Six Faces of Data Centric Networking

Eiko Yoneki

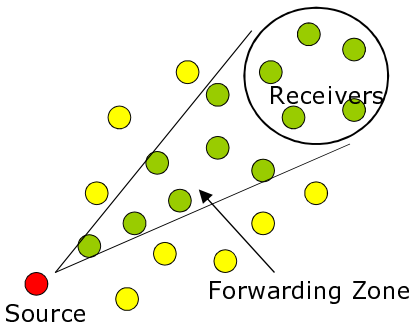
University of Cambridge Computer Laboratory

Data Centric Networking

- Shift of Communication Paradigm
  - From end-to-end to data centric
  - Data as communication token
  - Multipoint communication (Anycast and Multicast)
- Integration of complex data processing with networking
  - A key vision for future computing
  - A huge number of data sources and high volume of data accessible to applications

Geocast as an Example

- Data and Context decide Destination
- Data is forwarded when data is getting closer to the target region



Forwarding Zone in Geocast

3

What is Content Routing?

- Indirection point for multiplexing data messages based on content (semantic & syntactic) rather than network host addresses
- Features
  - Network address independence
  - Content based addressing
  - Asynchronous communication
  - Symmetric communication between source and sink
  - Cross layer (between middleware and network components)
  - Application and network level of programming paradigm
  - Integrate event correlation with networking

4

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Functional Point of View

- *Content routing from a functional point of view:*
  - Application layer
    - DNS tricks, HTTP redirects, P2P systems (routing on content hashes)
    - XML routers, ESB (Enterprise Service Bus), Publish/Subscribe systems, Application level of multicast
  - Transport layer
    - Load balancing HTML switches in data centres
  - Network layer
    - IP Multicast
  - Lower layer
    - Sensor networks data-centric routing

5

UNIVERSITY OF CAMBRIDGE Computer Laboratory

6 Faces of DCN

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

6

UNIVERSITY OF CAMBRIDGE Computer Laboratory

6 main Topics

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

7

UNIVERSITY OF CAMBRIDGE Computer Laboratory

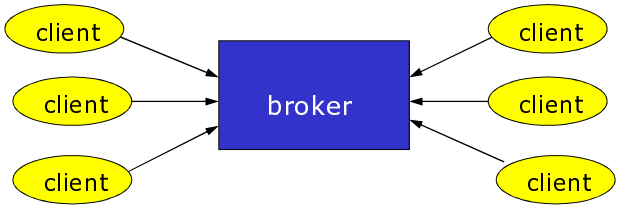
Multi-Point Communication

- Application level multicast
  - IP multicast is not supported well over wide area networks
  - Use DHT (Distributed Hashing Table)
  - Use tree routing in order to get logarithmic scaling
  - Bayeux/Tapestry and CAN
  - Service model of multicast is less powerful than content-based messaging system
- Research prototypes of messaging systems
  - Scribe (Topic-based system using DHT over Pastry)
  - SIENA (Content-based distributed event service)
  - JEDI (Content-based messaging system)
  - Gryphon (Topic/content-based message brokering system)

8

Content Based Networking

- Publish/Subscribe Paradigm
- Subscription model:
  - Topic-based (Channel)
    - Topics can be in hierarchies but not with several super topics
  - Content-based
    - Express interests as a query over the contents of data
    - How to turn subscriptions into routing mechanism in decentralised environments?



9

Publish/Subscribe over P2P

- Peer-to-peer techniques
  - Distributed hash tables (Pastry, CAN, Chord,..)
    - Overlay network of nodes with unique ids
    - Hash operation from key to node id
    - Scalable and efficient
- Advantages of P2P for publish/subscribe
  - High abstraction for building pub/sub systems
  - P2P overlay handles neighbouring set for brokers
    - Easy to manage
    - Dynamic mapping
    - Efficient routing
    - Fault-tolerance

10

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Publish/Subscribe Architecture

Subscription Types

Topic-Based | Content-Based | Type-Based

Routing Strategy

Simple Flooding	Parametric Flooding	Subsetting
Event Flooding	Gossiping	Rendezvous
Subscription Flooding	Adaptive Gossiping	Filter-Based

