Internet Traffic Matrices: A Primer

Paul Tune, Matthew Roughan *

April 3, 2013

1 Introduction

In the era of the telephone, voice traffic dominated physical telecommunication lines. With the birth of the Internet, and its subsequent adoption as a key means of communication, data traffic has taken over (though some of this data traffic is actually re-badged voice traffic using Voice over IP). With the advent of new applications such as video streaming, and the rapid growth of data traffic from mobile devices, we are witnessing a global data explosion.

Given the ever increasing importance of the Internet, knowledge of its traffic has become increasingly critical. The Internet, however, is just an all-encompassing term to describe the vast global collection of networks, and so it largely falls on individual network providers to determine their own traffic. This knowledge is vital for continued operations because it allows network operators to perform important tasks such as providing enough network capacity to carry the current traffic, as well as to predict and prepare for future trends. Traffic data is also important in network maintenance, necessary if services and content are to be provided to customers with minimal delay and interruption.

The focus of this chapter is on the *traffic matrix*, which, in a nutshell, is an abstract representation of the traffic volume flowing between sets of source and destination pairs. Each element in the matrix denotes the amount of traffic between a source and destination pair. There are many variants. Depending on the network layer under study, sources and destinations could be routers or even whole networks. "Amount" here, in the networking context, is generally measured in the number of bytes or packets, but could refer to other quantities such as connections.

Traffic matrices, as will be clearer below, are highly important for a variety of network engineering goals, such as prediction of future traffic trends, network optimisation, protocol design and anomaly detection. Considering the importance of these matrices, the first objective of this chapter is to provide an entry level for graduate students and new researchers to current research on traffic matrices. To that end, the material is organised in a tutorial-like fashion, so as to ensure key concepts can be easily understood.

1.1 Motivation

Why study Internet traffic matrices? Simply because their implications for network operators are vast. If the traffic matrix of a network is exactly known, then, armed with topology and routing information of the network, the operator knows *exactly* what is going on in the network, rendering network management tasks relatively easy. If the traffic matrix is completely unknown, then a network operator is blind. Subtle faults may plague their network, substantially reducing performance. Congestion may be rife, or sudden shifts in traffic may cause transient traffic losses.

The issues are becoming more important. The dominant philosophy in the early days of the Internet was best effort delivery. Most applications then did not have high quality of service (QoS) requirements.

*The authors are with the School of Mathematical Sciences, The University of Adelaide, Australia (Email: {paul.tune,matthew.roughan}@adelaide.edu.au).

These applications had some tolerance to packet drops and link failures, without drastically affecting QoS. In recent years, however, the landscape of the Internet is fast changing with the introduction of streaming content such as video, high definition television and Voice over IP (VoIP), with more stringent QoS requirements. For example, excessive packet drops and delays would produce highly noticeable artefacts in streamed video. These changes are driven primarily by user demands, with the introduction of new applications, such as online social networking, entertainment services and online multi-player gaming. Furthermore, with the increasing computational power of mobile devices and increasing wireless speeds, it is evident that a significant portion of future traffic will be generated by these devices. These trends are making measurements more and more critical.

For most operators, the state of their measurements is somewhere between extremes. Most operators have some traffic data (if only link counts). However, it is rare (in our experience) to find a network operator who knows everything about their traffic. As such, one of the tasks we shall consider here is how to measure these matrices[1], but also how to obtain estimates of traffic matrices when presented with incomplete data. The traffic matrix *inference* or *completion* or *recovery* problem is one of the major areas of research into these interesting creatures, and it is intricately related to modelling the matrices, both as measurements supply data to populate the models, and because models are used to perform the inference. Even those operators with extensive measurements and exact knowledge of today's traffic matrix may be interested in methods to predict their matrices for use in future planning, and this can be seen as another form of matrix completion.

Traffic matrices have many uses, apart from the simple fact that this type of business intelligence is critical to understanding your network. The more direct uses include network optimisation, anomaly detection and network protocol design.

There are three common optimisation problems on networks. Capacity planning is needed to ensure there is adequate bandwidth for users in the present and future, but at minimal cost. There are two types of network planning: evolutionary planning and green fields planning; see §5 for details. Traffic engineering tasks include day-to-day maintenance of the network as well as predicting growth trends and anticipating traffic demands. Routing involves organising traffic flow in the network. This includes functions such as finding the shortest paths for flows but also, importantly, load balancing to ensure links remain uncongested. In all these cases, the traffic matrix is a key input to perform the tasks effectively and efficiently.

Traffic matrices can also be used to detect sudden shifts in traffic due to anomalies. Anomalies include sudden unexpected events, such as network failures, or more malicious events, such as the September 11 World Trade Centre attack, worm infections and distributed denial of service (DDoS) attacks [74]. Regardless, these anomalies need to be detected so as to develop appropriate measures against possible threats to the network.

Traffic matrices may also be used to conduct reliability analyses, where the effect (on traffic) or network failures is considered. A basic task in most network design is to create redundant paths to carry traffic in case of failure, but if high reliability is required, then an operator should also ensure that there is sufficient capacity in the network to carry this traffic along its alternate paths.

Further, the performance of many network protocols depends on the traffic they carry, and the cross-traffic which forms their background. Design of new protocols therefore requires realistic measures, or models of such traffic. Models can be used to test protocols on artificially synthesised traffic. In this way, the limitations of a protocol may be understood in a controlled environment before running it on an actual network.

These issues will be examined in-depth in §5, where algorithms utilising traffic matrices to perform these tasks will be discussed.

[1] This chapter isn't really a primer on measurement tools as such, so much as the principles that underlie those tools. We won't tell you how to set up Netflow on your particular router, but instead we will aim to tell you what you could achieve with this type of flow-level measurements.

1.2 A primer on modelling

Motivated by a lack of understanding of Internet traffic and as a response to the recent shifting landscape of traffic demands, various traffic models have been developed over the last decade or so. There are literally hundreds of papers describing data traffic modelling, however, here we shall focus on one group of such models that are particularly useful to operators: models for traffic matrices.

A good model captures the essential characteristics of the underlying traffic, while being robust enough to cope with changes in traffic, for example, with regards to the time of the day, or the introduction of unexpected events, such as attacks on the network. Much can be gained from a good model, enabling the categorisation and analysis of the data in a systematic, and desirably, a simplified manner.

The ultimate goals are to use these models in various networking tasks, and the "essential" qualities that the model must capture depend critically on the task. For instance, traffic properties are highly dependent on time scale: in very short time scales (in seconds), the traffic volume distributions have been shown to exhibit complex statistical behaviour such as high variability, long-range dependence, and multi-fractal behaviour. On long time scales (hours to days to weeks) traffic exhibits strong periodicities: there are distinct diurnal and weekly cycles with traffic peaks occurring around mid-day or in the evening and troughs in the early morning. This pattern correlates to user behaviour, where they access the Internet during the day for work or school and sleep in the night. There is also an observable long-term growth trend underlying the traffic volume when measured over years, corresponding to increasing global traffic demand. However, most traffic matrices are measured at some intermediate time scale, often at 5 to 15 minute intervals.

The time scale is not just a property of the measurements though, the tasks we wish to perform usually have an associated timescale. In capacity planning, we are often concerned with a "busy hour" – certainly peak measures over some intermediate period are important. However, anomaly detection needs to act at a relatively fine time scale, in minutes, or perhaps even in seconds.

We also have to consider the *planning horizon* of a task. Capacity planning may be aimed at determining the correct network design six months in advance or more, where-as traffic engineering is sometimes conducted on scales as short as a day. The planning horizon determines how far in advance we must predict traffic matrices.

We can, perhaps, start to see that modelling traffic matrices is quite a challenging task. However, there are complexities layered on complexities. The Internet is designed in terms of layers, with different protocols overlaid atop each other. Such a paradigm was adopted to ensure that each layer works independently (in theory) to each other, although there are some overlaps between these layers in practice. The basic properties of traffic matrices will depend on the network level, but the congestion control mechanism at the transport layer can change traffic. Traffic can be measured between logical or physical source/destinations, and at different levels of aggregation, and at these different levels, new models may apply. Furthermore, changing trends in network usage, deployment of Content Distribution Networks (CDNs), and increasing mobile traffic mean that any model developed today may become outdated quickly.

Underlying all Internet traffic is the undeniable fact that all traffic is driven by consumer demands. It does not take a genius to realise that human behaviour is inherently difficult to model, let alone understand. A complicated model may be accurate for today, but fail in predicting traffic demands for the next few years or so, given fluctuations in demand and unexpected changes in traffic patterns [89]. Thus we believe there is no single model that captures all observed properties of once and future traffic matrices. It is often preferable to have simple, robust models, in preference to precise, but fragile ones.

Measurements of networks serve as the foundation in any model development. The caveat, however, is the measurements themselves are subject to errors and inconsistency, which may lead to an incorrect model. Moreover, several hypotheses may fit a particular observation of the network, leading to several possible models explaining the observation. To argue for the use of one model over another requires additional knowledge from new data, or from domain knowledge. Additionally, even if new information becomes available, there is a question of how to incorporate it into the model. There is a variety of

possible approaches, all equally valid [6].

Moreover, it is misleading to talk about a "correct" traffic matrix model. As pointed out in [6], just because a model replicates properties of the observed data does not necessarily mean the model is "correct", as there is a dangerous possibility of over-fitting. After all, better fitting is usually achievable simply by adding more parameters to the model. Several information criteria serving as guidelines do exist to prevent the over-fitting problem, such as the Akaike information criterion [5], Bayesian information criterion [77], Minimum Message Length [94] or the Minimum Description Length [69]. While these criteria are beyond the scope of discussion, the basic principle is to choose the simplest explanation (measured in an information metric) amongst all competing explanations of the data. The issue highlights the difficulty of choosing the number of parameters in a model, as it requires several tradeoffs between simplicity and its connection to reality.

It is for these reasons, models should be evaluated not just on their accuracy in making predictions of particular statistics, but also their simplicity, robustness and consistency with relation to the realities of network operations. Model assumptions should "make sense" to an operator as well as be empirically tested on various datasets to understand their reliability and pitfalls, which is not to say we can't learn new ideas and principles from measurements. We just need to keep in mind that scope of application, and the usefulness of these principles in practice may be limited.

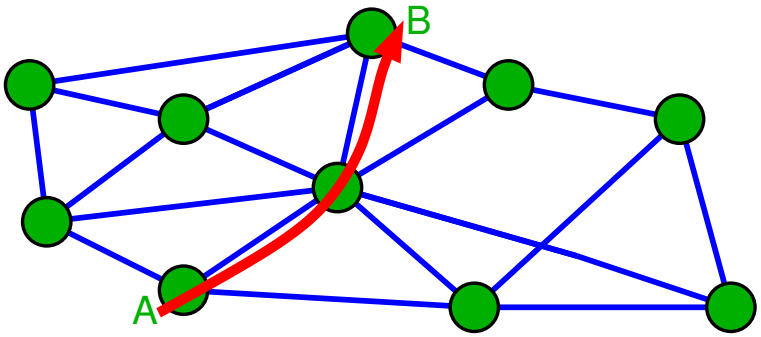
At a fundamental level we need to accept that models are all unrealistic in some way. A model describing the properties of a smaller scale network, such as a Local Area Network (LAN), may be unsuitable at the backbone network level. The underlying assumptions of one may not hold in the other. Models are simplifications of the glorious complexity that comprises humanity's primary means of telecommunication. We must, instead, reread the adage, by George E. P. Box: "All models are wrong, but some are useful". Some models have been more successfully used in real networks, and it is to these that we shall devote most time here. However, we shall endeavour to cover the majority of simple models with the view that individuals should use the best model *for their application* without fear or favour.

No doubt, the murkiness and apparent self-contradictions of this discussion have left our readers no wiser, as yet. Modelling is a topic that could be discussed endlessly. It is our aim that through consideration of the qualities of various traffic matrix models, we shall not just inform about these particular models, but also bring the reader to a new understanding of modelling, which makes these issues a little less opaque.

1.3 Chapter outline

The theme of this chapter is to directly report on all key works in the field. No attempt will be made to provide overt commentary on what techniques or models are good or bad, as the objective of this chapter is to present comprehensive summaries of existing work. It is important to note that the techniques, algorithms and models presented here were developed as tools to suit specific applications. Hence, it is the onus of the practitioner to evaluate and decide which of these are applicable to his or her situation. Whenever possible, the strengths and weaknesses of the inference techniques and models are discussed objectively, so as to minimise the problem of a practitioner treating a particular tool as the silver hammer for all proverbial nails.

The chapter is organised as follows. In §2, an overview is provided on traffic matrices: the basic definitions and some illustrations to give a better handle on the topic. §3 discusses how data on traffic matrices are collected in practice. §4 discusses the various models proposed in literature that aims to capture the statistical properties of traffic matrices. §5 goes into a more in-depth treatment of the applications of traffic matrices, in particular, how traffic matrix models are used for inference, network optimisation applications, anomaly detection and traffic matrix synthesis. Not everything is known about these matrices, and §6 summarises some open questions and concludes by giving some thoughts on what the future holds for traffic matrix research. The chapter is concluded in §7.



$$X = \begin{pmatrix} x_{AA} & x_{AB} & x_{AC} & \cdots \\ x_{BA} & x_{BB} & x_{BC} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Figure 1: *An example of a traffic matrix. Note that often the diagonal elements, $x_{AA}, x_{BB}, \ldots$ are zero as this traffic does not cross the network, however, in almost as many cases it is non-zero because a "node" in the graph represents an aggregation of devices such as a PoP or an AS. In these cases we often do wish to measure these diagonals, even though they may not cross the logical links pictured, because they affect traffic engineering within the PoP, for example.*

2 Definitions and Notation

In this section, a formal approach to networks and traffic matrices is defined. Traffic originates from a *source* and is delivered to a *destination* (or several destinations). The traffic traverses a set of *links* between some set of nodes. The links connecting nodes define the *topology* of the network, and the paths chosen by traffic flows determine the *routing*. Traffic may be split across multiple paths by load balancing, or may keep to a single path. Often, sources and destinations are identified with the network nodes (though they can also refer to a location in a logical space attached to the network, for instance IP addresses of prefix blocks).

Let Ω denote the non-empty set of all nodes in a network and let $|\Omega| = N$. The nodes often have a physical, geographic location, and so we regard the indices of the sources and destinations as *spatial variables*, even when nodes are actually logical entities, such as Autonomous Systems (ASes), or cannot be identified with the nodes, as with IP addresses.

A traffic matrix is naturally represented by a three dimensional, nonnegative hyper-matrix $\mathbf{X}(t)$, with i, j-th entry $X_{i,j}(t)$. Each entry represents the traffic volume, or demand, measured in terms of bytes or packets, from source i to destination j in time interval $[t, t + \Delta t)$, the full measurement interval denoted by $\mathcal{T}$. As an aside, a matrix representation is useful for the representation of other aspects of the network, for instance, delay, jitter, loss, bottleneck-bandwidth and distance [55], but throughout the chapter, traffic will be the focus. Whenever the context is clear, for example, when considering only the spatial structure of the matrix, the time index t is dropped.

A closely related concept is the *demand matrix*, distinct from the traffic matrix because the former is *offered load*, and the latter *carried load*. They may be the same, but may differ where congestion limits the carried traffic, or rate limiting is used on some traffic streams. In general we cannot measure offered traffic, only carried traffic, and so almost all empirical research has concentrated on traffic matrices, but it is important to note that many of the assumptions of traffic-matrix models are actually motivated by intuition about demand matrices, and these may not apply where the two differ substantially[2].

Large-scale, real-time monitoring of traffic is intractable at present, thus limiting measurements to the average traffic in a discrete time interval. Shorter time intervals *i.e.,* small Δt, benefit anomaly detection applications, the tradeoff being a possibility of uncertainty from traffic burstiness at shorter time scales combined with larger potential measurement or sampling errors[3]. Longer time intervals

[2]Many works make no distinction between the two types of matrices, leading to confusion. Here we shall try to keep the two distinct.

[3]Traffic is typically sampled at the backbone network level to cope with the tremendous volumes of data that could be collected.

result in "smoother" traffic demands, averaging out measurement errors. However, this smooths out real variability in the traffic as well, and can result in meaningless estimates in the presence of strong non-stationarity. Hence, the choice of Δt depends on the application and available measurements. Common choices range from 5 minutes to an hour. Further discussion on the temporal properties of traffic demands is found in §4.

There are two popular definitions of traffic matrices: the origin–destination (OD) matrix and the ingress–egress (IE) matrix.

(i) **OD traffic matrix**: this matrix measures traffic from true source to destination, *i.e.,* the point that generates a packet to the point that receives it. In the Internet, it is perhaps most reasonably defined in terms of Internet Protocol (IP) addresses. However, if Ω is defined over the entire IPv4 address space of 2^{32} addresses (with even more for IPv6), this poses storage and computational problems. Moreover, the matrix would be very sparse. What's more, protocols such as NAT, HTTP proxies and firewalls may obscure the true IP address mappings. One way to overcome some of these deficiencies is to aggregate the traffic matrix into blocks of IP addresses, frequently using routing prefixes.

(ii) **IE traffic matrix**: any single network operator sees only a small proportion of the Internet OD matrix. Thus this matrix is not just unknown, but unmeasurable (by a single operator). Instead, many operators find that using their edge routers (or even the edge links) as sources and destinations results in a local traffic matrix of great use. We call this the IE traffic matrix as the set Ω includes ingress, *i.e.,* traffic going into the network, and egress, *i.e.,* traffic flowing out of the network, points as proxies for sources and destinations. A single ingress or egress "node" may denote a router, a collection of physically co-located routers called a Point of Presence (PoP), or some other abstract collection of traffic ingress/egress points depending on the level of coarseness required in the modelling process. The PoP level convention is often adopted as it provides a simple visualisation of the network to network designers and operations management.

IE traffic matrices can be obtained in a number of ways. They can be formed from OD traffic matrices simply by mapping IP prefixes to ingress/egress locations in the network, but this assumes knowledge of all flows traversing the ingress/egress nodes. Traffic at egress nodes may be inferred from router data (see the next section) and measurements of ingress traffic, but typically, the converse is difficult. Likewise, it is usually difficult to form an OD matrix from IE matrices.

