance of considering these parameters in Section 3.4.2.5, however, since this is a model intended for application by a wide range of researchers and applications, we can only offer guidelines to the selection of these parameters based on the relevance to the datasets utilised in this thesis, as discussed in Section 3.4.2.

Regarding the metrics themselves, this thesis extends past work by first expanding the set of temporal metrics to include temporal centrality and temporally connected components. We also enhance the standard shortest path length metrics to han-

dle disconnected temporal graphs using a characteristic temporal path length and a temporal efficiency metric that naturally captures paths of infinite length. We then are able to use these single succinct values (a single value to characterise a whole temporal network) metrics to be normalised so we can accurately analyse real social and technological network traces and find that static metrics overestimate the number of available contacts and so underestimates the true shortest path length.

The novel contribution of this thesis is to advance this corpus of research by firstly, extending several well known complex network and social network metrics with temporal information, namely characteristic path length, local efficiency, centrality and connected components, and analysing real network using these new tools; secondly, fully exploring empirical time-varying networks by taking measurements at different start times; and thirdly, applying these temporal metrics to study universal properties of these real networks.

3.3.3.1 Alternative Representations

As we have seen, the consideration of time in all these studies have been motivated by real life network problems (though not all have used empirical datasets for evaluation) and since there is a range of different applications and requirements, several alternative temporal models are defined in addition to the temporal graph model employed in this dissertation. These can be categorised as:

- **static analysis on temporal graph model** providing a simple approximation where static metrics are independently calculated on each temporal snapshot of the network, however, this assumes independence between time whereas we are interested in analysis which takes account of temporal dependencies across time, for example temporal ordering. In other words, we are interested in *temporal analysis on a temporal graph model* as opposed to static analysis on either a static or temporal graph model[1];

- **time-stamped edges**, where edges in a static graph are labelled with the time of occurrence [KKK02]; and

[1]The fourth combination of temporal analysis on a static graph model is not well defined.

- **multi-slice networks** which refers to a recent approach to modelling time in networks was proposed by Mucha et. al. [MRM$^+$10] for identifying communities on a *multislice* network, where the extra dimension can be temporal, multi-scale or multiplex. The idea is to link the same node between time slices and hence can be collapsed into a single static representation.

Static analysis on a temporal graph does not satisfy time-order since there is a lack of time dependency between windows. However, the latter two representation are equivalent to the temporal graph model we employ in this thesis, since they all capture the four temporal properties identified in Section 1.1 of timestamps, duration, frequency and time order of edge interactions and hence the same logic can be followed to derive all our metrics presented in this thesis.

We choose our temporal graph representation for the simple reasons that it is intuitive and more natural for visually analysing the structure of the graph changing over time, just like an animation of the graph topology changing over time. As we have seen, there have been independent studies, performed in parallel which use a corresponding representation [Kos09, WWA11].

This list is by no means exhaustive as it is possible to define other representations suitable for different applications. For example, an alternative approach is to start from the other end of the spectrum and instill temporal information into static graphs with weighted static graphs where weights are link frequencies or duration. In this thesis we focus on static, *unweighted* graphs, though comparisons with weighted static graphs would be an interesting topic for future studies; we note, however, since any analysis on static graphs will inevitably miss time-order, temporal analysis would still be more appropriate in the analysis of real networks.

3.4 Application to Real Networks

3.4.1 Introduction

We now apply these definitions of temporal graphs and temporal distance metrics to real, empirically observed networks that exhibit time information. These case studies aim to, firstly, demonstrate the applicability of these metrics to a range of

	INFOCOM	REALITY	EMAIL
Start	2005-03-13	2004-07-26	2001-07-29
Duration	4 days	280 days	112 Days
Times	day1:6pm-12am	12am-12am	12am-12am
	day2:12am-12am		
	day3:12am-12am		
	day4:12am-5pm		
No. of nodes	41	100	59812
Contacts	avg. 4817	avg. 231	avg. 4000
Granularity	120 secs.	300 secs.	1 sec.

Table 3.3: Experimental Datasets.

real networks and, secondly, to demonstrate that temporal metrics can improve our understanding of dynamic processes on time-varying complex networks.

This section is split into two parts: in Section 3.4.2 we first study the differences between static and temporal shortest paths; in Section 3.4.3 we study the relationship between temporal shortest paths and the evolution speed of a time-varying network and uncover more general properties of time-varying graphs.

3.4.2 Importance of Time in Real Networks

Shortest paths in graphs are a fundamental concept in graph theory and, depending on the interpretation or application, measures the quickest, shortest or fastest path from a source to destination; this is directly related to the study of information dissemination in networks. Naturally these verbs all relate to some concept of time and brings us to the question that we wish to address in this section: *does time really matter and, if so, can we quantify this difference?*

In the introduction to this chapter we already gave a simple example of how time-order plays a part in accurately measuring shortest paths, we now quantify this difference in three real networks datasets, namely Bluetooth traces of people at the 2005 INFOCOM conference [HCS+05], campus Bluetooth traces of students and staff at MIT [EP06], email traces from Kiel University [EMB02] and interactions between a large group of members of a large online social network, namely Facebook

users affiliated with the London network [WBS+09]. We shall refer to these as INFOCOM, REALITY, EMAIL and FACEBOOK, respectively. Table 3.3 describes the characteristics of each set of traces.

The INFOCOM traces were collected in a conference environment using Bluetooth colocation scanning every 2 minutes. With 41 nodes it is quite a small trace but temporally dense in that there are a high number of contacts per day. The REALITY traces were collected at the MIT campus between Bluetooth phones sightings of students, research staff and professors, with Bluetooth scanning every 5 minutes. The EMAIL traces contain email server logs for 56,969 students at Kiel university. Due to the size, we only analyse 7 days of the trace during the Fall semester.

Also, as we identified in the survey of related work (Section 3.3.3), past work only takes measurements at a single time point; in this study to measure different start times of these three networks we take measurements at daily intervals.

3.4.2.1 Parameter Selection

An important choice is that of the window size w. As discussed in relation to related work (Section 3.3.3), we can only provide guidelines to the selection of this parameter. Past work has made simplifying assumptions about the window size through arbitrary selection [GH09, LB07] or ignored this parameter completely due to the use of artificial simulation that also relied on known time steps [FGM07]. Also, recall that the computational complexity of the calculation of temporal path length is $O(\tau.(|V|+|E|))$, where τ is the window count; this means that although we could use a very fine window size, say for example seconds or milliseconds, for large networks which also extend a long observation time unnecessarily small window sizes should be avoided.

Based on these observations and experience of handling several empirical datasets, we provide the following three guidelines. Firstly, the **dataset collection timescale** might provide a clear granularity to use, for example, the Bluetooth scanning rate in the REALITY dataset is five minutes and hence this provides a natural window size; the Gowalla friendship networks seen in Figure 1.2 was collected in monthly intervals and again no finer granularity is available; the same applies to the annual friendship questionnaire employed by Mercken et. al. [MSS+10]. Secondly, the **application timescale** might motivate an appropriate window size, for example *daily*

interactions between people in an office or the *seasonal* effects of predator-prey relationships in food webs [JOBL08]. Thirdly, as the complexity of computing temporal shortest paths and path lengths is defined in terms of the number of windows W, computational power might limit the **tractable window size**. This is a limitation of the temporal graph model but we shall see that any additional time information provides a better approximation to the real answer compared to a static graph (since increasing the window size eventually reduces down to a single window which is the definition of a static graph).

In the case studies presented in this section, the aim is to make the experimental results comparable and hence we fix the window size, w to 5 minutes which is equal to the longer Bluetooth scanning rate of the REALITY trace. The results of varying the window size will then be presented later in Section 3.4.2.5.

3.4.2.2 Importance of Time Order

			Static		Temporal	
Day	N	$\langle k \rangle$	L	Disc	L*	Disc
1	37	25.7	1.291	0	4.090	0.28
2	39	28.3	1.269	0	4.556	0.13
3	38	22.3	1.420	0	4.003	0.19
4	39	21.4	1.444	0	4.705	0.14

Table 3.4: INFOCOM: Static and Temporal Metrics (h=max=N-1, t_{min}=00:00, t_{max}=23:59, w=5 min).

Firstly, as a comparison between the temporal and the static metrics, we show the results calculated for the *INFOCOM* data set. As argued before, paths in static graphs ignore duration of contacts, inter-contact time, recurrent contacts and time ordering of contacts and so overestimate the number of connected node pairs and underestimate the path lengths. Table 3.4 shows calculations for both static and temporal path length, L. As a note, since our temporal L metric presented in Equation 3.4 is in real time, it is hard to compare with static L. Instead, we define separately the concept of a *shortest temporal hop length* which captures the time-respecting paths which minimise the hop count from node i to j as d^*_{ij} =

$min(|p_{ij}|), \forall p_{ij} \in Q_{ij}$ where $|p_{ij}|$ returns the number of hops in the temporal path p_{ij}. To bridge the gap we can then calculate temporal $L^* = \frac{1}{N(N-1)} \sum_{ij} d^*_{ij}$, which is the average shortest node to node hop that obeys time ordering of edges.

As we can see in the static results for Day 1, path length is low. Now looking at the temporal aspects, we have calculated the same metrics but obeying time ordering, duration and recurrence of contacts. The third column, *Disc* shows the ratio of disconnected node pairs. In the case of static graphs, there were no disconnected node pairs. As we can see temporal $L^* \gg$ static L and there are also much more disconnected node pairs due to the observed asymmetry and time ordering of paths. In other words, temporal L give us a better understanding of the network with respect to the temporal dimension since they can provide us an accurate measure of the delay of the information diffusion process that is not possible with traditional static metrics. In particular, since static shortest paths ignore time-order of contacts, it *over-estimate the availability of contacts and therefore under-estimates the true shortest path.*

3.4.2.3 Measuring Dissemination Efficiency

We now calculate temporal L from Equation 3.4 as a real time along with the temporal efficiency E. Each data set is measured individually by day, processed by window size $w = 5$ minutes. The left hand side of Table 3.5 ("Temporal Metrics") shows the temporal metrics calculated for all three datasets. The right hand side of the table ("Reshuffled") will be discussed in the next section.

INFOCOM: First looking at the INFOCOM dataset, recall in Table 3.4 that static L and temporal L^* only told us the average number of hops in a path but gave us no indication of how much time each hop took. Our temporal metrics give us a value that takes account of time and also captures disconnected nodes. From Table 3.5 we can see L for Day 1: if two people started gossiping at the start of the day, it would take 19 hours to spread the information to all participants. We also see that it is quicker to spread information in the second, third and final day of the conference at about 10 hours. From Table 3.3 this makes sense since on the first day participants did not start until 6pm (i.e., there is an initial delay equal to 18 hours).

What we see from the low values of E_{glob} and E_{loc} are that contacts between all participants and contacts between acquaintances did not allow a high capacity to

		Temporal Metrics			Reshuffled		
	Day	E_{loc}	L	E_{glob}	E_{loc}	L	E_{glob}
INFOCOM	1	0.033	19h 39m	0.003	0.077	5h 29m	0.100
	2	0.110	9h 6m	0.024	0.194	2h 45m	0.239
	3	0.077	10h 32m	0.018	0.114	4h 6m	0.167
	4	0.052	9h 55m	0.013	0.104	3h 3m	0.165
REALITY	08 Sep	0.000	23h 15m	0.000	0.003	21h 58m	0.010
	15 Sep	0.000	22h 47m	0.001	0.007	19h 55m	0.024
	22 Sep	0.000	22h 53m	0.001	0.007	20h 42m	0.019
	29 Sep	0.001	22h 20m	0.001	0.009	17h 44m	0.037
	06 Oct	0.000	22h 14m	0.001	0.011	16h 23m	0.041
	13 Oct	0.000	21h 37m	0.004	0.013	14h 57m	0.055
	20 Oct	0.001	21h 45m	0.003	0.007	17h 4m	0.031
	27 Oct	0.002	22h 1m	0.001	0.013	15h 19m	0.050
	03 Nov	0.001	21h 6m	0.004	0.012	16h 17m	0.043
	10 Nov	0.000	20h 5m	0.004	0.015	14h 25m	0.061
EMAIL	27Oct	$3.1E^{-8}$	86397.94s	$9.3E^{-7}$	$7.7E^{-8}$	86396.91s	$1.6E^{-6}$
	28 Oct	$4.0E^{-8}$	86399.78s	$1.4E^{-7}$	$4.1E^{-8}$	86399.71s	$1.5E^{-7}$
	29 Oct	$3.9E^{-8}$	86399.03s	$3.9E^{-7}$	$7.2E^{-8}$	86398.59s	$7.3E^{-7}$
	30 Oct	$5.8E^{-8}$	86398.76s	$5.5E^{-7}$	$6.9E^{-8}$	86398.48s	$7.5E^{-7}$
	31 Oct	$4.7E^{-8}$	86398.92s	$4.9E^{-7}$	$6.5E^{-8}$	86398.64s	$6.9E^{-7}$
	01 Nov	$5.8E^{-8}$	86399.03s	$4.9E^{-7}$	$6.6E^{-8}$	86398.85s	$6.0E^{-7}$
	02 Nov	$4.3E^{-8}$	86398.68s	$5.4E^{-7}$	$6.5E^{-8}$	86398.67s	$6.8E^{-7}$

Table 3.5: Temporal Metrics (h=1, t_{min}=00:00, t_{max}=23:59, w=5 min) compared with shuffled temporal graph (runs=50).

spread information. Since temporal local efficiency E_{loc} measures how people you meet interact amongst themselves we can drill in and examine on a local view, if the interaction between such acquaintances are any better at spreading information. In this case E_{loc} for each day is similar but slightly lower to E_{glob}: this tells us that acquaintances do not congregate together very often.

REALITY: This data set has many more days and can provide a better overview of day-to-day trends. We show 10 consecutive Wednesdays starting from the first

day of the Fall '04 semester (8th Sep to 9th Dec 2004)[2]. For the first day we can see that it is slow for information to spread since $L = 23$ hours. Since both local and global efficiency are at zero, participants infrequently interacted with eachother. This makes sense since relationships are unlikely to have formed and so there are less contacts. During the subsequent Wednesdays the information spreading process is quicker and there is also a steady decrease in the average temporal path length. However still compared to the conference environment, on a campus it takes twice as long for information to spread.

EMAIL: The final dataset is the poorest for data diffusion as seen by the zero value clustering and extremely low efficiency and high temporal path length, shown in Table 3.5. Since there are close to 57,000 nodes we have to consider this when examining these numbers as it contributes to the small normalised values. Classic metrics used on this dataset provide an overestimate of local efficiency since they assume that all links exist uniformly across time, when in fact in reality, e-mail exchanges take place at specific points in time. What differs from low values seen in REALITY is that now on some days E_{loc} is non-zero, albeit extremely small. This suggests that email users do not stay in groups or, in other words, do not use email as quick exchanges of messages to each other which makes sense since there are delays between replies.

3.4.2.4 Importance of Time Dependencies

We now turn our attention to a more general type of time-order, namely the time dependencies between windows of a temporal graph. As a null model, we compare the real data sets $\mathcal{G}_t$ with their randomised counterpart where we have randomly reshuffled the time windows $G_T \in \mathcal{G}_t$, destroying these temporal dependencies and any inherent window time order. The right hand half of Table 3.5 show the metrics calculated on reshuffled temporal graphs for all three datasets. As we can see in all three traces, the shuffled network gives a quicker data diffusion time and higher efficiency. The reason for this is down to the cyclic behaviour of human contacts. Humans as a collective congregate during the working hours and are more sociable during mid week. This means that there is a denser number of contacts at certain times which limits the opportunity for transitive meetings between friends to certain

[2]http://web.mit.edu/registrar/www/calendar0405.html

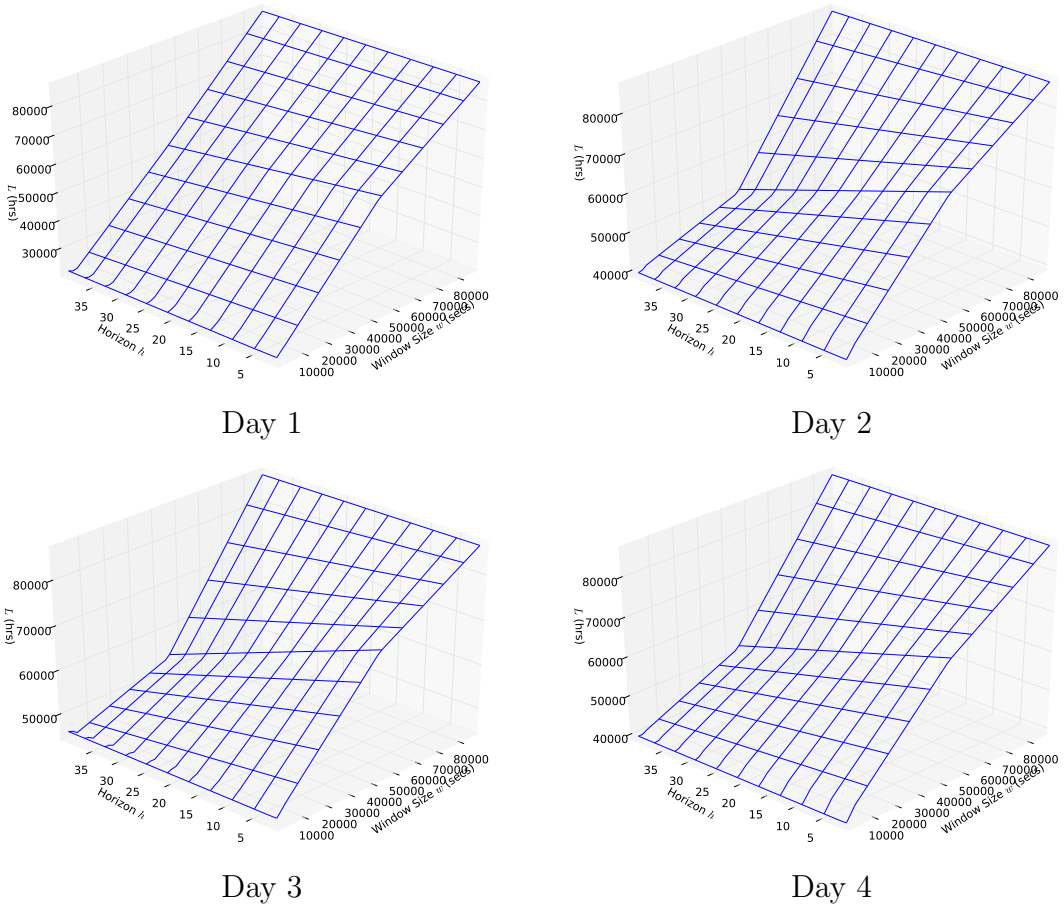


Figure 3.5: INFOCOM: Changes to average shortest temporal path length L when varying window size w and horizon h.

times of the day and decreases the speed of data diffusion. Reshuffling leads to the introduction of heterogeneity of contacts throughout a time period and introduces more opportunity for contacts throughout the day. This demonstrates that time dependencies are important in measuring the information dissemination efficiency of real time-varying networks; since static graphs aggregates all this time information there is no way to recover these temporal dependencies.

3.4.2.5 Varying Window Size and Horizon Parameters

We now return to the subject of window size w (number of windows W) and horizon h parameters. Figure 3.5 plots the average temporal shortest path length L for

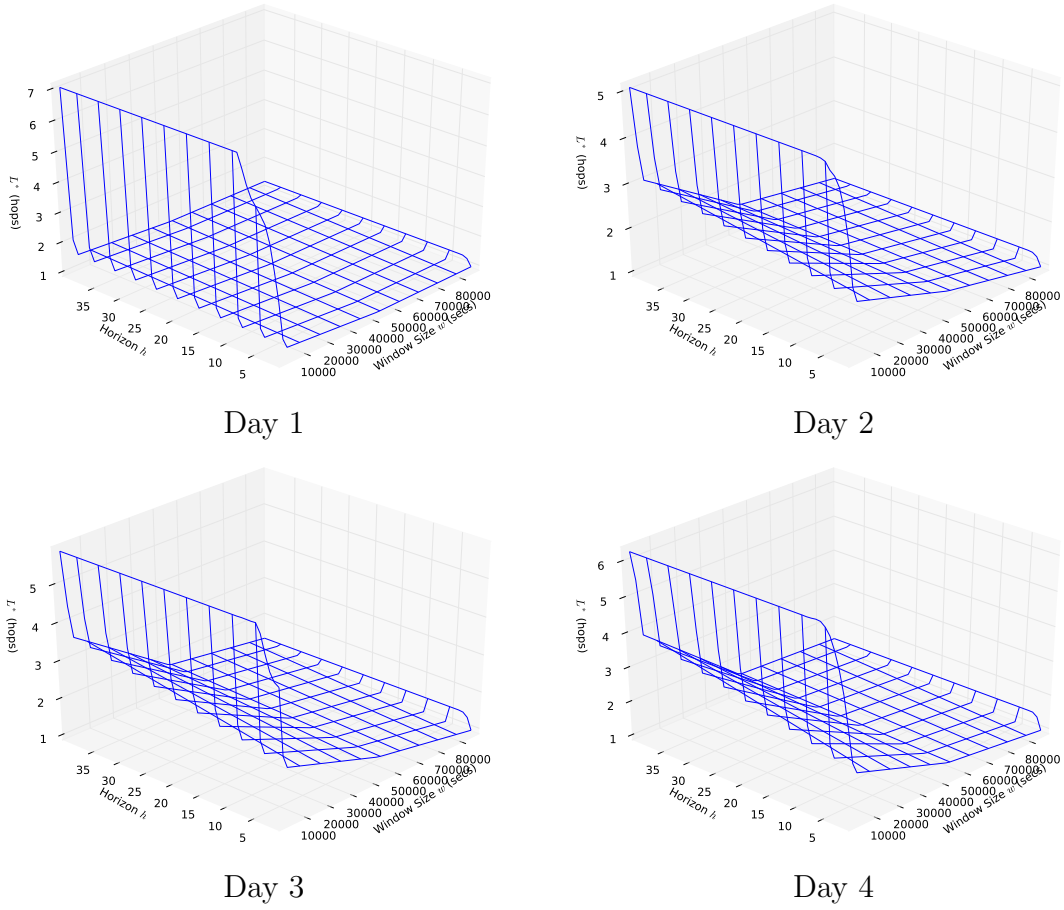


Figure 3.6: INFOCOM: Changes to average shortest temporal hops L^* when varying window size w and horizon h.

INFOCOM as we vary w from 2 minutes (τ=720) to 24 hours (τ=1 is equivalent to a static aggregated graph) and vary h from 1 to N-1. L is most affected by the window size, which is due to the granularity that that the average temporal path length is calculated at, for example, with window size of 12 hours then all messages will be treated as being delivered during this 12 hour window, however, with a finer window size of 2 minutes then shorter delivery times will be taken into account. We also note that the horizon is also affected by the window size; at both ends of the spectrum of w (τ=720 and 1) L is not sensitive to a varying horizon, however, with a mid-range window size then there is a linear decrease in L as h increases. This effect is intuitive since a smaller horizon limits the reachability of a node in the current window, delaying instead the path until a potentially later window.

Figure 3.6 plots the average time respecting hop count L^* for INFOCOM, again as we vary w from 2 minutes (τ=720) to 24 hours and vary h from 1 to N-1. Extrapolating from a static aggregated graph (τ=1, w=86400 seconds), as the number of windows increases, we see an increase in the average time respecting hop count L^*; this demonstrates that adding any extra time information will start to reveal the true shortest path length, which respects time order.

Turning our attention to the horizon, as h increases L^* also increases. At first this seems counter-intuitive as one might expect a similar relationship between these two variables seen for L in Figure 3.5, however, this can be explained with a reminder that the average time respecting hop count L^* does not give an indication of the duration of time to deliver messages. When h is high, long transitive paths to all nodes can indeed be formed in earlier windows, however this also means that L^* will be high; on the other end of the spectrum, when h=1 then this will delay delivery times since we cannot reach the destination in an early window, but this delay also means that a node is more likely to meet the destination node in the future within 1 hop and hence give a smaller L^*. Since L^* does not capture the time duration (as L does), this explains the counter-intuitive relationship between h and L^*. Taking another perspective, this means that the difference between the static average shortest path and temporal L^* gives a lower bound to temporal analysis, and increasing the horizon only serves to enhance the difference between static and temporal analysis.

We conclude that the selection of an appropriate window size w plays an important part in the accurate analysis of temporal graphs and with an appropriately fine window size then the horizon h parameter plays a small part in the calculation of the average temporal shortest path length. However, we also note that any increase the number of windows τ from single windows (equivalent to an aggregated static graph) improves the accuracy of temporal analysis and hence selecting a window size close to the collection interval gives a very good approximation to the true temporal path length. With this in mind, we return to our assumptions that the typical time for a message to pass from a node to one of its neighbours is of the same order as the typical time at which the graph changes and the rest of this thesis shall select values of w which reflect the collection interval and set h=1.

3.4.2.6 Discussion

We now return to the original question posed, namely *does time really matter and, if so, can we quantify this difference?* We have demonstrated that time does matter in two ways, firstly, since static analysis ignores time order of links then there is an over estimation of available links and hence an underestimation of the true static shortest path length; and secondly, temporal path length gives us an indication of the actual time elapsed as opposed to hops which can be misleading, as seen in our analysis of varying horizon and window size parameters. Key to quantifying this difference has been the definition of temporal shortest hop length L^* which has shown that a 4x underestimate exists in the best case scenario when the horizon h=1; also, the definition of temporal shortest path length L quantifies the global characteristics of a temporal network for information dissemination which aids in comparing different types of networks. In the following study we shall take this analysis one step further and find relationships between L and the speed at which a real network changes over time.

3.4.3 Small-world Behaviour in Temporal Graphs

3.4.3.1 Introduction

We now investigate the relationship between communication efficiency and the speed of temporal graph change. Intuition would suggest that a real network that changes slowly would also be slow for information dissemination (and vice versa), since paths between distant nodes are formed at a slower rate. We investigate this hypothesis by utilising the average temporal path length L and temporal correlation coefficient C. Low values of L (high values of E) indicate that the nodes of the graphs can communicate efficiently. In the following, we will show that temporal graphs from models and real-world systems can be, at the same time, temporally clustered and still have small temporal distances between their nodes. In analogy with the small-world analysis in static graphs [WS98, LM01], we will compare the actual values of temporal C, L and E of a given time-varying graph $\mathcal{G}$, with the corresponding values calculated by considering an ensemble $\{\mathcal{G}^{rand}\}$ of randomised versions of $\mathcal{G}$. Each sequence $\mathcal{G}^{rand}$ is obtained by randomly reshuffling the graphs in $\mathcal{G}$, i.e., by destroying the time order (and correlations) in the original sequence $G_1, G_2, \ldots, G_T$.