Overlay Types

Brokers Overlay | P2P Structured Overlay | P2P Unstructured Overlay

Network Protocols

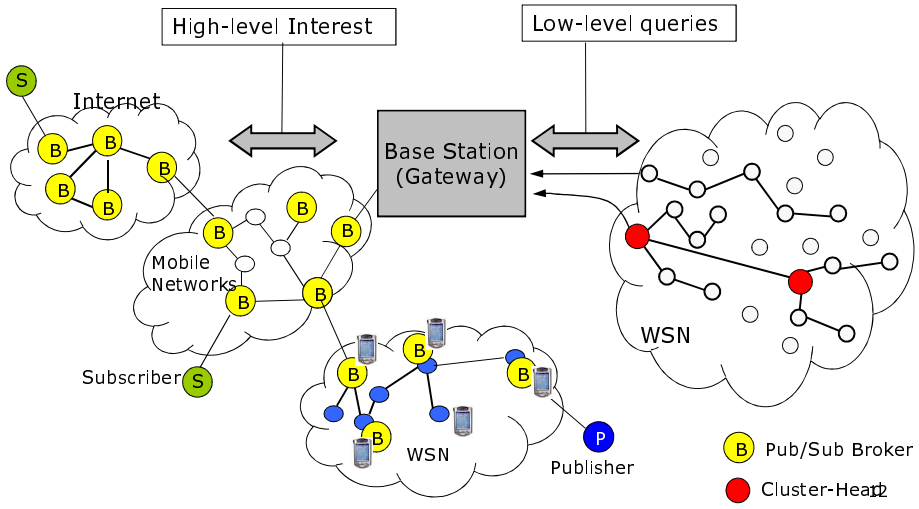
(TCP/IP, IP multicast, SOAP, 802.11g, MAC broadcast…)

11

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Publish/Subscribe System

- Content-Based Networking (CBN)



12

UNIVERSITY OF CAMBRIDGE Computer Laboratory

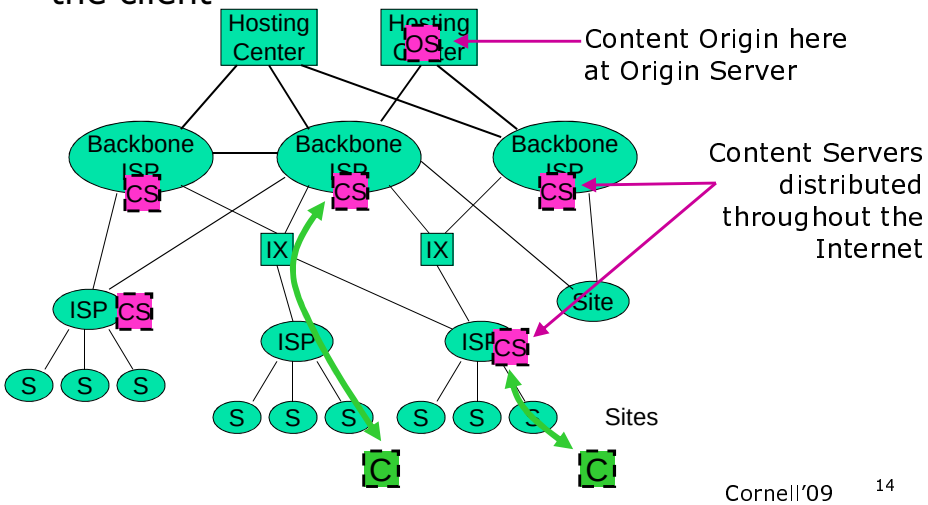
Content Distribution Networks

- Cache of data at various points in a network
- Content served closer to client
  - Less latency, better performance
- Load spread over multiple distributed systems
  - Robust (to ISP failure)
  - Handle flashes better (load spread over ISPs)
- Limitation
  - No mechanism with dynamic/personalized content, while more content is becoming dynamic
  - Difficult to manage content lifetimes and cache performance, dynamic cache invalidation
- CDN Providers
  - Coral Content Distribution Network
  - Akamai
  - BitTorrent
  - ...