Consequently, the IE traffic matrix is frequently adopted for network optimisation applications as it is more practical to measure, and because in aggregating traffic the OD flows are "bundled" together into locally meaningful groupings. A network may carry flows between billions of IP address pairs and millions of prefix pairs, but only thousands of router pairs, and hundreds of PoP pairs. In this way, the IE matrix is a more compact representation, but more importantly, the aggregation of the traffic into larger bundles results has a smoothing effect of the data, reducing the number of independent parameters that may have to be estimated. At the PoP level, the aggregation of flows results in averaging out sampling error (similar to the choice of Δt above). This is highly beneficial for numerical iterative algorithms used to estimate the traffic matrices, as aggregation leads to better conditioning of the traffic matrix. The trade-off, unfortunately, is the loss of fine grained data, as one can no longer observe IP level flow data, or examine application profile data.

Another consideration worthy of concern in applying traffic matrices concerns *invariance*. A good representation of the traffic has to be invariant to other network aspects, such as routing and the network topology. For instance, if the traffic matrix for a network changes in response to changes in link placement, then the matrix is not terribly useful for network design. IE matrices are subject, for example, to large changes due to routing shifts, and this means that they are less useful to operators compared to OD matrices. However, the practicalities of measurements mean that IE matrices may be all that is measurable.

That leads naturally to our next topic: measurement of traffic matrices.

3 Measurements

In theory it is possible to collect accurate data of Internet traffic from a network. In reality, however, many issues confound such measurements. Budget constraints whether from an economic viewpoint or from the massive data storage facilities required due to the sheer amount of data traversing a backbone network limit what can be achieved, and even good measurements systems can suffer from errors and missing data. Added to this, current operator practice rarely includes any significant calibration of measurement apparatus, so often the degree of accuracy of the measurements in unknown.

There are several well known strategies for collecting traffic measurements. A *packet trace* is a collection of packets headers (perhaps with some payload) and timestamps. A packet trace can be collected through various means, for example, through hardware such as a splitter placed strategically in optical fibre, adding a monitor port on a router or through software tools such as `tcpdump`, executed on several hosts in a shared network. An OD traffic matrix can be constructed from such a trace by simple consideration of the IP address in the packet headers (with the caveats mentioned earlier).

Such an approach is ideal in many respects: we have almost complete information, and the matrices may be drawn at almost any time resolution. However, collecting packet traces is expensive due to dedicated hardware, and the huge amount of storage required, with over a terabyte of data per hour on OC48 links (2.5 Gbps) being possible. It is rare for any but the smallest network to be able to completely instrument its network at this level of detail.

Fortunately, constructing a traffic matrix does not require such detailed information. Perhaps the most common alternative is *flow-level aggregation* where packets are aggregated according to a common flow key. One popular definition of the key is the 5-tuple comprised of the IP source and destination address, TCP source and destination port numbers and protocol number. A series of packets possessing a common key is called a *flow*, and we maintain simple statistics (byte and packet counters, and start and stop times) for each flow. The aggregation of packets into flows reduces the number of records needed to be stored by removing redundancies of data from a packet trace. Flow-level collection is generally performed in 15 minute time bins[4] and is often an in-built function of a router. The only additional infrastructure required is the Network Management Station (NMS) and flow records themselves are usually compressed by the router before being exported to the NMS. Despite this reduction, flows arrive at a router at rapid rates and the formation and storage of flow records at a router often burdens the router's CPU.

To further reduce the number of flow records at a router, sampling methods are employed. The most popular sampling method is packet sampling, where incoming packets to a router is sampled based on predetermined sampling patterns, used, for instance by Cisco's NetFlow [1]. Such pseudo-random patterns have a similar effect to independently picking incoming packets given sufficient mixing of traffic. The sampling rates can be adjusted depending on the capacity of the incoming links with recommendations such as 1 in 256 packets for OC192 links (10 Gbps). Higher capacity links require aggressive data reduction, and so lower sampling rates are used.

Packet sampling reduces the number of flows significantly by omitting many, but it is important to realise that although it may select *packets* in an unbiased way, it is not unbiased with respect to *flows*. Packet sampling has a strong bias towards long flows, since it is more likely to have sampled packets from a long flow than a short one. Furthermore, the sampled flow size is not the true size of the flow and there are several works [29, 30] proposing methods to sample and estimate the true size of a sampled flow. While the strong bias may be a problem for some applications, there is usually no problem in using sampled flows to form the IE traffic matrix, since the large volume of each entry and aggregation with other flows averages out the bias from sampling.

In addition to packet sampling, we may also sample a set of flows, and these sampled flows can then be used to create traffic matrices. The resulting reduction in intermediate storage and processing can be substantial, particularly if the sampling is done in a clever way [29, 30].

[4]The issue of timing of flows is actually somewhat more complicated, but readers should refer to detailed descriptions of specific flow-capture protocols for information on their particular flow capture.

It must be remembered that sampling is an inherently lossy process, regardless of the underlying sampling method used. The loss of information translates into errors or noise in the data. The size of these errors should, in best practice, be estimated for a given setup, but most operators do not undertake such procedures due to the difficulties involved in obtaining ground-truth data with which to compare the sampled data. In many cases it is simply assumed that these errors, once the data is aggregated further into traffic matrices, are negligible, but this assumption is rarely justified by data.

A less costly alternative are easily obtainable link counts. A link count, or link load measurement, gives the volume (in bytes or packets) of traffic on a link during a particular time interval. Link counts are obtainable from measurement data by the Simple Network Management Protocol (SNMP) [19], defined as part of the IETF standard and present on many Internet devices, including most routers. SNMP data from a single router provides two measurements for each interface, the incoming and outgoing byte counts. SNMP data is obtained by an NMS by periodically polling requests through an interface, typically UDP port 161. The polling period varies from 1 minute to several minutes, but the default seems to be 5 minutes. SNMP data is highly susceptible to error, due to the following factors:

(i) **missing link observations**: data is transmitted via unreliable UDP packets and there may be errors when copying the data into the observer's archive,

(ii) **incorrect link observations**: poor router vendor implementations causes inaccuracies, and

(iii) **sampling coarseness**: polling times are often inaccurate either due to poor NMS or SNMP agent implementations, high loads, or network delays.

As with flow-level data, link count data should be calibrated, but rarely is. There is only one experiment of which we are aware that does so [71]. The study showed that in one network, SNMP errors were typically low (90% of measurements had an error of less than 1%), but a small number of measurements were very large, some as large as 100%. This type of heavy-tailed distribution causes problems for some estimation approaches and should be considered in context.

Another drawback of SNMP data is that it only provides aggregate link statistics, omitting details such as types of traffic on the link and the traffic source and destination. Despite all these, SNMP data is, at present, the easiest way to obtain large-scale traffic data.

The observed link counts provide some information about the traffic matrix, but only in an indirect manner. Thus, the traffic matrix has to be inferred. Network tomography was first introduced by Vardi [91], with the inspiration taken from inference techniques for medical tomography, as both problems are similar in nature. Vardi's work was subsequently expanded upon by Tebaldi and West [87] and Cao *et al.* [18]. Various other works in network tomography measure other properties of the network, such as link delays (see [20, 23] for an overview) via active packet probing, but for traffic matrix estimation, we are only concerned with the link count observations from SNMP data.

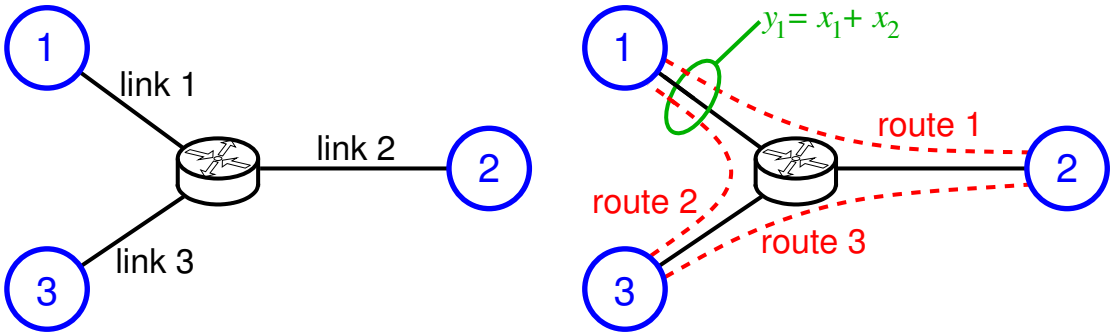
Mapping traffic to links requires topological and routing data in the form of a *routing matrix*. The routing matrix $\mathbf{A}$ is defined by

$$A_{i,r} = \begin{cases} F_{i,r}, & \text{if traffic for } r \text{ traverses link } i, \\ 0, & \text{otherwise.} \end{cases}$$

with $F_{i,r} \in (0, 1]$ defining the fraction of traffic from source-destination pair $r = (s, d)$ traversing link i. Fractional values occur in cases when some form of load balancing on traffic is performed. It is generally assumed, however, $F_{i,r} = 1$, resulting in $A_{i,r} \in \{0, 1\}$. The size of the routing matrix depends on the network, and with N nodes and L links, the routing matrix has size $L \times N(N-1)$ (traffic from a node is assumed not rerouted to itself).

Information on the routing matrix can be obtained from several different sources (router configuration files, traceroutes, or from the routing protocols themselves), but the collection of such information is not the topic of the chapter, and will be considered elsewhere in this book. A common assumption is that the routing matrix remains stable during the measurement interval, thus the temporal dependence is dropped, *i.e.,* $\mathbf{A}(t) = \mathbf{A}$ for all $t \in \mathcal{T}$. However, changes in routing may occur if there are

link or router failures, necessitating traffic reroutes. Generally, it is assumed the measurement interval is chosen over a period of time (minutes to hours) when $\mathbf{A}$ is stable enough to be considered static, justified by observations in [63], however in at least one case it is proposed that the changes be created, and exploited [83] for traffic matrix inference.



(a) *Link labels.* **(b)** *Traffic labels and routes.*

Figure 2: *A simplified network and traffic (the main simplifications are that we only consider a single router, and only consider unidirectional traffic, not bidirectional as in real networks).*

All the link counts may be grouped into an $L \times 1$ vector $\mathbf{y}$. Note that L is usually much smaller than $N(N-1)$. Then, based on link observations in one measurement interval, the SNMP link counts may be expressed as

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ is the $N(N-1) \times 1$ vectorised version of the traffic matrix $\mathbf{X}$, with its columns stacked upon one another.

Figure 2 presents an example of (1). It shows how the traffic on a single link y_1, is built from the sum of traffic routed across the link $x_1 + x_2$. We can see that by stacking each of these equations we would get

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{2}$$

which, written in matrix notation, is just (1). Note that in this case the routing matrix A is invertible, so the problem of inferring the traffic matrix from link measurements is easy, but this is rarely the case. Usually, the matrix is highly underconstrained.

There are two main assumptions implicit in this observation model. It assumes the traffic matrix is stationary, *i.e.*,, its statistical properties remain stable throughout the measurement interval and there are no errors in the measurement. Stationarity is preserved by choosing an appropriate measurement interval, say 1 hour (see §4). Moreover, in reality, errors do occur and to account for it, the model is extended to

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}. \tag{3}$$

The second model is a simple noise additive model often used to test the robustness of an inference technique. Each element of the additive noise term $\mathbf{z}$ typically chosen to be independent and identically distributed (i.i.d.) white Gaussian noise with mean zero and variance σ^2. Often σ^2 is kept small, as large values would result in some elements of $\mathbf{y}$ violating the non-negativity constraint. It is for this reason other distributions may be used, for example, log-normal or gamma distributions. Additionally, due to the problem of missing link information due to poor SNMP implementation, some of the elements of $\mathbf{y}$ may be missing. Finally, if the given routing matrix $\mathbf{A}$ is incorrect, the observations $\mathbf{y}$ would significantly depart from the true SNMP link counts. However, most works assume an accurate $\mathbf{A}$ because there are reliable methods for obtaining routing information [79].

There are usually many less links than the total number of IE pairs, and so the inverse problem above is highly underconstrained. Whether noise is present or not there may be an infinite number of solutions $\mathbf{x}$ that fit the observations (1). Figure 3 shows a picture of such a network, where we only measure at the bottleneck. Now, even in this very simple network the equations $y_1 = (1,1)^T(x_1, x_2)$ are underconstrained. In order to make progress, some additional information needs to be assumed, usually in the form of a traffic matrix model, and we shall consider some of these in the following section.

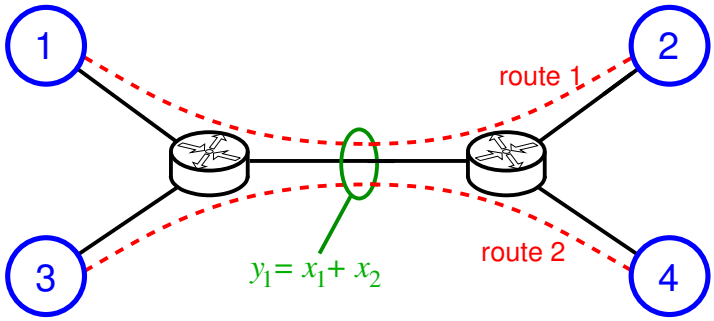


Figure 3: *A harder inference problem where we only have one measurement, but two traffic elements to estimate. There is therefore a fundamental ambiguity in this problem.*

There are other strategies for measurement. For instance, if MPLS is being used, this creates a set of tables (in many implementations) that can almost be read directly to obtain the matrix (*e.g.,* see [10]). Alternatively, the network operator may have more accurate local traffic matrices, obtained through specific functionality at the routers. It is, in principle, easy for a router to keep counts of its decisions [92], essentially amounting to a table of the volumes of traffic between pairs of interfaces. Locality here is defined in the sense of the matrix's restriction to a single router – we essentially see an IE traffic matrix of the single router's interface. These local matrices from all routers in the network can be used to improve the estimation of the IE traffic matrix; see §5. On some special cases, such as a star network, a single local matrix would be equivalent to the traffic matrix, serving to highlight the information gain from local traffic matrices. These matrices provide greater than a two-fold increase in accuracy of the tomography estimation schemes by [56, 97].

4 Models

In this section, several canonical as well as recently proposed models are presented. Modelling is divided into three categories: purely temporal modelling, spatial modelling and spatio-temporal modelling.

4.1 Temporal Modelling

Purely temporal models only focus on the time series properties of the traffic matrix. A key application of these models is in anomaly detection, so as to be able to pinpoint the time and location of the anomalous event. This is especially vital in detecting attacks on the network or worm outbreaks. However, temporal behaviour is also important in prediction, say for planning capacity of a future network.

Before delving into the models, some basic issues regarding the temporal properties of OD flows need to be understood. It is commonly agreed that IP data traffic is rising exponentially, and has been for more than a decade [2, 22, 40, 62]. There was much early controversy about such growth estimates, because the growth rate was vastly overestimated based on a small sample of data. However, these days, exponential growth is considered the common case (though with a much lower rate of growth), and

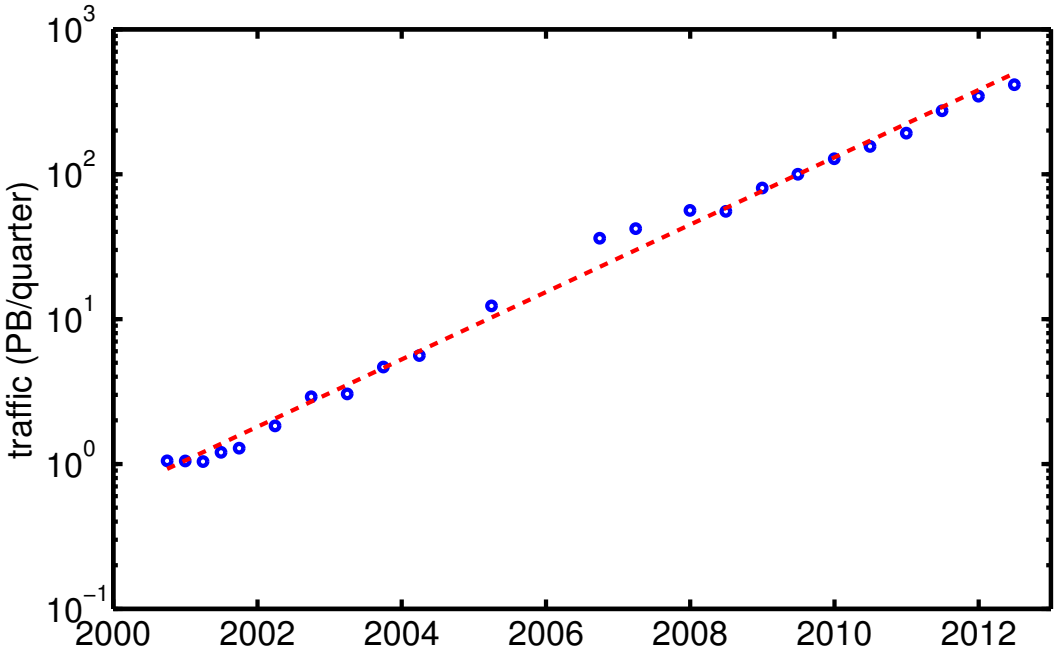


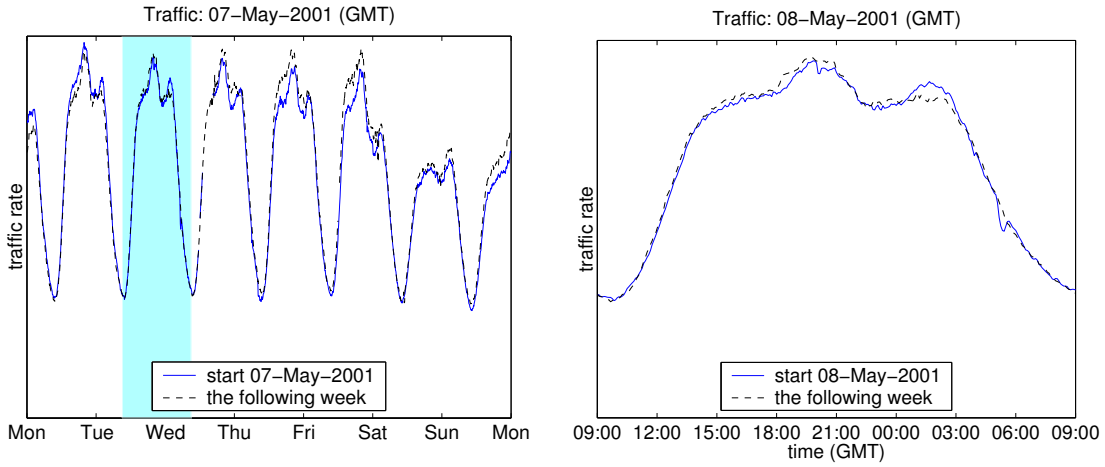
Figure 4: *Australian Internet traffic volumes from 2000-2012. The dashed line shows a linear fit to the data. Note the log y-axis, so the plot shows quite a reasonable fit to exponential growth with a doubling period of 473 days. Over the same period the growth in broadband subscribers has been almost exactly linear (soon this trend must decrease as a large proportion of Australia's population are now connected), so most of the growth has come from growth in the amount downloaded per customer.*

is justified both by data, and as a consequence of increasing computational and networking speeds, governed by Moore's and Gilder's laws respectively. As of the present, the introduction of mobile devices and the rapid growth of traffic from these devices are set to further increase data traffic in the years to come. Cisco (who admittedly have a vested interest in a high forecast) estimate that "Annual global IP traffic will surpass the zettabyte threshold (1.3 zettabytes) by the end of 2016" [2].