More precisely, we will show that some temporal graphs can have a value of C much larger than the correlation coefficient of the reshuffled sequence C^{rand}, and, at the same time a value of L as small as L^{rand}. We will refer to this behaviour as *small-world behaviour in time-varying systems*.

3.4.3.2 Random-walkers network model

To illustrate how this behaviour can be obtained in a network, we develop a simple network model of moving agents where the speed of evolution can be interpolated from slowly to quickly changing. We consider a system of N random walkers that move in a two-dimensional square of linear size D with a fixed velocity v, and additionally perform long-distance jumps to randomly chosen position of the square with a jump probability p_j [BFFL08]. For each fixed value of $p_j \in [0, 1]$, the temporal network $\mathcal{G}$ is constructed by linking, every second, all nodes having a distance in space smaller than a given value r_c. In Figure 3.7 we plot C and L as a function of p_j. The values reported are normalised to the maximum values of C and L obtained for $p_j = 0$, and respectively equal to $C(0) = 0.91$ and $L(0) = 442.8$.

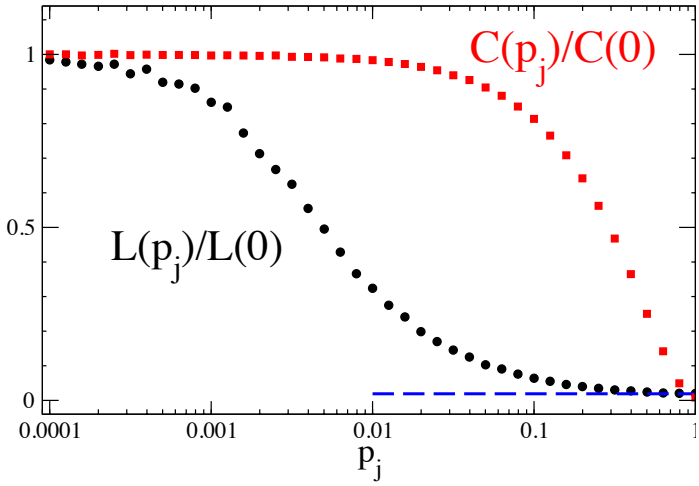


Figure 3.7: Characteristic temporal path length and temporal-correlation coefficient of temporal graphs produced by the model of moving agents, as a function of the probability p_j of long-distance jumps. In the simulations we have set $N = 100$, $D = 100m$, $v = 1\ m/s$, $r_c = 5m$ and produced sequences of length $T = 500$.

Confirming our hypothesis, we observe that when p_j=0.0001 the temporal correlation coefficient $C(p_j)/C(0)$ is at its peak i.e. the temporal graph is slowly changing, and the average temporal path length $L(p_j)/P(0)$ is also at its peak i.e. very slow for information dissemination. On the other end of the spectrum, when the graph is quickly evolving (p_j=1.0 and C 0), the network allows quick information dissemination (L 0). However, when slowly interpolating between these extremes we observe that a small percentage of jumps are sufficient to create links between nodes otherwise at large temporal distances and to produce a large drop in temporal L; this indicates that slowly evolving networks still allow for relatively quick information dissemination. When $p_j = 0.01$, L has reduced to one fourth of $L(0)$, and when $p_j = 0.1$, L has about the same value as for the reshuffled sequence. The value of L^{rand} obtained over $\mathcal{G}^{rand}$ (1000 realisations) is reported as a dashed line.

The relationship between the characteristic temporal path length and temporal correlation coefficient (Figure 3.7) draws parallels to the original relationship between the static characteristic shortest path length and static clustering coefficient presented by Watts & Strogatz (see Figure 2.1(a)). While $L(p_j)$ is rapidly decreasing, $C(p_j)$ is constant up to large values of $p_j \sim 0.1$, so that for intermediate values of p_j we have temporal graphs exhibiting small-world behaviour.

3.4.3.3 Empirical Networks

Brain cortical networks

We now explore real-world time-varying complex networks. We first consider time-varying functional cortical networks extracted from a set of high-resolution EEG recordings in a group of 5 normal subjects performing a task consisting in a foot movement [FLA+08]. For each subject and for each of four frequency bands $(\alpha, \beta, \gamma, \theta)$, we considered a time period of 0.5 sec corresponding to the final phase of execution of the foot movement. Each temporal graph has $N = 16$ nodes, representing cortical regions of interest and consists in a time sequence of $\tau = 100$ directed unweighted graphs, where the directed links represent causal influences between cortical regions (see the original study for details [FLA+08]). The original dataset was collected with a 200hz sampling frequency and hence this provides an appropriately fine granularity for the window size of w=5 milliseconds.

	C	C^{rand}	L	L^{rand}	E	E^{rand}
α	0.44	0.18 (0.03)	3.9 (100%)	4.2 (98%)	0.50	0.48
β	0.40	0.17 (0.002)	6.0 (94%)	3.6 (92%)	0.41	0.45
γ	0.48	0.13 (0.003)	12.2 (86%)	8.7 (89%)	0.39	0.37
δ	0.44	0.17 (0.003)	2.2 (100%)	2.4 (92%)	0.57	0.56
d1	0.80	0.44 (0.01)	8.84 (61%)	6.00 (65%)	0.192	0.209
d2	0.78	0.35 (0.01)	5.04 (87%)	4.01 (88%)	0.293	0.298
d3	0.81	0.38 (0.01)	9.06 (57%)	6.76 (59%)	0.134	0.141
d4	0.83	0.39 (0.01)	21.42 (15%)	15.55(22%)	0.019	0.028
Mar	0.044	0.007 (0.0002)	456	451	0.000183	0.000210
Jun	0.046	0.006 (0.0002)	380	361	0.000047	0.000057
Sep	0.046	0.006 (0.0002)	414	415	0.000058	0.000074
Dec	0.049	0.006 (0.0002)	403	395	0.000047	0.000059

Table 3.6: Temporal-correlation, characteristic temporal path length and efficiency for brain cortical networks (subject 1, and four band frequencies) [FLA+08], for the social interaction networks of INFOCOM'06 (time periods between 1pm and 2:30pm, four different days), and for messages over Facebook online social network (three different months of year 2007) [WBS+09]. Results are compared with those obtained for 1000 randomised (shuffled) sequences of the same length. The values in parenthesis next to C_{rand} are the respective standard deviations. The values in parenthesis next to L and L^{rand} are the percentage of pairs of nodes that are temporally connected and not considered in the averages.

We have computed the values of temporal C, L and E for each real sequence and for the reshuffled temporal network. In Table 3.6 we report the results for one of the subjects. For all the considered bands, the real sequence exhibits small-world properties, having a large value of C (significantly larger than C^{rand}) and, at the same time, a small characteristic temporal path length (a high efficiency), comparable to that observed in the shuffled sequence. Similar results (not reported) were obtained for the other four subjects.

Social interaction networks

The second real case study of our analysis is a time-varying social network based on a dataset of contacts among participants of INFOCOM'06, a major data communication conference which took place in a hotel. The contacts were collected by means of Bluetooth-enabled devices able to record interactions among people that are in proximity [SGC+09]. The Bluetooth scanning rate was set to 2 minutes and this

is use as an appropriately fine window size w in the temporal graph. In Table 3.6 we report the data for the interactions during lunchtime between 1pm and 2:30pm. This is the interval with the larger number of contacts during a day. Each sequence is made of $T = 45$ undirected unweighted graphs with $N = 78$ nodes each. The average path length and the efficiency are similar for the original and reshuffled traces (the number in parenthesis close to L and L^{rand} are the percentage of pair of nodes being temporally connected and hence considered in the computation of the average path length), whereas C is more than double that of C^{rand}. This can be considered as an indication of small-world behaviour in these traces according to our definition (Section 3.4.3.1).

Online social networks

The third system we study is based on interactions over an online social network. The original dataset contains the messages sent among 6 millions users in the London network of Facebook over one year (March 2007 to February 2008) [WBS+09]. We have divided the contacts according to the months of the year and, for each month, we have filtered out all contacts between pairs of nodes which exchange less than 10 messages per month. This allows us to consider only the subset of most active users, obtaining networks with about $N = 100,000$ users per month. For this dataset, there is no clear granularity for the window size w, however, as we are interested in the relative difference between the temporal graph to its randomly shuffled counterpart we select a window size w =1 hour which appropriately captures the time scale of social interactions between friends. For each month, the time varying graph is composed by $\tau = 720$ (for 30 days) or $\tau = 744$ (for 31 days) directed graphs, one for each hour of the month. As shown in Table 3.6 for four different months of the dataset, the average temporal path length of the temporal graph is close to the value obtained for the reshuffled sequences. However, the network under study is disconnected in several different components, and only an extremely small percentage (about 10^{-6}) of the node couples are temporally connected. Consequently, the characteristic temporal path length was evaluated as an average over a small number of node couples. A better characterisation of the system can be obtained by means of the temporal efficiency. The values of E and E^{rand} measured for Facebook are in general smaller than those observed in the other two networks, this being

due to the high disconnectedness of Facebook. Nevertheless, as for the case of the cortical networks and of INFOCOM'06, the real Facebook is almost as efficient as its reshuffled version. Finally, also for Facebook we observe a temporal small-world behaviour: while the length of the temporal paths of the temporal graph are not affected by the reshuffling procedure, the temporal correlation coefficient C is about one order of magnitude larger than in the reshuffled version C^{rand}.

3.4.3.4 Varying the Horizon Parameter

We have selected window size values w in line with the finest granularity available from the data source in the cortical and INFOCOM'06 networks; and selected a window size of an hour for our study of the FACEBOOK interactions. We now investigate the effect of varying the horizon parameter on these results.

We produce a temporal graph from real data by considering a system at its maximum resolution sampling time. This fixes the typical time τ_g at which the graphs in the sequence are changing. In our study, we have implicitly assumed that the typical time, τ_m, for information exchange from a node to one of its first neighbours, is of the same order as τ_g (this means setting the horizon h=1). However, the case $\tau_m < \tau_g$ can be simulated by increasing the horizon parameter, h (in order to have message propagation we have to assume that the time-varying graph changes slower than a message can propagate from a node to its neighbours, hence we discard the case $\tau_m > \tau_g$). Clearly as we increase the horizon, the temporal path length, L, will drop (or the Efficiency will increase) since a message can reach more nodes earlier. Note that firstly, this drop (increase) is proportional in both the original sequence and shuffled cases and that temporal correlation coefficient is not affected by h.

Starting with the random walker model, in Figure 3.8 we plot the values of $L(p_j)/L(0)$ for $h = \{1, 2, 3, 10, 25, 50, \infty\}$ where infinity is N-1. Since the $h = 1$ curve provides the upper bound this confirms that using a horizon of one gives the worst case scenario and increasing h makes the L curve in Fig 2. drop quicker, and increases the difference between the L and C curves.

Concerning the real networks, we first report in Table 3.9 the values of L and L_{rand} as we increase the horizon h for the Gamma EEG band of the cortical network. As we can see, when h increases, both L and L_{rand} drop proportionally, levelling off when $h \geq 4$.

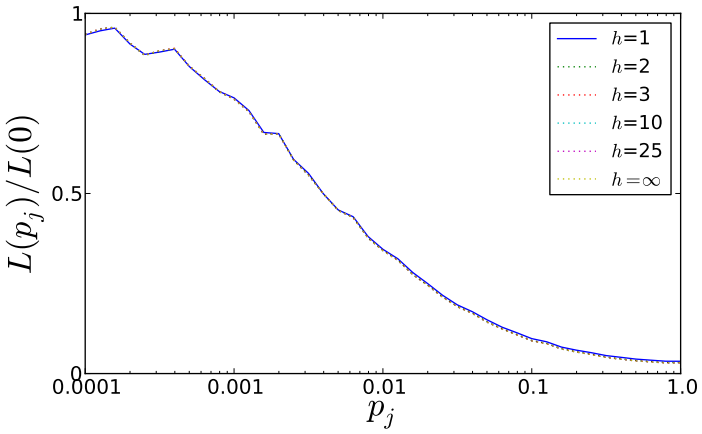


Figure 3.8: Varying the horizon h parameter on random jumper model in Figure 3.7.

Horizon	1	2	3	4	5	10	15
L	12.188	11.372	11.024	10.976	10.976	10.976	10.976
L_{rand}	08.807	07.953	07.699	07.611	07.596	07.594	07.594

Figure 3.9: L and L_{rand} calculated with different horizon values on cortical networks (gamma band).

To show the same effect for all cortical bands and the INFOCOM'06 networks, Figure 3.10 plots the ratio L/L_{rand} as we increase the horizon. For both datasets as we increase h, the ratio of L over L_{rand} always falls within the interval [1,2], which indicates that that L and L_{rand} remain similar.

We can conclude that the small-world property of $C \gg C^{rand}$ and at the same time a value of $L \sim L^{rand}$ still hold irrespective of the horizon parameter and that using $h = 1$ again gives us an upper bound (worst case scenario) to the the average temporal shortest path length L.

3.4.3.5 Discussion

In conclusion, our results suggest that time-varying networks, strongly clustered in time and, at the same time, with short temporal paths between their nodes, might

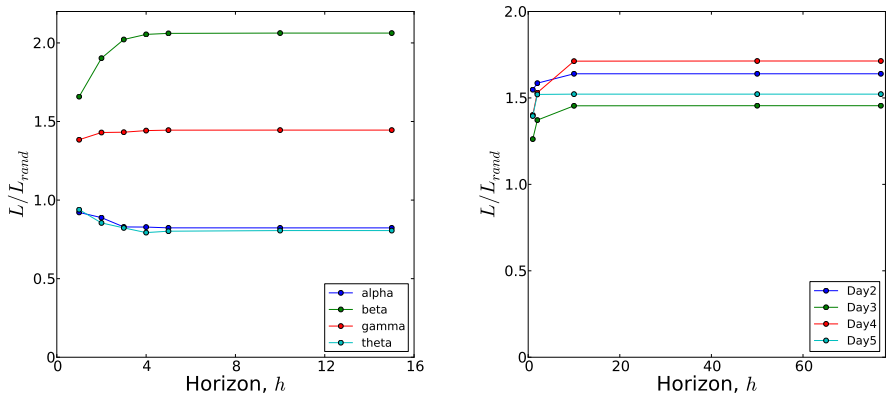


Figure 3.10: Varying the horizon h parameter on Cortical and INFOCOM'06 networks.

be widespread in biological, social and man-made systems, often with important dynamical consequences.

Another interpretation of the jumpers introduced in the random model is analogous to the concept of "shortcuts" introduced in the original Watts & Strogatz paper, where random rewirings of the lattice model created such long distance shortcuts. In the same way the jumpers in our temporal model capture the fact that people in social networks might have temporal shortcuts; speeding up the rate of mixing between and meeting of distant nodes.

3.5 Conclusions

In this chapter we have introduced the foundational temporal graph model and temporal distance metrics and applied them to two case studies on several empirically collected networks datasets. Firstly, we found real differences between static and temporal analysis of shortest paths in that since static graphs ignore time order, static shortest path lengths overestimates the available links and therefore underestimates the true shortest path length. Secondly, contrary to intuition, we found that even slowly evolving networks can exhibit properties that allow for fast information dissemination between nodes in temporal networks.

In these two studies, we also examined the part that window size w and horizon h parameters play in the analysis of temporal shortest path lengths L. We have found that the window granularity has the most effect on L, however, with any additional temporal information gives more accurate temporal analysis. We also found that with a fine grained window size, then the horizon plays little part in effecting L; this corroborates our initial assumptions that the typical time for a information to pass from a node to one of its neighbours is of the same order as the typical time at which the graph changes (i.e. h=1). In addition, when the window size does effect the horizon, we find that setting h=1 gives us the worst-case scenario when comparing to static graphs; this was observed in both studies over a synthetic model and across several different real networks. Since the metrics derived in the proceeding thesis are derived from temporal shortest paths we shall continue this worst case analysis and assume that the typical time for a message to pass from a node to one of its neighbours is of the same order as the typical time at which the graph changes, by implicitly setting h=1. In the next chapter, we shall see that these guidelines on window size and horizon are appropriate for the simulation of message spreading through contact-by-contact replay in mobile phone proximity networks (Section 4.2.2).

Following from these insights on the importance of time order in the calculation of shortest paths, the next chapter explores measures of important nodes for information dissemination, which are based on shortest temporal paths.

4

Temporal Centrality Measures

Introduction

Identifying important nodes in a network has become an essential part of analysing and understanding networked systems with application to a wide range of fields including finding the best person to target in a viral marketing campaign [KKT03, WF94], locating key neurons in cortical networks [BS09], protecting important species in ecological systems [JOBL08], finding bottlenecks in traffic networks [Hol03] and even in the hunt for an Iraqi dictator [Wil10].

The position of a node with respect to other nodes can be classified and exploited: one could argue that people with the most friends are popular and hence important; a node with high geodesic locality to other nodes could spread information quickly to high numbers of nodes; and a person who lies between the most paths of communication could act as a mediator among groups of people. These concepts are more commonly known as degree, closeness and betweenness centrality [WF94, BLM+06].

In particular, the calculation of closeness and betweenness centrality (defined in Section 2.2.3) on static graphs are based on shortest paths, however, as we have shown in the previous chapter, static shortest paths miss the vital time order of links which result in the underestimation of the true shortest path. With this in mind, the key contribution of this chapter is the introduction of *temporal centrality metrics* for the identification of key nodes in temporal graphs based on temporal shortest paths. Naturally, both these temporal extensions are associated to the identification of central nodes in the network with application to *dynamic processes* over a real network. In particular, temporal closeness quantifies how fast a user can disseminate a piece of information. Therefore, applications of this metric include viral marketing and the study of rumour spreading. On the other hand, temporal betweenness distinguishes individuals who act as key *mediators* between the most communication paths over time.

Chapter Outline

In the next section, we present the definitions of temporal closeness and betweenness derived intuitively from their static counterparts.

Evaluating the correctness or accuracy of a given centrality ranking is non-trivial since it is dependent on the intended application. However, through two case studies, we compare our temporal centrality formulations to their static counterparts and demonstrate the effectiveness of these temporal centrality rankings under their intended application.

Firstly, from a *semantic* perspective we discuss the node rankings given by temporal centrality and static centrality within the context of the three years leading up to a corporate bankruptcy, namely the Enron email dataset (Section 4.2.1). This dataset possesses the known corporate roles of each user and hence provides us useful insight into the actual roles that high centrality nodes played within the organisation during this period.

Secondly, from a dynamic communication *process* perspective, we evaluate the speed of information dissemination and mediation using high-ranking closeness and betweenness nodes, respectively. Effective information dissemination is measured by the time taken to deliver information to all other nodes starting from high ranked

temporal closeness nodes. Effective mediation is measured in a converse scenario of immunising the network against the spread of some contagion, through removal of high temporal betweenness nodes from the network and measuring the reduction in information dissemination speed between remaining nodes. Continuing with the Enron email dataset we demonstrate that the nodes selected by temporal centrality provides faster information dissemination and more effective protection against attack, when compared to static centrality (Section 4.2.1.3). Taking this one step further, we design two possible short-range mobile worm defence schemes and evaluate using proximity based mobile phone networks (Section 4.2.2).

4.1 Temporal Centrality

4.1.1 Temporal Betweenness Centrality

Betweenness is commonly used to discover nodes that are critical for mediating information flow. Such nodes represent individuals who negotiate between the different groups of parties; people in organisations who fall into middle management and balance reporting to senior management and also command a large workforce; and routers in the Internet which facilitate information flow between ASes. If such nodes provide an important mediatory role in a network then it stands that the complement would also hold; how does the removal of such nodes disrupt the overall efficiency for information dissemination across the network?

As described previously (Section 2.2.3.3), to identify these mediating nodes, the static betweenness centrality of a node i is defined as the proportion of shortest paths between all pairs of nodes that pass through i. This proportion is important in that it gives a higher weight to nodes which facilitate paths where there are no alternatives. To capture the notion of *temporal* betweenness it is important to take into account not only the proportion of shortest paths which pass through a node, but also the *length* of time for which a node along the shortest path *retains* a piece of information before forwarding it to the next node. For example, consider the 2-hop shortest temporal path from node A to D, (A, B, D). In terms of time, this path could be represented as (A, B, B, B, D) since a piece of information resides on node B for 3 time windows, and so we want to assign a higher value as removing

this node will have a greater impact in disrupting the network. From this, for a given time window T we define the *temporal betweenness centrality* of node i as:

$$\mathcal{B}_i(T) = \frac{1}{(N-1)(N-2)} \sum_{\substack{j \in V \\ j \neq i}} \sum_{\substack{k \in V \\ k \neq i \\ k \neq j}} \frac{U(i,T,j,k)}{|\sigma_{j,k}(i)|}, \tag{4.1}$$

where the function U returns the number of shortest temporal paths $p_{jk} \in \S_{jk}$ from j to k where there is an edge from a node $n \in p_{jk}$ to node $i \in p_{jk}$ at time window T or the edge from node i to the next hop is at a future time window; and $\sigma_{j,k}(i) \subseteq S_{jk}$ is the set of shortest temporal paths from node j to k which pass through node i, defined when $\sigma_{j,k}(i) \neq \emptyset$. In the case when $\sigma_{j,k}(i) = \emptyset$, i.e., node i is totally isolated, we set its betweenness to zero. Finally, the average temporal betweenness value across all time windows for each node i is:

$$\mathcal{B}_i = \frac{1}{\tau} \sum_{t=0}^{\tau-1} \mathcal{B}_i(t), \tag{4.2}$$

where τ is the number of time windows in the temporal graph.

4.1.2 Temporal Closeness Centrality

Two nodes of a static graph are said to be *close* to each other if their geodesic distance is small. In a static graph, an estimation of the global *closeness* of a node i is obtained as the average static shortest path length to all other nodes in the graph [WF94]. Similarly, we can extend the definition of closeness to temporal graphs using the temporal shortest path length between nodes, which is a measure of how fast a source node can deliver a message to all the other nodes of the network. Given the shortest temporal distance $d_{ij}(t_{min}, t_{max})$, *temporal closeness centrality* can then be expressed as:

$$\mathcal{C}_i(t_{min}, t_{max}) = 1 - \left(\frac{1}{\tau(N-1)} \sum_{j \neq i \in V} d_{ij}(t_{min}, t_{max}) \right) \tag{4.3}$$

so that nodes having, *on average*, shorter temporal distances to the other nodes are considered more *central*. Note that the subtraction from one is only required for a *descending* ranking.

4.1.3 Runtime Complexity

Calculating temporal closeness is equivalent to calculating the single source temporal shortest path length from a node i to all other nodes in the networks $O(\tau.(|V|+|E|))$ (Section 3.2.3.3), where τ is the number of time windows, and summing which takes linear time $|V|-1$, hence the asymptotic time complexity is $O(\tau.(|V|+|E|))$. Temporal betweenness requires first to calculate temporal shortest paths for *all* pairs of nodes i, j ($N.O(2^{\tau|V|}$, Section 3.2.3.3) before an individual node k's betweenness counter can be incremented based on the proportion of shortest temporal paths between all pairs of nodes i, j which pass through k. Incrementing betweenness takes $O(N^2)$ since we need to iterate over all pairs of node. Therefore, the asymptotic complexity is dominated by the calculation of *all pairs* temporal shortest paths and hence the complexity is $O(2^{\tau|V|})$. In practise, we find that the computational time is much better than this upper bound suggests[1], though future work could investigate an optimised algorithm, for example, based on Brandes [Bra01] algorithm for static betweenness centrality calculation where *counting* shortest paths is more efficient (polynomial time complexity) than enumerating all shortest paths.

4.2 Application to Real Networks

4.2.1 Corporate Email Dataset

4.2.1.1 Introduction

The Enron Energy Corporation started as a traditional gas and electrical utility supplier; however, in the late 1990s their main money making business came from trading energy on the global stock markets [EM04]. In December 2001, the Enron Energy Corporation filed for bankruptcy after it was uncovered that fraudulent accounting tricks were used to hide billions of dollars in debt [Fed08]. This led to the eventual conviction of several current and former Enron executives [Cal04, Joh04]. The investigation also brought to light the reliance of the company on traders to

[1]Especially as our implementation of the calculation of shortest temporal paths is parallelised between all pairs of nodes.

bring in profits using aggressive tactics culminating in intentional blackouts in California in Summer 2001. With both control over electricity plants and the ability to sell electricity over the energy markets, Enron trader's artificially raised the price of electricity by shutting down power plants serving the State of California and profiting by selling electricity back at a premium [Rob04].

During the investigation into the Enron accounting scandal, telephone calls, documents and emails were subpoenaed by the U.S. government and as such the email records of 151 user mailboxes were part of the public record consisting of approximately 250,000 emails sent and received during the period between May 1999 to June 2002 (1137 days), leading up to the bankruptcy filing. None of the emails were anonymised and so they provide unique semantic information of the owner of each mailbox.

In this section, we take advantage of this semantic information in the analysis of important nodes for information spreading and mediation in the context of this company.