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Content Routing Principle

- Content is served from content servers nearer to the client



Cornell'09

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Related Open Source Projects

- ***SIENA*** http://www.inf.usi.ch/carzaniga/cbn/
- ***Scribe*** http://research.microsoft.com/en-us/um/people/antr/overlays/overlays.htm
- ***CORAL*** http://www.coralcdn.org/
- ***Globule: an Open-Source Content Distribution Network*** http://www.globule.org/
- ***XML Blaster: Open Source XML event encoding with XPath expression subscription*** http://www.xmlblaster.org/

15

UNIVERSITY OF CAMBRIDGE Computer Laboratory

6 main Topics

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

16

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

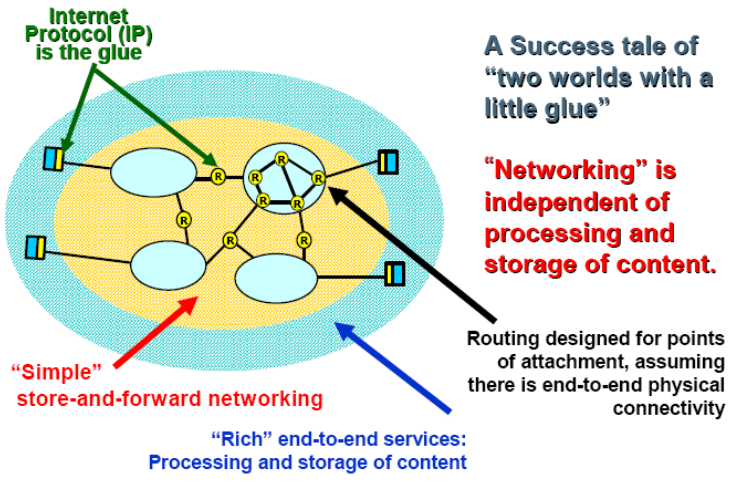
CCN and NDN

- Content-Centric Networking (CCN) and Named Data Networking (NDN)
- To networking that enables networks to self-organize and push relevant content where needed
- Pioneered by Van Jacobson

17

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

IP Internet Today



Garcia-Luna-Aceves'09

18

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Content Centric Networking

- Original Internet
  - 70s technology, conversational pipes, end-to-end
- Now, Internet use (>90%):
  - Content retrieval & Service access
  - Request & Delivery of *named data*
  - CDNs and P2P
- Shift to a content-centric view:
  - end-to-data
  - Content-awareness and massive storage
- Goals:
  - Remove the need to make DNS lookups
    - New naming system for services and data
    - Place the name lookup scheme in the network
  - Route to one of many possible service
  - Instances
    - Any-cast routing to a service instance
    - Find closest instance
  - Allow for service instances to move locations
  - Allow for self-certifying name

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Why CCN?

- Networks are used to access content
  - Source becomes less important – content itself matters
  - However there is no persistent content naming scheme
  - Different encodings/protection of same information, e.g. mp3, wav
- Efficiently handle increasing volume of information
  - No standard way to find and get *nearest* copy
  - Intelligent distribution of information (e.g. capacity, latency)
  - Include content inspection, filtering, video rendering
- Usable security is currently not content centric
  - Mainly based on securing channels (encryption) and trusting servers (authentication)

→ From CDNs to native *Content Networks*

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Existing Related Projects

- Next generation Internet proposals:
  - LNA, TRIAD, NIRA, ROFL, i3, DONA
- Van Jacobsen Content-Centric Networking
- PSIRP (Publish/Subscribe Internet Routing Paradigm)
- 4WARD - Architecture and Design for the Future Internet
  - NetInf
- Traditional Publish/Subscribe Systems, P2P and sensor networks

21

UNIVERSITY OF CAMBRIDGE Computer Laboratory

CCN in Practice

- Network delivers content from closest location
- Integrates a variety of transport mechanisms
- Integrated caching (short-term memory)
- Aggregation helps for right representation
- Search for related information
- Verify authenticity and control access

22

UNIVERSITY OF CAMBRIDGE Computer Laboratory

History of CCN I

- As early as 2002, Stoica et al (Berkeley) proposed the Internet Indirection Infrastructure (I3) in which routing is based on IDs with rendezvous-based communication
  - Sigcomm 2002, most CCN architectures today adopt ID-based routing
- In 2004, Balakrishnan et al (MIT-Berkeley) proposed a 3-Level Name Resolution: user-id to service-id, service-id to endpoint-id and endpoint-id to network address
  - Sigcomm 2004, today's locator-ID separation is a simplification of this
- In 2006, Caesar et al (Berkeley) proposed the Routing on Flat Labels (ROFL) approach which does not split identity and location but get rids of location for direct ID-based routing
  - Sigcomm 2006, superseded by the DONA approach of the same group