Relatively few countries appear to monitor their national traffic, but Australia is an exception. The Australian Bureau of Statistics have collected and published traffic statistics for many years [3]. Figure 4 shows the growth of traffic in Australia from 2000-2012.

Regardless of your belief about the rate of growth, and/or the best model (exponential is common, but linear and logistic models may also be appropriate in many cases), any model of long-term traffic needs to be able to incorporate such growth.

Second, most network traffic is human generated. Therefore, it stands to reason that traffic is influenced by human activity in a 24 hour cycle. In fact, distinct diurnal patterns have been observed, with peak traffic occurring around mid-day and throughs during the night. This can be seen in Figure 5, where the traffic rate of a large ISP is plotted over the span of a week in May 2001 and over a single day. Peak traffic rate is noticeably significantly less on the weekends. This correlates to the daily schedule of an average human being, where mid-day traffic is generated for work or school purposes, while the lack of traffic during the night correlates to sleeping periods. The regularity of this behaviour can be quite strong as shown in Figure 5 where successive weeks' data are overlaid so one can see how closely they match. This degree of regularity is only seen in large aggregates of traffic (the figure shows the traffic for a large PoP in North America), but most traffic measurements see some measure of cyclic behaviour.



(a) *Traffic rate over the span of a single week starting from the 7^{th} of May 2001, with the traffic rate of the following week overlaid.*

(b) *Traffic rate of a single day, 8^{th} of May 2001, on the same ISP, with traffic on the 15^{th} of May 2001 overlaid on top.*

Figure 5: *Observations of the cyclicality of the total traffic rate of a large PoP.*

Third, and leading from the last point, the traffic volume itself is dependent on the measurement period and the aggregation level of traffic. At very short scales, in milli-seconds or seconds, the traffic distribution is highly variable, and shows strong dependencies, making use of such measures statistically non-trivial, even if such measurements were easy to collect. A common paradigm is to consider a measurement interval of minutes to an hour, where measurement is easy. Also important is the fact that at these times scales *stationarity*[5] of the traffic volume distribution is often a reasonable assumption, though we can see the limits of this in Figure 5 (stationarity clearly doesn't hold for more than a couple of hours), however it was shown that *cyclo-stationarity* holds to a large extent [81].

Fourth, there are natural variations in traffic over time, and these are often modelled as a random (or stochastic) process. This random process could have all sorts of features, but there are some basics that should be observed by all reasonable models. For instance, Network operators aggregate traffic from multiple sources, which is known as *multiplexing*. Multiplexing is used to boost the efficiency of the links in a network by "smoothing" out variations in traffic. The apparent smoothness is a result of decreases in the relative variance, as predicted by the central limit theorem [73]. The more OD flows multiplexed on a link, the higher the efficiency and smoothing effect, provided the aggregated bandwidth does not exceed link capacity. Thus, any model for the large traffic rates in a network must be consistent under multiplexing, for example, when the number of flows being multiplexed is increased the relative variance should decrease in a predictable manner. Furthermore, the statistical properties of the aggregated traffic must also be consistent with the statistical assumptions of the traffic from a single user.

Finally, although rare, sometimes there may be sudden "spikes" in traffic. Such a component may arise from unusual traffic behaviour, such as DDoS attacks, worm propagation or BGP routing instability from misconfiguration. Flash crowds are also an example of this behaviour, which happens when there is a significant jump in the number of clients to a particular web server or content distribution network. Extreme unforeseen events, such as the September 11 attacks on the World Trade Centre in 2001 may instead cause a significant drop of traffic rates. In any case, a massive shift in traffic rates

[5]Stationarity refers to the concept that the statistics of the traffic (for instance the mean and variance, but in general including all statistics) are constant with respect to the time at which they are measured. In Internet traffic data it is only ever approximately true. Moreover, it is hard to test for stationarity when traffic has long-term correlations, and so we can only ever talk about the degree of stationarity.

would be of interest to a network operator.

One temporal model of OD flows traversing backbone routers was proposed by Roughan *et al.* [74] by generalising the Noros model [59], originally used for modelling LAN traffic. Each OD flow is assumed to be generated from an independent source. The model is characterised by the following components (at time t):

(i) $L(t)$, the long term traffic trend,

(ii) $S(t)$, the seasonal (cyclical) component,

(iii) $W(t)$, random fluctuations, and

(iv) $I(t)$, the anomaly component.

These components correspond to the observations of traffic described earlier.

The long term trend, $L(t)$, depends on the observed underlying traffic growth in the data. An exponential growth model for instance, could be found by fitting $L(t) = A\exp(ct)$ to the data, with the parameters A and c easily estimated via log-linear regression as in Figure 4. The cyclicality of the seasonal component is $S(t + kT_s) = S(t)$ for all integers k, with a period T_s. The component $W(t)$ is assumed to be a stochastic process with zero mean and unit variance, capturing the spurious components of the traffic. $I(t)$ captures the large variability of traffic from anomalies. These events were captured in an individual component to separate their influence from normal traffic.

Let $x(t)$ denote the volume of an OD flow at time t. The model takes the following form,

$$x(t) = m(t) + \sqrt{am(t)}W(t) + I(t), \tag{4}$$

where $m(t) = S(t) \cdot L(t)$ is the mean of the OD flow, assumed to be a product of the seasonal and long term trends, and a is the *peakedness* of the traffic. The average is modelled in such a way because as large OD flows have a larger range of variation in the size of their cycles. The parameter a controls the smoothness of the OD flow's volume in a way that is consistent given multiplexing of aggregated flows.

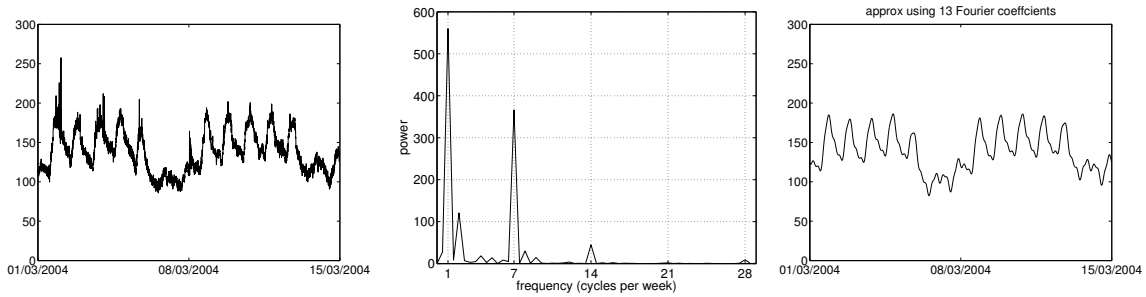
One nice feature is the preservation of the properties of the model through a linear combination, advantageous when looking at the aggregated behaviour of the OD flows. Consider K aggregated OD flows, then

$$x_{\text{agg}}(t) = \sum_{i=1}^{K} m_i(t) + \sum_{i=1}^{K} \sqrt{a_i m_i(t)} W_i(t) + \sum_{i=1}^{K} I_i(t).$$

The mean of $x_{\text{agg}}(t)$ is simply $m_{\text{agg}}(t) = \sum_{i=1}^{K} m_i(t)$, and the peakedness is the weighted average of the component peakedness, $a_{\text{agg}} = \frac{1}{m_{\text{agg}}(t)} \sum_{i=1}^{K} a_i m_i(t)$. The linearity properties allow $x_{\text{agg}}(t)$ to be expressed in the same form as (4) with the new parameters $m_{\text{agg}}(t)$ and a_{agg}. The linearity property enables the consistent computation of the variances of the aggregated traffic, which is useful for network planning and analysis. Besides this, [74] demonstrated the ease of estimating the parameters of model (4), via simple estimators and filtering.

The cyclical nature of the aggregated OD flows is also amenable to Fourier analysis. The Fourier transform decomposes a periodic signal into a weighted sum of sinusoids with distinct frequencies and phases. It would be reasonable to assume the observed cycle can be represented by a small number of Fourier coefficients, since the cycle is close to the shape of a sinusoid (see Figure 5 for instance). Indeed, it has been demonstrated this is the case with the traffic volumes [31, 82], where as little as just 5 Fourier coefficients were needed to achieve low error in fitting large OD flows of a Tier 1 network, demonstrating the relatively few significant frequencies present in a diurnal cycle of the OD flows. Figure 6 shows a similar analysis of Abilene data. It shows a simple example of the excellent degree of approximation to traffic we can obtain using only a very small number of Fourier coefficients corresponding to daily periods. The figure shows the important components of the power-spectrum of two weeks of Abilene data, clearly highlighting the importance of the daily and weekly cycles.

Figure 6a shows two weeks of data, and Figure 6b shows a zoom in of the important region of the Discrete Fourier Transform (DFT). We can see that only a few peaks (with frequencies of 1, 2, 7, and 14 cycles per week, corresponding to daily and weekly cycles and their harmonics) are large enough to matter for gross features. The approximation curve in Figure 6c is generated using only the largest 13 of these terms.



(a) *Observed bit rate for two weeks on the Abilene network.*

(b) *DFT focussed on the important region.*

(c) *Fourier approximation of the traffic data.*

Figure 6: *Fourier analysis of Abilene data.*

The choice of time interval to use in Fourier analysis/approximation is interesting. A longer interval provides more data, and hence better estimates if the data is truly *cyclostationary*[6]. However, as we earlier discussed, there are noticible trends in the overall volume, and so it is reasonable to assume that there will sometimes be significant shifts in the pattern (with respect to time of day or time of week). In these cases, extending the length of the dataset can confuse the statistical variability with non-stationary affects, resulting in biased estimates. The best tradeoff appears to depend on the dataset, but periods of perhaps a month seem to work reasonably well for estimating weekly cycles.

Principal components analysis (PCA) has also been employed to quantify the temporal correlations of the traffic matrix. If $\boldsymbol{\mathcal{X}}$ is a matrix where the rows represent a measurement (for instance a OD flow) and columns represented traffic volumes at time t, then temporal PCA decomposes the matrix $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}$ into its corresponding components of eigenvalues and eigenvectors. Often, each column is *centred*, simply by subtracting the mean vector $\bar{\mathbf{x}}$, the average of all columns, from each column in $\boldsymbol{\mathcal{X}}$. In what follows, $\boldsymbol{\mathcal{X}}$ is assumed to be centred.

The matrix $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}$ is *positive semidefinite*. Visualising this geometrically, if the columns of the matrix is reinterpreted as a set of points, then they trace out an ellipsoid. Alternatively, $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}$ may be viewed as the *empirical covariance matrix* of the columns of $\boldsymbol{\mathcal{X}}$, in effect computing temporal correlations in traffic.

PCA is used to find the directions of greatest variance of $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}$ by decomposing $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}} = \mathbf{W}\mathbf{D}\mathbf{W}^{\mathrm{T}}$, where $\mathbf{W}$ is an orthonormal matrix containing the eigenvectors of $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}$ and $\mathbf{D}$ the diagonal matrix containing the eigenvalues of $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}$. The eigenvectors are known collectively as *principal axes*. Thus, every column of $\boldsymbol{\mathcal{X}}$ can be expressed as $\mathbf{x}_k = \mathbf{a}_k^{\mathrm{T}}\mathbf{W}$, *i.e.,* a linear combination of a coefficient vector $\mathbf{a}_k$, called the *principal components*. Here, $\mathbf{W}$ is equivalent to a linear transform, post-multiplied to the data. Intuitively, if the size of the set of principal axes with large principal components are small, then this is evidence there are high temporal correlations between the traffic flows.

As an aside, PCA may be performed on $\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^{\mathrm{T}}$, in effect computing the spatial correlations of $\boldsymbol{\mathcal{X}}$ instead. Here, we have $\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{X}}^{\mathrm{T}} = \mathbf{V}\tilde{\mathbf{D}}\mathbf{V}^{\mathrm{T}}$, with each column $\mathbf{x}_k = \mathbf{V}\tilde{\mathbf{a}}_k$, equivalent to $\mathbf{V}$ pre-multiplied with the data. Spatial PCA was used in the context of anomaly detection [47–49] but there are problems with this approach. These discussions are deferred to §5.

[6] A cyclostationary process can be thought of as one who component processes formed from times embedded at multiples of the fundamental period form stationary sequences.

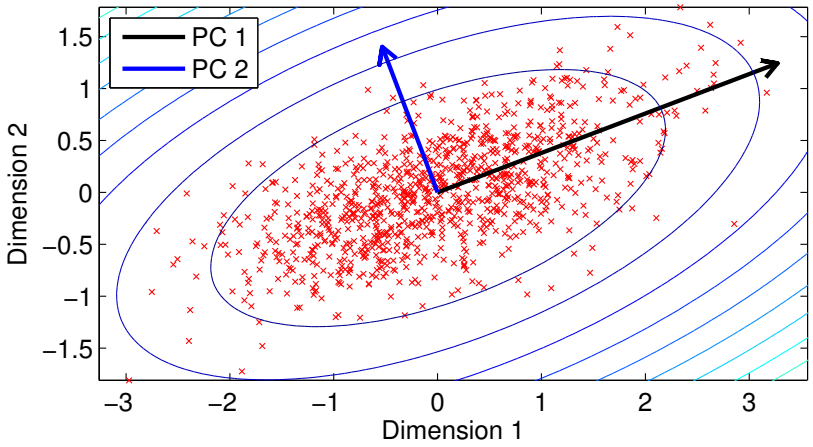


Figure 7: *Principal components analysis of the empirical covariance matrix of a two dimensional data matrix of 1000 centred points $\boldsymbol{\mathcal{X}}$. Here, "PC 1" and "PC 2" are the principal components. Note the elliptical shape of the contours of the density with semi-major and semi-minor axis given by the first and second principle components, respectively.*

Figure 7 demonstrates an example of PCA performed on the covariance matrix of 1000 two dimensional data points with zero mean, *i.e.,* $\boldsymbol{\mathcal{X}}$ has 2 rows and 1000 columns. The matrix $\boldsymbol{\mathcal{X}}^{\mathrm{T}}\boldsymbol{\mathcal{X}}$ formed by the data points vaguely resembles an ellipse. Here, there are two principal components, denoted by "PC 1" and "PC 2", with the higher variance captured by PC 1. This is clear from the way the points on the figure are distributed. The key point to takeaway is that both components captures the direction of highest variance and are orthogonal to each other. Moreover, the principal components matrix $\mathbf{W}$ has PC 1 and PC 2 as its first and second columns respectively. Each data point can be expressed as linear combination of these two components. The concept is easily extended beyond two dimensions to the larger dimensions typically encountered with traffic matrices.

PCA was performed by Lakhina *et al.* on empirical data from two backbone networks show that OD flows are a combination of no more than 35 "eigenflows" (the principal axes), and in fact, fewer than this in general [50]. These eigenflows belonged to one of three categories, depending on their properties:

(i) **deterministic or d-eigenflow**: generally the significant diurnal component of the largest OD flows. Although present in smaller OD flows, these eigenflows are less significant. These eigenflows have a cyclo-stationary property and suggests that these eigenflows may be approximated by a small number of Fourier coefficients. These eigenflows account for the majority of the total traffic of the OD flow.

(ii) **spike or s-eigenflow**: medium sized eigenflows with a spikiness behaviour in time, with values ranging up to 5 standard deviations from the mean of the OD flow. This suggests these contributions come from bursty processes and may be modelled by a wideband Fourier process.

(iii) **noise or n-eigenflow**: small eigenflows behaving like stationary additive white Gaussian noise. These eigenflows have small energy and their contribution to overall traffic is negligible. The majority of eigenflows from Lakhina *et al.*'s datasets belong to this category.

There are several eigenflows belonging to two or more categories, but these eigenflows are rare [50]. For the most part, these categories are very distinct for almost all eigenflows. The low number of eigenflows compared to the dimension of the traffic matrices under study suggests low intrinsic dimensionality of traffic matrices. In many senses PCA confirms the previous analysis and modelling, but it is interesting

because its approach simply looks for correlations across different sets of measurements, and uses a different set of assumptions from, for instance, Fourier analysis which can be performed on a single time series.

Finally, it is important to note that the full data needs to be available (no missing entries in $\boldsymbol{\mathcal{X}}$) in order to perform PCA. Furthermore, PCA is not a robust method, since it is an entirely data driven method and is therefore sensitive to outliers. Robust variants have been proposed but they necessarily complicate the basic version of PCA presented here, since these modifications entail constructing methods to identify and exclude outliers. Despite these disadvantages, in its purest form, PCA is a useful tool to learn the temporal structure of traffic flows.