4.2.1.2 Temporal Graph Construction

There are a number of versions of the Enron email dataset in various formats[2][3]; we use the dataset prepared by Shetty & Adibi [SA05] since it is in a convenient SQL format and the authors have some partial information of the corporate roles of each user. In addition, we manually find background information of unknown users using professional OSN, such as LinkedIn[4] along with results of search engines. Since we do not have a complete picture of the interactions of users outside of the subpoenaed mailboxes we concentrate on email exchanges between the core 151 users only. Taking this email dataset, we process the complete temporal graph over a three-year period from 1999 to 2002. There is no clear window size but considering the temporal time scale of the dataset and the context, we choose to investigate the dataset on the granularity of a business day and hence w=24 hours. If an email was exchanged between two individuals in a temporal window, an undirected link

[2] http://www.cs.cmu.edu/~enron

[3] http://http://enrondata.org/content/research

[4] http://www.linkedin.com

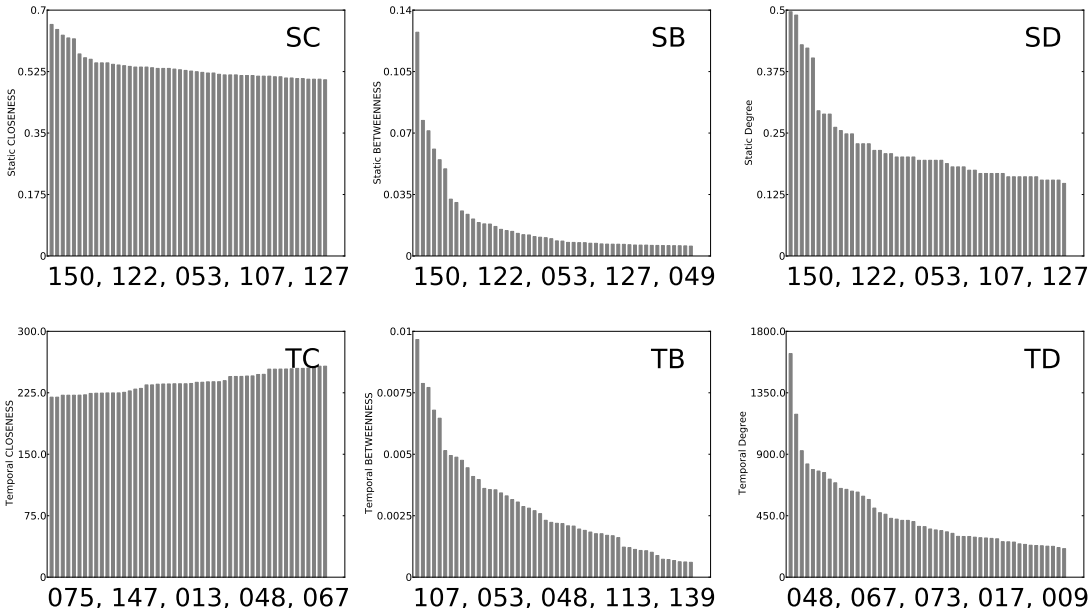


Figure 4.1: Ranked distribution of top 50 centrality nodes. Rows: static (S) & temporal (T) centrality. Columns: Closeness (C), Betweenness (B) & Degree (D). Top 5 node ID's listed under each plot. TC values: shown as windows (days).

between the two nodes representing those individuals will be added to the graph representing the temporal snapshot for that business day.

4.2.1.3 Semantic Value of Temporal Centrality

Figure 4.1 plots the static and temporal centrality rankings of employees calculated using closeness and betweenness. Examining the static centralities (left column) we note that there is little difference between the top five employees using static closeness or betweenness. Also plotting the static degree centrality of each node, we notice similar rankings suggesting that static analysis favours employees who interacted with the most number of other people. Temporal closeness and temporal betweenness yield different rankings amongst the top five and the calculated Kendall-tau correlation coefficient[Ken38] (Table 4.2) confirm that static-to-static metrics are strongly correlated ($\simeq 0.7$). Also, note that there is low correlation (< 0.4) between temporal metrics and static degree demonstrating that temporal analysis is not dependent on the number of people with which individuals interact.

ID	Name	Role	Notes
9	Stephanie Panus	(Unknown)	
13	Marie Heard	Legal	Senior Legal Specialist
17	Mike Grigsby	Manager	
48	Tana Jones	Executive	
53	John Lavorato	Trader	
54	Greg Whalley	President	Former Head of Trading
67	Sara Shackleton	Vice President	Enron Wholesale Services
73	Jeff Dasovich	Trader	
75	Gerald Nemec	Director of Trading	
107	Louise Kitchen	Trader	Head of Online Trading
122	Sally Beck	Managing Director	
127	Kenneth Lay	Chairman & CEO	
139	Mary Hain	Director	
147	Carol Clair	Trader	
150	Liz Taylor	Secretary	Assistant to Greg Whalley

Table 4.1: Roles of top centrality nodes.

Cross referencing the top two employee identifiers with their position within the organisation (Table 4.1) we identify a secretary (150) and managing director (122) as central nodes for both static closeness and betweenness; however, both temporal closeness and betweenness consistently selected employees in trading roles (053, 075, 107, 147). A secretary and a managing director are certainly important for information dissemination and central to many communication channels, as detected by static measures. However, instead the top trading executives are exclusively favoured by temporal analysis. Moreover, cross-referencing with media reports [CNN02], we find a correlation between the top two bonuses received and the two employees identified by temporal betweenness. To show that temporal analysis does not simply uncover nodes with the most interactions with other people, we also plot the temporal degree (TD) calculated as the total number of emails sent and received by each node i. Since there is a low correlation (< 0.4) with temporal closeness and betweenness this shows that temporal analysis is not dependent on the number of emails sent and received by each individual.

	SB	SC	SD	TB	TC	TD
SB	1.00	0.57	0.69	0.41	0.24	0.43
SC	-	1.00	0.70	0.36	0.22	0.31
SD	-	-	1.00	0.39	0.28	0.48
TB	-	-	-	1.00	0.43	0.34
TC	-	-	-	-	1.00	0.40
TD	-	-	-	-	-	1.00

Table 4.2: Kendall-tau correlation coefficients between centralities.

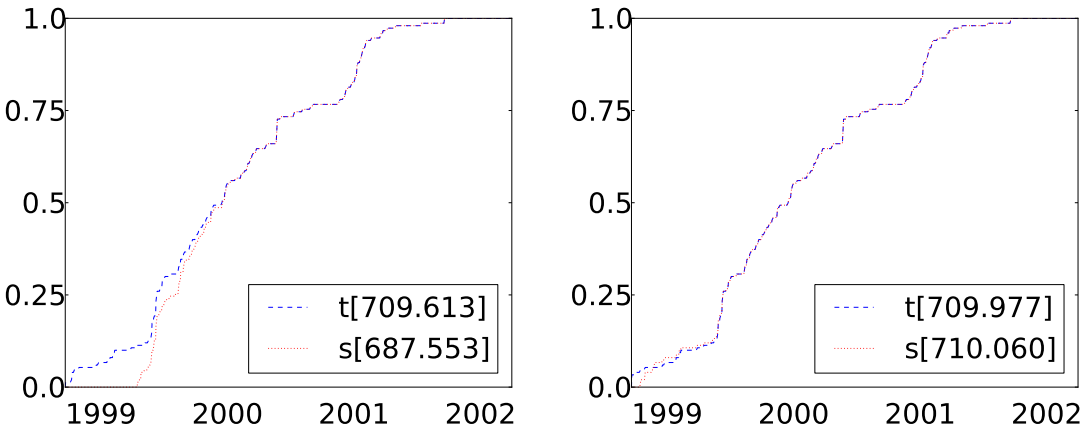


Figure 4.2: Dissemination Process: Dissemination ratio starting from top 2 (left) and top 10 (right) closeness source nodes. Area under curve reported in legend for temporal (t) and static (s) centrality.

4.2.1.4 Effectiveness of Central Nodes on Dynamic Processes

Trace-driven Simulation Setup

To evaluate the role and the centrality of the employee's identified by temporal and static analysis, we consider two dynamic processes. First, we simulate a simple information *dissemination process* over the temporal graph constructed from the Enron traces. The process is simulated as follows. We select the top N nodes from the ranking based on temporal closeness centrality. We place an identical message m into their (infinite) buffers. We refer to any node that has received a copy of this message as *reached*. We then replay the contact trace through time and as reached nodes make contact with an unreached node u, the message is replicated into the

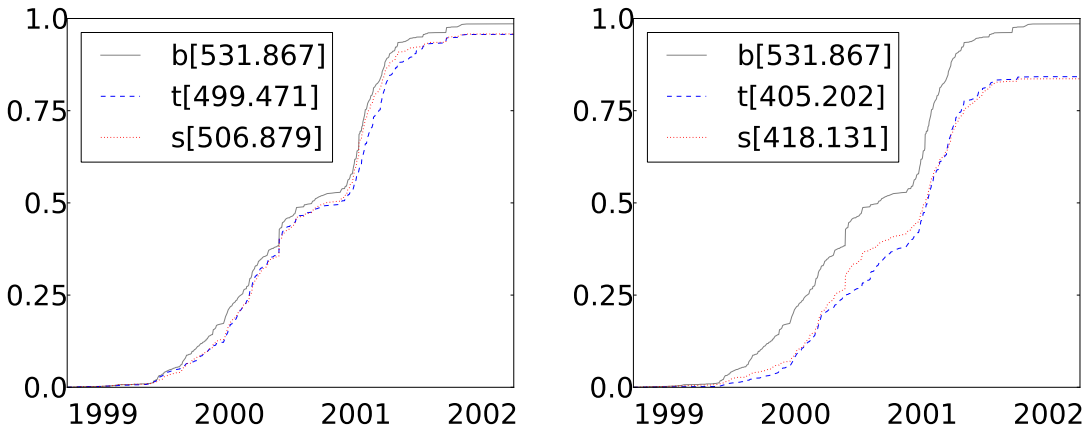


Figure 4.3: Mediation Process: Dissemination ratio after removing top 2 (left) and top 10 (right) betweenness nodes. Area under curve reported in legend for temporal (t), static (s) and baseline (b) where no nodes are removed.

buffer of node u. We assume that messages are transferred instantaneously and only the first neighbour in a time window can be reached. We then repeat this for static closeness centrality and plot the dissemination ratio across time for both.

Second, to model the role of individuals as part of an information *mediation process*, we borrow concepts from the more commonly known epidemic immunisation process where the dissemination ratio of a contagion spreading throughout a static network is measured before and after certain nodes are immunised against the contagion [BBV08]. This is analogous to measuring the spread of information (the contagion) before and after important individuals are removed from the network (such as going on holiday or being discharged) since our conjecture is that removing mediators will affect the network communication efficiency greatly.

In the trace-driven simulation, instead of a single message spreading within the organisation, we seed all employees with a different message that needs to be delivered to all other employees. This models multiple channels of communication. In order to derive a baseline performance, we start by calculating the dissemination ratio when no nodes are removed. We then remove the top N individuals identified by temporal betweenness and rerun the information spreading process. Nodes that are removed cannot receive or pass on messages. We then repeat the same process for comparison using static betweenness centrality for the ranking.

Evaluating Information Dissemination & Mediation

We present plots using $N = \{2, 10\}$ for information dissemination (Figure 4.2) and information mediation (Figure 4.3). As we can see the different pairs of traders identified by temporal analysis are better than the arbitrary nodes selected by static analysis for both disseminating information through the organisation and acting as mediators between communication channels. In the information dissemination case, although the final dissemination is the same across the long period of time, the two traders selected by temporal analysis disseminate information quicker. Only after increasing to 10 nodes, the static analysis presents similar results. In the information mediation case, the final dissemination ratios for both temporal and static centrality nodes slightly decreases by removing the nodes but are comparable. However, removing the two traders gives an overall more prolonged drop in information dissemination. In the case of the removal of 10 nodes, the individual's identified by means of the temporal metrics slow the dissemination process further compared to static ones.

4.2.1.5 Insights into Temporal Dynamics

To gain some insight into the interactions of individuals over time selected by temporal and static analysis, Figure 4.4 plots the number of emails sent and received over time, again by the top [Kos09] two centrality nodes. Moreover, we recall, from Section 4.2.1.4, that there is a strong correlation between static closeness and betweenness with degree. Such strong correlation between static closeness and betweenness with degree has been well documented in [New05, Bar04].

By comparing the contact distribution between static analysis (top row) and temporal analysis (bottom row), we observe that the trader's identified as important individuals by temporal analysis have sent and received more emails earlier in time, compared with the nodes identified by static analysis that interact with the highest number of different people. This fits the intuition that earlier interactions are key to faster dissemination and hence temporal metrics are more accurate at identifying key individuals. This also confirms our arguments that static analysis ignores time information such as duration, frequency, time ordering and, at the simplest level, earlier interactions.

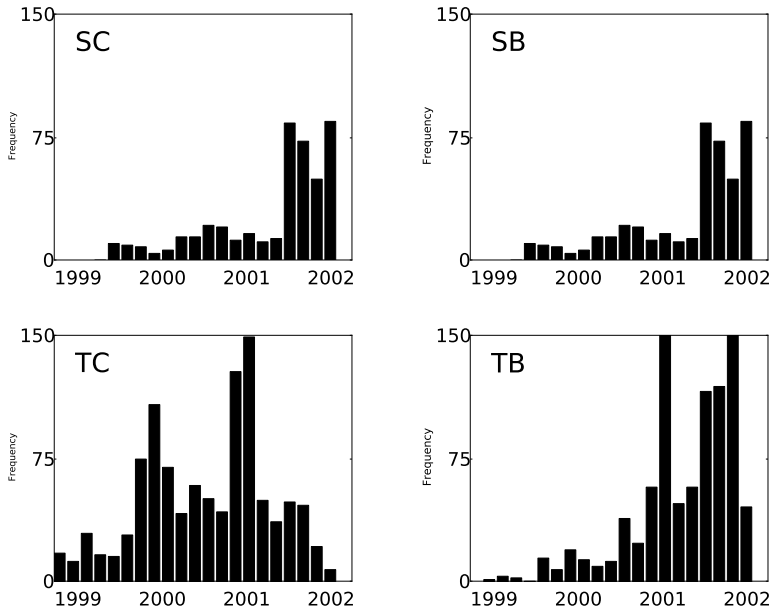


Figure 4.4: Distribution of total emails sent & received over time by top 2 centrality nodes. Bin size=50 days. From top-left: Static Closeness (SC), Temporal Closeness (TC), Static Betweenness (SB), Temporal Betweenness (TB).

4.2.1.6 Discussion

We have applied temporal centrality to a real network with semantic information of each users corporate role. From this, temporal centrality has identified nodes which have more intuitive connection to the context of the corporate before their bankruptcy filing. We also studied basic dynamic information spreading and mediation processes using these key nodes. Firstly, closeness centrality nodes should spread information quickly and to the most number of nodes, however, we observed that static closeness centrality is highly correlated with nodes with high degree and identifies nodes form paths which occur at later in time; compared to temporal closeness centrality which selects nodes which are uncorrelated with degree and send and receive more emails earlier in time. Secondly, temporal betweenness identifies nodes which mediate the most information flows as demonstrated by their removal from the network.

In the next section, we shall investigate how the features of temporal closeness and betweenness can help in the application to short-range mobile malware containment.

4.2.2 Short Range Mobile Malware Containment

4.2.2.1 Introduction

Smartphones are not only ubiquitous, but also an essential part of life for many people who carry such devices through their daily routine. It comes as no surprise then, that recent studies have shown the mobility of such devices mimic that of their owners' schedule [EP06, WGHB09]. This fact constitutes an opportunity for devising efficient protocols and applications, but it also represents an increasing security risk: as with biological viruses that can spread from person to person, mobile phone viruses can also leverage the same social contact patterns to propagate via short-range wireless radio such as Bluetooth and WiFi. For example, when security researchers downloaded *Cabir* [cab04] – a proof-of-concept mobile worm – for analysis, they discovered the full risk as it broke loose, replicating from the test device to external mobile phones [Hyp06]. This prompted the need for specially radio shielded rooms to securely test such malicious code [Hyp05].

Until recently though, mobile malware has been developed only for proof-of-concept experiments with very limited and non-malicious effects on users [Str08, Sch09]. However, the immense popularity and improvements in smartphone technology have attracted the attention of a growing number of attackers. In particular, increasing economic incentives have been the motivation of more recent exploits, for example stealing private data such as phone contacts [Liu06]; transferring call credit to other accounts [Lab09]; and traditional exploits such as premium rate number dialling [ter10].

Unlike desktop computers, mobile malware can spread through both short-range radio (i.e., Bluetooth and WiFi) and long-range communication (i.e., SMS, MMS and email) [Lea05]. Long-range malicious traffic can potentially be contained by the network operator by scanning every message against a database of known malware [RCSS07], however, short-range propagation might fall under the radar of centralised service providers: effective schemes to defend against short-range mobile malware spreading are necessary.

There are several reasons why naively sending security patches directly to every device is not efficient in cellular environments. Firstly, the *cost* in mobile data service plans is not widely in favour of the end user, hence users may resist to

update patches via 2G/3G networks if they do not subscribe to unlimited wireless data access plans; secondly, many mobile devices, such as tablets, do not have 2G/3G radios and hence rely on Bluetooth or WiFi to receive data; finally, *service coverage* is not guaranteed in certain areas (e.g., rural or underground metro systems). In scenarios where we may not be able to solely rely on the mobile network operator to deliver the patch to every device simultaneously, we study two alternative and complementary methods for patch distribution based on both social and temporal information, namely temporal centrality measures.

Being highly correlated with human contacts, understanding how such malware propagates requires an accurate analysis of the underlying time-varying network of contacts amongst individuals. State-of-the-art solutions on mobile malware containment have ignored two important temporal properties: firstly, the time order, frequency and duration of contacts; and secondly, the time of day a malicious message starts to spread and the delay of a patch [ZCZ+09, ZVL+09]. Instead, we argue that the temporal dimension is of key importance in devising effective solutions to this problem.

With this in mind, the focus of this study is to investigate the effectiveness of two containment strategies based on targetting key nodes, taking into account these temporal characteristics. We firstly investigate a traditional strategy, inspired by studies on error and attack tolerance of networks [AJB00], exploiting a static and a time-aware enhanced version of *betweenness centrality* which provide the best measure of nodes that mediate or bridge the most communication flows (as defined in Section 4.1.1). According to this strategy, the nodes that act as mediators are patched to *block* the path of a malicious message. However, due to temporal clustering and alternative temporal paths such strategies merely *slow* the malware and does not *stop* it; this was also observed in the previous study of mediators in the corporate email dataset (Section 4.2.1.6). In other words, a scheme based solely on immunisation of key nodes is not sufficient, instead *quick spreading* of the patch is necessary for most networks. We propose a solution based on opportunistic *spreading* of patches through Bluetooth, i.e., exploiting the same mechanism used by the malware itself. The key issue in this approach is to select the right nodes as starting points of the patching process. Temporal betweenness only provides a quantitative measure of the number of communication paths over time that go through a certain node and it proves to be sub-optimal metric for this. A metric capable of identifying nodes that

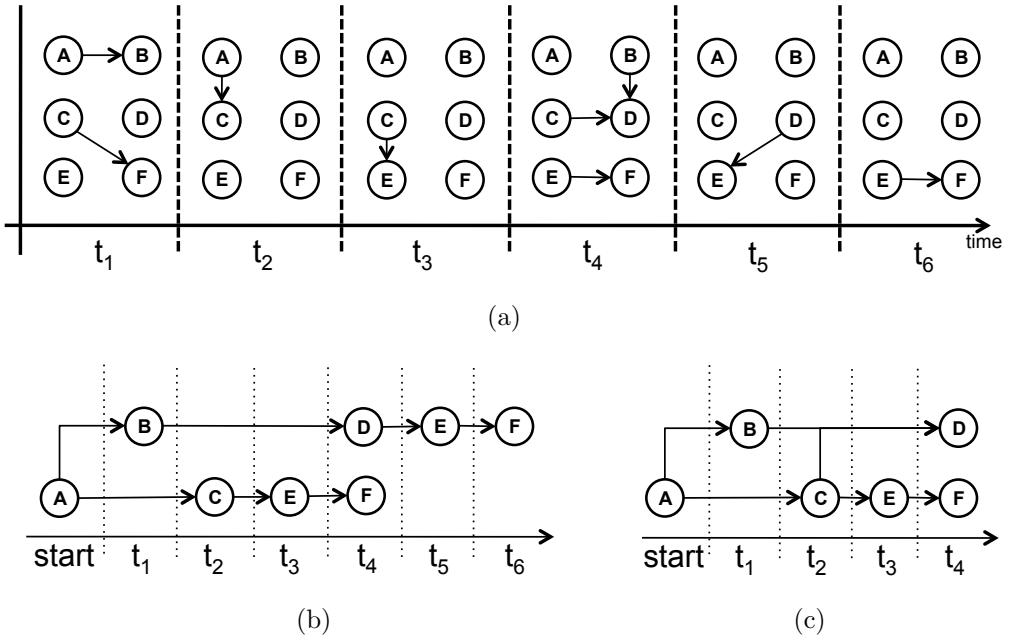


Figure 4.5: (a) Example Temporal Graph. (b) Two temporal paths from node A to node F. (c) Temporal minimum spanning tree with source node A showing shortest temporal paths to all other nodes.

can reach a large quantity of other nodes quickly is *temporal closeness centrality* (Section 4.1.2) which ranks nodes by the speed at which they can disseminate a message to all other nodes in the network. We show that this strategy can reduce the cellular network resource consumption and associated costs, achieving at the same time a complete containment of the malware in a limited amount of time.

4.2.2.2 Exploiting Temporal Centrality for Malware Containment

Let us consider a simple scenario where a person receives a malicious message on their device in the early hours of the morning and the malicious program replicates itself to any devices it meets during the day, for example at work and while socialising in the evening. A simple strategy consists of immunising only the nodes that mediate the most communication flows. *Betweenness centrality* (Section 4.1.1) can potentially help identify such nodes, however, we will show that this strategy is ineffective either through using a static or temporal metrics to find these path mediators. The intuition behind this is given through the example in Figure 4.5(a). Consider the

shortest temporal paths from node A to node F, namely (A, C, E, F) and a longer (both in terms of hops and time of delivery) temporal path (A, B, D, E, F), also illustrated in Figure 4.5(b). If we consider the simple case of patching a single node in an attempt to block the malware from spreading, the best choice would be node C, as the one on most temporal paths, however notice that node B provides an *alternative* path to F albeit a longer path.

Our second strategy relies on the ability to opportunistically spread a patch message quickly throughout the network; we utilise closeness centrality (Section 4.1.2), which is able to capture this property.

Exploiting Temporal Betweenness Centrality to Block the Paths of Mobile Malware

By definition, temporal betweenness centrality finds nodes that mediate between the most communication channels and, hence, their removal will have the greatest impact on the network overall communication efficiency. It follows that the first containment scheme can utilise this information to send a patch to these mediating devices, *blocking* a malicious message from using paths, which pass through these devices. As already mentioned, we will show in Section 4.2.2.5 that such a scheme is not effective due to many alternative paths which exist in real human contact traces. The presence of these alternative paths is due to social clusters during the day that requires a high number of nodes to be patched in order to stop and contain the malware.

Exploiting Temporal Closeness Centrality to Spread a Competitive Patch

An alternative scheme can be based on the selection of the best devices to start opportunistically *spreading* a patch message; the intuition is that a patch message, if started at the right device(s), can propagate faster than the malicious message. Closeness centrality fits this specification since it ranks nodes by their ability to spread a message quickly to the most nodes. Intuitively, this can be thought of as a *temporal* minimum spanning tree (see Figure 4.5(c)). We will show in Section 4.2.2.6 that such a scheme is indeed effective.

	CAMBRIDGE	INFOCOM	MIT
Environment	Office	Conference	Campus
N	18	78	100
Start Date	3 Feb '10	23 Apr '06	26 Jul '04
Duration	10 Days	5 days	280 days
Avg. contacts per day	1927	25796	231
Scanning Rate	30 sec	2 min	5 min

Table 4.3: Experimental Datasets

4.2.2.3 Evaluation Setup

We evaluate the design space of a time-aware containment scheme through a *trace-driven* simulation using as input the three datasets summarised in Table 4.3. We will examine the effects of four key factors: the starting time of the malware spreading process t_m and of the corresponding patching time t_p, the initial number of the infected nodes N_m and the initial number of patched nodes N_p. The top N_p devices are chosen according to the calculated temporal betweenness or temporal closeness centrality ranking from the temporal graph $\mathcal{G}^w(t_p, t_{max})$, where w is set to the finest window granularity, corresponding to the scanning rate of the devices in each dataset (e.g., 30 second windows for CAMBRIDGE). The N_m nodes that are initially infected with malicious messages are chosen uniformly randomly. The results are obtained by averaging over 100 runs for each N_p. The static centralities from the static aggregated graph over the time interval $[t_p, t_{max}]$ are also calculated for comparison.

Our evaluation is based on the following assumptions: firstly, when a node receives a patch message, it is immunised for the rest of the simulation (i.e., we assume that the malware does not mutate over time); secondly, there is always a successful file transfer between devices (errors in transmission can be taken into consideration in the assessment of the contention scheme without changing significantly the results of our work, assuming random transmission failures); thirdly, an attacker chooses nodes at random; and finally, we have no knowledge of which devices are compromised (otherwise the best scheme is to patch those devices immediately).

4.2.2.4 Effects of Time on Malware Spreading

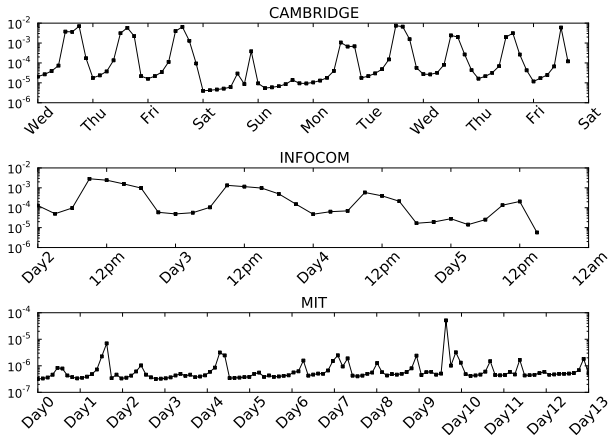


Figure 4.6: Temporal efficiency (y-axis) as a function of time (x-axis). Note the logarithmic y-axis.

Firstly, we briefly analyse the effects of the time of day have on mobile malware propagation. Let us consider Figure 4.6 where we measure the temporal efficiency (Formula 3.5) as a function of time. This *sliding* temporal efficiency is calculated for all three datasets. As we can see there are oscillations corresponding to the natural human periodic daily and weekly behaviour. For example, the CAMBRIDGE dataset is spread over 10 days, and it is apparent from the traces that a (malicious) message can spread more efficiently during the daytime, as opposed to evenings and weekends.