23

UNIVERSITY OF CAMBRIDGE Computer Laboratory

History of CCN II

- In 2007, Koponen et al (Berkeley) proposed the Data-Oriented Network Architecture (DONA) in which name resolution and routing are combined, based on IDs
  - Sigcomm 2007, key influence on emerging CCN architectures today
- In 2009, Jokela et al (Ericsson) proposed the Line Speed Pub/Sub Inter-Networking (LIPSIN) approach which employs an Internet-scale pub/sub approach for content access
  - Sigcomm 2009, emanated from the EU FP7 project PSIRP
- In 2009, Jacobson et al (Xerox Parc) – confirmed speaker for FIA Valencia - proposed the Networking Named Content (NNC) approach which treats content as a network primitive and retrieves content directly by name
  - CoNEXT 2009, to be tested by FP7 instead of US NSF projects

24

UNIVERSITY OF CAMBRIDGE Computer Laboratory

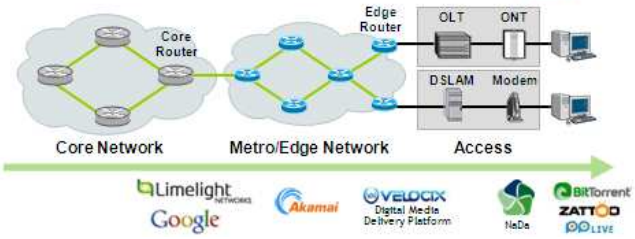
CDN approach

Pushing content closer to the users

- Hop count reduction (overall network traffic reduction) ➲ energy savings

CDN Strategies:

- Limelight — placing CDN servers near a small # of ISP core nets
- Akamai — placing CDN servers deep into a large # of ISP networks' sites
- Nano Data Center (NaDa) — home gateways (STBs/modems) as CDN servers
  (peer-to-peer delivery among NaDa servers)



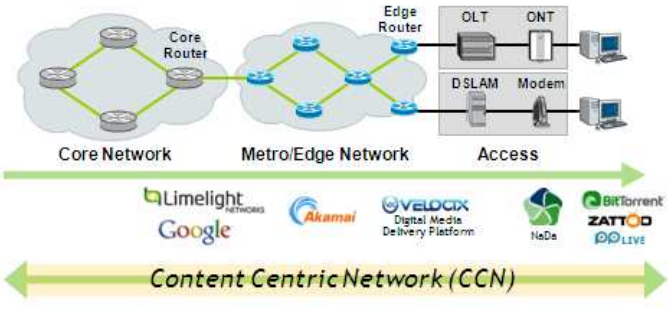
UNIVERSITY OF CAMBRIDGE Computer Laboratory

CCN approach

CCN enables efficient content dissemination to users

Energy efficiency of CCN content routers

- Extra memory hierarchy that requires minimal power draws
- Ex) Memory 4G ~ 10W, SSD 32G ~ 1W, Disk ~ 12W (if needed)



UNIVERSITY OF CAMBRIDGE Computer Laboratory

Related Open Source Projects

- ***CCN*** http://www.ccnx.org/ (http://www.named-data.net/)

27

UNIVERSITY OF CAMBRIDGE Computer Laboratory

6 main Topics

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

28

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Programming in Data Centric Environment

- Programming in Data Centric Environment
  - Cloud: Programming is becoming a data-centric fashion (e.g. transformations to data sets)
  - Network meets data flow programming
- Data Centre and Cloud environments
  - Applications: as a service
  - Components: Platform as a service (e.g. Google AppEngine, MS Azure)
  - Processes: Infrastructure as a Service (e.g. Amazon EC2)
  - Challenges:
    - Programming Model (exposure of concurrency, parallelism) and its implementation
    - Physical architecture (new communication protocols, structures)
    - High volume (e.g. billions of entities and terabytes of data) of data management in cloud infrastructure → Data oriented perspective