4.2 Spatial Modelling

Spatial models only focus on the properties of traffic between source and destination pairs, typically within a single measurement interval, without regard to how the traffic changes in time. The major models presented here are the gravity model and its generalisation, the discrete choices model, the independent connections model and low rank spatial models, but we shall start with the simplest test models. For ease of exposition in this section, the set of sources and destinations are assumed to be sets of PoP ingress and egress nodes, denoted by $\mathcal{I}$ and $\mathcal{E}$ respectively. The set Ω represents the set of all nodes in the network, *i.e.,* $\Omega = \mathcal{I} \cup \mathcal{E}$.

4.2.1 Simple *test* models

We must remember that the purpose of models is not always to "realistically" represent a network's traffic. Their purpose is to provide inputs to other tasks. One common task is to assess the sensitivity of a network to different types of traffic, and to that end, engineers can consider the affect of various artificial *test* models.

Three such are the uniform traffic model, peak load model, and focussed overload model. They are extremely simple:

uniform this simple model simply assigns the same value to all traffic matrix elements. It is used to provide a base load in some experiments, or to see the behaviour of a network under one extreme (the most uniform extreme) of traffic.

peak load this model is equally simple, and equally extreme. It has zero for all loads except one OD flow. It simulates the opposite extreme where the aim is to see the affect of one dominant flow.

focussed overload this type of TM simulates the affect of a *focussed overload*, of *flash crowd*[7], where many users become interested in one location or resource and the traffic to this single location from all other sources is the dominant affect in the network. As a result, the focussed over can be represented by a matrix with all elements zero, except for one row. We can likewise represent a focussed traffic load arise from a single point (say as response traffic to a focussed set of queries) by a matrix with a single non-zero column.

The advantage of each of the models lies in its simplicity. The simplicity means that the affect of the traffic is easy to interpret, and thus gain insights from these models where a more complex model would perhaps confound us with multiple potential causes for some results. For instance, in each of the above models we can gradually increase the traffic to see when capacity bounds are reached, and where those bounds would be reached in order to identify potential bottlenecks in a network.

4.2.2 Gravity model

The gravity model is perhaps the next simplest type of model, but it has a great deal to offer. In its simplest form it assumes, for any given packet, the source and destination nodes are *independent.*

[7] See also the slashdot effect.

Depending on context, this could be the origin and destination, or ingress and egress nodes respectively. Consequently, the traffic between two nodes is *proportional* to the total traffic from the source node to the destination node. The gravity model is amongst one of the most well-studied models and is considered a canonical first generation model.

The name of the model derives from Newton's model of gravitation, where the gravitational force is proportional to the product of the mass of two objects divided by the distance between them squared. The general formulation of the gravity model is defined by two forces: the *repulsive* force (factor) R_i, associated with "leaving" from i and the *attractive* force (factor) A_j, associated with "going" into j. Its general form is described by the following equation:

$$X_{i,j} = \frac{R_i \cdot A_j}{f_{i,j}}, \tag{5}$$

where $f_{i,j}$ represents the *friction factor*, which describes the weakening of the forces (akin to distance in Newton's model), depending on the physical structure of the modelled phenomenon. The model has been used extensively in various fields, for instance the modelling of street traffic [64].

In the context of Internet traffic matrix modelling, the friction factors have typically been taken to be constant. That is, distance is assumed to have little effect on network traffic. That certainly seemed to be true even at a fairly large scale in the past, but it is unknown to what extent the deployment of CDNs (Content Distribution Networks) over the last few years has changed distance dependence, or how inter-country matrices are affected by distance (for instance through language barriers). Where distance is ignored, equation (5) becomes

$$X_{i,j} = \frac{X_i^{\text{in}} \cdot X_j^{\text{out}}}{X^{\text{total}}}, \tag{6}$$

where X_i^{in} is the total traffic entering the network through i, X_j^{out} is the total traffic exiting the network through j and X^{total} is the total traffic across the network [97]. The model can be expressed succinctly as the single rank matrix

$$\mathbf{X} = \frac{\mathbf{x}^{\text{in}} \cdot \mathbf{x}^{\text{out}\,T}}{X^{\text{total}}}. \tag{7}$$

The popularity of the model stems from the ease of estimating the X_i^{in} and X_j^{out} for each node pair (i, j), and especially at the PoP or backbone level, since the level of traffic aggregation mitigates errors in the estimation of these quantities from sampled traffic.

The gravity model only captures the spatial structure of the traffic. The key assumption of the gravity model is the independence between each source i and destination j. Coupled with the assumption that none of the nodes act as a source or sink of traffic (*i.e.,* that traffic is conserved in the network) $X^{\text{total}} = \sum_{k\in\mathcal{I}} X_k^{\text{in}} = \sum_{\ell\in\mathcal{E}} X_\ell^{\text{out}}$. Under normal operating conditions in most backbone routers, where congestion is kept to a minimum, the conservation assumption appears reasonable. With this assumption,

$$X_{i,j} = X^{\text{total}} p_i^{\text{in}} p_j^{\text{out}}, \tag{8}$$

where

$$p_i^{\text{in}} = \frac{X_i^{\text{in}}}{\sum_{k\in\mathcal{I}} X_k^{\text{in}}},\ p_j^{\text{out}} = \frac{X_j^{\text{out}}}{\sum_{\ell\in\mathcal{E}} X_\ell^{\text{out}}},$$

are the proportions of traffic entering the ingress and exiting the egress nodes respectively, called *fanouts*. The formulation (8) is known as the *fanout* formulation because it describes how a packet entering via node i is distributed to several nodes $j \in \mathcal{E}$. Fanout has been demonstrated to be close to a constant over several measurement intervals, compared to the traffic matrix [56], suggesting the fanout may be a better alternative to measure and use in, for instance, anomaly detection, than the raw traffic volumes.

Observe the implication of independence between the source and destination in (8): $\Pr(\mathcal{I}, \mathcal{E}) = p_{\mathcal{I}}^{\text{in}} p_{\mathcal{E}}^{\text{out}}$. An immediate consequence is $\Pr(\mathcal{E} \,|\, \mathcal{I}) = P_d(\mathcal{E})$, where $P_d(\mathcal{E})$ is the marginal distribution

of the traffic demand distribution at the destinations. The assumption of independence between the source and destination leads to two important properties of the gravity model making it well suited to traffic matrix modelling.

Theorem 1 (Independence)**.** *Independence between the source and destination holds for any randomly chosen submatrix of the model.*

Proof. The independence property implies $\Pr(s,d) = p_s^{\text{in}} p_d^{\text{out}}$, holding for every $s \in \mathcal{I}$ and $d \in \mathcal{E}$. This condition would also hold for a subsample of locations in $\mathcal{I}$ and $\mathcal{E}$. □

Theorem 2 (Aggregation)**.** *An aggregate of the gravity model is itself also a gravity model.*

Proof. Let all nodes be partitioned into N subsets $\{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_N\}$, with $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^{N} \mathcal{S}_i = \Omega$. The aggregated traffic matrix is defined as

$$X_{\mathcal{S}_i,\mathcal{S}_j} = \sum_{i \in \mathcal{S}_i} \sum_{j \in \mathcal{S}_j} X_{i,j}. \tag{9}$$

The independence condition implies

$$X_{i,j} = \frac{X_{i,\Omega} \cdot X_{\Omega,j}}{X^{\text{total}}}. \tag{10}$$

Substituting (10) into (9),

$$\begin{aligned} X_{\mathcal{S}_i,\mathcal{S}_j} &= \sum_{i \in \mathcal{S}_i} \sum_{j \in \mathcal{S}_j} \frac{X_{i,\Omega} \cdot X_{\Omega,j}}{X^{\text{total}}} \\ &= \frac{1}{X^{\text{total}}} \sum_{i \in \mathcal{S}_i} X_{i,\Omega} \sum_{j \in \mathcal{S}_j} X_{\Omega,j} \\ &= \frac{X_{\mathcal{S}_i,\Omega} \cdot X_{\Omega,\mathcal{S}_j}}{X^{\text{total}}}. \end{aligned}$$

which is also a gravity model. □

These are not just theoretical results. Any model should be consistent in the sense that if the data to which it applies is viewed in a different way (for instance by sampling or aggregation) then the model should still apply (though its parameter values may change). It seems like an obvious requirement, and yet there are many model to which it does not apply.

The utility of the gravity model is not just restricted to network measurement. It is used in various areas: teletraffic modelling [45,51], economy and trade [65,90], epidemiology [36,57,95], sociology [85], the retail industry, specifically Reilly's law of retail gravitation [24, 43, 67], and in vehicular traffic modelling [32]. More advanced discussion on the gravity model (albeit with an economics flavour) is found in [78].

The gravity model can be interpreted in terms of the *principle of maximum entropy*. Entropy here is the Shannon entropy from information theory parlance [25]. The principle is closely related to Occam's Razor, essentially choosing the most parsimonious explanation of the data amongst competing explanations. With little information regarding the traffic matrix besides the total traffic information, it turns out that the best one can do, according to the principle, is to describe the observations with a model promoting independence and symmetry, consistent with known constraints. In this way, the model enjoys robustness compared to other models, as the gravity model seeks to minimise deviation from what has already been observed.

The model, however, is not without its drawbacks. The main critique against the gravity model is in its main assumption: the independence of the ingress and egress nodes[8]. It has been pointed out in several papers [33] that this assumption does not hold true. Most traffic between node pairs are

[8]The difference between OD and IE traffic matrices becomes critical here.

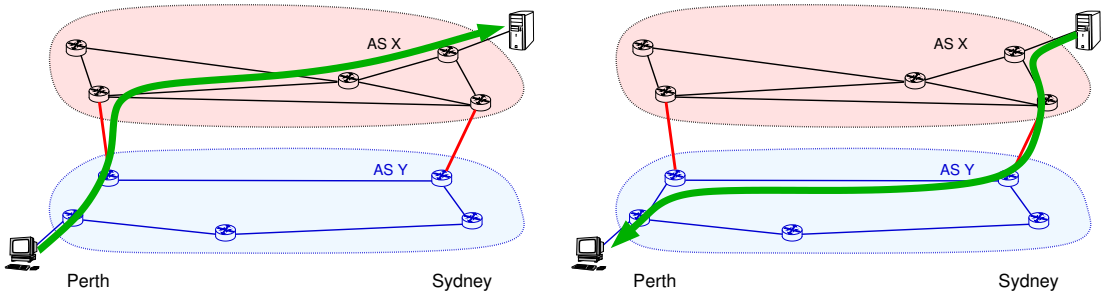


Figure 8: *Traffic flow between two ASes, one in Perth and the other in Sydney. Note the asymmetry in traffic: due to the action of hot potato routing, the path taken by a traffic flow from Perth to Sydney differs from the reverse path, since the closest router is always chosen.*

determined by connections, for example TCP initiated sessions, so there exist dependencies between node pairs. The second is the violation of the conservation of traffic assumption, for *e.g.,* when there is high congestion, causing packets to be dropped from router queues.

Actual traffic matrices are generally asymmetric, violating gravity models. For example, forward traffic volumes of a source-destination pair of nodes do not typically match up with the volume of reverse traffic. Even if the OD traffic matrix matches the gravity model well, the corresponding IE traffic matrix may be distorted by hot potato routing [63], where the egress point closest to the ingress point is chosen to route the traffic, in effect causing asymmetric traffic and distance dependency between ingress and egress points [6]. In short, hot potato routing aims to dump traffic off the network as quick as possible, so to speak. An example of hot potato routing is in Figure 8. Here, there is a clear asymmetry since the paths taken by traffic flows from Perth to Sydney differ from Sydney to Perth. The shortest path computed depends on the closest router to the source and destination.

Thus, although the source-destination independence assumption may hold for OD traffic matrices, it may not necessarily hold for IE traffic matrices, due to distortion by inter-domain routing. Consider a simple toy example of a network in Figure 9 (originally from [6]). The ASes A, B and C are assumed to be connected, with A having three routers: 1, 2 and 3. The inter-domain routing protocol between these ASes uses hot potato routing, seeking the shortest path between these ASes.

Suppose $X^{\text{total}} = 9$. Consider an OD traffic matrix with the form of a gravity model, with even spread of traffic over each internal router 1, 2 and 3, with $\mathbf{x}^{\text{in}} = \mathbf{x}^{\text{out}} = \mathbf{x}$. The OD traffic matrix has form $\mathbf{X}_{\text{OD}} = \mathbf{x}\mathbf{x}^{\text{T}}/X^{\text{total}}$, with $\mathbf{x} = (1, 1, 1, 3, 3)^{\text{T}}$, and written explicitly as

$$\mathbf{X}_{\text{OD}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & B & C \end{array} \\ \begin{pmatrix} 1/9 & 1/9 & 1/9 & 1/3 & 1/3 \\ 1/9 & 1/9 & 1/9 & 1/3 & 1/3 \\ 1/9 & 1/9 & 1/9 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 & 1 & 1 \\ 1/3 & 1/3 & 1/3 & 1 & 1 \end{pmatrix} \end{array} . \tag{11}$$

By Theorem 2, the gravity model for the aggregated OD matrix, comprising OD traffic volumes between ASes A, B and C, is given by

$$\mathbf{X}'_{\text{OD}} = \begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array} , \tag{12}$$

simply by summing the traffic in the internal nodes. In this case, $\mathbf{X}'_{\text{OD}} = \mathbf{x}\mathbf{x}^{\text{T}}/X^{\text{total}}$, with $\mathbf{x} = (3, 3, 3)^{\text{T}}$, still a gravity model.

In order to construct the IE traffic matrix, the ingress and egress points of the network in A needs to be determined. The following assumptions are made:

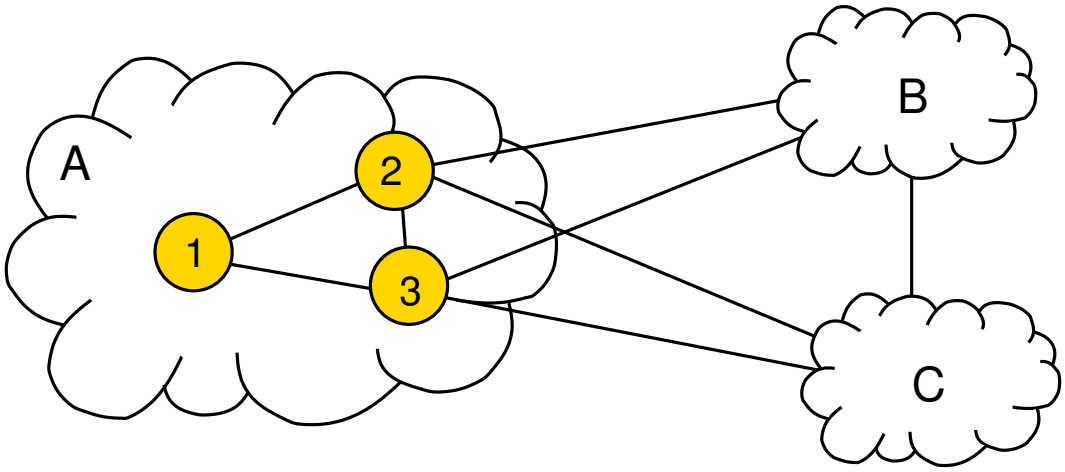
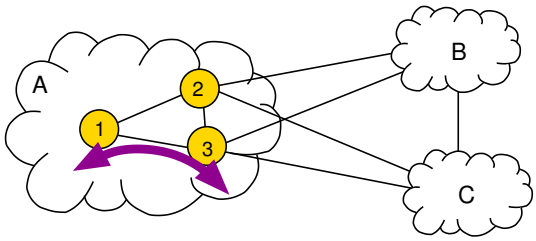
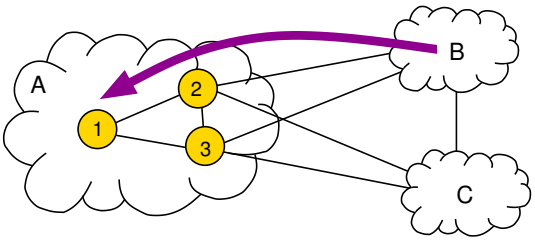


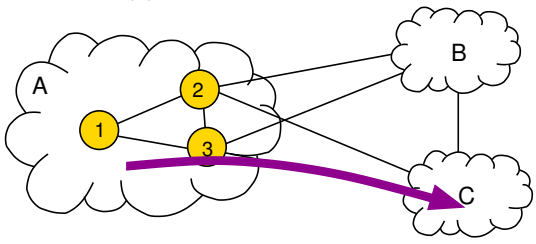
Figure 9: *Example toy network with three ASes: A, B and C are all assumed to be peers. The routers 1, 2 and 3 are internal to A.*



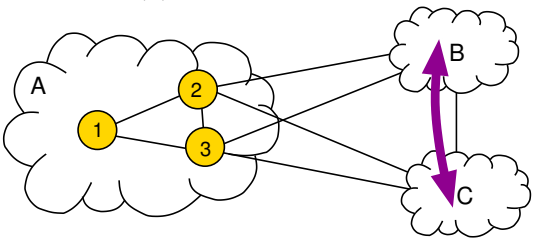
(a) *Internal traffic within A.*



(b) *Incoming traffic to A.*



(c) *Outgoing traffic from A.*



(d) *Traffic external of A, between B and C.*

Figure 10: *Traffic flows within the network of Figure 9, classified into four components.*

(i) A, B and C are peers,

(ii) the shortest AS path protocol is used for inter-domain routing,

(iii) hot potato routing is used internally by A, and

(iv) the Interior Gateway Protocol (IGP) weights are all equal.

Suppose ingress and egress points are defined by the following routing table ($*$ represents a wildcard character)

Origin router	**Destination**	**Egress router**
1	B	2
1	C	3
2	*	2
3	*	3

The path for each traffic flow in the network, therefore, differs depending on its source and destination.