4.2.2.5 Non-Effectiveness of Betweenness based Patching

Starting from the results of the analysis of the effects time of day has on message spreading, we now evaluate the *best case* scenario for the containment scheme based on patching nodes (without spreading the patch) and we show that this is highly inefficient since it requires a very large number of nodes to be patched via the cellular network to be effective.

Using Day 4 of the INFOCOM trace for this example, a piece of malware is started at the beginning of the day (t_m=12am) and the device(s) are patched at the same time (t_p=12am). This is the best case scenario for two reasons: first, the temporal

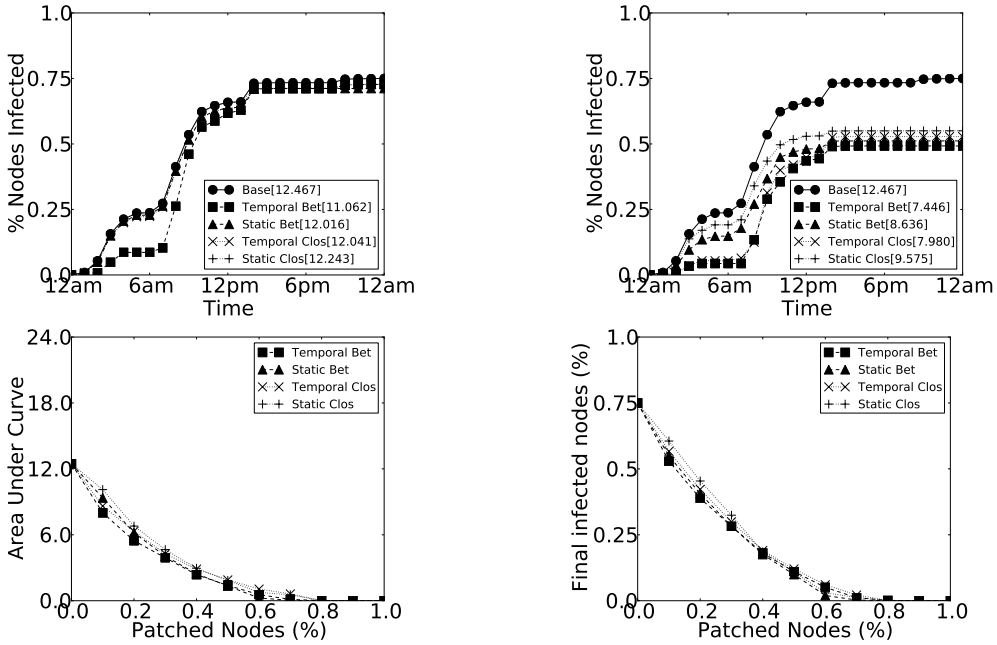


Figure 4.7: INFOCOM day 4: Immunising 1 (top left) & 10 source nodes (top right). Area under curves shown in the legend. Area (bottom left) and final % of infected nodes (bottom right), as we increase the % of nodes immunised (x-axis).

graph in the morning is characterised by low temporal efficiency since there are very few contacts, therefore, the malware spreads slowly (as we have seen in Figure 4.6); secondly, devices that are immunised immediately have the best chance of blocking malware spreading routes.

Figure 4.7 shows the ratio of compromised devices across time when the top 1 (top left panel) and top 10 (top right panel) devices are patched after being selected using betweenness and closeness. As we can see, temporal betweenness initially performs better than static betweenness and both temporal and static closeness (quantified by the difference in the area under each curve, shown in the legend). However, by 7am we observe a steep rise in the number of compromised devices and by the end of the day, all curves converge to the same point. We also note that *in both cases it is not possible to totally contain the malware, suggesting that more devices need to be patched.* Taking a broader view, Figure 4.7 shows the area under the curve (bottom left) and final ratio of nodes infected (bottom right) as we increase the number of patched devices. Clearly, even when the malware is started at the slowest time of day for communication, we still need to patch 80%

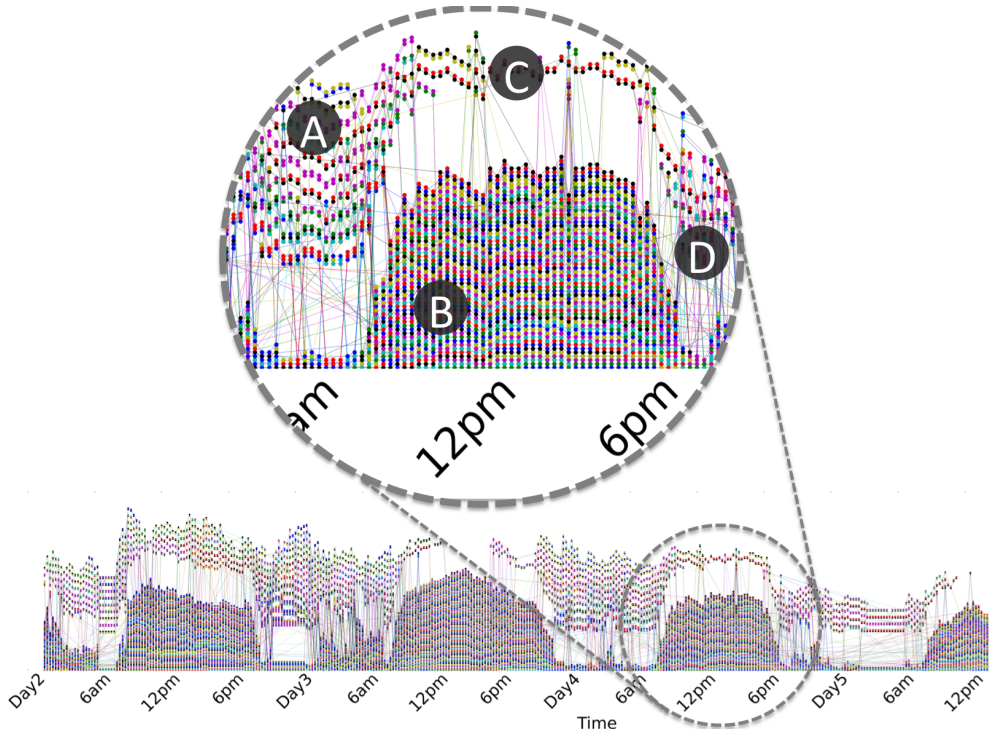


Figure 4.8: INFOCOM: Temporal clustering provide four types of alternative paths: (A) inflowing paths to temporal cluster; (B) redundant nodes in cluster; (C) alternative flows around temporal cluster; (D) many outflows to next temporal cluster.

of the devices before we can completely stop the malware from spreading; this can be considered an impractically high number of devices to patch. Similar high percentages are also required in the MIT trace with a minimum of 45% patched nodes. We can also conclude that in human contact networks, even with blocked nodes, it is only a matter of time before a (malicious) message disseminates to all nodes. To understand the reason for the effectiveness of (malicious) message propagation, we take a visual analysis approach: Figure 4.8 shows the *temporal activity diagram*[5] for the INFOCOM experiment across all four days. This gives a bird's eye view of proximity between individuals as they move between groups of colocated people across time, where the trajectory of the same node is given by a straight line. The horizontal axis is time and the vertical groupings of nodes represents people that are in the same static connected component such that there

[5]This plot was inspired by `http://xkcd.com/657`

is a path between every node in that cluster. The main feature to note is the *temporal cluster* of remarkable size that appears from around 7am until 7pm every day, coinciding with the main activities at the INFOCOM conference[6]. By means of this infographic, what we see are periodic clusters of nodes during the daytime and smaller disparate clusters during the evening. Figure 4.8 also zooms into Day 4, highlighting the four types of activity which give rise to temporal clustering and, more importantly, to alternative paths providing link redundancy for a message to pass through a network over time. *Since this strategy cannot deal with these alternative paths effectively, the propagation of a malicious message can merely be slowed down.* Hence, the rapid increase of infected nodes that can be observed in Figure 4.7 around 7am can be attributed to the presence of this large temporal cluster starting at 7am where many alternative paths are present and, therefore, the spreading cannot be stopped just patching some of the nodes. We conclude that this containment strategy is not efficient given the large number of patch messages it requires.

4.2.2.6 Effectiveness of Closeness based Patching (Worst Case Scenario)

Since the blocking based containment scheme is not effective, we now evaluate the closeness based *spreading* scheme with the aim of disseminating a patch message throughout the network more quickly than a malicious message. We start our analysis by examining a *worst case* scenario using the CAMBRIDGE dataset: a researcher receives a malicious message on their device in the early hours of Friday morning (t_m=Fri 12am) and the malicious program replicates itself to any devices it meets during the day. A patch message is started a day later to try patching all the compromised devices (t_p=Sat 12am). Again referring to Figure 4.6, this can be considered as a worst case since the malware is started during a day with high spreading efficiency and the patch is delayed until the weekend when the efficiency is low.

Figure 4.9 shows the spreading rate for the malicious message versus the best (left) and worst device (right) to start the patching message. These results were obtained by running simulations considering every single device as a starting point of the patching process, and then ranking them based on three *performance metrics*:

[6] `http://www.ieee-infocom.org/2006/technical_program.htm`

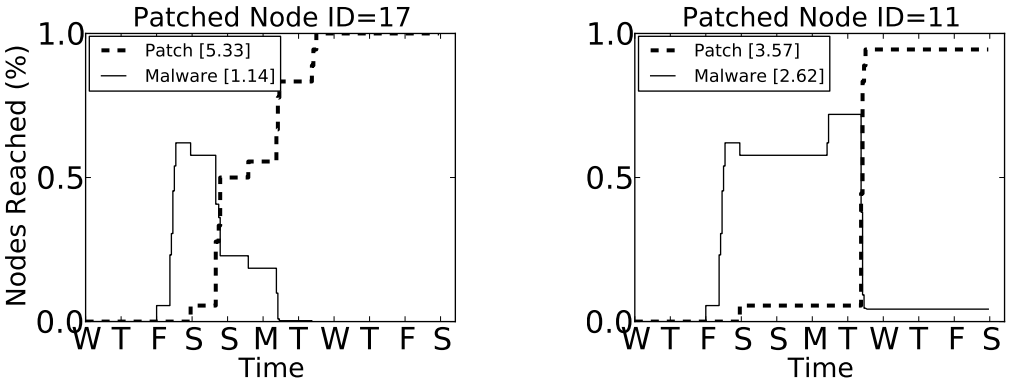


Figure 4.9: CAMBRIDGE [t_m=Fri 12am, t_p=Sat 12am] delivery rate (y-axis) starting a mobile worm from single node. Best case (left) and worse case patching node (right) shown. Area under curve presented in legend.

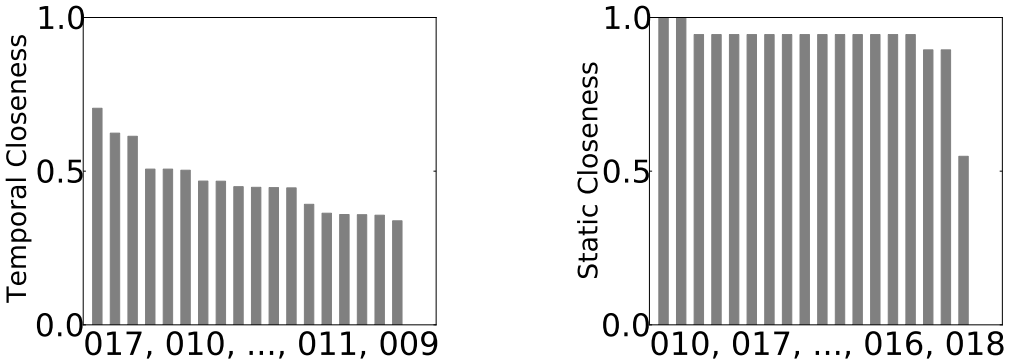


Figure 4.10: Temporal (left) and static (right) closeness centrality ranking for Figure 4.9. Top two and bottom two device IDs shown on x-axis. Nodes ranked left to right.

- the area under the curve (AUC), which captures the behaviour of the infection over time with respect to the number of infected devices[7];
- the peak number of compromised devices (I_{max});
- the time in days necessary to achieve total malware containment (τ).

Since the AUC captures both the I_{max} and τ, the best and worst initial devices that were patched were selected using the AUC. Comparing all three measures, the case related to the selection of the worst device (right panel) is characterised by double AUC (2.62 vs. 1.07); a higher peak in compromised devices I_{max} (68%

[7]The AUC is commonly used in epidemiology and medical trials [EI01].

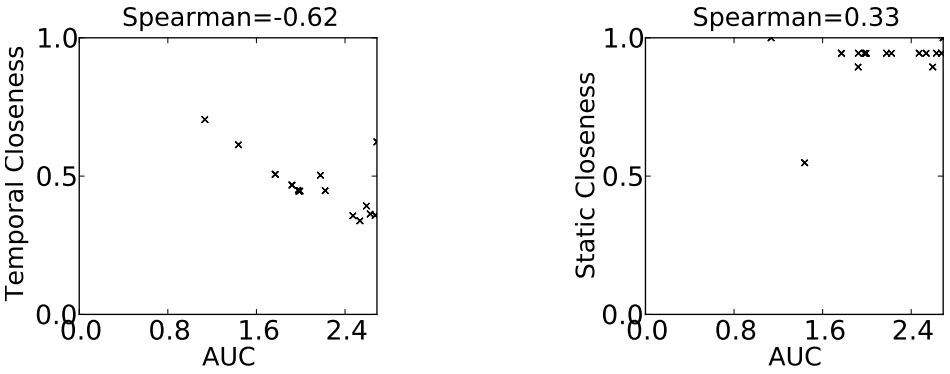


Figure 4.11: Correlation between AUC with temporal (left) and static (right) closeness centrality.

vs. 60%) and by the fact that it is not possible to fully contain the malware in a finite time τ (∞ vs. 3.3 days). Now comparing these observations with centrality, in Figure 4.10 we observe that the node characterised by the highest temporal closeness centrality (ID=17) is also the optimal one for spreading the patch and the node that leads to the worst performance (ID=11) is ranked within the bottom two nodes. This should be compared with static centrality, which ranks the best device to start the patching process (ID=17) in second place and the worst device (ID=11) seventh from the bottom (not shown). Also, the value of static centrality of each node is more uniformly distributed; a fact that can be attributed to the dense static graph previously observed in Figure 3.1(a). The stronger correlation between temporal closeness centrality and an effective malware containment scheme can be seen more clearly by plotting these rankings against the AUC in Figure 4.11. We expect a strong negative correlation since centrality values are ranked in *descending* order; by using temporal closeness centrality, we can identify the best node to start disseminating a patch message to contain a piece of mobile malware which fits our intuition that spreading a patch message quickly is the best containment strategy.

4.2.2.7 Effects of Temporal Variability

Thus far, we have only considered a single malware start time. We now take a broader view and examine the effects of varying malware start time (t_m) and patch delay (t_p). For each dataset the AUC, I_{max} and τ are exhaustively calculated for different malware start times at hourly intervals and increasing patch delays starting

from zero (i.e., patch messages start at the same time as malicious messages) to up to 2 days. We compare node selection based on temporal and static closeness to that of temporal and static betweenness. As a baseline, a naive method of randomly selecting patching nodes is also calculated, averaged over 100 runs.

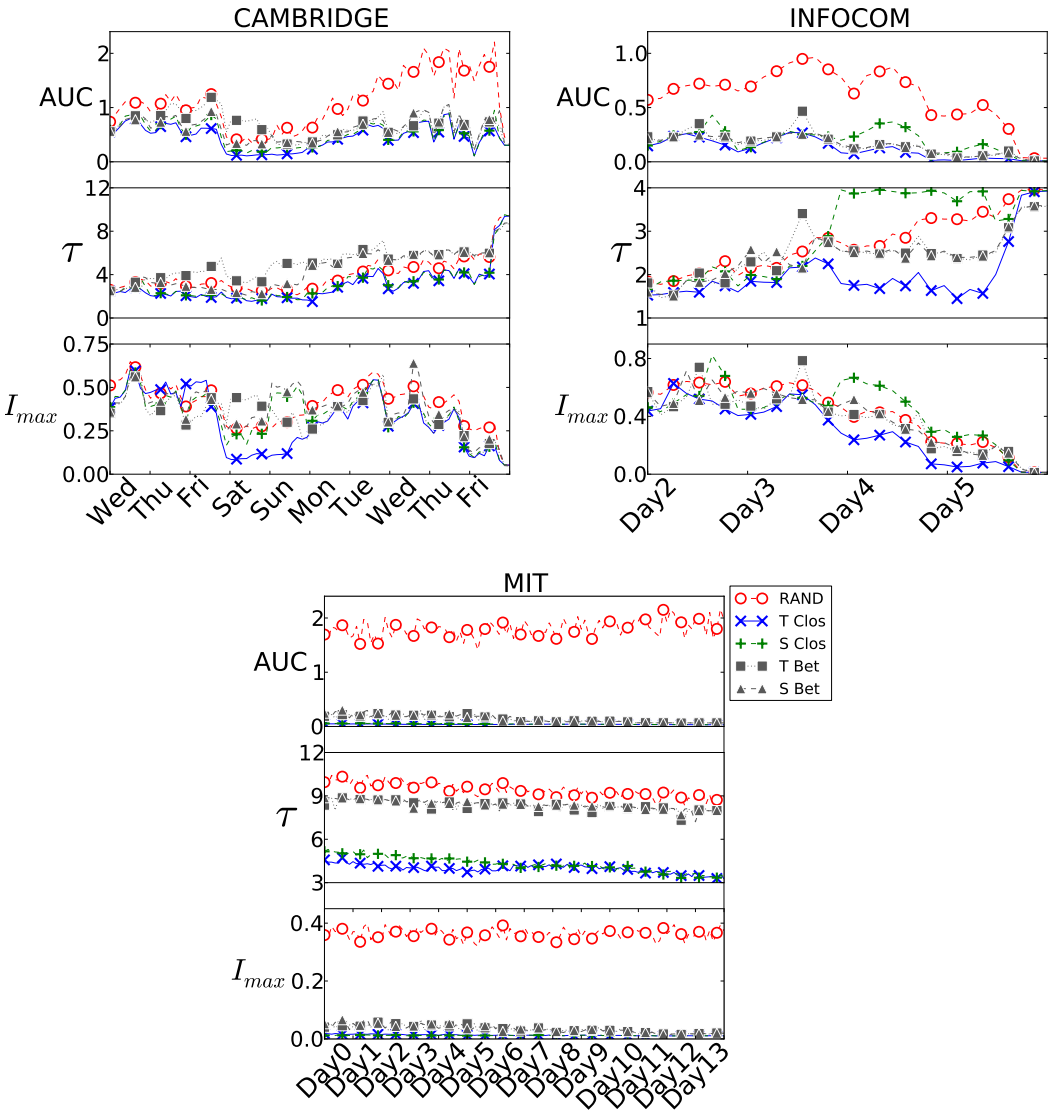


Figure 4.12: Performance of temporal, static and naive node selection, across different malware start times (x-axis), averaged over all patch delays.

Sensitivity to Malware Start Time

To understand the effects of a malicious message starting at different times, Figure 4.12 shows, for each dataset, the performance metrics as a function of the malware start time t_m, averaged over all patch delays. Firstly, referring back to the temporal efficiency from Figure 4.6, which exhibited daily peaks and troughs during the weekend, the AUC and the maximum number of infected nodes I_{max} tend to follow these same patterns (strictly related to human circadian rhythms); however, the total time of containment (τ) remains stable across all start times. These results demonstrate that this time-aware containment scheme is an effective method of quickly containing malware, irrespective of when the malware started. Now analysing the AUC and I_{max}, the temporal closeness centrality curve is consistently lower than static closeness, betweenness (both temporal and static) and naive methods. Further, betweenness (both static and temporal) generally take longer to fully contain the malware (higher values of τ) and static closeness centrality performs worse than the naive method at some points of time; more specifically:

- For the CAMBRIDGE dataset, during the weekend a static closeness method has a higher peak number of compromised devices (I_{max}) than the naive method, which shows that a static method is not effective at slowing down the malware from spreading.

- For the INFOCOM dataset, again I_{max} is higher than the naive method, during days 2 and 4. In addition, the AUC curve for a static method peaks with temporal efficiency during days 2, 4 and 5: this means that the malware is not contained effectively in these scenarios. Also, the total containment time (τ) is greater than that of the naive method during days 3, 4 and 5. This shows that temporal closeness centrality is more consistent at identifying the best nodes to start the patching process, compared to both static and naive methods.

- Finally, for the MIT dataset, the naive method performs extremely poorly (with high values of AUC, I_{max} and τ across all malware start times), compared to either a static or temporal methods. However, we also see that during the first week of the Fall semester, temporal closeness centrality identifies nodes

with lower AUC and τ, exhibiting over half a day quicker malware containment compared to static closeness centrality.

Sensitivity to Patch Delay

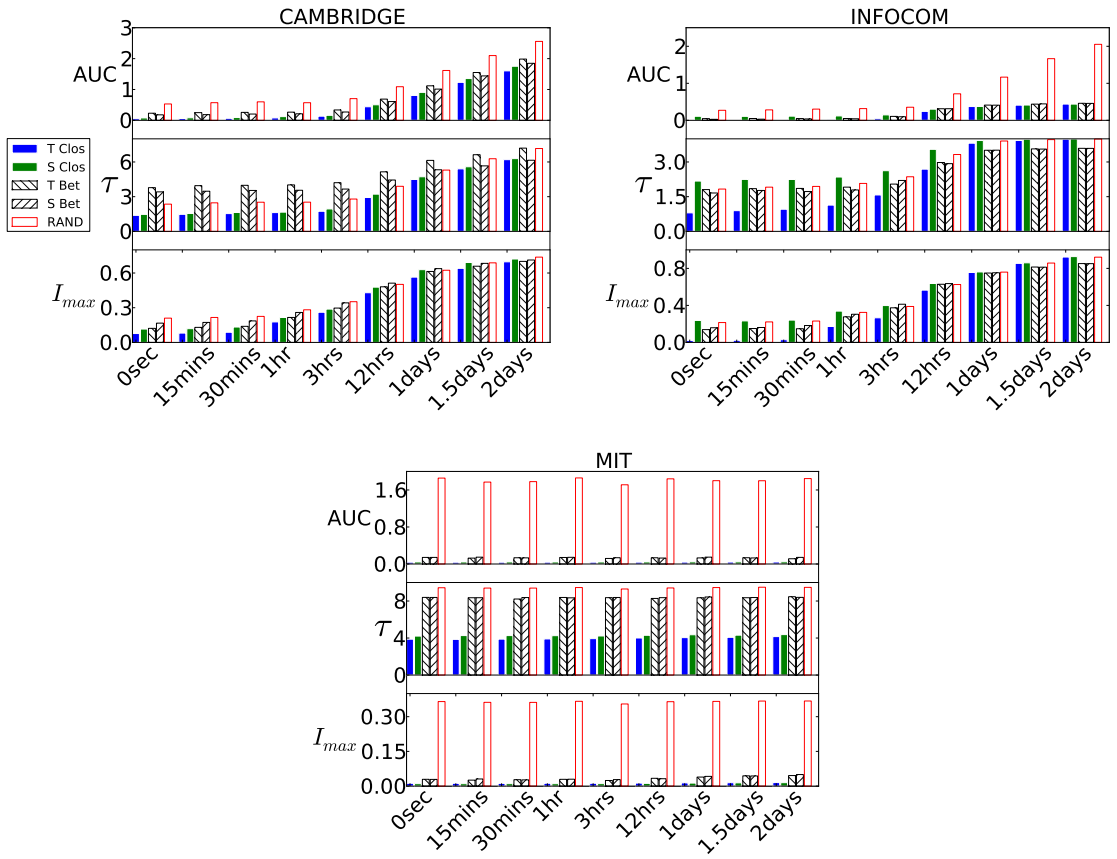


Figure 4.13: Performance of temporal, static and naive node selection methods, as a function of patch delay (x-axis), averaged over all malware start times.

To understand the effects of delaying a patch message after a malware outbreak, Figure 4.13 plots the performance metrics for a representative sample of patch delays, averaged over all malware start times. As the patch delay increases, all the performance indicators also increase. However, we note that across all three datasets, temporal closeness centrality (left most bar) exhibits the best results: smallest AUC, fastest total containment time (τ) and smallest peak compromised devices (I_{max}). We also observe that in the INFOCOM dataset, static closeness node selection gives

higher values of I_{max} and τ up to a 12 hour delay, showing that static centrality does not consistently capture the true *speed* at which a node can spread a message, compared to temporal closeness centrality. In addition, these plots demonstrate that betweenness (both static and temporal) are not suited to a spreading process and hence perform worse than closeness based node selection. Again, from these observations, we conclude that a containment scheme based on temporal closeness centrality provides the best performance as the patch delay increases.

Impact of the Initial Number of Compromised and Patching Devices

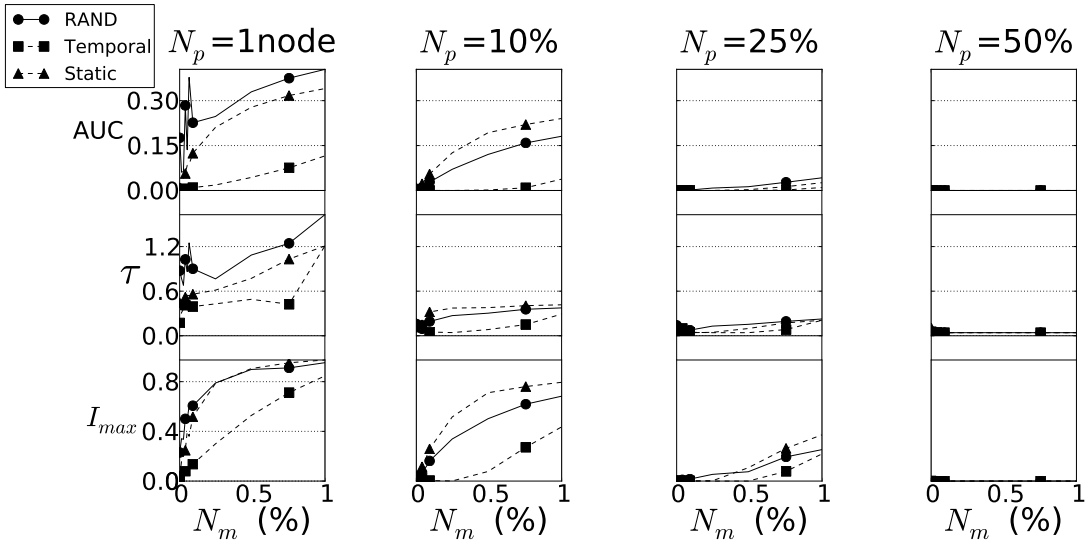


Figure 4.14: INFOCOM: Effect of increasing number of initial devices with malware (x-axis). From left to right, each column plots an increasing number of devices from which a patch is started (t_m=t_p=Day 4 12am).