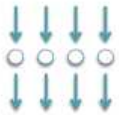
29

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Cloud Programming Model

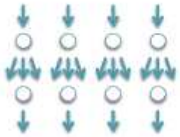
Batch computing: Condor, Grid Engine, Amazon SQS

- Programming Model: Relatively easy, but restricted
- Challenges: Scheduling, Load Balancing, Fault Tolerance
- Resources: Sufficient local memory & cores, fast file system



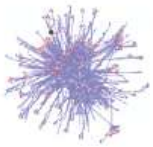
Loosely coupled: Hadoop, Dryad, Amazon EMR

- Programming Model: More complicated, more expressive
- Challenges: Parallel Communication
- Resources: 4+ cores, 1 TB / core disk, 4 GB / core RAM
  - Cloudera Recommendations: http://bit.ly/hi2lec



Tightly coupled: MPI, Pregel, Hadoop

- Programming Model: Most complicated, most expressive
- Challenges: Parallel Algorithms
- Resources: High Bandwidth, low latency interconnects
  - Amazon Cloud Compute Instance Type



30

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Data Flow Programming

- Data parallel programming (e.g. MapReduce, Skywriting)
- Declarative networking (e.g. P2)
  - Declarative language: "ask for what you want, not how to implement it"
  - Declarative specifications of networks, compiled to distributed dataflows
  - Runtime engine to execute distributed dataflows
  - Adopting a data centric approach to system design and by employing declarative programming languages → simplify distributed programming

31

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Skywriting

- JavaScript-like job specification language
  - Supports functional programming
  - Data-dependent control flow
- Distributed execution engine (Ciel)
  - Assigns tasks to devices
  - Publish/subscribe for results

32

D³N Data-Driven Declarative Networking

- How to program distributed computation?
- Use Declarative Networking
  - Use of Functional Programming
    - Simple/clean semantics, expressive, inherent parallelism
  - Queries/Filer etc. can be expressed as higher-order functions that are applied in a distributed setting

http://www.cl.cam.ac.uk/~ey204/pubs/2009_MOBIHELD.pdf

D³N and Functional Programming I

- Functions are first-class values
  - They can be both input and output of other functions
  - They can be shared between different nodes (code mobility)
  - Not only data but also functions flow
- Language syntax does not have state
  - Variables are only ever assigned once; hence reasoning about programs becomes easier

  (of course message passing and threads → encode states)
- Strongly typed
  - Static assurance that the program does not 'go wrong' at runtime unlike script languages
- Type inference
  - Types are not declared explicitly, hence programs are less verbose

D³N and Functional Programming II

- Integrated features from query language
  - Assurance as in logical programming
- Appropriate level of abstraction
  - Imperative languages closely specify the implementation details (how); declarative languages abstract too much (what)
  - Imperative – predictable result about performance
  - Declarative language – abstract away many implementation issues

Related Open Source Projects

- ***Boom*** https://trac.declarativity.net/
- ***Ciel*** http://www.cl.cam.ac.uk/netos/ciel/
- ***Apache Hadoop*** http://hadoop.apache.org/
- ***DryadLINQ*** http://research.microsoft.com/en-us/projects/dryadling/
- ***MapReduce Online*** http://code.google.com/p/hop/
- ***P2*** http://p2.berkeley.intel-research.net/
- ***Opis*** http://perso.eleves.bretagne.ens-cachan.fr/~dagand/opis/