All traffic flows between the PoPs may be decomposed into four components: internal traffic within A, traffic departing A, traffic coming into A and traffic external to A, shown in Figure 10. The internal traffic of A (Figure 10a) is just the top-left 3×3 submatrix of $\mathbf{X}_{OD}$, which is

$$\mathbf{X}_{\text{internal}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \end{array} \tag{13}$$

Traffic bound for A, as seen in Figure 10b to be specifically for router 1 in this instance, from its peers has entry points controlled by B and C, given the above routing table. Hence, from A's point of view, the traffic behaves as if the traffic randomly distributed across ingress links. Assuming the traffic is evenly spread, the traffic matrix is

$$\mathbf{X}_{\text{arriving}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} \end{array} \tag{14}$$

Traffic departing from A, seen in Figure 10c as originating from router 1, and routed by hot potato routing, is described by

$$\mathbf{X}_{\text{departing}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1/3 & 1/3 \\ 0 & 2/3 & 0 \\ 0 & 0 & 2/3 \end{pmatrix} \end{array} \tag{15}$$

Since A does not provide transit for B and C, traffic external to A, *i.e.,* between B and C, should not appear on A, the traffic will remain unseen by A (Figure 10d). Thus, the total IE traffic matrix is the sum of the component traffic above, so that the entry and exit points match, and is given by

$$\mathbf{X}_{IE} = \begin{pmatrix} 1/9 & 4/9 & 4/9 \\ 4/9 & 10/9 & 4/9 \\ 4/9 & 4/9 & 10/9 \end{pmatrix} . \tag{16}$$

The matrix $\mathbf{X}_{IE}$ is not $\mathbf{X}'_{OD}$ in (12), simply due to traffic asymmetry resulting from hot potato routing. Moreover, the assumption of the conservation of traffic no longer holds, since the total traffic of $\mathbf{X}'_{IE}$ is not equal to X^{total}. The diagonal terms, for example, are much larger than in $\mathbf{X}'_{OD}$. This example demonstrates that even if the OD traffic matrix is generated from the gravity model, the IE traffic matrix not necessarily has a structure that conforms to the gravity model.

Assumptions on the gravity model are more likely to hold for large backbone networks where large aggregates of traffic are being observed. On smaller, local area networks, its effectiveness is limited. Finally, the friction factor $f_{i,j}$ may not necessarily be constant in actual traffic matrices, possibly due to different time zones [72], especially for a global spanning network, language barriers, or the increased deployment of CDNs.

The gravity model by itself incurs significant estimation error as the estimates obtained typically do not match the observed link counts. Due to violations of these assumptions, the gravity model turns out to be inaccurate when used in traffic matrix estimation. For example, it was reported to have $\pm 39\%$ accuracy when used in estimating traffic matrices [70].

Despite the flaws mentioned, the gravity model was reported to be a good initial estimate to more sophisticated methods. The model was paired with SNMP link measurements to develop the so-called *tomogravity* technique [97]. The gravity model is also surprisingly useful in the synthesis of traffic matrices. When proposed as a method for synthesising traffic matrices by Roughan [70], the gravity model serves as an excellent first order model for generating the cumulative distribution function of the traffic demands, closely mimicking the statistical properties of actual traffic matrices. While the basic gravity model may not necessarily be an optimal model, it is a simple and good first order model for estimation and synthesis purposes, and it can be improved to take into account the factors described above.

4.2.3 Generalised gravity model

In order to improve the efficacy of the basic gravity model and to address its deficiencies, a generalisation of the gravity model was developed [98, 99]. In a nutshell, the assumption of independent ingress and egress nodes was relaxed by dividing traffic into several classes of ingress and egress nodes, evident from the example in the previous section. Independence only applies to traffic belonging within a certain class, effectively enforcing a *conditional independence* criterion. Such an assumption is closer to actual conditions between ingress-egress pairs in a network.

In particular, the model now accounts for asymmetry of the IE traffic matrix. To account for the effect from hot potato routing, traffic is separated into classes based on peering and access links. Consider again the network in Figure 9. From the figure, two classes can be defined: internal and external classes. There are then four types of source-destination links (see Figure 10): *internal to internal, internal to external, external to internal* and *external to external*.

In the generalised gravity model, independence between nodes are only assumed between the internal to internal class and the external to external class. Thus, routers 1,2 and 3 in ASes A are independent to each other, and so are ASes A, B and C to one another, but not traffic from 2 to B, for instance.

Thus, in the generalised gravity model, a modification is made by ensuring the independence assumption still holds, but only when conditioned within each traffic class. In terms of probabilities, traffic is *conditionally independent*, as formulated below for the joint fanout distribution of the sets of access nodes of the network of interest $\mathcal{A}$ and and peering nodes $\mathcal{P}$ respectively:

$$p_{S,D}(s,d) = \begin{cases} \frac{p_S(s)}{p_S(\mathcal{A})}\frac{p_D(d)}{p_D(\mathcal{A})}(1 - p_S(\mathcal{P}) - p_D(\mathcal{P})), & \text{for } s \in \mathcal{A}, d \in \mathcal{A}, \\ p_S(s)\frac{p_D(d)}{p_D(\mathcal{A})}, & \text{for } s \in \mathcal{P}, d \in \mathcal{A}, \\ \frac{p_S(s)}{p_S(\mathcal{A})}p_D(d), & \text{for } s \in \mathcal{A}, d \in \mathcal{P}, \\ 0, & \text{for } s \in \mathcal{P}, d \in \mathcal{P}. \end{cases} \tag{17}$$

The four probabilities corresponds to the four cases in Figure 10. In particular, as per intuition, peering traffic is set to zero, since this class does not transit the network of interest.

The stratification of traffic into several classes results in an improved model. Its performance in the traffic matrix estimation results is significantly better than the basic gravity model [98]. Further stratification beyond separating peering and access nodes is possible. For example, the origin of the traffic, whether from a fixed location or mobile device, or the destination of the traffic, depending on

application profiles, may be defined as new classes in the model. However, further classification in this manner is only possible with more side information available.

The generalised gravity model is superior to the basic model, but gravity models in general have been somewhat tarnished by the same brush. Most works benchmarking the performance of various models, for instance for estimation, compare against only the simple gravity model, but make confusing statements that could lead one to believe that all such models are faulty. In fact, the generalised gravity model is vastly superior, but rarely used outside of the company at which it was first developed — AT&T. The chief reason is that the model requires additional topological and routing data, and for the external traffic flows to be mapped using this data. This is a non-trivial task. In addition, in many external studies researchers have not have access to, in particular, knowledge on access and peering links in the network under study network. Network operators are not open to releasing information on their networks to the public, however, the Abilene dataset used in [99] is publicly available, and contains enough information to make such comparisons.

4.2.4 Discrete choice models

Another proposed model is the *choice model*, introduced by Medina *et al.* [56]. The basis of the Discrete Choice Model (DCM) is the theory of choice models for decision behaviour, originally developed in psychology, and later expanded upon by researchers in other fields, more recently in economics, by Daniel McFadden, for which he won the Nobel Prize in Economics in 2000 (see for example, [54]).

Choice models are popular in econometric applications as the model is used to describe a simplified underlying mechanism of rational decision behaviour. It has been used for transportation analysis, econometrics, marketing and consumer theory. The main inspiration for its use in Internet modelling comes from [86], where a choice model is used in the context of modelling the behaviour of travellers between the cities of Maceio and Sao Paulo, two cities in Brazil, as it parallels traffic traversing PoPs.

The choice model is defined by four elements:

(i) the decision makers,

(ii) the set of alternatives (choices),

(iii) the attributes of the decision maker and the set of alternatives, and

(iv) the decision rules.

All these elements play a key role in ultimately determining the decision process. The *decision makers* represent the agents making the decisions on which choices to go for. The *set of alternatives* characterise the set of possible actions the agents can choose. Each decision maker executes several choices based on its own inherent properties, or *attributes*, as well as the attributes of the set of alternatives. These attributes predispose a decision maker to certain alternatives. Finally, the *decision rules* determine how choices are made. How good a choice is, is measured by a standard based on a set of criteria. The rules establish constraints on the choices of the decision makers, enforcing consistency in the entire system. All four elements of the model aim to capture how agents would naturally decide on several differing choices in a system, in a rational and consistent way, based on a set of rules.

In the context of network traffic modelling, there are two interdependent factors influencing choices. First, the network users' behaviour determine much of how traffic flows are generated, as discussed §4.1. Second, the network design and configuration plays a very important role in how traffic flows are delivered on the network. Routing protocols, policies, QoS as determined by the network operator and the geographical local of routers and PoPs determine how traffic is transported within the network and between networks. One could visualise this as a two level process: users generate the traffic flows, determining the source and destination of each flow, whereupon it is routed through the network based on its design and the policies imposed on it.

All four elements have direct analogies in the context of network traffic modelling. The decision makers are the set of ingress nodes, which aggregates all information about user behaviour and network

design and policies. The set of alternatives are the set of egress nodes, which aggregates the information about the users connected to these nodes. Thus, each decision maker i has a choice set $\mathcal{C} \subseteq \mathcal{E}$. Each node i, a decision maker, is modelled by the equation, for all $j \in \mathcal{C}$,

$$U_j^i = V_j^i + \varepsilon_j^i, \tag{18}$$

where U_j^i denotes the utility between the node pair i and j, V_j^i aggregates the information from the user behaviour and network design, which is deterministic, and ε_j^i is a random component to account for missing information from unknown factors. The term V_j^i can be thought of accounting for the level of attractivity of a destination node j. In [56], the authors proposed M-attributes per decision maker-choice pair, such that

$$V_j^i = \sum_{m=1}^{M} \mu_m \omega_j^i(m) + \gamma_j, \tag{19}$$

where $\omega_j^i(m)$ denotes the m-th attribute, μ_m are weights to account for the relative importance of the m-th attribute, and γ_j is a scaling term for other factors for attractivity, besides all M attributes. An attribute $\omega_j^i(m)$ could be the size of the destination node PoP, since a large egress PoP is more likely to have traffic exiting from it, or the number of peering links the destination node j has.

Based on the above, the work [56] proposed a traffic matrix model that assumes the decomposition

$$X_{i,j} = O_i \alpha_{i,j}. \tag{20}$$

The parameters O_i and $\alpha_{i,j}, \forall j$ denote the total outgoing traffic volume from node i and the *fanout* of node i respectively. For each i, $\sum_j \alpha_{i,j} = 1$. The total traffic from a node O_i is known from SNMP data. Observe that the traffic matrix is now parameterised by the fanout distribution which has a direct analogy in the gravity model. In inference applications, it is the fanout distribution being estimated, thus indirectly inferring the traffic matrix, rather than directly estimating the traffic demands. Fanouts have been shown to be generally stable over a measurement period (several hours), compared to traffic demands [42], which is advantageous in traffic matrix estimation, since the stability contributes to more accurate inference.

The fanout distribution is determined by a decision rule. In [56], a utility maximisation criterion was used,

$$\alpha_{i,j} = \Pr(U_j^i = \max_{k \in \mathcal{C}} \{U_k^i\}). \tag{21}$$

Now, $\alpha_{i,j}$ is a random quantity as it depends on $\epsilon_{i,j}$, as observed from equation (18). A natural starting point is to assume ε_j^i is i.i.d. Gaussian distributed with mean 0 and variance 1. This transforms (21) to the well-known multiple normal probability unit or m-probit model [53]. However, there is no closed form for (21) under this assumption. Instead, by assuming ε_j^i is i.i.d. distributed following the Gumbel distribution, the m-probit model can be approximated, with (21) now having a closed form. This model is popularly called the multiple logistic probability unit or m-logit model [53]. The closed form is simply

$$\alpha_{i,j} = \frac{\exp(V_j^i)}{\sum_{k \in \mathcal{C}} \exp(V_k^i)}, \tag{22}$$

implying that

$$X_{i,j} = O_i \frac{\exp(V_j^i)}{\sum_{k \in \mathcal{C}} \exp(V_k^i)}. \tag{23}$$

The difficulty lies in determining what attributes should be included. The authors considered two models which they empirically validated:

(i) $V_j^i = \mu_1 \omega_j(1) + \gamma_j$, where $\omega_j(1)$ denotes the total incoming bytes to an egress PoP j, and

(ii) $V_j^i = \mu_1 \omega_j(1) + \mu_2 \omega^i(2) + \gamma_j$, where in addition, $\omega^i(2)$ denotes the total bytes leaving the ingress PoP i.

In general, the second model is more accurate, owing to the additional attribute, it is not know if it is just a case of overfitting or the new parameter is truly useful.

The choice model is a variation of the gravity model. In particular, looking back at equation (20), the total traffic outflowing from ingress i may be regarded as the *repulsion factor*, while the parameters $\alpha_{i,j}$ combining both the *attractiveness factor* and the *friction factor*. A quick comparison of the choice model to (8) will show the strong link between both models. The choice model, however, has a larger number of parameters to account for the attributes of the decision maker and set of alternatives.

4.2.5 Independent connections model

The independent connections model (ICM) was introduced in [33, 34]. Unlike the gravity model, this model discards the assumption of independence between the ingress and egress nodes, and instead focuses on the *connections* between nodes. More specifically, the model differentiates between *initiators*, nodes that initiate a traffic connection, such as a TCP connection, and *responders*, the nodes that accept these connections. The independence assumption comes in by assuming that each initiator and responder are independent, in effect, resulting in independent *connections*.

The inspiration for the ICM comes from traffic characterisation studies, specifically on TCP behaviour. TCP creates two-way connections in response to a SYN packet. Although it is common for the majority of traffic to flow in one direction, there is also a smaller reverse flow. Common examples include an HTTP query, which involves query packets flowing in one direction, and a much larger set of data flowing in the other as a response, or an FTP transaction which may involve mainly data flow in one direction, but the forward packets require acknowledgement packets in the reverse direction. Therefore, the model uses the notion of a connection: a two-way exchange of packets between an *initiator* and a *responder*, corresponding to the ingress and egress nodes.

Three parameters were defined as a product of these studies. The first parameter, the forward traffic proportion $f_{i,j}$ is the normalised proportion of forward traffic from a connection between ingress i to egress j, measured in packets or bytes and $0 \leq f_{i,j} \leq 1$, $\forall i \in \mathcal{I}$ and $j \in \mathcal{E}$. The second parameter A_i describes the activity level of the users at i (the A stands for 'activity'). Finally, some nodes may be chosen for connection more than others, and thus, P_j (stands for 'preference') denotes the preference for node j.

The main assumption of the model is that the probability that a connection responder belongs to node j depends on j only. The values of P_j for $j \in \mathcal{E}$ are unnormalised. They are divided by the sum $\sum_{k\in\Omega} P_k$ in order to treat them as the probability a node j is a connection responder. The parameters A_i and P_j were shown to be uncorrelated on empirical data, providing some evidence these parameters describe two very different underlying quantities.

The model is expressed by

$$X_{i,j} = \frac{f_{i,j} \cdot A_i \cdot P_j}{\sum_{k\in\Omega} P_k} + \frac{(1 - f_{j,i}) \cdot A_j \cdot P_i}{\sum_{k\in\Omega} P_k}. \tag{24}$$

The first term captures the forward traffic of the connection between initiator i and responder j while the second term its reverse traffic, generated by the users from i and j respectively. The model may be viewed as a weighted sum of two gravity models, with one gravity model characterising the forward traffic, while the other the reverse traffic. Thus potential asymmetries in traffic can be accounted for.

The model is sufficiently flexible to accommodate variations. For example, the *simple IC model* modifies one parameter of model (24) by setting $f_{i,j} = f$, where f is a constant as it has been observed that f is fairly stable from week to week (at least on the Abilene dataset [34]) simplifying the model considerably. Another variation, the *time-varying IC model* includes temporal variation of the parameters, *i.e.,*

$$X_{i,j}(t) = \frac{f(t) \cdot A_i(t) \cdot P_j(t)}{\sum_{k\in\Omega} P_k(t)} + \frac{(1 - f(t)) \cdot A_j(t) \cdot P_i(t)}{\sum_{k\in\Omega} P_k(t)},$$

and the *stable-fP IC model* removes the time dependency of f and the preferences $\{P_j\}_{j\in\mathcal{I}}$, while the *stable-f IC model* only removes the temporal dependence of f. These variations allows trade-offs between the degrees of freedom of the model and computational complexity, especially when used for the synthesis or inference of traffic matrices. With less parameters, which was shown to be less than the basic gravity model, the model is easier to compute.

The parameters $\{A_i\}_{i\in\mathcal{I}}$ and $\{P_j\}_{j\in\mathcal{E}}$ were validated on actual data. Activity levels $\{A_i\}_{i\in\Omega}$ possess diurnal patterns, corresponding to user access patterns, and a periodic pattern on a weekly timescale. In particular, activity levels are higher on weekdays compared to the weekend, matching observations such as Figure 5. There is also a more prominent periodic pattern when considering larger nodes, as this effect is due to aggregation, as it captures the users with higher activity levels. These observations are consistent with the temporal properties discussed of traffic matrices discussed in §2. The model was shown to be effective in estimating traffic matrices, improving over the basic gravity model by $20\% - 25\%$ for the GÉANT dataset and almost 10% for the Totem dataset [33,34]. These results and observations show that average user behaviour is largely stable and predictable, a great boon to traffic modelling development.

In some ways, the ICM is similar to the DCM, in that both models include parameters to describe the underlying user behaviour, unlike the basic gravity model. For example, both models have a parameter to quantify the level of attractiveness of one node (connection) to another. The DCM is also able to incorporate features of the ICM as well. The differences end there, however. For one, the ICM has a slightly richer description of the flow connections between nodes, such as the forward and reverse traffic flows between nodes, whereas the DCM aggregates the information in a single parameter. The ICM seeks to capture the behaviour of each connection made, rather than merely model the relationship between nodes, emphasising a different focus compared to the DCM. Thus, the ICM may account for hot potato routing and other asymmetries in traffic flow.