We now look at the effects of starting malware messages (N_m) and patch messages (N_p) from more than one device. This corresponds to the case, for example, when a group of people download a malicious program at the same time, or an attacker has programmed the replication to be time-triggered. Since we have observed that betweenness based node selection is not suited to patch spreading scheme, we now focus on closeness based node selection only. To make comparisons with the first containment scheme (Section 4.2.2.5) we discuss result for the same malware start and patch delay times. Similar trends were found for different start times and other datasets. Figure 4.14 shows the effect of starting a patch from an increasing number

of initial devices N_p (increasing column left to right) as the number of initially compromised devices N_m (reported on the x-axis) is increased for the INFOCOM dataset.

First, in the case when a single initial patch message (N_p=1) is used (left panel), we observe that the AUC corresponding to the scheme based on temporal centrality is lower than that corresponding to the cases of static and naive methods of node selection even as N_m increases; the total containment time (τ) remains below half a day up to N_m=75% of the total number of nodes (which we indicate with N_{tot}) and the peak compromised devices (I_{max}) rises slowly as N_m increases. When increasing to N_p=10%N_{tot}, using temporal centrality the total containment time (I_{max}) drops below 2.5 hours (about 0.1 of a day) up to N_m=75%N_{tot}. Only at N_p=25%N_{tot} both the naive and static methods start to match the performance of the temporal method. These observations suggest that our time-aware containment scheme using temporal centrality is more accurate at ranking important nodes and hence a viable option for a network operator since less devices are required to receive a patching message in order to achieve an effective containment strategy.

4.2.2.8 Discussion

This study has motivated and investigated the effectiveness of a time-aware mobile malware containment scheme using temporal centrality to identify the best node to start a competitive patch message. The evaluation on three real human contact traces has shown that this time-aware scheme can more consistently identify the best devices to start such a patch across different malware start times and patch delays, compared to static and random node identification. As we discussed earlier, dynamic processes are intrinsically linked to the underlying dynamic network topology. Since we do not have the real information of a short range malware spreading, we have simulated this on top of the known topological contact sequence of mobile devices. For this to work we have modelled the malware spreading as shortest paths using epidemic spreading; this is reasonable since we imagine that the goal of many types of malicious worms is to spread quickly and to infect as many devices as possible. We have also assumed that the malware spreading process is independent from the contact process; for short range mobile malware, the user is unlikely to know if their device is infected and hence their future contact behaviour is unlikely to be affected.

An possible direction for future work is the study of how the underlying contact process affects the spread of a virus, for example, in a real biological viruses, such as influenza, the effects of people changing their daily meetings to avoid friends who are infected may impact the spread of the virus. This has been recently studied using a mean-field model in static networks [KL11, KL10]; an interesting extension would be to time-varying networks, possibly starting with empirical data collection of node perception and changes to regular contact processes.

4.3 Related work

In his work on temporal paths, Moody [Moo02] first mentioned the possibility of temporal extensions to centrality measures as possible future work, though this was never formalised. More recently in the study of ecological networks where coarse grained seasonal snapshots of predator-prey networks have been available for some time, Jordán et. al. [JOBL08] examined the relationship between static aggregated graphs and temporal snapshots. In their study, degree, closeness and betweenness centralities were calculated on the static aggregated graph and again independently on each topological snapshot. They found large variations in centralities between static and snapshot graph models and trends over time were missed by static analysis. This study is different from our techniques since centralities are calculated separately on each time window, whereas our proposed technique captures the time dependencies across time windows, however, their insights into the inaccuracy of static analysis corroborates the results seen in this thesis.

Grindrod et. al. [GPHE11] formalise an eigenvector centrality, namely *katz centrality*, on temporal networks. Katz centrality is similar to closeness centrality in that it measures the paths from a source node to other nodes in the network, however, katz centrality captures paths of *all* lengths in addition to the shortest paths. In this thesis, the focus has been on the shortest path of dissemination, which is appropriate in the application under study, for example in short-range mobile malware propagation where the malware spreads via the shortest route; however, it would be interesting in future work to apply and compare temporal katz centrality to temporal closeness and temporal betweenness in a wider range of applications.

Within computer science research into DTN's, there are two notable studies [DH09,

HCY11] into exploiting social properties of human contact networks for message delivery in "pocket switched networks" (PSN). The goal of these studies is the delivery of a message from a source to known destination node through decentralised algorithms. Related to our thesis is the use of social network analysis for message delivery in PSN's, both techniques rely on some measure of node important (to help bridge between separate clusters of nodes) and a measure of destination node similarity (to guide the message to the right node). Daly et. al. [DH09] utilise ego-centrality version of betweenness to find cluster bridges and a simple overlapping neighbour measure (i.e. Jaccard index) of destination similarity, both in a decentralised manner. Hui et. al [HCY11] proposed a similar solution but tackled the problem from a different perspective using both community detection and centrality. Their algorithm uses the most important nodes both globally and within communities to decide on the next hop. They propose an algorithm to identify the most central nodes (RANK) using the number of shortest delay paths that pass through a node, however this does not take into account the fraction of alternative paths and also they present a strong correlation between such central nodes with degree. In the end, their proposed algorithm favoured an ageing degree centrality since it is suited for a decentralised algorithm.

In this chapter, we have evaluated two different types of centrality, namely betweenness taking into account alternative paths and closeness to find nodes that can propagate messages quickest in the context of containing mobile malware. The goal of mobile malware containment or rumour spreading is different from opportunistic forwarding, in that the latter is targets delivery of a message whilst minimising overheads (such as power and memory); in malware containment flooding to all nodes in the fastest possible time is required. Future work could study the limits of resource aware flooding for mobile malware patch dissemination.

4.4 Conclusions

In this chapter, we have introduced the notion temporal closeness and betweenness centrality for the study of key information spreaders and mediators. We have shown that firstly, from a contextual perspective, temporal centrality identifies nodes that intuitively fit the context of the dataset; and secondly, from a dynamic process

perspective, these highly ranked nodes can be exploited in containing short range mobile malware. Clearly both patching schemes rely on centralised knowledge and require "oracle" knowledge of future contacts between devices. In the next section, we shall develop tools that eliminate this latter requirement by predicting nodes that possess high temporal closeness now, based on past observations.

5

Predicting Information Spreaders in Temporal Graphs

Introduction

In this chapter, we develop a technique to help analyse the predictability of temporal centrality in dynamic human contact networks. Our previous evaluations of temporal distance metrics were sampled across different points over the whole of the network dataset [1] this has enabled us to uncover clear patterns in the efficiency of information dissemination at different points of time (Figure 4.6). This in turn has motivated this chapter, to understand whether we can take advantage of these patterns in the predictability of temporal centrality rankings.

Forming the core of this technique is the ability to apply a well studied *descriptive analysis* visualisation, namely correlograms [Cha03], to find patterns in centrality

[1] As discussed in Section 3.3.3, past work only took a single measurement from the beginning of a network dataset.

rankings over time. Given a time-series of values, a correlogram plots the self similarity between two time-points as we vary the time difference, known as a *lag*; this allows us to uncover any patterns over time. The main contribution of this chapter is to apply the same analysis, but on a time-series of rankings, namely temporal centrality rankings calculated at each time window. The technique is general enough for analysing many different types of time-varying networks, however, we continue our study of short range mobile malware in human contact networks; this is founded on the hypothesis that a central node yesterday is highly likely to be central today. From a practical point of view, the ability to predict highly ranked temporal centrality nodes is useful in real applications of information dissemination.

To allow us to plot a correlogram using a time-series of lists, we need to make an assumptions: given a time-series with a ranking associated with each time-point, we relax this ranking requirement within the top-k ranked elements. By relaxing this condition, we can treat the top-k elements as a *set* and then the similarity function between two given time-points is the number of intersecting nodes (e.g., the Jaccard index); from this, we can plot the correlogram without modification. This simplification makes sense in the application to information dissemination: once we have selected k nodes to start spreading a message then the ordering is irrelevant.

Chapter Layout

To analyse the predictability of temporal centrality in dynamic networks we first define the top-k prediction model which enables us to plot a correlogram between time windows in a dynamic graph (Section (5.1). This allows us to analyse the predictability of temporal centrality rankings and we see that there are simple ageing and periodic correlations (Section 5.2) which inform our prediction function design (Section (5.2.3). We then evaluate the predicted nodes in short range mobile malware containment with those found with full knowledge of future contacts and random node selection in Section (5.3). Since we found that a containment scheme based on opportunistically spreading a patch starting from highly ranked temporal closeness nodes is effective, we focus our analysis on temporal closeness prediction and this type of containment scheme.

We find that, firstly, the set of top-k temporal closeness centrality nodes are cor-

related with past time windows (up to two days); and secondly, that simple and efficient prediction functions can be designed to select the set of top-k nodes, optimal for patch spreading. We compare the predicted devices with those found in the previous chapter with full knowledge of future contacts.

5.1 Top-k Prediction Model

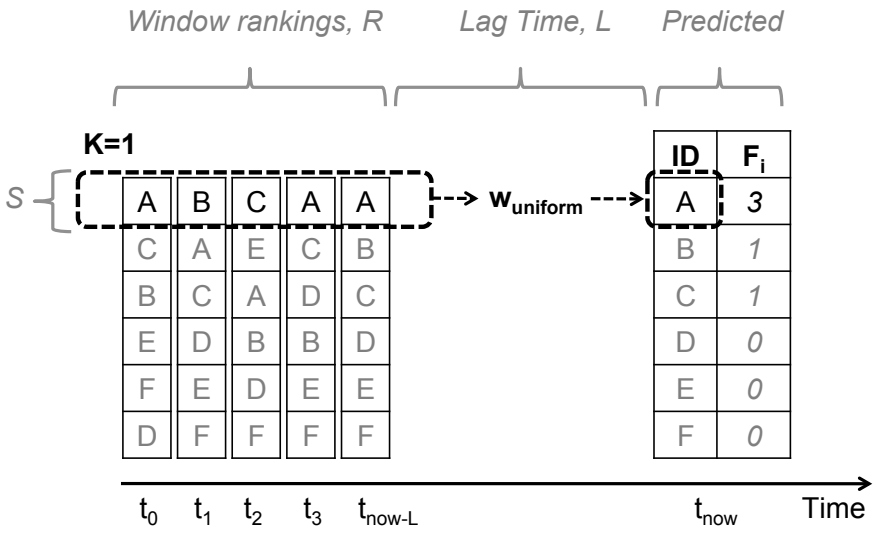
The top-k prediction model captures the problem of identifying the top-k nodes to start spreading a patch, starting from the current instance of time. Intuitively, using past observations, this prediction is based on the number of times a node i is in the set of top-k central nodes in the previous intervals of time. This frequency of observation can also be weighted by considering the time difference relative to the current time (i.e., more recent observations could have higher weighting or vice versa, etc.). More precisely, in Section 5.2.2 we will provide experimental evidence that frequency can be used as an estimator for predicting the likelihood of having a certain node as one of the top-k central nodes in the future.

5.1.1 Example

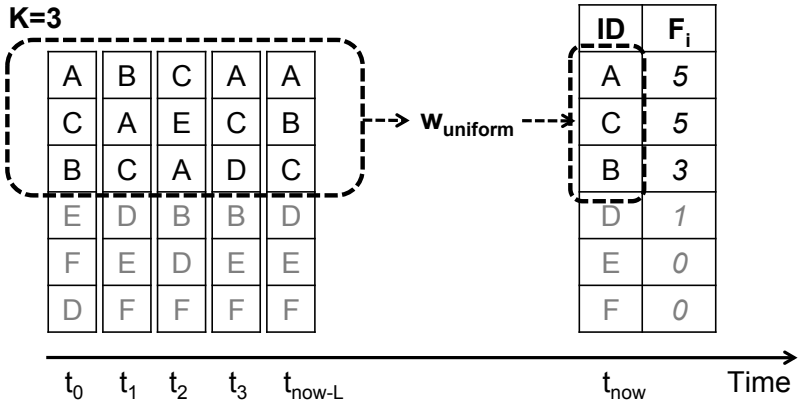
To illustrate this idea, Figure 5.1(a) depicts the problem of predicting the top $k = 1$ node at the current time t_{now}. For each time window (x-axis), we construct an ordered list of node ids, for example, at time t_0, nodes are ranked by a centrality measure as (A, C, B, E, F, D). Since we may not have the most recent information of contacts and node rankings, there is a lag time L between t_{now} and the last training window at t_{now-L}.

Using this model, we define a suitable weighting function on the top-k set of nodes in these past windows; this shall be discussed in further detail in Section 5.2.3, however, for now, consider a simple *uniform* weighting function W_{uniform}, where all training windows are treated equally. In this case since node A appears three times across the training windows, it has the top weight and would be sent the patch.

Extending this to $k = 3$, again we iterate over all training windows, weighting the top 3 nodes accordingly. Notice again that node A is predicted to be in the top 3 nodes, along with node C and B; a patch is sent to all three devices.



(a) Example for k=1



(b) Example for k=3

Figure 5.1: Example of the top-k set membership prediction problem, using *uniform weighted* frequency selection.

Finally, patching additional nodes might provide a limited benefit in some cases. For example, if there are two temporal connected components, the top-2 nodes may belong to the same component. If the infection is started also from this additional node, the benefit will be incremental, since both nodes are members of the same connected component. However, this proposed scheme does allow for *redundancy* that might be very useful given the inherent uncertainty of predictions. We shall see in Section 5.3.5 that temporal centrality requires a smaller value of k for an effective containment scheme.

5.1.1.1 Definitions

More formally, given a top-k, lag time L and current time t_{now}, we first construct the temporal graph $\mathcal{G}(t_0, t_{now} - L)$ from the uploaded contact data. Next for every graph $G_t \in \mathcal{G}$ at time t, we calculate the temporal centrality C_t using $\mathcal{G}(t, t_{now} - L)$. From this we construct the list of *window centrality rankings* $R(t_0, t_{now-L})$ for each time window in the interval $[t_0, t_{now-L}]$. Each window centrality ranking $r^t \in R$ at time t is an ordered list of N node identifiers ranked by temporal centrality using C_t. Next, we construct the list of *top-k window centrality rankings* $S^k = (s^0 ... s^{W-1})$, where s^t corresponds to the ordered set of the top-k centrality nodes in the window ranking r^t.

From this, given the top-k sets $S^k(t_0, t_{now-L})$, for each node i, its *weighted frequency value* F_i is defined as:

$$F_i = \sum_{t=0}^{W-1} z_i^t w(d), \qquad z_i^t = \begin{cases} 1 & \text{if } i \in s^t \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where z_i^t is used to count the presence of node i in the top-k set s^k, $d = t_{now} - t$ is the difference between the time to be predicted and the training window and $w(d)$ is an aging function used to assign different values to the presence of the node in the set of the top-k nodes in a certain window.

Then the nodes are sorted in descending order by their value of F_i and the top-k are selected for patching. In the previous example a uniform weighting $w(d) = 1$ was described. Note that although contact uploads could be staggered between different devices, we consider a uniform lag time (for example, all nodes uploaded at the same time yesterday). This is reasonable since any extra recent information increases prediction accuracy. In Section 5.2.3 we shall describe more refined prediction functions.

5.1.2 Parameters

There are two related parameters that are fundamental to the setup of the prediction framework, firstly, the *training interval* size defines the interval $[t_0, t_{now-L}]$; and, secondly, the *upload interval* defines how frequently mobile devices upload contacts

to the server. A larger upload interval will decrease the freshness of the contact data and increase the lag time L. We envisage that devices can connect to WiFi or desktop sync managers to reduce data costs of upload, however since a patch needs to be distributed as early as possible, the cellular network is utilised instead to target the top-k set of devices. In our simulations (Section 5.3) we investigate hourly and daily uploads.

5.2 Predictability of Human Contact Traces

The derivation of our prediction functions is founded on the hypothesis that since human mobility is highly regular [CE07], a central person today is highly likely to be central at some point in the future. To test this, we utilise the same three mobile phone contact traces evaluated in the previous study on mobile malware (Table 4.3), namely CAMBRIDGE, INFOCOM, MIT, which can be classified as office, conference and campus environments, respectively. Again, for the CAMBRIDGE dataset, all 10 days are used as part of the evaluation; for the INFOCOM dataset, since devices were not handed out to participants until late afternoon during the first day, only the last 4 days are used; and for the MIT dataset, we show results for the first two weeks of the Fall semester[2] representing a typical fortnight of activity. The most important characteristic is the density, described by the average number of contacts per day. Indeed, since the INFOCOM dataset is extracted from a confined conference environment with scheduled talks, they are *temporally denser* compared to the campus and office settings.

5.2.1 Top-k Correlation Function

To test our hypothesis, we first define a *correlation function* to measure the similarity of top ranking nodes between different windows. Building on definitions in Section 5.1.1.1, given a sequence of top-k centrality window sets $S^k(t_{min}, t_{max})$, we simply use the Jaccard index between any two given window sets $s^a, s^b \in S^k$, where $k = |s^a \cup s^b|$:

$$A^k_{a,b} = \frac{|s^a \cap s^b|}{|s^a \cup s^b|} \tag{5.2}$$

[2]http://web.mit.edu/registrar/www/calendar0405.html

(a) CAMBRIDGE

(b) INFOCOM

(c) MIT

Figure 5.2: Plotting self-similarity between the rankings related to all windows a with all windows $b \leq a$, averaged by time difference $d = a - b$ (x-axis). Static (left column) and temporal (right) centralities plotted. Legacy correlation is observed with both static and temporal centrality.

5.2.2 Testing for Top-k Correlations

We now measure the self-similarity between the rankings for different time windows, by first calculating the complete sequence of window centrality rankings $S(t_{min}, t_{max})$ for each dataset, and then plotting the correlation function $A_{a,b}$ for every training window $s^a \in S$ against a past window $s^b \in T$ where $b \leq a$. We repeat this for different values of k. Figure 5.2 plots the time difference $d = a - b$ across the x-axis against $A_{a,b}$ on the y-axis, averaged by d. First, we notice that, as expected, as we increase k the correlation function A also increases. However, we also notice

Function	$w(d)$
Uniform	1
W-log(d)	$\log(d+2)^{-1}$
W-sqrt(d)	$(\sqrt{d+1})^{-1}$
W-exp(d)	$(2^d)^{-1}$

Table 5.1: Prediction functions with time difference, d.

across both static and temporal closeness centralities, there is a clear *legacy* effect in that top-k nodes are stable for some consecutive time windows (around a day in all traces). The peak at around 10 days in the CAMBRIDGE dataset can be attributed to the devices being collected and physically colocated at the end of the experiment. We also tested these correlations against a null model where we randomly shuffle the windows and calculate the same correlation function A: we found $< 2\%$ correlation for top-75% nodes uniformly across different time differences.

5.2.3 Prediction Function Design

Our aim is to predict the top-k ranked nodes from which to spread the patch by taking advantage of the knowledge about the previous evolution over time of the network. By making use of past observations, this prediction is based on the number of times a node i is in the set of top-k nodes which can also be weighted by the time difference relative to the current time. Since we have observed both a strong correlation with recent past windows (in all centralities) we design empirical functions that weight past windows by distance in time.

We now describe four possible prediction functions based on a weighted average characterised by different complexity. These functions are summarised in Table 5.1.

From our observations of a strong correlation with recent time windows, we can assign an age weighted function to a nodes membership in a previous time window t_i with time difference $d = (t_{now} - t_i)$: **W-log**(d), **W-sqrt**(d), **W-**(d), and **W-exp**(d). In addition, we also compare to a simple option that weights all previous set membership equally: **Uniform**. Note that these functions can be computed in

$O(M)$ for one prediction of $w(d)$ where M is the number of training time windows used.

Our approach has two key advantages: (1) it is simple to implement and deploy since we only require the past centrality values of *mobile services*, rather than tracing the whole past geometric positions of nodes; (2) it requires linear time to approximate network centrality. Our strategies are thus useful for large-scale and online computation – training data can be frequently updated in real time.

5.3 Application to Real Networks

We return to short range mobile malware containment application and compare with the results obtained with full knowledge of contacts and random node selection. As such, the simulation setup is the same as described in Section 4.2.2.3.

5.3.1 Parameters and Evaluation Metrics

We employ the same three performance metrics, namely the area under the curve, AUC; the total malware containment time, τ (days); and the peak number of compromised devices, I_{max}.

These three performance metrics are utilised to investigate several parameters:

- Malware Start: the time at which malware is deployed, starting every 3 hours of each trace.
- Patch Delay: the delay before a patch is ready to be deployed from {1 hr, 3 hrs, 24 hrs, 48 hrs}.
- Upload Interval: the frequency of mobile device contact uploads {1 hr, 24 hrs}
- Initial number of compromised devices and number of devices we start a patch from.

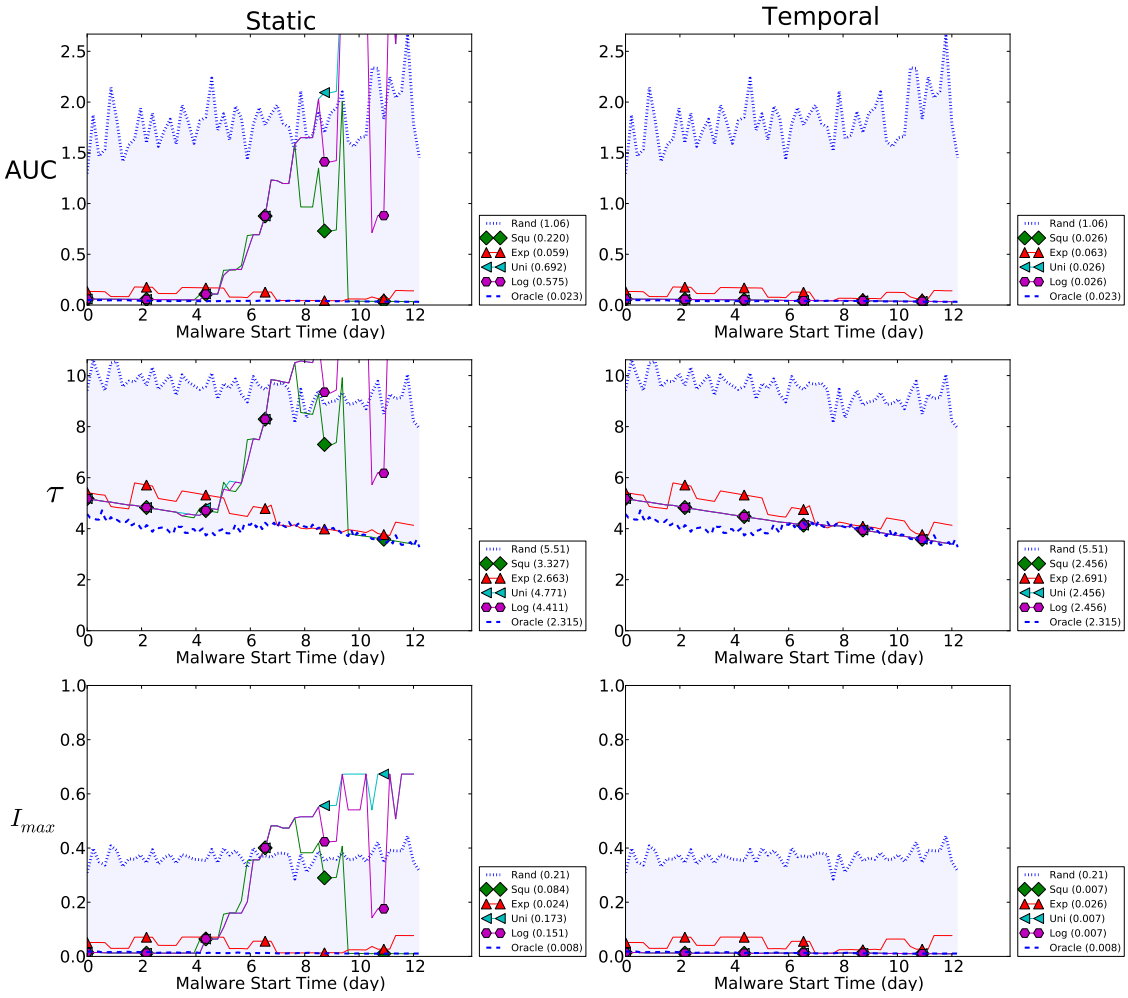


Figure 5.3: MIT Traces: Comparison of Centrality type vs. Prediction Function, as a function of Start Time (x-axis). Rand and Oracle node selection provide upper and lower performance bounds.