36

6 main Topics

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

37

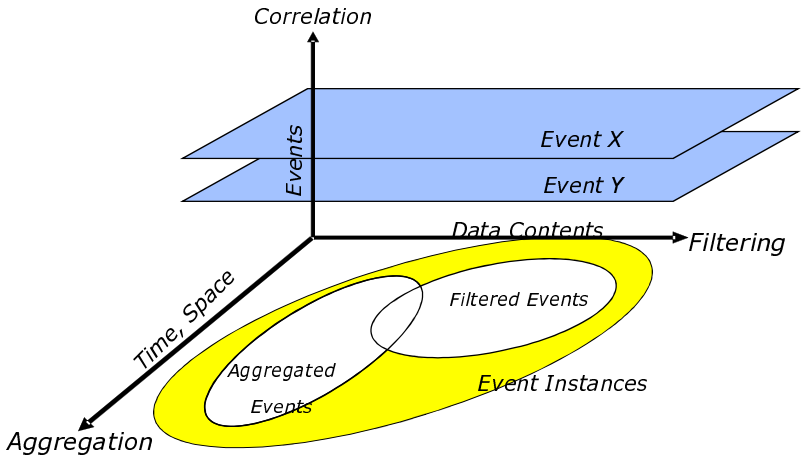
Stream Data Processing

- Stream Data Processing and Data/Query Model
  - Stream: infinite sequence of {tuple, timestamp} pairs
  - Continuous query is result of a continuous query is an unbounded stream, not a finite relation
- Data stream processing emerged from the database community (90's)
- Database systems and Data stream systems
  - Database
    - Mostly static data, ad-hoc one-time queries
    - Store and query
  - Data stream
    - Mostly transient data, continuous queries
- Stream data processing is analogue to Complex Event Processing
  → Composite events

38

Filtering, Aggregation, and Correlation

- Composite events represent complex patterns of activity from distributed system



39

Sensor Networks

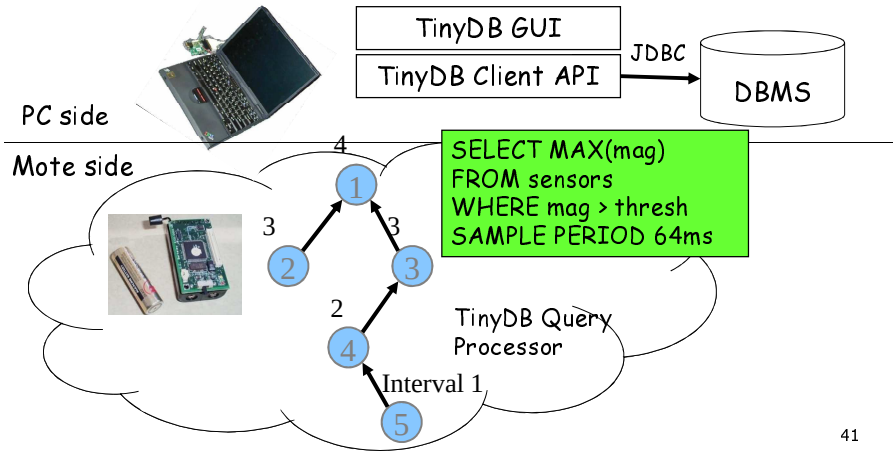
- Programming models
  - TinyOS
  - The need to move beyond node centric programming
- Macro-programming examples
  - State-space, EnviroTrack, Hood, Abstract region
  - Declarative/query: TinyDB
- Common interfaces

40

UNIVERSITY OF CAMBRIDGE Computer Laboratory

TinyDB

- Declarative SQL-like query interface
- Multiple concurrent queries and persistent storage,
- In-network, distributed query processing
- Fault mitigation: redundancy



41

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Related Open Source Projects

- ***Borealis*** http://www.cs.brown.edu/research/borealis/public/
- ***Cayuga*** http://www.cs.cornell.edu/bigreddata/cayuga/
- ***STREAMS*** http://infolab.stanford.edu/stream/
- ***TelegraphCQ*** http://telegraph.cs.berkeley.edu/telegraphcq/v0.2/
- ***DSN*** http://db.cs.berkeley.edu/dsn/
- ***TinyDB*** http://telegraph.cs.berkeley.edu/tinydb/software.html
- ***Yahoo scalable streaming query system*** http://www.globule.org/
- ***Flask*** http://www.eecs.harvard.edu/~mainland/projects/flask/

42

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

6 main Topics

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Delay Tolerant Networks

- Delay Tolerant Networks (DTN)
  - Network holds data
  - Path existing over time
  - Store and forward paradigm
- Weak and episodic connectivity - Eventual connectivity
- Non-Internet-like networks
  - Stochastic mobility
  - Periodic/predictable mobility
  - Exotic links
    - Deep space [40+ min RTT; episodic connectivity]
    - Underwater [acoustics: low capacity, high error rates & latencies]
- DTN routing takes place on a time-varying topology
  - Links come and go, sometimes predictably