4.2.6 Low-rank spatial models

The very noticeable feature of both DCM and ICMs is that they can better represent traffic matrices, but are more highly parameterised. It is, in general, possible to fit a data set more accurately when more parameters are available, but this presents a difficulty – does one accept the more complex, more highly parameterised model, or the simpler, perhaps more robust model?

In the previous cases, this was an "all or none" decision (at least we had to decide on the type of model we used, of not the exact number of choices involved), whereas the gravity model is fixed in its parameterisation. However, there are concepts that easily extend the gravity model.

The low-rank model somewhat new, made popular by its use in matrix completion problems [14,15, 17,66]. Low rank models assume the traffic matrix is well-represented by the low rank approximation

$$X_r = \sum_{i=1}^{r} \sigma_i^2 \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}, \tag{25}$$

where σ_i denotes the i-th singular value, with all singular values arranged in order of descending order, *i.e.,* $\sigma_i \geq \sigma_2 \geq \cdots \geq \sigma_r$. The famous Eckhart-Young theorem [84, Theorem 4.32, p. 70] states this is the best rank-r approximation, in the sense of the Frobenius norm[9], of a matrix $\mathbf{A}$ given by retaining the largest r singular values of its Singular Value Decomposition (SVD). The theorem, however, assumes that the target matrix for approximation is already known. In low-rank matrix recovery, however, the target matrix is unknown.

In the context of traffic matrix modelling, low-rank models are a relatively recent introduction, beginning with work in [100]. However, the choice is strongly suggested by the earlier results of PCA applied to the data, for instance in [50,96]

In essence, we can see (26) as expressing a traffic matrix as a weighted sum of gravity models, *i.e.,* each single rank component looks exactly the same as that expressed in (7). It seems a logical

[9]The Frobenius norm is the Euclidean norm applied to matrices *i.e.,* $\|\mathbf{X}\|_F = \sqrt{\sum_{i,j} x_{i,j}^2}$.

approach simply because the Internet is not a homogenous entity. In particular there are many types of applications running across the network: from interactive session, to voice, to HTTP, to streamed video. We might imagine that a class of traffic, say streaming video, satisfies the gravity law, but with different row and column sums to, say, voice traffic. Given this, it seems that a weighted sum of gravity matrices is a natural extension.

Previous models actually turn out to be special cases of this low-rank model. The gravity and discrete choice model are spatial rank-1 models. The generalised gravity model and the ICM are spatial rank-2 models, the latter of which can be observed from the summation of the forward and reverse traffic contributions in equation (24). The low-rank model may be viewed as a general model for providing a fundamental framework for further model development. We shall consider this idea in more detail below in the context of spatio-temporal modelling.

4.3 Spatio-Temporal modelling

Spatio-temporal models aim to describe spatial and temporal structure jointly. Considering the rich structure traffic matrices have both spatially and temporally, these models would be more sophisticated than their purely temporal or spatial counterparts. There has been relatively little work performed on this type of modelling as yet, but there is considerable suggestion that it will be fruitful in the future.

4.3.1 Low-rank spatio-temporal models

The main idea in spatio-temporal modelling of traffic matrices so far has been to exploit the low-rank models mentioned above, but in this context to apply it to the stacked representation of a series of traffic matrices denoted here by $\boldsymbol{\mathcal{X}}$.

Low-rank models assume the traffic matrix is well-represented by the low-rank approximation

$$\boldsymbol{\mathcal{X}}_r = \sum_{i=1}^{r} \sigma_i^2 \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}}, \tag{26}$$

where σ_i denotes the i-th singular value, with all singular values arranged in order of descending order, *i.e.,* $\sigma_i \geq \sigma_2 \geq \cdots \geq \sigma_r$. As before, the Eckhart-Young theorem [84, Theorem 4.32, p. 70] applies. The theorem, however, assumes that the target matrix for approximation is already known. In low-rank matrix recovery, the target matrix is unknown.

In the context of traffic matrix modelling, low-rank models are a relatively recent introduction, beginning with work in [100]. Besides spatial correlations (exploited by the models proposed previously), traffic matrices are known to exhibit temporal correlations, resulting in a low-rank structure both spatially and temporally, justifying the rationale behind the model. The objective of the work is to approximate the time series of traffic matrices $\boldsymbol{\mathcal{X}}$ by a rank-r model $\boldsymbol{\mathcal{X}}_r$. The model proposed here is *spatio-temporal*, in contrast to the models discussed previously, which are only spatial in nature.

Simply insisting on low rank, however, is missing another important point, which is that matrices also exhibit locality, *i.e.,* elements that are close in time (where this might mean time of day, not absolute time), or space exhibit strong correlation. It turns out that the model (26) is greatly enhanced with additional simple constraints on the temporal and spatial structure to reflect the smoothness property of Internet traffic, under normal operating conditions.

The low-rank construction also proved relatively easy to use in practical applications such as matrix completion, and [100] showed that it could be used to do matrix inference from link data, impute missing data (from as little as a few percent of extant values), or be used to predict matrices into the future.

Despite the demonstration of its effectiveness in traffic matrix estimation, low-rank models are still not well understood. Unlike the previous models, where the parameters are, by design, quantitative measures of an underlying network property, low rankedness (in spatio-temporal matrices) does not correspond to any particular network aspect, such as user behaviour. It is just a measure of the spatio-temporal correlation between traffic flows. It does, however, hint that OD traffic flows are *clustered*,

if one considers the allocation of IP prefixes. A better interpretation is necessary to understand the properties of the model, and work in [9, 50] may provide clues in the right direction.

Furthermore, in the recovery of traffic matrices using the low-rank model, theoretical work on the minimum number of measurements required for recovery under structured losses of rows and columns of $\boldsymbol{\mathcal{X}}$, which occurs frequently in the networking context, is left open. At present, the current focus is on random erasures of the elements of $\boldsymbol{\mathcal{X}}$ [14, 15, 17, 66]. Overcoming structured losses is far more important than random erasures, as such a scenario is frequently encountered in real networks. For example, a router failure may result in missing data for an entire row of the traffic matrix. The results of [100] show much promise, as the method is largely immune to structured losses. The challenge now is to construct a theory as to why this is so and as to what extent structured losses may be recovered.

Low-rank models hold much promise for the development of more sophisticated models. More work is required to understand the spatio-temporal properties of the traffic matrix, but as preliminary results indicate, there is a potentially rich structure to exploit.

4.3.2 Tensors and hyper-matrices

A time series of purely spatial traffic matrices is simply a 3-dimensional array, which is sometimes also called a hyper-matrix. Such a representation would be a more natural representation as it would theoretically preserve spatio-temporal properties better than the stacked matrix, as well as track the evolution of the traffic demands throughout the measurement interval. A tensor representation of traffic is much better as it is invariant to changes of basis, unlike hyper-matrices. The difficulty, however, is identifying the type of decomposition of the tensor that would produce low-rank structures, or a beneficial, exploitable structure. There are many proposed methods for tensor decomposition, but the two most popular are the Canonical Polyadic (CP) or PARAFAC decomposition and the Tucker or multilinear decomposition [44]. Tensor decomposition requires a large number of computations, which may be an obstacle to its adoption in traffic matrix recovery. At present the one work exploiting the tensor structure of network traffic to impute missing entries of network traffic tensor is found in [4].

5 Applications

5.1 Traffic Matrix Recovery

As noted earlier it may be difficult to measure traffic matrices directly, but we need to recover traffic matrices from measurements before they can be used. The technique will obviously depend on the available measurements, but there is a glut of works on the recovery of traffic matrices, whether at the OD level, IE level or AS level. Amongst these traffic matrices, IE traffic matrices are relatively easier to recover, as measurements of these matrices are available through SNMP link counts. Recovery of OD level traffic matrices are fraught with challenges because at any point in time, only a subset of IP traffic is seen by a network. There is no way to know what goes on in the entire IPv4 address range, unless all measurements of the global network were combined, but even so, the data from such an endeavour would be massive and computationally intractable to analyse. Similarly, for AS level traffic matrices, the lack of measurements as well as error prone measurement tools lead to inaccurate recovery of these matrices.

The major challenge in recovering the IE traffic matrix from SNMP measurements is that the problem is highly *underconstrained*. The set of linear equations (1) is under-determined, *i.e.,* there are many solutions to the observations. Moreover, the measurements themselves are subject to error possibly due to poor data collection methods and poor vendor implementation of SNMP polling. They are also not fine-grained since polling of the measurements is performed every five minutes (and the polling intervals may not be exactly synchronised across a whole network). Clearly, any inference method is required to be robust against these errors and uncertainties.

The underconstrainedness of the problem may be mitigated by active measures. One is direct measurement, using dedicated monitors or in-built measurement software on routers such as NetFlow

[1]. Direct measurements at even a single point of ingress results in measurement of an entire row of the traffic matrix, drastically reducing the number of missing matrix entries. Another interesting proposal is to change the IGP (Interior Gateway Protocol) link weights over several snapshots within the measurement interval to provide fresh sets of observations, thereby resulting in a system of linear equations with a unique solution (full rank) out of the SNMP measurements [60]. Both these techniques may be impractical, either being too costly in the case of direct measurements, or requiring direct intervention by the network operator for IGP weight changes. Most proposals simply avoid these by settling on a passive approach of inferring the traffic matrix straight from SNMP data.

There are two main approaches to traffic matrix inference. The first is the deterministic approach, where $\mathbf{y}$ is assumed to provide hard constraints, rather than statistical data. Goldschmidt [39] formulated this as a Linear Program (LP) where the objective was designed to push the to find bounds on traffic matrix elements. In simple terms, the LP finds the traffic matrix with the worst case upper and lower bound on the traffic demand subject to constraints. Recall the vectorised traffic matrix $\mathbf{x}$ has size $N(N-1)$. For the upper bound, the LP model is defined with the objective function

$$\max_{\mathbf{x}} \sum_{j=1}^{N(N-1)} \omega_j x_j, \tag{27}$$

where ω_j is a weight for an OD pair j, also called the coefficient of demand. There are three constraints to satisfy, namely,

(i) **observation constraints**:

$$\sum_{j=1}^{N(N-1)} A_{ij} x_j \leq y_i, \ i = 1, 2, \cdots, L, \tag{28}$$

(ii) **flow conservation constraints**:

$$\sum_{\substack{\ell_1=i,\ell_2=j,\\ \ell_1\neq\ell_2}} y_{\ell_1} A_{\ell_2 k} - \sum_{\substack{\ell_1=j,\ell_2=i,\\ \ell_1\neq\ell_2}} y_{\ell_1} A_{\ell_2 k} = \begin{cases} x_k, & \text{if } j \text{ is the source of } k, \\ -x_k, & \text{if } j \text{ is the destination of } k, \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

(iii) **positivity constraints**: $x_j \geq 0$, $j = 1, 2, \cdots, N(N-1)$.

Similarly, the lower bound is found by substituting the maximisation operation in (27) with a minimisation operation. The LP only produces a nontrivial solution if the lower bound and upper bound on the traffic demand is greater than zero and less than the observed total link count, *i.e.,* $\sum_j y_j$.

Unfortunately, the utility of the LP is only restricted to small toy problems. First, two linear programs have to be solved each time to obtain the upper and lower bounds on traffic demands, which is computationally expensive for large n. Second, the LP was shown to have terrible performance when tested on several types of traffic matrices [56]. Estimates of some traffic matrix entries were in excess of 200%, with most in excess of 100% error, proving that while the LP may useful for certain small topologies, in general it is not considered a practical estimation method. The reason for this is because the LP sets many estimated values to zero, resulting in overcompensation for the rest of the estimated values in order to meet the total traffic constraints. Third, there is a high sensitivity of the solution to weight choices, which implies that different solutions will be obtained depending on the chosen weights.

Instead, a more successful alternative is the use of statistical models and regularisation, *i.e.,* treating the traffic matrix as a realisation of a random process generated from a model. Regularisation refers to the inference technique of imposing additional structural assumptions on the problem to reduce underconstrainedness. Regularisation methods are defined by four components:

(i) a **prior solution**, generated from a *model*,

(ii) a **model deviation** measure, used to compute the deviation of a feasible solution from the model,

(iii) a **distortion** measure, used to compare the deviation of the model with the observations, and

(iv) an **adjustment step**, to ensure the constraints on the total traffic entering and exiting all ingress and egress nodes respectively, as well as non-negativity constraints, are satisfied.

In terms of an optimisation procedure, solving the tomography problem is equivalent to

$$\mathbf{x}^\star = \underset{\mathbf{x}\in\mathbb{R}^{N(N-1)}}{\operatorname{argmin}} \quad R(\mathbf{x},\mathbf{y}) + \lambda d(\mathbf{x},\mathcal{M}), \tag{30}$$

where $R(\cdot,\cdot)$ denotes the distortion measure, $d(\cdot,\cdot)$ denotes the model deviation measure and $\lambda \geq 0$ is the penalty constant that amplifies the penalisation of a feasible solution which strays too far away from the model[10]. Typically, $R(\mathbf{x},\mathbf{y}) = \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2$. Regularisation techniques are *biased* to a particular prior model. Thus, if the model is inconsistent, then the estimator (30) would be inconsistent as well. However, if the prior model chosen describes the final solution somewhat accurately, then it is expected that the final estimate would be fairly accurate.

As an example, suppose the prior model, $\mathbf{x}^{(0)}$, used is the gravity model, which can be derived from link measurements by calculating the ingress and egress traffic volumes (by summing link measures on the edge-links of the network). One proposed penalty [99] is defined as

$$d(\hat{\mathbf{x}},\mathbf{x}^{(0)}) = H(\hat{\mathbf{x}}) + H(\mathbf{x}^{(0)}) - H(\hat{\mathbf{x}},\mathbf{x}^{(0)}), \tag{31}$$

where

$$H(\mathbf{x}) = -\sum_{j=1}^{N(N-1)} \frac{x_j}{\sum_{k=1}^{N(N-1)} x_k} \log \frac{x_j}{\sum_{k=1}^{N(N-1)} x_k}, \tag{32}$$

is the empirical entropy, while

$$H(\mathbf{x},\mathbf{x}^{(0)}) = \sum_{j=1}^{N(N-1)} \frac{x_j}{\sum_{k=1}^{N(N-1)} x_k} \log\Big(\frac{x_j}{\sum_{k=1}^{N(N-1)} x_k} \Big/ \frac{x_j^{(0)}}{\sum_{k=1}^{N(N-1)} x_k^{(0)}}\Big), \tag{33}$$

is the joint empirical entropy, between the estimate and the prior model. The penalty function (31) measures the uncertainty between the quantities $\mathbf{x}$ and $\mathbf{x}^{(0)}$, and is commonly known as the *mutual information* [25]. The joint entropy term $H(\mathbf{x},\mathbf{x}^{(0)})$ quantifies the uncertainty between $\mathbf{x}$ and $\mathbf{x}^{(0)}$. If $H(\mathbf{x},\mathbf{x}^{(0)}) = 0$, then $\mathbf{x}$ is statistically independent of $\mathbf{x}^{(0)}$.

The approach is highly flexible: it can deal with the generalised gravity model simply by using a new prior model and constraints (17) are added to account for the different traffic classes (access and peering traffic).

The penalty can be rewritten and thought of as the Kullback-Leibler distance [25] between the estimate $\hat{\mathbf{x}}$ and the prior model $\mathbf{x}^{(0)}$, implying that the estimation objective is seeks to preserve as much prior information from $\mathbf{x}^{(0)}$ as possible, while minimising $R(\mathbf{x},\mathbf{y})$. This can be used directly, or approximated, for instance as a weighted quadratic [97, 98].

Using suitable models, most of the existing inference methods can be described in this framework (see [99] for details). Or, other penalties can be used, such as the nuclear norm, given by

$$d(\mathbf{X}) = \|\mathbf{X}\|_* = \sum_{i=1}^{r} \sigma_i \tag{34}$$

[10]Technically, $\mathbf{x}$ comprises non-negative integers, but a relaxation to real numbers is used as it is easier to compute, especially when considering large traffic matrices.

for low rank model recovery.

The solution of the optimisation procedure is often adjusted after regularisation using Iterative Proportional Fitting (IPF) [27], so as to satisfy the observed total traffic constraints and non-negativity constraints (those that weren't included in the regularisation for computational reasons). In practice, the IPF is a very simple algorithm, performing fast even on large traffic matrices.

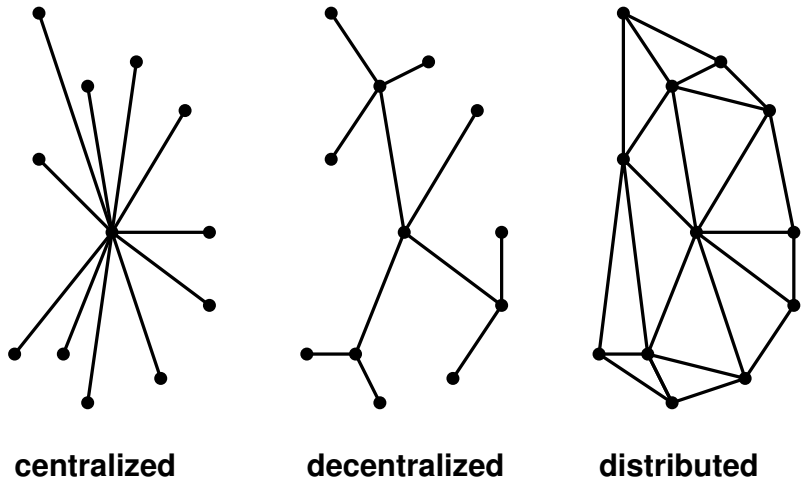


Figure 11: *Three example topologies where local traffic matrices provide benefits (motivated by the seminal figure in [7]).* ***Left****: centralised, or star, topology,* ***Centre****: decentralised topology,* ***Right****: distributed topology.*

Additional information can also be used, for instance, if some rows of the matrix are known from measurements, then this eases the number of variables to be estimated, making the problem a little simpler. Another source of potential data is the collection of *local traffic matrices* [92], providing information on traffic between interfaces of routers. We can see why this is useful by considering the three network topologies in Figure 11, with a centralised or star, a decentralised and a distributed topology. If the network has a star topology, then the entire traffic matrix is known if the local traffic matrix of the router right in the centre is obtained. For the other two topologies, collection of local traffic matrices in strategic places of the topology is likely to reduce the underconstrainedness of the original inference problem, though less (relative) information is provided the more distributed the topology. Local traffic matrices have been demonstrated to provide a significant information boost in [98], especially if the interfaces are well-connected, and that is highly dependent on the underlying network topology. If direct flow measurements from dedicated monitors are available, they provide a huge boon as an entire row of an IE traffic matrix would be revealed. In practice, however, these are generally not available as they are deemed expensive. The advantage of the regularisation method is that these additional information may be incorporated easily via constraints.