5.3.2 Effect of Malware Start Time

Figures 5.3, 5.4 and 5.5 plots for MIT, INFOCOM and CAMBRIDGE datasets, respectively, the effects of disseminating malware starting from a single device at different times (x-axis) during the trace. We fix the upload interval to 1 day and average across all delay times. For static and temporal closeness centrality measures, each plot shows how different prediction functions perform when selecting a single nodes to start spreading the patch. We plot curves for naive random patch

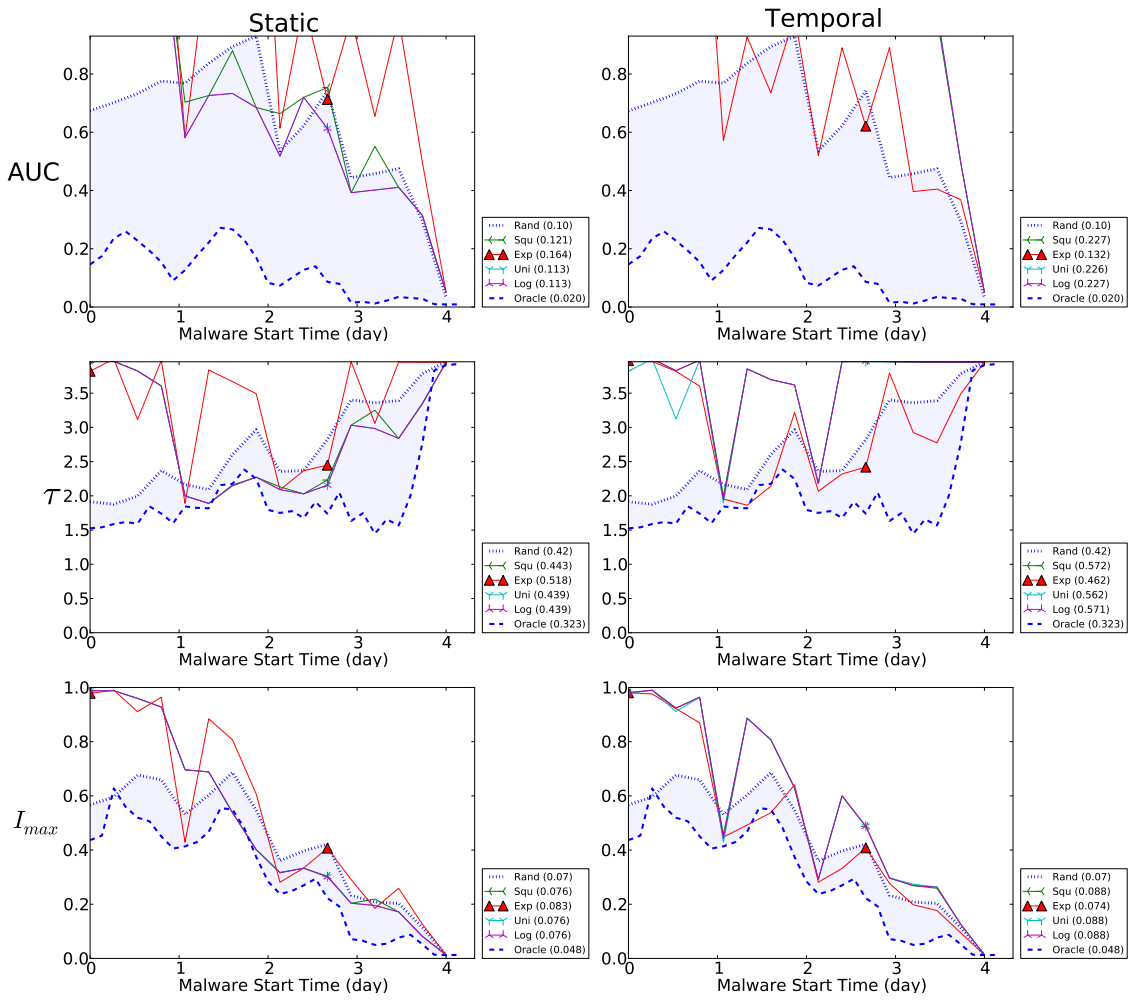


Figure 5.4: INFOCOM Traces: Comparison of Centrality type vs. Prediction Function, as a function of Start Time (x-axis).

device selection and an oracle device selection, corresponding to the case of temporal closeness with knowledge of all future contacts which was previously shown to be the most effective for opportunistic patch dissemination 4.2.2. Note that these two curves provide an upper and lower bound to the performance we would expect from an accurate prediction function for with static or temporal centrality. As such, to compare between curves, we present the area under each curve in the legend.

First, notice that there is a significant improvement over a random node selection and that the performance of devices selected approaches that of the oracle. Second, notice that both static and temporal centrality are highly accurate across all predic-

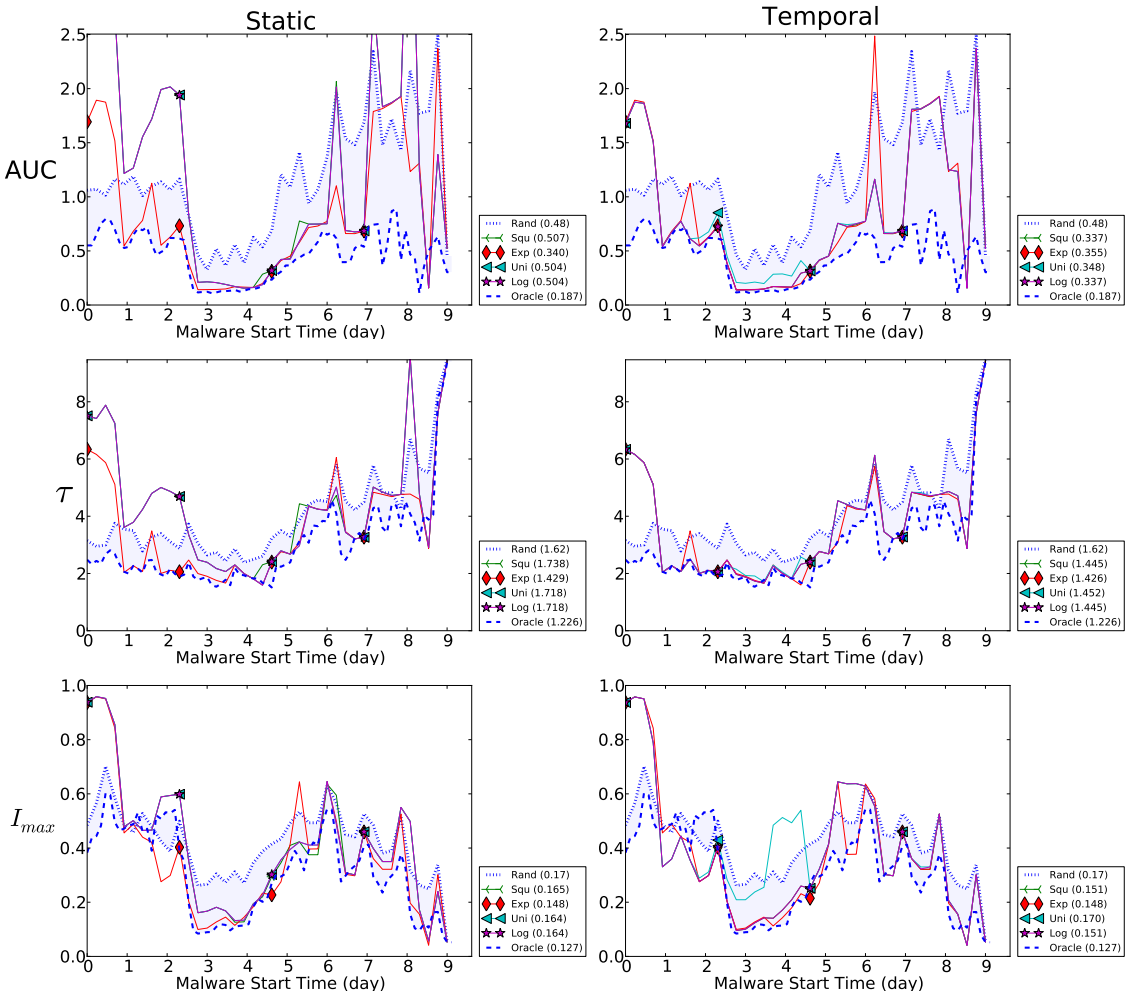


Figure 5.5: CAMBRIDGE Traces: Comparison of Centrality type vs. Prediction Function, as a function of Start Time (x-axis).

tion types, however, static is only accurate when using W-exp. Third, comparing the best prediction function between static and temporal centralities, temporal has a lower AUC, translating to better patching performance across different start times. Fourth, in the MIT and CAMBRIDGE datasets, all methods take around 150 hours (6.25 days) to fully contain the malware and around 60 hours (2.5 days) for INFOCOM. In addition, notice that for MIT, $I_{max} < 10\%$ when using W-log weighting for static and temporal and $I_{max} < 50\%$ for CAMBRIDGE and INFOCOM: this fits our aim of spreading the patch to as many nodes quickly and rely on the natural chain of human contacts to eventually trickle the patch to remaining devices over time.

	CAMBRIDGE		INFOCOM		MIT	
Model	Static	Temporal	Static	Temporal	Static	Temporal
Uniform	0.504	0.348	0.113	0.226	0.692	0.026
W-Exp	**0.340**	0.355	0.164	**0.132**	**0.059**	0.063
W-Log	0.504	**0.337**	**0.113**	0.227	0.575	**0.026**
W-Squ	0.507	0.337	0.121	0.227	0.22	0.026
Best	W-Exp	W-Log	W-Log	W-Exp	W-Exp	W-Log
Oracle	0.187		0.020		0.023	
Overhead	1.817x	1.802x	5.812x	6.774x	2.563x	1.155x

Table 5.2: Comparing Centrality vs. Prediction function, measured by AUC of all start times averaged over all lag times.

Finally, common across all centrality types, there are small peaks around noon for τ and I_{max} and troughs during the evening, which demonstrates that a time-aware approach is required since malware has more opportunity to spread during the daytime; this is most apparent when observing random node selection.

We enumerate in Table 5.2 the AUC for all centrality prediction pairs for all datasets. There is no single perfect choice prediction function that is best for all centralities; however, the centrality prediction pairs that minimise the AUC can be used as a first approximation (shown in bold). Note that there is more than one best prediction function for temporal centrality in the MIT dataset; however, we use W-log since it is the best performing overall for temporal centrality across all datasets. Now, comparing the best performing centrality prediction combination between datasets, we observe that the temporal approach performs best to minimise AUC in CAMBRIDGE and MIT datasets, however, static has more accurate prediction for INFOCOM. This suggests that, in confined spaces, with denser contacts a static model may be best suited; however, temporal can still be relied on across all scenarios to contain malware in a finite time and in most cases perform better than static. Quantifying the overhead of the best centrality prediction combination with the oracle, using temporal centrality we can achieve up to 1.155x accuracy in the best case and also on average across the three scenarios temporal centrality has the lowest overheads.

5.3.3 Increasing Patch Delay

Figure 5.6 plots the best centrality-prediction pairs, binned by increasing patch delays (x-axis), for the CAMBRIDGE, INFOCOM and MIT datasets, respectively. Increasing the patch delay is detrimental to malware containment, increasing the AUC, time of total containment and peak infected devices. Across all datasets, this is most prominent in the conference (INFOCOM) environment that again can be attributed to the confined space that increases the malware spreading rate and again suits a static model better than temporal centrality. However, for CAMBRIDGE and MIT, temporal centrality outperforms static device selection.

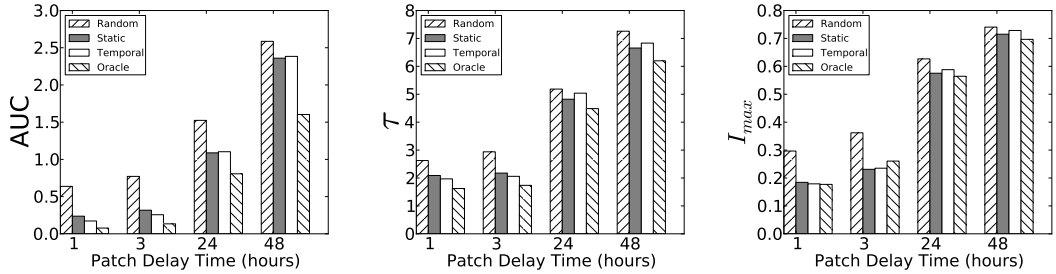
5.3.4 Effects of Contact Upload Interval

Thus far, we have considered a daily upload interval; Figure 5.7 again plots an increasing patch delay for the CAMBRIDGE dataset (compared with Figure 5.6(a)) but for an hourly upload interval. We note two things: firstly, there is very little improvement from a daily upload, and secondly, static methods have improved more than temporal. This suggests that these prediction functions are still able to perform accurately even with missing data (increased lag time, L).

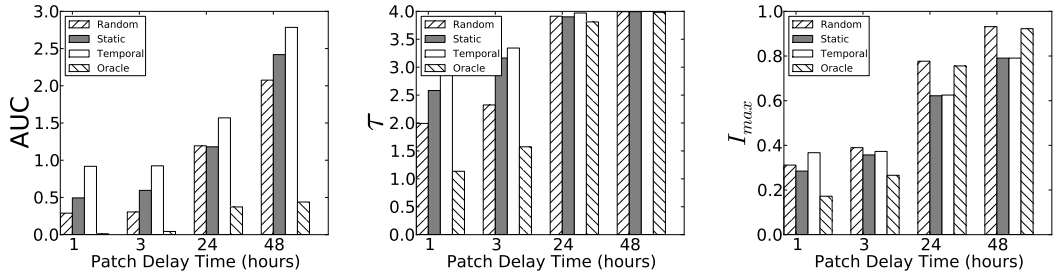
5.3.5 Varying initial compromised and patched devices

To understand the effects of increasing the number of initially infected devices I_n and increasing top-k patched devices, we fix the malware start time to day 2 at midday (the most damaging time of day for malware spreading), upload time to 1 hour and patch delay to 3 hours. Figure 5.8 plots the percentages of initially infected devices (equal to 10% (left column), 25% (middle) and 50% (right)) with increasing top-k patched devices (x-axis) using the MIT dataset.

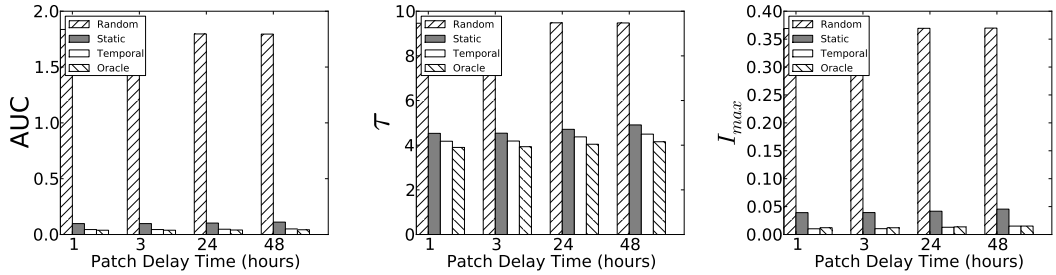
Starting with a low $I_n = 10\%$, containment is effective with an equally low value for $k = 5\%$. Increasing the number of devices to spread the patch past $k = 10\%$ does not add performance gains. Notice that temporal centrality is able to contain the malware within $\tau = 10$ hours, compared with static centrality which take around 75 hours.



(a) CAMBRIDGE



(b) INFOCOM



(c) MIT

Figure 5.6: Best centrality-prediction binned by patch delay (upload interval 24 hours).

With $I_n = 25\%$, again temporal offers advantages using a low value of k. However, as we increase the value of k to 10% then static and temporal are very similar; this is more apparent when $I_n = 50\%$. From this, we observe that temporal can more effectively select a smaller set of devices compared to static methods. This is a useful property, because one of the requirements is the minimisation of the number of devices required to receive the initial patch.

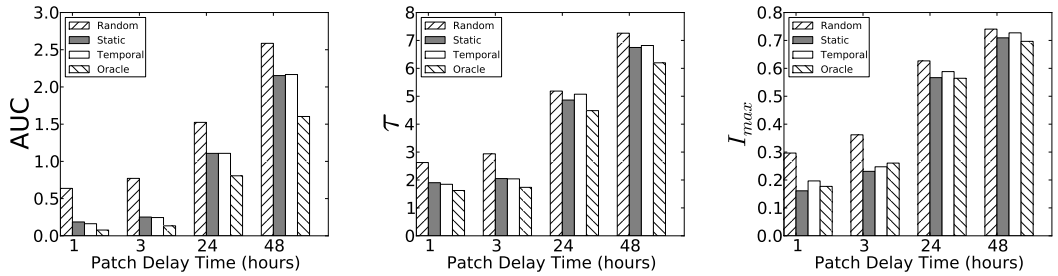


Figure 5.7: CAMBRIDGE: Best centrality-prediction binned by patch delay (upload interval 1 hour).

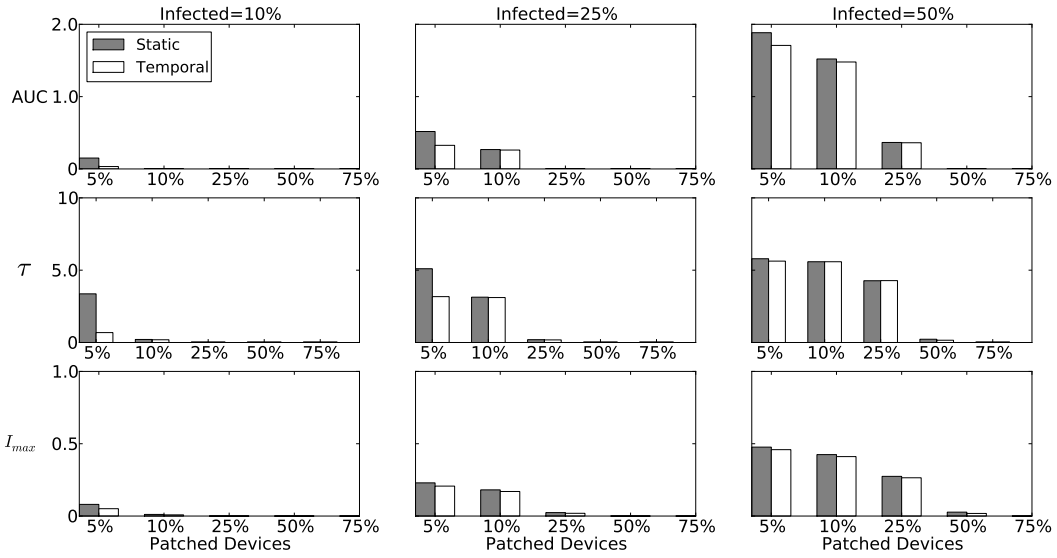


Figure 5.8: MIT: Effects of increasing number of patched devices against initial infected devices. Midday infection, 3 hours patch delay.

5.4 Related work

Predicting the topology of a network in the future has fuelled many studies into how complex network phenomena, such as small world or scale free topologies, form. Indeed early models in *generative graph* models, such as preferential attachment [BA99] attempt to capture the essence of how real networks are formed and potentially predict their structure in the future.

More direct studies of predicting link formation was formalised by Liben-Nowell

& Kleinberg [LK03], which required only topological information to predict future links based on measures of proximity. In the empirical dataset of co-authorship, the hypothesis evolved around the likelihood that your co-authors are highly likely to meet and write a research paper together. Although, centrality could potentially be estimated on top of predicted links, our study takes a more direct approach by predicting highly ranked centrality nodes.

Similar to our framework are studies on opportunistic networking in mobile devices such as BubbleRap [HCY11] and SimBet [DH09], as discussed in the previous chapter (Section 4.3). Both studies calculate and maintain centrality (degree and betweenness, respectively) to help decide the best next hop to pass a message nearer to the destination. Similar to our observations of ageing rules in centrality correlations, the effectiveness of an sliding window degree centrality were observed in the BubbleRap study although no empirical reason was given to why this may; in our study we have formalised this more directly through the ability to understand the centrality correlations over time using a correlogram.

More related to dynamic graphs, another set of studies, which bears similarities to our work, is that of frequent subgraph prediction in temporal networks [LB07]. This work utilises online machine learning algorithms to find and predict subgraphs (or links) based on subgraphs observed in the past. Our work also uses a notion of temporal graphs however focuses on temporal centrality prediction using time-series analysis which offers insights into how the predictive algorithm can be designed, rather than a black box approach given by machine learning techniques.

5.5 Conclusions

In this chapter, we have introduced a technique for analysing the correlations of centrality rankings which helps eliminate the requirement of knowledge of future contacts. To achieve this, we have assumed that the order of the top-k is not relevant; this is applicable to information spreading since starting information dissemination from all k nodes does not require any ordering. We have applied this prediction scheme to a case study where centrally managed message dissemination is required, namely short range mobile malware containment, which was introduced in the previous chapter.

There are many fruitful avenues for future investigation to understand the practicalities of such a scheme in other applications. Firstly, from the previous chapter on short range mobile malware containment, we found that opportunistically spreading a patch starting from highly ranked temporal closeness nodes was most effective, for this reason we have concentrated on predicting temporal closeness; clearly different centralities for other applications can be studied. Secondly, we have not set out to address the optimal k, but we conjecture this would be application specific. Thirdly, in addition to utilising correlograms, we could employ other time-series analysis techniques such as calculating auto-correlation coefficients. Fourthly, we wish to study a real deployment using a set of controlled devices examining, in particular, the relationship between opportunistic message spreading in conjunction with infrastructure based message delivery.Finally, the current study was limited to human contact networks though the techniques presented could be applied to many different types of empirical temporal networks.

6

Reachability in Temporal Graphs

Introduction

In the previous chapters, we have learnt about the importance of time order in information dissemination and how this can be used to enhance measures of centrality. We now turn our attention to the topic of *reachability* in graphs, which is fundamental to the study of real networks; the main issue is connectedness of the graph and whether the topology of the graph allows a source node to form a path to another node in the network.

Returning to the analogy described in Section 2.2.4.1, the United States, United Kingdom and Australia all have their own well-connected road networks however there is no method to drive from one country to another. Generalising this, many studies on real networks have found islands of connectivity (highly intra-connected) but which are separated from on another (not inter-connected). In particular, where mobility of nodes is present, this inherently introduces time-varying connectedness i.e. in satellite delay tolerant communications [BHT+03] and opportunistic commu-

nications in mobile phone pocket switched networks [HCY11, DH09]. We previously defined connectedness in static networks (Section 2.2.4), however, in this chapter we shall investigate how time adds additional constraints on the reachability between nodes in a temporal network, in particular, time order naturally introduces directionality and affects the connectedness of a real time-varying network.

Chapter Outline

We first define the notions of temporal connectedness and components in temporal graphs (Section 6.1). Next, we introduce an abstract graph model which captures, which we call the *affine graph*, the reciprocal reachability between nodes; this captures in a convenient static graph representation the connectedness of the real network taking into account the time order (Section 6.2). We then apply this to a real network where temporal contextual information (for example, exact dates with significant events) exists, namely the Reality Mining human contact network between mobile phones [EP06] (Section 6.3). Finally, we conclude in Section 6.5.

6.1 Temporally Connected Components

The problem of defining connectedness and components in temporal graphs looks more similar to the case of directed static graphs than to the case of undirected static graphs (Section 2.2.4.1). In fact, even if each graph G_m, $m = 1, \ldots, M$ in the sequence is undirected, the temporal ordering of the graphs naturally introduces a directionality.

In order to define node connectedness for a temporal graph, we first need to introduce a mathematical definition of *reachability* for an ordered couple of nodes i and j. We say that i can reach j, if i can send a message to j directly or through a time-ordered sequence of contacts.

In other words, we can use our definition of temporal paths (Definition 2, Section 3.2). A node i of a temporal graph $\mathcal{G}_{[t_1,t_M]}$ is *temporally connected* to a node j if there exists in $[t_1, t_M]$ a temporal path going from i to j. This relation is not symmetric: if node i is temporally connected to node j, in general node j can be either temporally connected or disconnected to i. In the graph $\mathcal{G}_{[t_1,t_4]}$ of Figure 6.1, node 5 is temporally

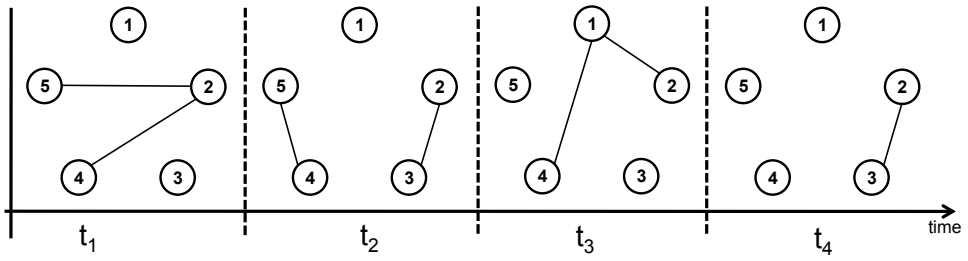


Figure 6.1: Temporal graph $\mathcal{G}$ consisting of a sequence of $M = 4$ graphs with $N = 5$ nodes.

connected to 1 but node 1 is not connected to node 5. For this reason, we introduce the definition of *strong connectedness*, which enforces symmetry:

Definition 3 (Strong connectedness) *Two nodes i and j of a temporal graph are* strongly connected *if i is temporally connected to j and also j is temporally connected to i.*

Strong connectedness is a reflexive and symmetric relation, so that if i is strongly connected to j, then j is strongly connected to i. However this definition of strong connectedness lacks transitivity, and therefore it is not an equivalence relation. In fact, if i and j are strongly connected and j and l are strongly connected, nothing can be said, in general, about the connectedness of i and l.

In the example shown in Figure 6.1, node 5 and 2 are strongly connected and also 2 and 1 are strongly connected, but nodes 5 and 1 are not strongly connected, since there exists no temporal path which connects node 1 to node 5.

It is also possible to introduce the concept of weak connectedness for a pair of nodes. Similarly to the case of static directed graphs, given a temporal graph $\mathcal{G}$, we construct the underlying undirected temporal graph $\mathcal{G}^u$, which is obtained from $\mathcal{G}$ by discarding the directionality of the links of all the graphs $\{G_m\}$, while retaining their time ordering.

Definition 4 (Weak connectedness) *Two nodes i and j of a temporal graph are* weakly connected *if i is temporally connected to j and also j is temporally connected to i in the underlying undirected temporal graph $\mathcal{G}^u$.*

Also weak connectedness is a reflexive and symmetric relation, but it is not transitive. This definition of weak connectedness is quite similar, but not identical, to that given for directed static graphs. In fact, two nodes in $\mathcal{G}$ can be weakly connected even if there is no temporal directed path which connects them, but the temporal ordering of links breaks the transitivity so that if i and j are weakly connected and j and l are weakly connected, then nothing can be said about the weak connectedness of i and l. All these subtleties are due to the fact that temporal graphs have a much richer structure compared to static graphs, so that the existence of a temporal path between two nodes crucially depends on the time ordering of links, and does not guarantee the existence of the backward path. Notice that the definitions of strong and weak connectedness given above for temporal graph are consistent with those given for static graphs, so that if two nodes are strongly (weakly) connected in a temporal graph, then they are also strongly (weakly) connected in the corresponding aggregate static graph. The vice-versa is trivially not true, so that two nodes which are strongly connected in the aggregate graph can be temporally disconnected in the temporal graph.