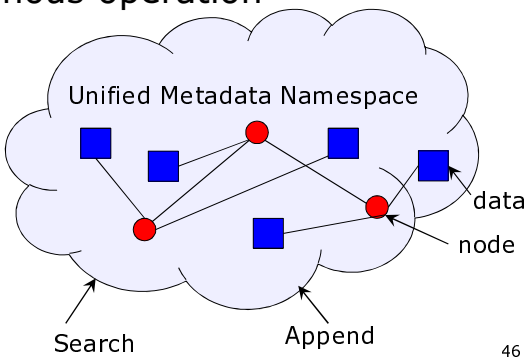
Prototypes: Architecture

- Providing Connectivity to Developing Countries: DakNet
- Vehicular Communications: DriveThru, DieselNet
- Wildlife Tracking: ZebraNet
- Haggle: Pocket Switched Networks, Social Networking
- DTNRG and the Bundle Protocol (RFC 5050)
  - Mostly an engineering approach to implement the InterPlaNetary Internet

45

Haggle Node Architecture

- Each node maintains a data store: its current view of global namespace
  - Persistence of search: delay tolerance and opportunism
- Semantics of publish/subscribe and an event-driven + asynchronous operation
- Multi-platform
  (written in C++ and C)
  - Windows mobile
  - Mac OS X, iPhone
  - Linux
  - Android



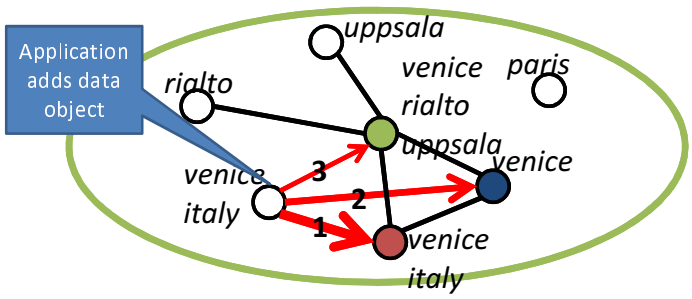
46

Search-based Networking

- Matching keywords against metadata
  - Non-boolean (e.g., not filtering)
  - Ranking, sorting out low-quality matches
  - Limits (e.g., '10 results per page')
- Finding data
  - Flood based request-response (e.g., Gnutella) does not work
    - Requires synchronous connectivity
    - Queries time out (non-persistency)
- Publish/Subscribe inspired

47

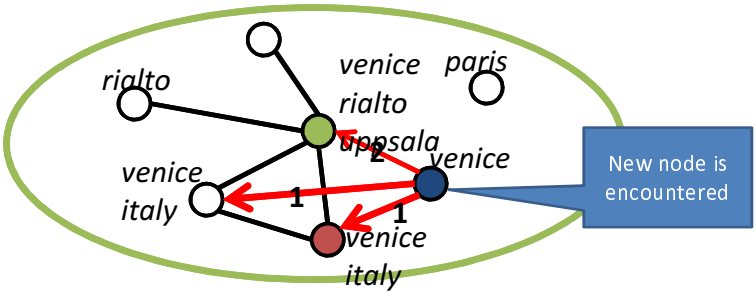
Relation Graph



- A node's view of the world
- Data object relations based on attributes
  - Weighting and ranking of relations

48

Relation Graph



- Graph updated as nodes are encountered
  - Common interests determine data exchange
- Node descriptions exchanged as any other data objects

49

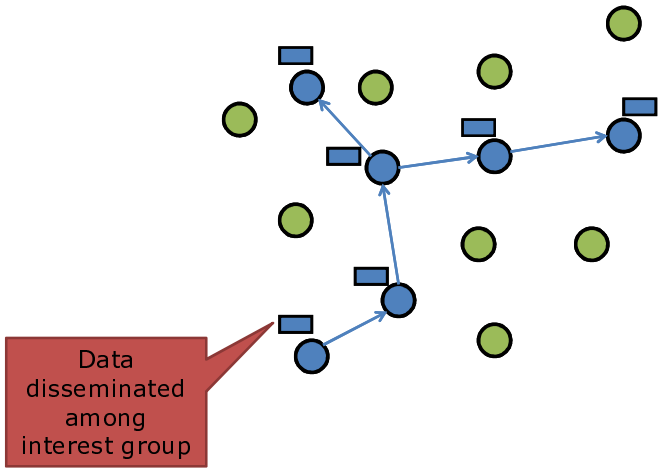
Summary of Haggle Primitives

- Resolution – the search aspect of Haggle
  - Find the "target nodes" in relation graph matching a data object, or vice versa
  - Data objects (and nodes) are ranked
- Interest forwarding
  - Give data object to neighbor with matching interests
- Delegate forwarding
  - Delegate data object to neighbor with higher forwarding metric but no interest in the data object