Another issue is their computational tractability. Speed is an issue for these algorithms, since traffic matrices are often large. Most model deviation and distortion measures are chosen to be convex, with linear constraints. In this way, problem (30) is a convex optimisation problem, where many fast and efficient algorithms have been developed to solve such problems [11].

The discussions here only consider point-to-point traffic matrices. For IE matrices, the point-to-multipoint matrix may be more useful instead. Recall from the above that an ideal traffic matrix is invariant to other network aspects to be useful for network design. Unlike the point-to-point traffic matrix, the point-to-multipoint matrix contains records on the amount of traffic from one ingress point to a set of egress points. These sets are chosen to preserve invariance under changes in the egress point, a property much more useful for network planning. Inference of the point-to-multipoint traffic matrices may be done in a similar fashion to point-to-point IE matrices [98].

5.2 Network Optimisation

An Internet Service Provider (ISP) of a backbone network must ensure each link in the network has adequate capacity. The consequences of failure to provide such capacity is congestion, and a resulting loss of quality service, which in severe cases would result in loss of customers. However, over-provisioning can be wasteful, and so optimisation is used to strike the right balance between cost and capacity.

Network optimisation involves several tasks with varying planning horizons. Assuming node locations are fixed, in the long term, we plan a network by considering the link locations, and capacity. We refer to this as *network planning*. It may be categorised into two common scenarios: *incremental planning* on existing networks, or *green-fields planning* [72]. In the former, the planning takes an evolutionary route, since the network designer is constrained by the current existing network. Any upgrades to the network are deliberately incremental, so as not to disrupt current operations. Green-fields planning, as the name suggests, begins from scratch: the entire network is designed from ground up. Shorter-term network optimisation tasks include traffic engineering [75] and some potential routing schemes.

In all of these tasks, one of the key ingredients is the traffic matrix. The reason these matrices are so useful comes down to *invariance*. The traffic on a particular link obviously varies as the links in the network change. However, an ideal traffic matrix is invariant to other aspects of the network such as the network topology and underlying routing protocols. Invariance allows the design to be varied without the inputs to the process changing. To a certain extent, the IE traffic matrix satisfies the invariance property, but it is far from perfect for some tasks, most notably is highly sensitive to external routing changes, and some internal changes [88,89]. The OD matrix is in some sense preferable [6], but harder to measure in most cases. The point-to-multipoint IE matrix (discussed in the previous section) is a useful compromise.

Furthermore, a network operator would require a prediction of the traffic matrix out to the level of the planning horizon for a task. Any forecast depends on the time scale involved and the underlying model used. At short time scales, say minutes stationarity may be a reasonable approximation, and there are therefore many time series approaches the problem. On time scales of hours to days to weeks, the cyclostationarity nature of the data must be included. The temporal models presented earlier can provide such predictions. For instance, with the model (4), we can estimate the mean traffic at some time in the future by simply extrapolating the mean. Longer-term prediction often focusses purely on the large-scale trend $L(t)$, which is often captured using a simple growth model (linear or exponential) and regression. In all cases, historical data is needed, usually several times as long as the prediction interval.

In addition, whenever performing prediction we should provide estimate variances, or confidence intervals, though this component of the problem has not been well-studied in the specific context of traffic matrices.

5.3 Reliability Analysis

Traffic matrices may also be used to conduct reliability analyses, where the affect (on traffic) or network failures is considered. A basic task in most network design is to create redundant paths to carry traffic in case of failure, but if high reliability is required, then an operator should also ensure that there is sufficient capacity in the network to carry this traffic along its alternate paths. For more details of this task see [72].

5.4 Anomaly Detection

Unfortunately, in reality, not all incoming traffic to a network is legitimate. Various attacks may be launched on a network: DDoS attacks or worm outbreaks such as the Nimda worm. Non-malicious, but equally violent spikes in traffic may be caused by a flash crowd or implementation bugs. We call

these surges anomalies, and if they catch a network operator by surprise they can congest networks, causing untold damage to daily network activities. Other types of anomalies may cause drops in traffic, again resulting in performance problems.

All these anomalies may be rare, but the potential damage can be tremendous. It is for these reasons network operators strife to detect anomalies, with the hope of protecting their networks from these harmful effects.

Although network equipment vendors do provide some form of fast detection and diagnosis mechanisms, these features are generally not adequate for the problems listed above. Consequently, methods were developed to counter anomalies. One approach is to use detailed (packet level) traces and signature-based detection to detect known attacks, but this does not help if the attack is unknown (in advance) or if the necessary measurements are not available. Other techniques infer statistical anomalies from the traffic flow data or SNMP measurements. Traffic matrices play an important role in this respect, since these matrices record traffic volumes across a whole network.

The basic principle of anomaly detection is to define a *baseline operating condition* of the network, by establishing normal conditions of the traffic. The baseline could be from a model, say a gravity model, for example. There are many approaches, such as entropy-based methods [12, 41], as network anomaly detection itself is a vast topic, but here, the focus is on the direct use of traffic matrices for anomaly detection.

Deviations from the baseline predictions of the traffic are quantified with a chosen norm, and one is flagged as an anomaly if it exceeds a predefined threshold. There are two sources of error: *false positives*, when normal traffic is flagged as an anomaly, and *false negatives*, when a detector does not flag an anomaly. The latter type of error, where we miss a potential anomaly, are apparently a more serious problem (given the serious nature of anomalies). However, if too many false positives occur, then operators can be overwhelmed, and will typically ignore the alarm system. The false positive problem is exacerbated if the number of tests is large, and in traffic matrix analysis (where we might conduct one test per traffic matrix element, per time interval) that number can be very large, requiring a very low false positive rate.

We consider the tradeoff between the two in a ROC (Receiver Operating Characteristic) curve which shows the two types of errors plotted against each other as a function of the chosen threshold (or other suitable tuning parameter). However, proper assessment of an approach requires ground-truth data, which is, by the nature of anomalies, hard to obtain in the volumes required.

Models themselves can be modified to account for anomalous traffic. In §4.1, the model (4) itself has a term to account for sudden spikes in traffic, which was demonstrated empirically to be useful in detecting large shifts in traffic. Low rank models [100] were shown to be highly effective in detecting anomalies as well.

Many anomaly detection proposals may be broadly classified as methods to preprocess measurement data via a linear transformation, in order to separate normal traffic from anomalous traffic. This was observed in [96]. Their *anomography* (a portmanteau of "anomaly" and "tomography") framework is easy to understand and is aimed at providing a framework for discussing these types of techniques. It proceeds as follows: start by assuming the routing matrix is static in the entire duration of the measurements. Given a series of SNMP measurements, $\mathbf{Y} = \mathbf{A}\boldsymbol{\mathcal{X}}$, a new inference problem is obtained by multiplying $\mathbf{Y}$ with a linear transform $\mathbf{T}$ to obtain $\tilde{\mathbf{Y}} = \mathbf{A}\tilde{\boldsymbol{\mathcal{X}}}$, which are the anomalous link loads. Whether the focus is on spatial or temporal anomalies depends on whether $\mathbf{Y}$ is pre- or post-multiplied with $\mathbf{T}$:

(i) *spatial anomography*: pre-multiplication, *i.e.,* $\tilde{\mathbf{Y}} = \mathbf{TY}$, uses the spatial relationships between traffic at particular points in time to find traffic that is unusual with respect to other flows at the same time; and

(ii) *temporal anomography*: post-multiplication, *i.e.,* $\tilde{\mathbf{Y}} = \mathbf{YT}$; uses the relationships between traffic at different times to determine if traffic is unusual for its point in time.

The two have been combined to create spatio-temporal anomaly detection [100], though the full details of this go beyond the scope of this article.

The above assumes the routing matrix is static over the series of measurements. The models themselves have to be modified to account for possible route changes. Some models are less amenable to modification, requiring a high number of constraints that scale with the number of measurements [96], which makes them undesirable for practitioners.

Anomaly detection employing SNMP data took off with a series of papers [47–49,52], where the low intrinsic spatial dimensionality of traffic matrices was exploited via a PCA-based anomaly detector. In the spatial PCA-based method, the principal components and axes of $\mathbf{Y}$ are computed from its columns and ordered from most to least significant component, to obtain a subspace $\mathbf{P} = [\mathbf{v}_1\, \mathbf{v}_2\, \cdots \mathbf{v}_m]$. The traffic space is then divided to a *normal subspace* and an *anomalous subspace*. The traffic time series is then projected on each principal axes, starting from $\mathbf{v}_1$ and so forth, and the projection magnitude is compared to a simple hard threshold of three standard deviations from the mean. Once there exists a projection exceeding this threshold, say at some $\mathbf{v}_K$, this component and subsequent components are classified as belonging the anomalous subspace $\mathbf{P}_A = [\mathbf{v}_K\, \mathbf{v}_{K+1}\, \cdots\, \mathbf{v}_m]$. The anomalous traffic is identified by projecting the time series onto the anomalous subspace and projecting the traffic back to obtain $\tilde{\mathbf{Y}}$.

Lakhina's spatial PCA method fits in the framework since the last step of extracting the anomalous traffic involves the projection $\tilde{\mathbf{Y}} = (\mathbf{P}_A^{\mathrm{T}}\mathbf{P}_A)\mathbf{Y}$ so the linear transformation is $\mathbf{T} = (\mathbf{P}_A^{\mathrm{T}}\mathbf{P}_A)$. However, its shortcomings have been the subject of scrutiny. Implementation of spatial PCA to network traffic is likely to be ineffective due to several drawbacks [68, 96]. Spatial PCA can be contaminated with a large anomaly[11], rendering it unable to detect the anomaly. In fact, PCA has been known to flag an entire measurement interval although there is only one anomaly present [96]. Additionally, PCA is very sensitive to the underlying data: adequate measurements are required and there must be a sufficient level of traffic aggregation before underlying trends can be detected by PCA. It is not robust enough in practice, requiring much fine tuning. Finally, there is a high computational cost involved in computing the principal components of a traffic matrix.

Other alternatives exist: wavelet transformations [8], Fourier transformations, autoregressive integrated moving average or ARIMA [96], and temporal PCA [96], where PCA is applied to the rows of $\mathbf{Y}$ (the temporal dimension) instead. In all these techniques, the baseline traffic flows are assumed to follow the prescribed model. In the Fourier model, baseline traffic is assumed to be composed of low frequencies. High frequencies may potentially indicated the presence of anomalies since these correspond to sudden changes in the traffic. Thus, the transformation filters out low frequencies and examines the remaining high frequencies to determine if any of these frequencies exceed a predetermined threshold. A similar rationale holds for the wavelet transform model. The ARIMA model [13] is very well-known in time series analysis, providing flexibility in the choice of parameters. The model generalises popular models such as the Holt-Winters model, the random walk model and exponentially weighted moving average models. It also allows memory and long range dependency [93] to be built into the model via fractional ARIMA, as evidenced and used to great effect in [76][12].

After $\tilde{\mathbf{Y}}$ is obtained, the anomalous traffic $\tilde{\boldsymbol{\mathcal{X}}}$ has to be recovered. The choice of a particular inference algorithm would depend on the model. The spatial PCA method uses a greedy algorithm to find the largest anomaly in each time bin [47]. Other methods include the use of ℓ_1 regularisation, inspired by compressive sensing [16, 28], which was shown, when coupled with the ARIMA model, outperforms other methods, including PCA and wavelet-based anomaly detection [96].

The bottom-line is that the model of the traffic matrix matters in anomaly detection. It serves as a baseline. However, it also needs to consistently allow for anomalies. One problem with approaches such as PCA is the models implied by the approach are often left unstated (implicit) and do not allow the anomalies to be separated as part of estimation (thus they can pollute the estimation process). Good techniques, going on into the future, need to be able to perform such separation consistently.

[11] Though in fact this is a problem in general for anomaly detection, and has not received the attention it deserves.
[12] See [13] for a good introduction to time series analysis.

5.5 Traffic Matrix Synthesis

Synthesis of the traffic matrix is an important area, motivated by the lack of real world traffic matrices available, due to the proprietary nature of most traffic data. Publicly available data is often obtained from networks operated and maintained by research institutions and universities, such as GÉANT [26] and Abilene [58], which may be limited in scope and is certainly biased towards educational and research networks.

Thus, network operations stand to gain much from artificially synthesised traffic matrices, provided a good model is available. Artificial traffic matrices may be used in capacity planning to stress test network topologies to see if they stand up to heavy loads without ending up with congestion. Monte Carlo-type simulations may be used to produce estimates of the behaviour of networks. Synthesised traffic matrices provide an avenue to explore the limitations of a protocol in a controlled environment before running it on an actual network. A good model simplifies these simulations, as parameters of the model can be tuned to generate a variety of scenarios for testing protocols.

Unfortunately, there is a dearth of work on traffic matrix synthesis, apart from [61,70] and a brief mention regarding synthesis of matrices from the independent connections model [33,34]. The problem of synthesising traffic matrices is the inverse of the inference problem. In synthesis, the topology of the network matters, as the generated entries of the artificial traffic matrix must not exceed the link capacity it is mapped to. Some models, such as the gravity model, automatically satisfies bandwidth constraints naturally [70]. However, if the generated entries do not conform to these constraints, there are algorithms to solve these problems [61]. Computational complexity of the model is the other important issue, dependent on the number of parameters. Hence, there is an inherent tradeoff between the descriptive power of the model and the ease of synthesising traffic matrices. A guideline is to preferably choose the model with as little parameters as possible but enough proven descriptive power, measured via an information criterion, such as the AIC [5].

We here describe the simple approach of [70] motived by works such as [37], in order to provide a starting point for future work on such synthesis. We start by taking $\mathbf{x}^{\text{in}}$ and $\mathbf{x}^{\text{out}}$ to be vectors of N i.i.d. exponential random variables with mean one. The TM is then generated using (7). We can then adjust the total traffic to match the desired total by simple scaling. This method is extremely simple (an exponential distribution has only one parameter to estimate), and we need generate only $2N$ random variables. Yet it matches observed statistics for both Abilene and GÉANT data extremely well [6].

6 Future

There are some interesting tasks left for traffic matrix research. The various algorithms and techniques described here could be improved, though in many cases the improvements may be relatively incremental given the success of existing approaches. More interest may be found in extending the ideas and techniques used here to new domains, and to evolving Internet traffic.

There are a few obvious cases (and no doubt many less obvious cases that we have not thought of), for instance: multicast traffic has not, to our knowledge, been studied in this way. Multicast is interesting because it violates the *traffic conservation* assumption that lies underneath many techniques for estimation and modelling of traffic matrices. We could image modelling it by considering the "flow" to be the traffic on a multicast group, from say one source, to a set of destinations, and then stacking a vector with these. The routing matrices now include elements for every link used (no longer following a single path). The traffic "matrix" could then be the a column vector of the traffic on each of these flows. So the idea of multicast traffic can fit into the structure we have talked about here, but appropriate models for performing tasks such as inference do not seem to exist.

It would also be very interesting to understand the way that CDNs are affecting network traffic. A CDN's typical goal is to bring content closer to the user, thereby reducing network traffic. However, that explicitly violates the "friction free" assumption in most gravity models, and introduces distance as something to be modelled.

That leads naturally to consideration of global traffic. Almost all studies of traffic have concentrated on a single network no larger than the national scale. That may still be very large – for instance several studies looked at Tier 1 providers in the USA, which for some time dominated Internet traffic. However, although large, it was still relatively homogeneous traffic between people speaking much the same language(s) from place to place in the network. When we consider the Internet globally, we may see that there are language or cultural clusters where large groupings of traffic a focussed by these issues.

On a large scale, time zones also play a significant role. Traffic patterns show strong cyclic behaviour based on user activity, but such activity is strongly dependent on the local time zone. If traffic is flowing from user to user, then this can result in strong apparent locality effects, simply because people in the same time zone are more likely to be awake at the same time [38].

Language and cultural focussing may be geographic in nature, but it might also be considered per network, and that leads to another topic of some interest. Very few papers have tried to consider inter-AS (also known as inter-domain) traffic in any detail. Exceptions are Chang *et al.* [21] (which presents suggestions for estimating traffic based on models of business models and resulting usage); Bharti *et al.* [9] (which considered inference of hidden elements of this matrix using a subset of data), Feldman *et al.* [35] (which aimed to estimate a global traffic matrix, but only in the limited domain of WWW traffic), and Labovitz *et al.* [46] (which looked at inter-domain traffic from 110 network operators over a two-year period, though not in the form of a matrix). Study of the Internet's global traffic matrix is made difficult by the sheer scale of the project: Labovitz *et al.* studied 110 network operators over a two-year period[13] and to do so, collected over 200 Exabytes of data. Many network operators do not collect or store data of the type required for such a study, and many more regard it as proprietary or covered by privacy legislation with provisions such that no researcher is ever likely to see it. So we can see that study of the inter-domain matrix is likely to be a long-term, and rather challenging project.

In addition, we know that the *traffic profile* (or mix of applications) has changed fairly rapidly over time. It is likely this trend will continue, and there are bound to be effects on traffic patterns as a result. Peer-2-peer traffic significantly altered traffic patterns when it appeared because it was more symmetric than traditional (at the time) WWW traffic. However, in addition, peer-2-peer applications have the potential to exploit locality information to download from sources closer to the destination. This could potentially change the "no friction" assumption in much the same way that CDNs can, though in the early days it did not appear to be the case [38]. An example traffic matrix (drawn from [38]) showing normalised[14] traffic between regions in a cable-network operator is given in Table 1. The major deviation, in this data, from a pure gravity model seemed to be based more on time-zone differences than other aspects of locality.