We are now ready to give the definitions of components associated to a node of a temporal graph $\mathcal{G}$:

1. The *temporal out-component of node* i, denoted as $\mathrm{OUT}_T(i)$, is the set of nodes which can be reached from i in the temporal graph $\mathcal{G}$.

2. The *temporal in-component of a node* i, denoted as $\mathrm{IN}_T(i)$, is the set of nodes from which i can be reached in the temporal graph $\mathcal{G}$.

3. The *temporal weakly connected component of a node* i, denoted as $\mathrm{WCC}_T(i)$, is the set of nodes which i can reach, and from which i can be reached, in the underlying undirected temporal graph $\mathcal{G}^u$.

4. The *temporal strongly connected component of a node* i, denoted as $\mathrm{SCC}_T(i)$, is the set of nodes from which node i can be reached, and which can be reached from i, in the temporal graph $\mathcal{G}$.

Differently from the case of directed static graphs, it is not possible to define the strongly (weakly) connected components of a temporal graph starting from the definition of connectedness for pairs of nodes. As we explained above, this is because

the relation of strongly (weakly) connectedness for couples of nodes is not an equivalence relation. For this reason, we give the following definition of strongly connected component of a temporal graph:

Definition 5 (Strongly connected component) *A set of nodes of a temporal graph $\mathcal{G}$ is a temporal strongly connected component of $\mathcal{G}$ if each node of the set is strongly connected to all the other nodes in the set.*

Similarly, a set of nodes is a *weakly connected component*, if each node in the set is weakly connected to all the other nodes in the set. The definitions of strongly and weakly connected components enforce transitivity, but the check of strong (weak) connectedness has to be directly performed for every couple of nodes. Suppose for instance that we want to verify if the five nodes in the graph $\mathcal{G}$ shown in Figure 6.1 form a strongly connected component. In the static aggregate graph this check has $O(K)$ computational complexity, where K is the total number of links in the graph. In fact, we have only to check that 2, 3, 4 and 5 are connected to 1, which can be done by a *depth first* visit of the graph started at node 1, since node connectedness is an equivalence relation for static graphs and a component of a node is also a component for the whole graph. On the contrary, for a temporal graph we should check the connectedness of all the possible couples of nodes, so that a procedure to verify that a set of N nodes form a strongly connected component has computational complexity $O(N^2)$ instead of $O(K)$, for every check. Moreover, while static directed graphs admit only one partition into strongly connected components, for a temporal graph there exists, in general, more than one possible partition, as we shall see in the next section.

6.2 The affine graph of a temporal graph

We show in the following that the problem of finding the strongly connected components of a temporal graph is equivalent to the well-known problem of finding the maximal-cliques of an opportunely constructed static graph [Kar72]. We call such static graph the *affine graph* corresponding to the temporal graph. It is defined as follows:

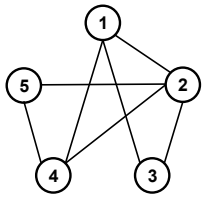


Figure 6.2: The affine graph $G_{\mathcal{G}}$ associated to the temporal graph $\mathcal{G}$ reported in Figure 6.1. The affine graph is static and undirected, and each of its maximal-cliques correspond to a strongly connected component of the original temporal graph $\mathcal{G}$

Definition 6 (Affine graph of $\mathcal{G}$) *Given a temporal graph $\mathcal{G} \equiv \mathcal{G}_{[t_1,t_M]}$, the associated affine graph $G_{\mathcal{G}}$ is an undirected static graph with the same nodes as $\mathcal{G}$, and such that two nodes i and j are linked in $G_{\mathcal{G}}$ if i and j are strongly connected in $\mathcal{G}$.*

In practice, the affine graph of a temporal graph can be obtained by computing the temporal shortest paths between any two pairs of nodes, and then adding a link between two nodes i and j of the affine graph only if the temporal distance from i to j and the temporal distance from j to i are both finite. Another method to construct the affine graph makes use of the out-components of all the nodes. We start by considering the out-component of the first node, let us say $i = 1$, and then we check, one by one, if for each node $j \in OUT_T(i), j > i$ then also $i \in OUT_T(j)$. If this is true, we put a link between i and j in the affine graph. We then repeat this procedure for the second node, $i = 2$, for the third node, $i = 3$ and so on. We obtain the affine graph by iterating over the out-components of all the nodes. In Figure 6.2 we report the affine graph corresponding to the time varying graph shown in Figure 6.1. In this graph, node 1 is directly connected to nodes $\{2, 3, 4\}$, since it is temporally strongly connected to them in the temporal graph. Similarly, node 2 is connected to nodes $\{1, 3, 4, 5\}$, node 3 is connected to $\{1, 2\}$, node 4 is connected to $\{1, 2, 5\}$ and node 5 is connected to $\{2, 4\}$. Hence, the affine graph $G_{\mathcal{G}}$ has only 7 of the 10 possible links, each link representing strong connectedness between two nodes.

We briefly report here some definitions about graph cliques. Given an undirected static graph, a *clique* is a complete subgraph, i.e. a subgraph in which all the nodes are directly linked to each other. A *maximal-clique* is a clique that is not

included in any larger clique, while a *maximum-clique* is a *maximal-clique* whose size is equal or larger than those of all the other cliques [Wes01]. By construction, a clique of the affine graph $G_{\mathcal{G}}$, contains nodes which are strongly connected to each other, so that the *maximal-cliques* of the affine graph, i.e. all the cliques which are not contained in any other clique, are temporal strongly connected components (SCC_T) of $\mathcal{G}$. Similarly, all the *maximum-cliques* of the affine graph $G_{\mathcal{G}}$, i.e. its largest maximal-cliques, are the largest temporal strongly connected components (LSCC_T) of $\mathcal{G}$. Therefore, the affine graph can be used to study the connectedness of a temporal graph, and the properties of the strongly connected components of a temporal graphs can be obtained from known results about maximal-cliques on static graphs. For instance, the problem of finding a partition of $\mathcal{G}$ that contains the minimum number of disjoint strongly connected components is equivalent to the well–known problem of finding a partition of the corresponding affine graph $G_{\mathcal{G}}$ in the smallest number of disjoint maximal-cliques [Kar72]. Unfortunately, this problem is known to be NP–complete, and in practice can be exactly solved only for small graphs. In the case of the affine graph in Figure6.2, it is possible to check by hand that there are only three possible partitions of $G_{\mathcal{G}}$ into maximal-cliques, namely:

1. $\{1,2,3\} \bigcup \{4,5\}$
2. $\{1,2,4\} \bigcup \{3\} \bigcup \{5\}$
3. $\{2,4,5\} \bigcup \{1,3\}$

Notice that the second partition contains two isolated nodes, which are indeed degenerated maximal-cliques. Therefore, the original temporal graph admits only two different partitions into a minimal number of non-degenerated strongly connected components, namely into two components containing at least two nodes each. One possible partition of our network $\mathcal{G}_{[t_1,t_4]}$ is made by the components $\{1,2,3\}$ and $\{4,5\}$, while the other partition consists of $\{2,4,5\}$ and $\{1,3\}$. If we discard the temporal ordering of links, we obtain different results. In fact, the aggregate static graph shown in Figure 6.1, has only one connected component, which includes all the five nodes.

Other interesting results stem from the mapping into affine graphs and from the following well-known results for cliques in graphs.

1. Checking if a graph contains a clique of a given size k has polynomial computational complexity, and precisely $O(N^k k^2)$ [Dow95].

2. The *clique decision problem*, i.e., the problem of testing whether a graph contains a clique larger than a given size $\overline{k}$, is NP–complete [Kar72]. Therefore, any algorithm which verifies if a temporal graph has a strongly connected component whose size is larger than a fixed value $\overline{k}$, has exponential computational complexity.

3. Listing all the maximal-cliques of a graph has exponential computational complexity, namely $O(3^{N/3})$ on a graph with N nodes [MM65, BK73]. Consequently, finding all strongly connected components of a temporal graph with N nodes, requires an amount of time which exponentially grows with N.

4. The problem of finding a maximum-clique for an undirected graph is known to be hard–to–approximate [FGL$^+$91, AS98, ALM$^+$98], and an algorithm that finds maximum-cliques requires exponential time. The best algorithm works in $O(\sim 1.2^N)$ for a graph with N nodes [TT77, Rob86].

5. The problem of determining if a graph can be partitioned into K different cliques is NP–complete, and consequently the problem of finding the minimum number of cliques that cover a graph, known as the *minimum clique cover*, is NP–complete [Kar72]. This means that there exists no efficient algorithm to find a partition of a temporal graph made by a set of disjoint strongly connected components. Moreover, there is in general more than one partition of a graph into maximal-cliques, so that a temporal graph cannot be uniquely partitioned into a set of disjoint strongly connected components.

The existence of a relation between the strongly connected components of a temporal graph and the maximal-cliques of its affine graph implies that it is practically unfeasible to find all the strongly connected components of large temporal graphs. The problem can be exactly solved only for relatively small networks, for which it is computationally feasible to enumerate all the maximal-cliques of the corresponding affine graphs. Even if, in many practical cases, it is possible to find only the maximal-cliques up to a certain size $\overline{k}$, we can still obtain some information about the maximum value of $\overline{k}$ to be checked. First, in order to have a clique of size $\overline{k}$ the

graph should have at least $\overline{k}$ nodes having at least $\overline{k}$ links. Moreover, each clique of order $\overline{k} > 3$ has exactly $\binom{\overline{k}}{3}$ sub–cliques of order 3, so that in order for a subgraph to be a clique of order $\overline{k}$, the graph should have at least $\binom{\overline{k}}{3}$ triangles. This means that there is a relation between the number of triangles of the affine graph and the size of its maximum-cliques. In particular, the number of existing triangles in the affine graph fixes an upper bound for the size of the largest admissible maximal-cliques of the graph.

6.3 Application to a Real Network

As a practical example, in this section we extract and analyse node and graph components of real temporal social networks. We return to the Reality Mining dataset as discussed in Sections 1.1.1.3 and 3.4.2. We report results of component analysis performed on *a)* graphs corresponding to the first half and to the second half of a week, *b)* graphs corresponding to different days of a week and *c)* graphs corresponding to different weeks. In particular, we will focus our attention on the Fall term (namely from start of September to mid of December), which corresponds to weeks from 10 to 19 in the dataset. We chose this dataset for two very simple reasons. First, due to the relatively small number of nodes, it is possible to extract all the maximal-cliques of the corresponding affine graphs by using a limited amount of computational resources. Secondly, this dataset represents a real human interaction network and, as we shall see in the following, the approximation made representing it as a static graph, i.e. considering all the links as concurrent in time, is a very poor and unrealistic representation of the system.

In Figure 6.3 we consider week 11. For each node, we report the size of temporal in-component (panel a) and temporal out-component (panel b) during the beginning of the week (WB), namely from Monday 12:00am to Thursday 12:00pm (red circles), and during the end of the week (WE), namely from Thursday 12:00pm to Sunday 11:59pm (blue squares). During WB almost all nodes have temporal in-components and out-components of similar sizes. In fact, the majority of nodes have in-component of size 72 and out-component of size 74. Conversely, during WE, we observe a wide distribution of the sizes of temporal in- and out-components. In particular (panel a) we notice a group of nodes having an in-component of size

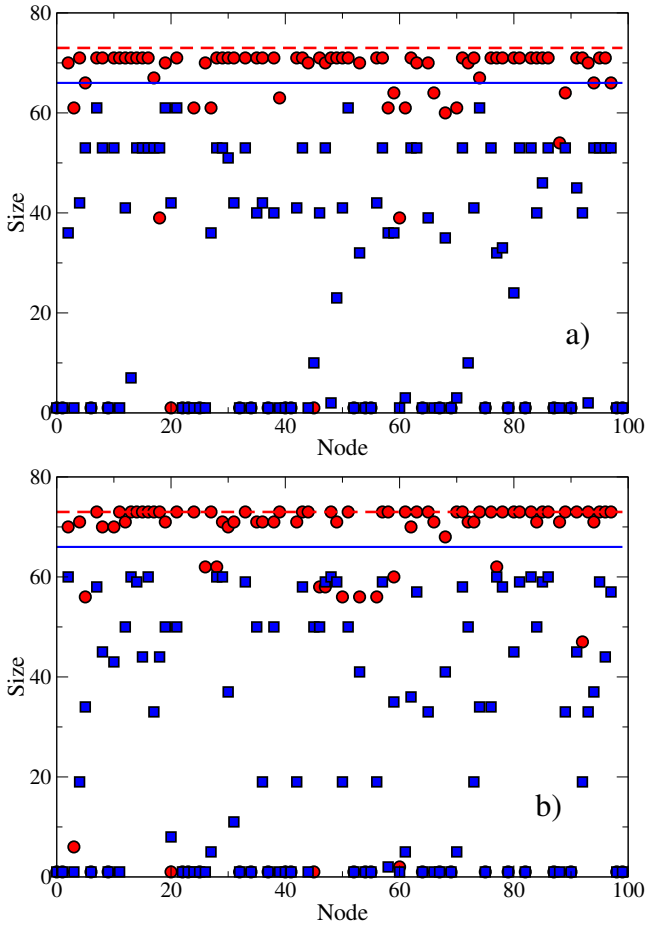


Figure 6.3: Size of the temporal in-component (a) and out-component (b) for each of the $N = 100$ individuals during week 11 of the Reality Mining dataset. Red circles and blue squares correspond, respectively, to the beginning of the week (WB) and to end of the week (WE). For comparison, the size of the largest connected component of the corresponding aggregate static graph is reported as dashed red line (WB) and solid blue line (WE), respectively.

53, another group whose in-component contains around 40 nodes, and other nodes with in-component of size smaller than 30. Similarly (panel b), there is a group of nodes whose out-component contains around 60 nodes, a second group of nodes with out-component sizes between 40 and 50, and many other nodes having out-component with less than 40 nodes. The observed small variability in the size of node components during WB, is due to the fact that students and faculty mem-

bers have more opportunities to meet and interact at lectures during WB. Even if not all students attend the same classes, and not all professors teach to all the students, there is a high probability that two individuals are connected by longer temporal paths. Conversely, during WE, the students usually meet other students in small groups, and they usually do not meet professors and lecturers, except for the classes held on Thursday afternoon and on Friday. As a result, the size of the in- and out-components during WE exhibits large differences from node to node. Such fluctuations are lost in a static graph description, which aggregates all the links independently of their time ordering. In fact, the two static aggregate graphs corresponding respectively to WB and WE, have only one giant connected component, which contains the majority of the nodes, while the remaining nodes are isolated. As comparison, the size of the giant component of the aggregate static graphs for WB and WE are also reported in Figure 6.3, respectively as dashed red line and solid blue line. Notice that the static aggregate graph corresponding to a co–location temporal graph is intrinsically undirected. Therefore, the in- and out-components of a node in this graph coincide and correspond to the component to which the node belongs. Moreover, in a static aggregate graph all the links (and consequently also all the paths) are always available, so that all the nodes in the same connected component have the same component size. As a result, the variability in the node connectedness of the temporal network, which is evident from the distribution of circles and squares in Figure 6.3, is flattened down in the aggregate static graph. In the latter case, all information about network connectedness is represented by a single value, namely the size of the largest connected component, which indeed says nothing about the mutual reachability of two generic nodes of such a component. In particular, the size of the giant connected component of the static aggregate graph is equal to 74 during WB and to 66 during WE, despite the fact that in the same intervals the majority of nodes have much smaller temporal in- and out-components.

In Table 6.1 we report some relevant structural properties of the affine graphs. We consider and compare the temporal graphs constructed in the first 24 hours (Monday) of ten consecutive weeks (from week 10 to week 19). We observe large fluctuations in the measured values. The number of links K ranges from 105 in week 12 to 1485 in week 15, while the number of triangles T is in the range $[307, 22096]$, with a mean value around 10000 and a standard deviation equal to 6932. This variance is due to the fact that, even if the daily activity of each individual is, on

Week #	K	T	N_s	$\langle s \rangle$	S	N_S	N_U	N_I	C
10	646	4341	22	10.3	27	1	27	27	62
11	554	4414	15	9.1	29	1	29	29	54
12	105	307	11	4.1	13	1	13	13	22
13	772	8322	16	10.6	36	1	36	36	59
14	815	6481	20	12.7	27	1	27	27	62
15	1485	22096	23	23.7	44	1	44	44	67
16	1022	9033	22	16.5	29	1	29	29	70
17	1284	15572	19	22.3	38	1	38	38	67
18	1417	18430	16	20.7	44	1	44	44	67
19	1106	13531	13	20.9	38	2	42	34	60

Table 6.1: Structural properties of the affine graph corresponding to the temporal graph of the first 24 hours of the week (Monday), for each week of the Fall term: number of links (K), number of triangles (T), number of maximal cliques (N_s), average size of maximal cliques ($\langle s \rangle$), size of the largest maximal clique (S), number of largest maximal cliques (N_S), number of nodes in the union (N_U) and in the intersection (N_I) of all largest maximal cliques. The size of the giant component of the corresponding static aggregate graph (C) is reported in the rightmost column.

average, almost periodic, in a particular day we can observe a peculiar temporal pattern of connections, for instance because some students decide to skip a class or because the lessons are suspended for public holidays. In particular, this is exactly what happens on week 12. Monday of week 12 is September 11^{th} 2004, and corresponds to the *Patriot Day*, a national holiday introduced in the US in October 2001, designated in memory of the 2977 killed in the September 11^{th}, 2001 attacks. Therefore, we observe the minimum connectivity and the minimum number of triangles on week 12, because all teaching activities were suspended, and students did not participate to lessons as usual. In addition, the number N_s and the average size $\langle s \rangle$ of maximal cliques of the affine graphs change from one week to another. In particular, during weeks 10 to 14 we observe relative smaller values of N_s and $\langle s \rangle$ than in weeks 15 to 19, which is probably due to the relatively lower number of links and triangles. Conversely, if we consider the size S of the largest strongly connected component (i.e. the largest maximal-clique of the affine graph), we notice that it is not strongly correlated with K and T. For instance, the size of the largest

strongly connected component found at week 11 ($S = 29$) is equal to that observed at week 16. However, at week 11 the affine graph has a much smaller number of links and triangles than at week 16. Moreover, on Monday of week 14 we have a maximal-clique of size 27, even if the number of links and triangles is higher than on Monday of week 11. These results confirm that the size of the largest strongly connected component of a temporal graph is mainly due to the actual configuration of links and triangles of the corresponding affine graph, and not only to their relative number. We notice also that every affine graph reported in Table 6.1 admits a single LSCC_T, except at week 19 where two LSCC_Ts of size $S = 38$ emerge. For this reason we also looked at the number of nodes N_U which participate to *at least one* LSCC_T, and at the number N_I of nodes which participate to *all* LSCC_Ts. These numbers correspond, respectively, to the number of nodes found in the union and in the intersection of all LSCC_Ts. An interesting result is that $N_I = 34$ on week 19, so that 34 nodes participate to both maximal 42-node cliques. These 34 nodes play a very important role in the structure of the network. If we remove just one of them, then the resulting affine graph does not have a clique of size 42 any more, and consequently the size of the LSCC_T of the remaining temporal graph is smaller than 42. At the same time, removing all these N_I nodes will cause a significant reduction in the size of LSCC_Ts, in the number of triangles of the affine graph and, consequently, in the number of SCC_Ts. The nodes that participate in at least one LSCC_T are important in the diffusion of information throughout temporal graphs. In fact, it is sufficient to pass a message to one of the nodes in a LSCC_T early in the morning, to assure that at least N_U nodes will receive the message before the end of the day.

Finally, in the rightmost column of Table 6.1 we report the size C of the giant component of the corresponding static aggregate graph. Notice that for any of the ten weeks under consideration, the value of C is much larger than S, as a consequence of the fact that the static representation of the temporal graph systematically overestimates node connectedness and paths availability. In panel (a) of Figure 6.4 we plot the value of S and C for each Monday of the Fall term. We notice that both C and S are able to capture the anomalous behaviour at Monday of week 12 (*Patriot Day*). If we focus our attention on the period from week 13 to week 19, the size of the giant connected component of the aggregate static graph is in the range $[59, 70]$, while the size of the LSCC_T of the temporal graphs in the same interval exhibits

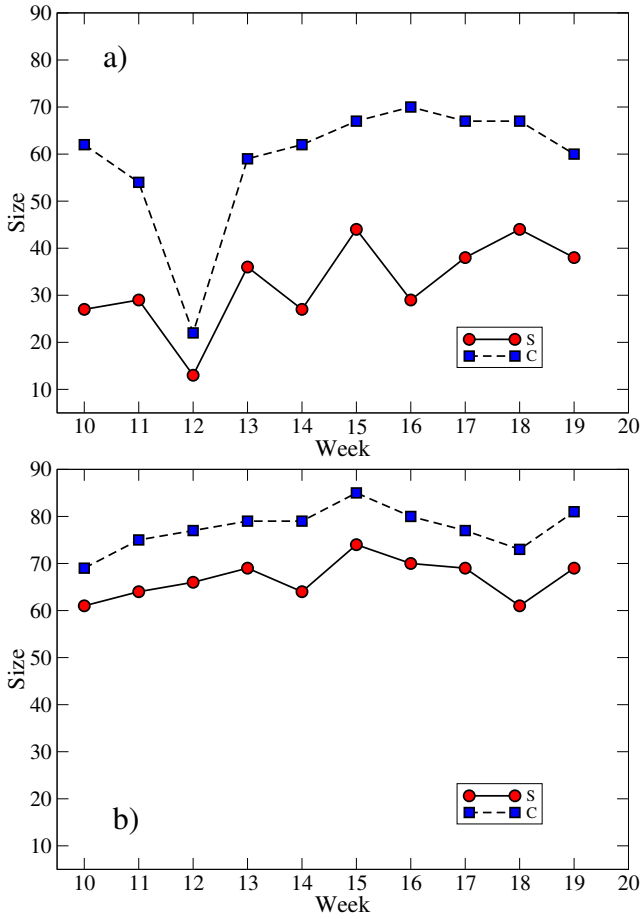


Figure 6.4: Panel a): size of the LSCC_T of the temporal graph on Monday (red circles) and of the giant component of the corresponding static aggregate graph (blue squares). Panel b): the same as panel a) but for the temporal graph corresponding to the whole week.

wider fluctuations between $S = 27$ (week 14) and $S = 44$ (week 15 and week 18). This variability is due to the intrinsic fluctuations observed in human contact networks. For instance, some of the students that attended a given class on Monday of week 13, might have decided to remain at home on week 14, and this eventually had an impact on the availability of links and paths, producing smaller strongly connected components. This intrinsic variability is somehow flattened down if we use the standard static component analysis and compute the largest connected component of a static graph that aggregates all the links of one day. Furthermore, we notice the lack of correlation between C and S. (the linear correlation coefficient

Week #	K	T	N_s	$\langle s \rangle$	S	N_S	N_U	N_I	C
10	2200	45428	10	44.0	61	1	61	61	69
11	2506	54500	12	46.8	64	1	64	64	75
12	2598	57913	12	43.5	66	1	66	66	77
13	2965	71561	9	62.5	69	1	69	69	79
14	2590	56826	15	39.3	64	1	64	64	79
15	3321	85348	9	54.7	74	1	74	74	85
16	2927	69452	9	53.2	70	1	70	70	80
17	2802	66247	10	57.9	69	1	69	69	77
18	2298	47429	12	40.0	61	2	62	60	73
19	2966	70963	13	53.8	69	3	72	68	81

Table 6.2: Structural properties of the affine graph corresponding to the temporal graph of the whole week, for each week of the Fall term. Legend as in Table 6.1.

between C and S from week 13 to week 19 is equal to $r = 0.12$). For instance, at Monday of week 16 we observe the maximum value of C, namely $C = 70$, while the temporal graph has a largest strongly connected component of size $S = 29$, which is relatively small compared to the other weeks. Conversely, at Monday of week 13 we observe a relatively small giant component, with $C = 59$ nodes, while the size of the largest strongly connected component is $S = 36$.

In order to show the results of our analysis when applied at a larger temporal scale (weeks instead of days), we have reported in Table 6.2 the structural properties of the affine graphs constructed from the contacts observed during a whole week. As in Table 6.1, we compare the 10 weeks in the Fall term. We observe a variance in the number of links and triangles: K is in the range $[2200, 3321]$ and T is in the range $[45428, 85348]$, and still there is no appreciable correlation between the average size $\langle s \rangle$ of SCC_Ts and K or T. If we look at panel (b) of Figure 6.4, where we report S and C for the temporal graph corresponding to the whole week, we notice that the size of the LSCC_T at each week is still lower than the size of the giant component of the corresponding aggregate graph. Differently from the case of single days, at a scale of the entire week we observe a clear correlation between S and C. The linear correlation coefficient between C and S, from week 10 to week 19, is now equal to $r = 0.89$. These results confirm that the number and size of strongly connected components in temporal graphs depend on the length of the period during which we

observe the system. In our system at the scale of a week, almost all the nodes are in the largest strongly connected component because longer temporal paths appear, so that the affine graphs at different weeks are more similar to each other and the information extracted from a temporal analysis is similar to that obtained by plotting static measures on aggregated graphs as function of time. On the contrary, at the scale of a day, our system has affine graphs that are disconnected or similar to trees, with very few triangles and relatively small cliques. In this case, as shown in Figure 6.4, a temporal component analysis of temporal graphs reveals interesting details about the dynamics of contacts, which cannot be detected by a static graph analysis.

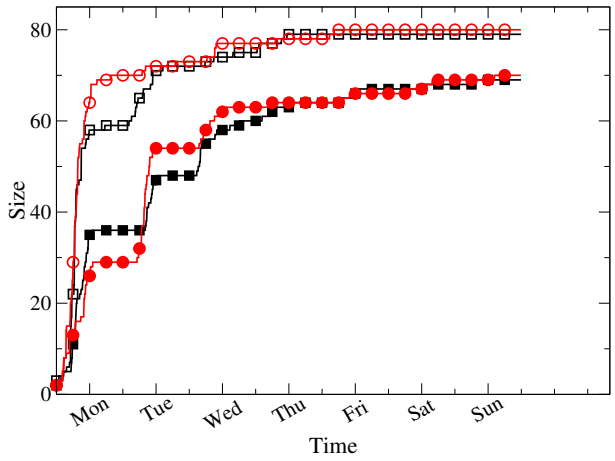


Figure 6.5: Size of the largest strongly connected component of the temporal graph (lines with filled symbols) and size of the giant component of the corresponding static aggregate graph (lines with empty symbols). The black lines with squares correspond to week 13, while red lines with circles correspond to week 16. Large ticks on the x-axis indicate 12:00pm of each day.