50

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

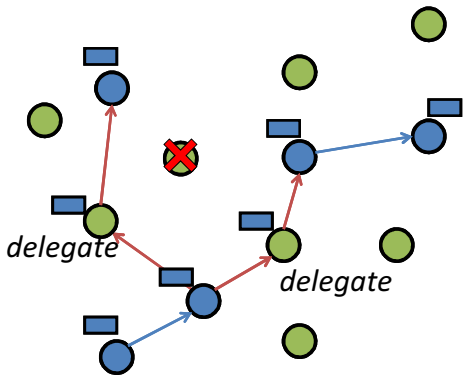
Interest Forwarding



51

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Delegate Forwarding



52

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Related Open Source Projects

- ***Haggle*** http://code.google.com/p/haggle/, http://www.haggelproject.org
- ***DTN at TKK Comnet*** http://www.netlab.tkk.fi/~jo/dtn/

53

UNIVERSITY OF CAMBRIDGE Computer Laboratory

6 main Topics

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

54

Network Structure

- Build network structure/topology for data dissemination (e.g. overlay construction) for improving performance or reliability
  - What context should be used for building a topology?
  - How to decide next hop (e.g. k random selection)?
- With given network graph/topology, how does data diffuse?
  - Data flow in network graph
  - Based on node capacity
- Understanding graph in networking

55

Example: Opportunistic Networks

- Opportunistic Contacts
- 1st effort: Epidemic Routing to deal with lack of knowledge
  - Minimum delay IF infinite buffer/bandwidth
  - Prohibitive resource usage
- 2nd effort: How to achieve epidemic routing delays with much less overhead?
- One answer: Smarter routing schemes
  - Controlled replication
  - Utility-based forwarding
  - Using logical backbone network

56

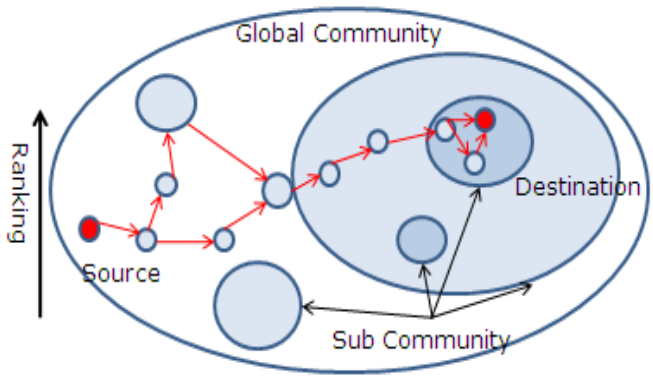
BUBBLE RAP Forwarding

- Optimisation of Epidemic Forwarding
  - Epidemic forwarding - highly robust against disconnection, mobility, and node failures; simple, decentralised, and fast
  - Control Flooding is necessary (e.g. Location, Count-base, Timer, History)
  - Exploit contextual information
- Use of Social Structure (e.g. Topology)
- Social hubs (e.g. celebrities and postman) as betweenness centrality and combining community structure for improved routing efficiency

57

BUBBLE RAP Forwarding

- LABEL Community based
- RANK Centrality based: Global and Local ranking of popularity
- BUBBLE RAP Combination of centrality and community



58

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Summary: 6 Faces of DCN

1. Content-Based Networking (CBN) and Content Distribution Networks (CDN)
2. Content-Centric Networking (CCN) and Named Data Networking (NDN)
3. Programming in Data Centric Environment
4. Stream Data Processing and Data/Query Model
5. Delay Tolerant Networks (DTN)
6. Network Structure/Characteristics and Contexts

59