From/To	R1	R2	R3	R4	R5	R6	R7	R8
R1	-	0.180	0.140	0.126	0.174	0.128	0.124	0.127
R2	0.172	-	0.141	0.126	0.190	0.132	0.118	0.120
R3	0.132	0.120	-	0.189	0.135	0.145	0.139	0.140
R4	0.107	0.111	0.182	-	0.124	0.163	0.155	0.158
R5	0.161	0.180	0.136	0.132	-	0.135	0.127	0.129
R6	0.107	0.108	0.145	0.155	0.125	-	0.187	0.173
R7	0.107	0.106	0.137	0.157	0.127	0.182	-	0.184
R8	0.109	0.111	0.127	0.161	0.128	0.178	0.185	-

Table 1: Normalised inter-regional traffic matrix from [38].

Other applications may equally change traffic matrices in the future, so there are lots of new, or

[13] To put this in context, there are tens of thousands of ASes in the Internet.

[14] The elements have been normalised by dividing each row by the row-sum, so that each element actually represents the probability that a packet enters the network at a given region i will departs the network at region j.

changing traffic classes to consider in modelling Internet traffic matrices. On the other side, the tasks of interest, the work on anomaly detection is likely to continue due to its immediate benefit to operators, but the most overlooked task is traffic matrix synthesis.

Synthesis means generating artificial traffic matrices, typically for use in simulations. There are only, to our knowledge, two papers [61, 70] on synthesising Internet traffic matrices, but there are already quite a few where synthetic traffic matrices were used, and this demand for such matrices will continue.

Synthesis is not demanding in some ways. Traffic matrices are usually relatively small (compared to other types of traffic data), when measured at a reasonable level of aggregation and time scale. However, in other ways these matrices are quite challenging. For instance:

- we have few sets of traffic matrix data, and even fewer that are public, and somehow need to use these to estimate properties of these complex, high-dimensional objects;
- there is a real relationship between topology and traffic (although we would like a traffic matrix to be invariant to the topology, there are clear cases where, particularly IE matrices are not);
- traffic matrices come in a wide variety of types (at different levels of aggregation, for particular applications and so on) and it is unlikely that one model fits all; and
- there are a number of conflicting goals in synthesis, *e.g.,* to generate variability, but well "matched" to real traffic matrices.

However, there is considerable hope that progress can be made in terms of generating synthetic matrices, both for green-fields network design [45], and for simulation in general.

The major use of synthesis is in simulation. In many cases a traffic matrix is enough for a simulation, but in others, we need to translate this into packets (or at least connections). The analogue in transportation modelling is often called a *micro-simulation* model. Here, the problem becomes one of taking a *demand* matrix (remember, most of the work here is related to traffic, not demand), and translating this into carried load. We know how to do that (using simulation tools such as `ns`) but doing it efficiently is difficult. One paper [80] starts to tackle this problem, but as in the work of transportation modelling, there is considerable scope for advanced scalable micro-simulation of traffic.

Another use for synthetic traffic matrices is in the further task of synthetic topology generation, but we shall leave discussion of this topic to another section of this book.

7 Conclusion

This chapter has been aimed at introducing the reader to the state-of-the-art in Internet traffic matrix modelling and applications. It is not a complete survey of all research into Internet traffic matrix modelling as such a survey would necessarily consume a much larger amount of space, and be less digestible, and we apologise to those whose work has not been referenced.

The chapter has also aimed to clarify a set of common terminology in a field which has occasionally been confounded by ambiguous or confusing terms. It is our aim to also provide, as an adjunct to this chapter, links to the most commonly used datasets in this domain, and code to perform some of the commonest tasks. In this way, we hope to provide a firm foundation for future work in the area, and to help those who just want to use traffic matrices in their research.

References

[1] Cisco NetFlow. http://www.cisco.com/go/netflow.

[2] Cisco visual networking index: Forecast and methodology, 2011-2016. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.

[3] Internet Activity, Australia. http://www.abs.gov.au/AUSSTATS/abs@.nsf/DOSSbyTopic/0444532C5EBD3B76CA256BD0002833B6, 2013.

[4] E. Acar, T. G. Kolda, D. M. Dunlavy, and M. Mørup. Scalable tensor factorizations for incomplete data. In *SIAM International Conference on Data Mining (SDM) 2010*, pages 701–712, April 2010.

[5] H. Akaike. A new look at statistical model identification. *IEEE Trans. Autom. Control*, 19(6):716–723, December 1974.

[6] D. L. Alderson, H. Chang, M. Roughan, S. Uhlig, and W. Willinger. The many facets of Internet topology and traffic. *Networks and Heterogeneous Media*, 1(4):569–600, December 2006.

[7] P. Baran. On distributed communications: 1. introduction to distributed communications network. RAND Memorandum, August 1964.

[8] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, pages 71–82, 2002.

[9] V. Bharti, P. Kankar, L. Setia, G. Gürsun, A. Lakhina, and M. Crovella. Inferring invisible traffic. In *Proceedings of the 6th International Conference*, Co-NEXT '10, pages 22:1–22:12, 2010.

[10] R. Blili and A. Maghbouleh. Best practices for determining traffic matrices in IP networks V 4.0. Tutorial in NANOG 43, http://www.nanog.org/meetings/nanog43/presentations/Blili_trafficmatrix_N43.pdf, 2008.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian. Anomaly extraction in backbone networks using association rules. In *ACM SIGCOMM IMC 2009*, pages 28–34, 2009.

[13] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2nd edition, March 2002.

[14] E. Candes and Y. Plan. Matrix completion with noise. *Proc. IEEE*, 98(6):925–936, June 2010.

[15] E. Candes and B. Recht. Exact matrix completion via convex optimization. *Found. of Comput. Math.*, 9:717–772, 2008.

[16] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Info. Theory*, 52(12):5406–5425, December 2006.

[17] E. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Info. Theory*, 56(5):2053–2080, May 2010.

[18] J. Cao, D. Davis, S. V. Wiel, and B. Yu. Time-varying network tomography: Router link data. *J. Am. Statist. Assoc.*, 95(452):1063–1075, December 2000.

[19] J. D. Case, M. Fedor, M. L. Schoffstall, and J. R. Davin. A simple network management protocol (SNMP). Technical Report RFC 1157, IETF, May 1990. http://www.ietf.org/rfc/rfc1157.txt.

[20] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network Tomography: Recent Developments. *Statistical Science Magazine*, 19(3):499–517, August 2004.

[21] H. Chang, S. Jamin, Z. Mao, and W. Willinger. An empirical approach to modeling inter-AS traffic matrices. In *ACM/SIGCOMM Internet Measurement Conference (IMC) 2005*, pages 139–152, October 2005.

[22] K. Claffy, H.-W. Braun, and G. Polyzos. Tracking long-term growth of the NSFNET. Cooperative Association for Internet Data Analysis - CAIDA, http://www.caida.org/publications/papers/1994/tlg/, 1994.

[23] M. Coates, A. Hero, R. Nowak, and B. Yu. Internet tomography. *Signal Processing Magazine*, 19(3):47–65, May 2002.

[24] P. D. Converse. New laws of retail gravitation. *The Journal of Marketing*, 14(3):379–384, October 1949.

[25] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 2nd edition, 2006.

[26] DANTE. GÉANT Trace Data. http://www.geant.net/.

[27] W. E. Deming and F. F. Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Ann. Math. Stat*, 11(4):427–444, 1940.

[28] D. Donoho. Compressed sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, April 2006.

[29] N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. *IEEE/ACM Trans. Networking*, 13(5):933–946, 2005.

[30] N. Duffield, C. Lund, and M. Thorup. Learn more, sample less: control of volume and variance in network measurement. *IEEE Trans. Info. Theory*, 51(5):1756–1775, 2005.

[31] B. Eriksson, P. Barford, R. Bowden, M. Roughan, N. Duffield, and J. Sommers. BasisDetect : A model-based network event detection framework. In *ACM SIGCOMM Internet Measurement Conference*, Melbourne, Australia, 2010.

[32] S. Erlander and N. F. Stewart. *The gravity model in transportation analysis: Theory and extensions*. Topics in Transportation. International Science, 1990.

[33] V. Erramilli, M. Crovella, and N. Taft. An independent-connection model for traffic matrices. In *ACM IMC 2006*, pages 251–256, October 2006.

[34] V. Erramilli, M. Crovella, and N. Taft. An independent-connection model for traffic matrices. Technical Report BUCS-TR-2006-022, Department of Computer Science, Boston University, September 2006.

[35] A. Feldmann, N. Kammenhuber, O. Maennel, B. Maggs, R. De Prisco, and R. Sundaram. A methodology for estimating interdomain web traffic demand. In *IMC '04*, pages 322–335, October 2004.

[36] M. J. Ferrari, O. N. Bjornstad, J. L. Partain, and J. Antonovics. A gravity model for the spread of a pollinator-borne plant pathogen. *American Naturalist*, 168(3):294–303, September 2006.

[37] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, 40(10):118–124, October 2002.

[38] A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen. P2P The Gorilla in the Cable. In *National Cable & Telecommunications Association(NCTA) 2003 National Show*, Chicago,IL, June 2003.

[39] O. Goldschmidt. ISP backbone inference methods to support traffic engineering: Methodology and experience. In *Internet Statistics and Metrics Analysis (ISMA) Workshop*, December 2000.

[40] N. K. Groschwitz and G. C. Polyzos. A time series model of long-term NSFNET backbone traffic. Cooperative Association for Internet Data Analysis - CAIDA, http://www.caida.org/publications/papers/1994/tsm/, 1994.

[41] Y. Gu, A. McCallum, and D. Towsley. Detecting anomalies in network traffic using maximum entropy estimation. In *ACM SIGCOMM IMC 2005*, pages 32–32, 2005.

[42] A. Gunnar, M. Johansson, and T. Telkamp. Traffic matrix estimation: A comparison on real data. In *ACM IMC 2004*, pages 149–160, October 2004.

[43] A. F. Jung. Is Reilly's law of retail gravitation always true? *The Journal of Marketing*, 24(2):62–63, October 1959.

[44] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.

[45] J. Kowalski and B. Warfield. Modeling traffic demand between nodes in a telecommunications network. In *ATNAC '95*, 1995.

[46] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet inter-domain traffic. In *ACM Sigcomm*, 2010.

[47] A. Lakhina, M. Crovella, and C. Diot. Characterization of network-wide anomalies in traffic flows. In *ACM IMC 2004*, pages 201–206, 2004.

[48] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM 2004*, pages 219–230, 2004.

[49] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. In *ACM SIGCOMM 2005*, pages 217–228, 2005.

[50] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural analysis of network traffic flows. *SIGMETRICS Perform. Eval. Rev.*, 32(1):61–72, June 2004.

[51] D. Lam, D. Cox, and J. Widom. Teletraffic modeling for personal communications services. *IEEE Communications Magazine: Special Issues on Teletraffic Modeling Engineering and Management in Wireless and Broadband Networks*, 35:79–87, February 1997.

[52] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and identification of network anomalies using sketch subspaces. In *ACM IMC 2006*, pages 147–152, 2006.

[53] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman and Hall, 2nd edition, 1989.

[54] D. McFadden. Modeling the choice of residential location. In A. Karlqvist et al., editor, *Spatial Interaction Theory and Planning Models*, pages 75–96. North Holland, 1978.

[55] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, and C. Diot. A taxonomy of IP traffic matrices. *Proc. SPIE*, 4868(200), July 2002.

[56] A. Medina, N. Taft, K. Salmatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *ACM SIGCOMM 2002*, 2002.

[57] G. D. Murray and A. D. Cliff. A stochastic model for measles epidemics in a multi-regional setting. *Trans. Institue British Geographers*, 2:158–174, 1977.

[58] NLANR. Abilene Trace Data. http://pma.nlanr.net/Special/ipls3.html.

[59] I. Noros. A storage model with self-similar input. *Queueing Systems*, 16(3-4):387–396, 1994.

[60] A. Nucci, R. Cruz, N. Taft, and C. Diot. Design of IGP link weight changes for estimation of traffic matrices. In *INFOCOM 2004*, volume 4, pages 2341–2351, March 2004.

[61] A. Nucci, A. Sridharan, and N. Taft. The problem of synthetically generating IP traffic matrices: Initial recommendations. *SIGCOMM Comput. Commun. Rev.*, 35:19–32, July 2005.

[62] A. M. Odlyzko. Internet traffic growth: Sources and implications. In B. B. Dingel, W. Weiershausen, A. K. Dutta, and K.-I. Sato, editors, *Optical Transmission Systems and Equipment for WDM Networking II*, volume 5247, pages 1–15. Proc. SPIE, 2003.

[63] V. Paxson. End-to-End routing behavior in the Internet. *IEEE/ACM Trans. Networking*, 5(5):601–615, October 1997.

[64] R. B. Potts and R. M. Oliver. *Flows in Transportation Networks*. Academic Press, 1972.

[65] P. Pöyhönen. A tentative model for the volume of trade between countries. *Weltwirtschaftliches Archive*, 90:93–100, 1963.

[66] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum–rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[67] R. B. Reynolds. A test of the law of retail gravitation. *The Journal of Marketing*, 17(3):273–277, January 1953.

[68] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for traffic anomaly detection. In *Proceedings of the 2007 ACM SIGMETRICS*, pages 109–120, 2007.

[69] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Ann. Statis.*, 11(2):416–431, June 1983.

[70] M. Roughan. Simplifying the synthesis of Internet traffic matrices. *SIGCOMM Comput. Commun. Rev.*, 35(5):93–96, 2005.

[71] M. Roughan. A case-study of the accuracy of SNMP measurements. *Journal of Electrical and Computer Engineering*, 2010, 2010. Article ID 812979.

[72] M. Roughan. Robust network planning. In C. R. Kalmanek, S. Misra, and R. Yang, editors, *The Guide to Reliable Internet Services and Applications*, chapter 5, pages 137–177. Springer, 2010.

[73] M. Roughan and J. Gottlieb. Large-scale measurement and modeling of backbone Internet traffic. In *SPIE ITCom*, Boston, 2002.

[74] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in measuring backbone traffic variability: Models, metrics, measurements and meaning. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[75] M. Roughan, M. Thorup, and Y. Zhang. Traffic engineering with estimated traffic matrices. In *ACM IMC 2003*, pages 248–258, October 2003.

[76] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, and P. Abry. Non-Gaussian and long memory statistical characterizations for Internet traffic with anomalies. *IEEE Trans. Depend. Secure Computing*, 4(1), JAN–MAR 2007.

[77] G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 1978.

[78] A. K. Sen and T. E. Smith. *Gravity models of spatial interaction behavior*. Springer, 1995.

[79] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb. A case study of OSPF behavior in a large enterprise network. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement*, IMW '02, pages 217–230, 2002.

[80] J. Sommers, R. A. Bowden, B. Eriksson, P. Barford, M. Roughan, and N. G. Duffield. Efficient network-wide flow record generation. In *IEEE Infocom*, pages 2363–2371, 2011.

[81] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft. How to identify and estimate the largest traffic matrix elements in a dynamic environment. *SIGMETRICS Perform. Eval. Rev.*, 32(1):73–84, June 2004.

[82] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft. How to identify and estimate the largest traffic matrix elements in a dynamic environment. In *ACM SIGMETRICS 2004*, pages 73–84, June 2004.

[83] A. Soule, A. Nucci, R. L. Cruz, E. Leonardi, and N. Taft. Estimating dynamic traffic matrices by using viable routing changes. *IEEE/ACM Transactions on Networking*, 15(3):485–498, 2007.

[84] G. W. Stewart. *Matrix algorithms Volume 1: Basic decompositions*. SIAM, 1998.

[85] J. Q. Stewart. Demographic gravitation: Evidence and applications. *Sociometry*, 11(1/2):31–58, February 1948.

[86] J. D. Swait. Probabilistic choice set information in transportation demand. Technical report, Department of Civil and Environmental Engineering, MIT, June 1984.

[87] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. *J. Am. Statist. Assoc.*, 93(442), June 1998.

[88] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan. Traffic matrix reloaded: Impact of routing changes. In *Proc. Passive and Active Measurement Workshop*, pages 251–264, April 2005.

[89] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. *SIGMETRICS Perform. Eval. Rev.*, 32:307–319, June 2004.

[90] J. Tinbergen. Shaping the world economy: Suggestions for an international economic policy. *The Twentieth Century Fund*, 1962.

[91] Y. Vardi. Network Tomography: Estimating source-destination traffic intensities from link data. *J. Am. Statist. Assoc.*, 91:365–377, 1996.

[92] G. Varghese and C. Estan. The measurement manifesto. In *HotNets-II*, November 2003.

[93] D. Veitch and P. Abry. Wavelet analysis of long-range dependent network traffic. In *Proceedings of the 9th INFORMS Applied Probability Conference*, page 57, Cambridge, Massachusetts, 30 June - 1 July 1997. INFORMS Technical Section on Applied Probability. http://appliedprob.society.informs.org/.

[94] C. S. Wallace and D. M. Boulton. An information measure for classification. *Computer Journal*, 11(2):185–194, August 1968.

[95] Y. C. Xia, O. N. Bjornstad, and B. T. Grenfell. Measles metapopulation dynamics: A gravity model for epidemiological coupling and dynamics. *American Naturalist*, 164(3):267–281, 2004.

[96] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network anomography. In *ACM IMC 2005*, pages 317–330, 2005.

[97] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *ACM SIGMETRICS 2003*, pages 206–217, 2003.

[98] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *ACM SIGCOMM 2003*, pages 301–312, 2003.

[99] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. Estimating Point-to-Point and Point-to-Multipoint traffic matrices: An information-theoretic approach. *IEEE/ACM Trans. Netw.*, 13(5):947–960, October 2005.

[100] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu. Spatio-Temporal compressive sensing and Internet traffic matrices. In *ACM SIGCOMM 2009*, pages 267–278, August 2009.