Finally, in Figure 6.5 we show the temporal evolution of S and C during the week. In particular, we compare week 13 and week 16. A point of the plot at time t is obtained by considering the temporal graph constructed from the events occurred in the interval $[0, t]$, where $t = 0$ corresponds to Monday at 00:00. For each of these temporal graphs we construct the corresponding affine graph to compute $S(t)$, and then we consider the static aggregate graph to obtain $C(t)$. We observe that $S(t)$ is always smaller than $C(t)$, $\forall t$. In particular, until Tuesday at midnight the size of the largest strongly connected component in week 16 is around $S = 30$, which

is less than 50% of the size reached on Sunday. Moreover, at Wednesday midnight the maximal-clique contains $S = 54$ nodes, and the size continues to grow until the end of the week. Conversely, the size of the giant component of the corresponding aggregate graph on Tuesday at midnight is $C = 73$, which is more than twice larger than the largest strongly connected component at the same time and corresponds to 90% of the size of the giant component at the end of the week. On Friday at midnight, the size of the giant component has already reached its maximal value, and does not change any more until the end of Sunday. Notice that the temporal evolution of the size of the giant component over the week looks similar in the two cases, while we observe interesting differences in the temporal evolution of the size of largest strongly connected component. In fact, the size of the LSCC_T at the end of Monday of week 13 is $S = 36$, while at the same time the size of LSCC_T for week 16 is $S = 29$. This indicates that during Monday of week 13 there was a higher number of contacts than during Monday of week 16. On the contrary, at the end of Tuesday the size of LSCC_T of week 13 is $S = 48$, which is smaller than the value observed at the same time in week 16, i.e. $S = 54$. All these variations, which are due to the temporal correlation and fluctuations in the individuals' connection patterns, disappear in an aggregate static representation.

6.4 Related Work

The notion of *reachability* in a graph taking into account time has been studied in the past. Holme [Hol05] studied the reachability between pairs of nodes through time-respecting paths in time-stamped email and a online dating message exchanges and found that these graphs were highly disconnected; however, this study did not focus on connected components and hence does not capture the reachability between a set of nodes. Similar to our affine graph construction, Moody [Moo02] constructed a static *reachability graph* from a time-stamped graph to study possible communication channels between nodes; however, this was only for one-way, unreciprocated temporal paths. This loosely corresponding to our notion of a temporally weak connected component; however, as we have discussed, due to temporal directionality, a temporal path from A to B does not imply that a temporal path from B to A is possible. Furthermore, by ignoring the temporal directionality of paths in a

weakly connected component, the reachability between a pair of nodes is overestimated. Instead, our study quantifies the "islands" of communications between a *set* of nodes in reciprocated, two-way time-respecting paths and, consequently, uncovers the computational complexities associated with calculating strongly connected components in temporal graphs.

Within computer science, the notion of connectedness is inherent in opportunistic and delay-tolerant networking, where connectivity between mobile devices is highly intermittent [JFP04]. This makes communication between devices more challenging since such time-varying connectivity requires more sophisticated message forwarding protocols. Previously (Section 4.3), we have discussed two such protocols based on social network analysis, namely BubbleRap and SimBet, however, such studies have not quantified the connectedness of nodes in a time-varying network due to the lack of a formal model for time-varying networks. Using temporal graphs, we have formalised the reachability between nodes in a temporal graph in terms of the well studied concept of connected components. Through this, future work could apply this formalisation to understand the performance upper limits and quality-of-service guarantees which can be made in such message delivery protocols taking into account the connectedness of the network over time.

6.5 Conclusions

Conventional definitions of connectedness and components proposed so far have only considered aggregated, static topologies, neglecting important temporal information such as time order, duration and frequency of links. In this chapter, we have extended the concepts of connectedness to the case of temporal graph, and we have introduced definitions of node and graph components which take into account times of appearance and temporal correlations of links. The proposed temporal measures are able to capture variations and fluctuations in the linking patterns, typical of many real social and biological systems; this was not captured by static component analysis. As a first application we have studied a dataset of human contacts, showing that variations in the pattern of connections among nodes produce relevant differences in the size and number of temporal strongly connected components. We pointed out the important role played by nodes that belong to many strongly con-

nected components at the same time, and we have analysed how temporal strongly connected components evolve over time. We hope that our formalism could find useful to study other empirically collected temporal networks and to better characterise dynamical processes, which take place on these networks, such as diffusion of information and spreading of diseases. In addition, the robustness of real networks to attack could be better characterised through the study of temporally connected components and would be an interesting direction for future work.

7

Summary and Outlook

Real networks inherently exhibit rich temporal information and only recently has the technology to collect temporal data and computational power to process such data been available to researchers.

Returning to our original thesis, we have demonstrated that this extra temporal information is important for the analysis of information dissemination in real networks. We first identified four important pieces of time information, namely timestamps, duration, frequency and time order of links, however we have found that the most important detail is that of time order. Next, we studied the fundamental measure of shortest paths in networks and found that, since static aggregated graphs ignore time order, the available links are over estimated and the true shortest path length is underestimated. This led us to find important consequences in the accurate identification of important nodes in a network, which play a role in information spreading and information mediation. Also, since we have considered sliding time points in these real networks we have noticed patterns in correlations over time which was instrumental in conceiving a technique for predicting key information spreaders to

eliminate the need for knowledge of future contacts. Finally, the study of time order in real networks enlightened our study of temporal directionality in the study of reachability and connectedness in real networks.

We also remarked on the generality of the techniques proposed in this dissertation. Although we have been limited to the availability and suitability of empirical networks datasets to certain studies, the techniques studies in this thesis have been purposefully selected for their applicability to a range of different disciplines and networks; this is informed by their range of uses in current static network analysis. However, due to the wealth of tools and techniques available to researchers who wish to uncover important properties of real networks, we have focussed on metrics related to information dissemination.

My experience during this thesis has successfully followed the following recipe, select a static graph metric, redefine using temporal information and evaluate on a real network. Indeed this recipe also informs the wide possibilities for future work.

Firstly, in this on going project, we wish to widen our study to other topics in static network analysis, for example in examining whether power law degree distributions still hold over time; if temporal motifs can aid in predicting future links; and reformalising the notion of node similarity taking into account time.

Secondly, the study of complex interactions between nodes over time requires manual analysis and this is aided by *visualisations*. We have introduced some novel visualisations that help capture certain aspects of node contacts, for example, Figure 4.8 helped us understand the robustness of real temporal networks due to many alternative paths that a (malicious) message can propagate. We believe that a substantial contribution can be made through the design of static (2- and 3- dimensional) and interactive visualisation tools.

Thirdly, our applications have been limited to the currently available empirical network datasets. With our understanding of the importance of time-order, this would focus future efforts in collecting network data. This also leads us in to the range of applications which could be studied using these techniques. For example, in the study of DTN and opportunistic networks, we can formally measure the information dissemination properties of a time varying mobility model or mobility trace; rather than proposing new routing algorithms, perhaps there are more fundamental properties of the underlying link sequence and network topology which are important for

information dissemination. For instance, our preliminary studies into the relationship between random and periodic mobility suggest that mixing mobility models is detrimental for information dissemination [TZLM10]. We also envisage applications in targeted marketing in evolving online social networks, identifying suspicious activity over time and the effects of a spreading process on the underlying network topology.

All in all, this thesis has made a substantial step in addressing the natural initial inquisition of any researchers into the advantages of extra temporal information in the study of real networks and, in doing so, opened the door to a vast range of future possibilities in the study of real time-varying networks.

Bibliography

[AB02] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, June 2002.

[AB09] N. J. Allen and B. A. Barres. Neuroscience: Glia - more than just brain glue. *Nature*, 457(7230):675–677, February 2009.

[ACG+09] F. A. C. Azevedo, L. R. B. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. L. Ferretti, R. E. P. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513(5):532–541, April 2009.

[AG51] S. E. Asch and H. Guetzkow. Effects of group pressure upon the modification and distortion of judgments. In *Groups, leadership and men; research in human relations.*, pages 177–190. Carnegie Press, 1951.

[AJB00] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45:501–555, May 1998.

[AS98] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45:70–122, January 1998.

[BA99] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[BAAS09] J. Bonneau, J. Anderson, R. Anderson, and F. Stajano. Eight friends are enough: Social graph approximation via public listings. In *Proc. of the 2nd ACM Workshop on Social Network Systems, SNS '09*, pages 13–18. ACM, March 2009.

[Bar04] M. Barthélemy. Betweenness centrality in large complex networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):163–168, March 2004.

[BBPV04] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *Proceedings of the Nationl Academy of Science, PNAS*, 101(11):3747–3752, March 2004.

[BBV08] A. Barrat, M. Barthélemy, and A. Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, November 2008.

[BC03] M. Balazinska and P. Castro. CRAWDAD data set ibm/watson (v. 2003-02-19). Downloaded from `http://crawdad.cs.dartmouth.edu/ibm/watson`, February 2003.

[BFFL08] A. Buscarino, L. Fortuna, M. Frasca, and V. Latora. Disease spreading in populations of moving agents. *Europhys. Lett.*, 82(38002), 2008.

[BHT+03] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, June 2003.

[BK73] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16:575–577, September 1973.

[BLM+06] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, February 2006.

[Bol01] B. Bollobás. *Random graphs*. Cambridge University Press, 2nd ed. edition, 2001.

[Bra01] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[BS09] E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, March 2009.

[BU89] D. Baird and R. E. Ulanowicz. The seasonal dynamics of the chesapeake bay ecosystem. *Ecological Monographs*, 59(4):329–364, December 1989.

[cab04] Virus description: Bluetooth-worm:symos/cabir. `http://www.f-secure.com/v-descs/cabir.shtml`, 2004.

[Cal04] L. B. Calkins. Enron fraud trial ends in 5 convictions. `http://www.washingtonpost.com/wp-dyn/articles/A23034-2004Nov3.html`, November 2004. Washington Post.

[CE07] A. Clauset and N. Eagle. Persistence and periodicity in a dynamic proximity network. *Proc. of DIMACS Workshop on Computational Methods for Dynamic Interaction Network*, 2007.

[CFQS10] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-Varying graphs and dynamic networks. *1012.0009*, November 2010.

[CH66] K. L. Cooke and E. Halsey. The shortest route through a network with time-dependent internodal transit times. *Journal of Mathematical Analysis and Applications*, 14(3):493–498, June 1966.

[Cha03] C. Chatfield. *The Analysis of Time Series: An Introduction, Sixth Edition*. Chapman and Hall/CRC, 6th edition, 2003.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, 2nd edition, September 2001.

[CNN02] CNN. Enron paid hefty bonuses before bankruptcy. `http://archives.cnn.com/2002/LAW/02/09/enron.bonuses/index.html`, February 2002.

[DA05] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72(2):027104, 2005.

[DGB+04] B. Draganski, C. Gaser, V. Busch, G. Schuierer, U. Bogdahn, and A. May. Neuroplasticity: Changes in grey matter induced by training. *Nature*, 427(6972):311–312, January 2004.

[DH09] E. M. Daly and M. Haahr. Social network analysis for information flow in disconnected Delay-Tolerant MANETs. *IEEE Transactions on Mobile Computing*, 8(5):606–621, 2009.

[DMS01] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Giant strongly connected component of directed networks. *Physical Review E*, 64(2):025101, July 2001.

[Dow95] R. Downey. Fixed-parameter tractability and completeness II: on completeness for w[1]. *Theoretical Computer Science*, 141:109–131, April 1995.

[DW10] P. De Wilde. Modelling interacting networks in the brain. In *Modelling and Analysis of Networked and Distributed Systems, A SICSA Workshop*, University of Stirling, June 2010.

[EI01] C. H. Evans, Jr. and S. T. Ildstad. *Small Clinical Trials: Issues and Challenges*. National Academy of Sciences Press, January 2001.

[EK10] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, July 2010.

[EM04] P. Elkind and B. McLean. *The Smartest Guys in the Room: The Amazing Rise and Scandalous Fall of Enron*. Penguin, new ed edition, September 2004.

[EMB02] H. Ebel, L.-I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *Physical Review E*, 66(3):035103, 2002.

[EP06] N. Eagle and A. Pentland. Reality Mining: Sensing Complex Social Systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[EPL09] N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009.

[fac] The facebook project. `http://www.thefacebookproject.com`.

[Fed08] Federal Energy Regulatory Commission. Addressing the 2000-2001 Western Energy Crisis. `http://www.ferc.gov/industries/electric/indus-act/wec.asp`, December 2008.

[Fer04] A. Ferreira. Building a reference combinatorial model for MANETs. *IEEE Network*, 18(5):24–29, 2004.

[FF58] L. R. Ford and D. R. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6(3):419–433, May 1958.

[FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *Proc. of the conference on Applications, technologies, architectures, and protocols for computer communication, SIGCOMM '99*, pages 251–262. ACM, 1999.

[FGL+91] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proc. of 32nd Annual Symposium on Foundations of Computer Science*, pages 2–12. IEEE, October 1991.

[FGM07] A. Ferreira, A. Goldman, and J. Monteiro. On the evaluation of shortest journeys in dynamic networks. In *Proc. of the 6th IEEE International Symposium on Network Computing and Applications, NCA '07*, pages 3–10, 2007.

[FLA+08] F De Vico Fallani, V. Latora, L. Astolfi, F. Cincotti, D. Mattia, M G Marciani, S. Salinari, A. Colosimo, and F. Babiloni. Persistent patterns of interconnection in time-varying cortical networks estimated from high-resolution EEG recordings in humans during a simple motor act. *Journal of Physics A: Mathematical and Theoretical*, 41(22):224014, 2008.

[GH09] P. Grindrod and D. J. Higham. Evolving graphs: dynamical models, inverse problems and propagation. *Proc. of the Royal Society A*, 466(2115), 2009.

[GH11] P. Grindrod and D. J. Higham. Models for evolving networks: with applications in telecommunication and online activities. *IMA Journal of Management Mathematics*, February 2011.

[GLMP08] J. Gómez-Gardeñes, V. Latora, Y. Moreno, and E. Profumo. Spreading of sexually transmitted diseases in heterosexual populations. *Proceedings of the National Academy of Sciences*, 105(5):1399 –1404, February 2008.

[Gol99] A.R. Golding. Automobile navigation system with dynamic traffic data. *US PATENT 5,933,100*, August 1999.

[GPHE11] P. Grindrod, M. C. Parsons, D. J. Higham, and E. Estrada. Communicability across evolving networks. *Physical Review E*, 83(4):046120, April 2011.

[Gua90] J. Guare. *Six degrees of separation : a play*. Random House, New York, 1st ed. edition, 1990.

[Hal77] J. Halpern. Shortest route with time dependent length of edges and limited delay possibilities in nodes. *Mathematical Methods of Operations Research*, 21(3):117–124, June 1977.

[HBZ73] C. Haney, C. Banks, and P. Zimbardo. Interpersonal dynamics in a simulated prison. *International Journal of Criminology & Penology*, 1(1):69–97, 1973.

[HCS+05] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 244–251. ACM, 2005.

[HCY08] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proc. of the 9th ACM international*

symposium on Mobile ad hoc networking and computing, MOBIHOC '08, MobiHoc '08, pages 241–250. ACM, 2008.

[HCY11] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE rap: Social-based forwarding in delay tolerant networks. *IEEE Transaction on Mobile Computing*, 10:1576–1589, November 2011.

[Hil96] B. Hillier. *Space is the machine: A configurational theory of architecture*. Cambridge University Press, 1996.

[Hol03] P. Holme. Congestion and centrality in traffic flow on complex networks. *Advances in Complex Systems*, 6(2):163–176, 2003.

[Hol05] P. Holme. Network reachability of real-world contact sequences. *Physical Review E*, 71(4):046119–8, April 2005.

[HP74] J. Halpern and I. Priess. Shortest path with time constraints on movement and parking. *Networks*, 4(3):241–253, 1974.

[HS11] P. Holme and J. Saramäki. Temporal networks. *arxiv:1108.1780*, August 2011.

[Hyp05] M. Hypponen. F-secure weblog: The grand opening!, May 2005.

[Hyp06] M. Hypponen. Malware goes mobile. *Scientific American*, pages 70–77, November 2006.

[ISB+11] L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1):166–180, February 2011.

[JFP04] S. Jain, K. Fall, and R. Patra. Routing in a Delay Tolerant Network. In *Proc. of the 2004 ACM conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '04*, pages 145–158, 2004.

[JOBL08] F. Jordán, T. A. Okey, B. Bauer, and S. Libralato. Identifying important species: Linking structure and function in ecological networks. *Ecological Modelling*, 216(1):75–80, August 2008.

[Joh04] L. Johnston. Former Enron Trader Pleads Guilty. `http://www.cbsnews.com/stories/2004/06/16/national/main623569.shtml`, August 2004. CBS News.

[Kar72] R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, 40(4):85–103, 1972.

[Ken38] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[KHAY09] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo. CRAWDAD data set dartmouth/campus (v. 2009-09-09). Downloaded from `http://crawdad.cs.dartmouth.edu/dartmouth/campus`, September 2009.

[KHS+11] M. G. Kitzbichler, R. N. A. Henson, M. L. Smith, P. J. Nathan, and E. T. Bullmore. Cognitive effort drives workspace configuration of human brain functional networks. *The Journal of Neuroscience*, 31(22):8259 –8270, June 2011.

[KKK02] D. Kempe, J. Kleinberg, and A. Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64:820–842, June 2002.

[KKT03] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03*, pages 137–146, 2003.

[KKW08] G. Kossinets, J. Kleinberg, and D. Watts. The structure of information pathways in a social communication network. In *Proc. of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08*, pages 435–443, 2008.

[KL10] S. Kitchovitch and P. Liò. Risk perception and disease spread on social networks. *Procedia Computer Science*, 1(1):2345–2354, May 2010.

[KL11] S. Kitchovitch and P. Liò. Community structure in social networks: Applications for epidemiological modelling. *PLoS ONE*, 6(7):e22220, July 2011.

[KLPM10] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media. In *Proc. of the 19th conference on the World Wide Web, WWW '10*, April 2010.

[Kos09] V. Kostakos. Temporal graphs. *Physica A: Statistical Mechanics and its Applications*, 388(6):1007 – 1023, 2009.

[Lab09] Kaspersky Lab. Kaspersky lab reports a new malicious program for mobile phones that steals money from mobile accounts. `http://www.kaspersky.com/about/news/virus/2009/Kaspersky_Lab_reports_a_new_malicious_program_for_mobile_phones_that_steals_money_from_mobile_accounts`, 2009.

[LB07] M. Lahiri and T.Y. Berger-Wolf. Structure prediction in temporal networks using frequent subgraphs. In *Proc. of IEEE Symposium on Computational Intelligence and Data Mining. CIDM '07.*, pages 35–42, 2007.

[LEA+01] F. Liljeros, C. R. Edling, L. A. N. Amaral, H. E. Stanley, and Y. Aberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, June 2001.

[Lea05] N. Leavitt. Mobile phones: the next frontier for hackers? *Computer*, 38(4):20–23, 2005.

[LGP07] M. Lenczner, B. Grégoire, and F. Proulx. CRAWDAD data set ilesansfil/wifidog (v. 2007-08-27). Downloaded from `http://crawdad.cs.dartmouth.edu/ilesansfil/wifidog`, August 2007.

[LH08] J. Leskovec and E. Horvitz. Planetary-scale views on a large instant-messaging network. In *Proc. of the 17th international conference on the World Wide Web, WWW '08*, pages 915–924. ACM, 2008.

[Lie03] H. Lieu. Revised monograph on traffic flow theory. *US Department of Transportation Federal Highway Administration*, 2003.

[Liu06] Y. Liu. SymbOS.Pbstealer.D | symantec. `http://www.symantec.com/security_response/writeup.jsp?docid=2006-011915-4557-99`, 2006.

[LK03] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559. ACM, 2003.

[LKF05] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. of the 11th ACM SIGKDD international conference on Knowledge discovery in data mining, KDD '05*, pages 177–187. ACM, 2005.

[LM01] V. Latora and M. Marchiori. Efficient behavior of small-world networks. *Physical Review Letters*, 87(19):198701, October 2001.

[Mil63] S. Milgram. Behavioral study of obedience. *The Journal of Abnormal and Social Psychology*, 67(4):371–378, 1963.

[Mil67] S. Milgram. The small world problem. *Psychology Today*, 1:61–67, May 1967.

[MM65] J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3:23–28, March 1965.

[Moo02] J. Moody. The importance of relationship timing for diffusion. *Social Forces*, 81(1):25–56, 2002.

[MRM+10] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela. Community structure in Time-Dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, May 2010.

[MSS+10] L. Mercken, T.A.B. Snijders, C. Steglich, E. Vartiainen, and H. de Vries. Dynamics of adolescent friendship networks and smoking behavior. *Social Networks*, 32(1):72–81, January 2010.

[New01a] M. E. J. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132, June 2001.

[New01b] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404 –409, January 2001.

[New05] M.E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005.

[New10] M. Newman. *Networks: An Introduction*. Oxford University Press, March 2010.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, February 2004.

[PLW+09] S. Porta, V. Latora, F. Wang, E. Strano, A. Cardillo, S. Scellato, V. Iacoviello, and R. Messora. Street centrality and densities of retail and services in bologna, italy. *Environment and Planning B: Planning and Design*, 36(3):450 465, 2009.

[RCSS07] E. Van Ruitenbeek, T. Courtney, W.H. Sanders, and F. Stevens. Quantifying the effectiveness of mobile phone virus response mechanisms. In *Proc. of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN '07*, pages 790–800, 2007.

[RMM+10] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas. EmotionSense: a mobile phones based adaptive platform for experimental social psychology research. In *Proc. of the 12th ACM international conference on Ubiquitous computing, UbiComp '10*, September 2010.

[Rob86] J.M Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, September 1986.

[Rob04] J. Roberts. Enron traders caught on tape. `http://www.cbsnews.com/stories/2004/06/01/eveningnews/main620626.shtml`, June 2004. CBS News.

[SA05] J. Shetty and J. Adibi. Discovering important nodes through graph entropy the case of Enron email database. In *Proc. of the 3rd International Workshop on Link Discovery, LinkKDD '05*, pages 74–81. ACM, 2005.

[Sch09] M. Schipka. Dollars for downloading. *Network Security*, 2009(1), 2009.

[Sed88] R. Sedgewick. *Algorithms*. Addison-Wesley Pub, Reading MA, April 1988.

[SFF+10] S. Scellato, L. Fortuna, M. Frasca, J. Gómez-Gardeñes, and V. Latora. Traffic optimization in transport networks based on local routing. *The European Physical Journal B*, 73(2):303–308, 2010.

[SGC+09] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD data set cambridge/haggle (v. 2009-05-29). May 2009.

[SMMC11] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proc. of the 20th international conference on the World Wide Web, WWW '11*, pages 457–466. ACM, 2011.

[SMML10a] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora. Distance matters: geo-social metrics for online social networks. In *Proc. of the 3rd conference on online social networks, WOSN '10*, pages 8–14. USENIX Association, 2010.

[SMML10b] S. Scellato, M. Musolesi, C. Mascolo, and V. Latora. On nonstationarity of human contact networks. In *Proc. of the 2nd Workshop on Simplifying Complex Networks for Practitioners, SIMPLEX '10*, June 2010.

[SNM11] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proc. of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '11*, 2011.

[Ste95] L. Steen. *Counterexamples in topology*. Dover Publications, New York, 1995.

[Str08] R. Stringer. Six months of cyber-crime. *Infosecurity*, 5(5):11, 2008.

[ter10] Virus description: Trojan:WinCE/Terdial. `http://www.f-secure.com/v-descs/trojan_wince_terdial.shtml`, 2010.

[TMP11] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of facebook networks, February 2011.

[TT77] R. E. Tarjan and A. E. Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6:537, 1977.

[TZLM10] J. Tang, M. Zafer, K.-W. Lee, and C. Mascolo. Towards understanding the compound behaviour of periodic and random mobility on data dissemination. In *Annual Conference of ITA*, September 2010.

[VB00] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, July 2000.

[WBS+09] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. pages 205–218, 2009.

[Wes01] D. West. *Introduction to graph theory.* Prentice Hall, Upper Saddle River N.J., 2nd ed. edition, 2001.

[WF94] S. Wasserman and K. Faust. *Social Networks Analysis.* Cambridge University Press, 1994.

[WGHB09] P. Wang, M. C. González, C. A. Hidalgo, and A.-L. Barabási. Understanding the spreading patterns of mobile phone viruses. *Science*, 324(5930):1071–1076, May 2009.

[Wil10] C. Wilson. Searching for saddam: A five-part series on how social networking led to the capture the iraqi dictator. - by chris wilson - slate magazine. `http://www.slate.com/id/2245228/`, February 2010.

[WS98] D. J. Watts and S. H. Strogatz. Collective Dynamics of 'Small-world' Networks. *Nature*, 393(6684):440–2, June 1998.

[Wu10] C. W. Wu. Evolution and dynamics of complex networks of coupled systems. *Circuits and Systems Magazine, IEEE*, 10(3):55–63, 2010.

[WWA11] M.J. Williams, R.M. Whitaker, and S.M. Allen. Decentralised detection of periodic encounter communities in opportunistic networks. *Ad Hoc Networks*, 2011.

[XFJ03] B. B. Xuan, A. Ferreira, and A. Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(2):267 – 285, April 2003.

[ZCZ+09] Z. Zhu, G. Cao, S. Zhu, S. Ranjan, and A. Nucci. A social network based patching scheme for worm containment in cellular networks. In *Proc. of the 28th IEEE Conference on Computer Communications, INFOCOM '09*, pages 1476 – 1484. IEEE, April 2009.

[ZVL+09] G. Zyba, G. M. Voelker, M. Liljenstam, A. Mehes, and P. Johansson. Defending mobile phones from proximity malware. In *Proc. of the 28th IEEE Conference on Computer Communications, INFOCOM '09*, pages 1503–1511. IEEE, April 2009.