UNIVERSITY OF CAMBRIDGE

Design and Realization of an Information-centric Networking Architecture

Dirk Trossen, Computer Laboratory

Outline

- WHY are we actually doing ICN?
- WHAT are the main principles?
- HOW could we do it?
  - one example for a core function
  - Prototype
- Demo

We All Know About Video: Staggering Numbers

- Over 4 billion hrs of videos watched on YouTube every month
  - 72 hrs uploaded on YouTube every minute
  - 70% of traffic from outside US
- The 2012 Olympics broke all records
  - BBC delivered 2.8 petabytes on its busiest day, 700Gb/s during the B. Wiggins' gold
- 74 mins average BBC iPlayer TV usage per week
  - 1.6 mio daily iPlayer viewers in July 2011
- …in all this, mobile usage just started to take off!
  - YouTube mobile traffic tripled in 2011

…With Staggering Forecasts (Cisco)

- Annual global IP traffic will reach the zettabyte threshold by 2015
- The average smartphone will generate 1.3 GB of traffic per month in 2015 (26x)
- In 2015, there will be 6 million Internet households worldwide generating over a terabyte per month in traffic
- By 2012 Internet video will account for over 50 percent of consumer Internet traffic

SO WHAT: The Internet Has Always Been About Information – And It Copes Well With It!

That is correct… (to a point to be discussed)

BUT: Economics have changed the possible starting points for a design

- Computing and storage resources are NOT scarce anymore; this led to an almost ubiquitous availability of processing and memory
- Information availability has changed attitude of users
  - WHAT is primary, WHO and WHERE mostly secondary! Information is often not locked anymore behind portals

⇒ **Location looses its meaning!**

⇒ **There is desire to fully optimize usage of resources (wherever they are)**

Hypothesis

*A systems approach that operates on **graphs** of **information** with a **late** (as late as possible) binding to a location at which the **computation** over this graph is going to happen, enables the full potential for **optimization**!*

This systems approach requires to marry information & computation (and with it storage) into a single design approach for any resulting distributed system

What Are The Promises of this?

- More Resilient and robust
- More Flexible & more Efficient, possibly greener
- Better aligning interests (e.g., economic, security, social)
  - What about more private (if wanted)?
- New services, such as in lifestyle management in pervasive healthcare
  - E.g., Android **AIRS** app for activity recording at https://play.google.com/store/apps/developer?id=Dirk+Trossen&hl=en_GB

TO BE CLEAR: We have NO conclusive evidence on any of this…**but early indications!**

Starting Point: Solving Problems in Distributed Systems

- One wants to solve a **problem**, each of which might require solving another problem
  - Examples:
    - Send data from A to B(s), involving fragmentation along the link(s)
    - Disseminate a video over a local network
- Problems involve "*a collection of **information** that*" an implementation *"can use to decide what to do"*, which is to implement a problem solution (*)

-> Computation in distributed systems is all about *information dissemination* (pertaining to a task at hand)

*REF: S. J. Russell, P. Norvig, "Artificial Intelligence: A Modern Approach", 2nd Edition, Pearson Educ., 1998

Starting from Desired System Properties…

- **Manipulation of (structured) information flows for computational purposes**
  - Expose service model and provide late binding (*WHAT->WHO*)
- **Modularity within a single computational problem**
  - Provide modular core functions (*enable optimization*)
- **Modularity across computational problems**
  - Provide rigorous but flexible layering (*deconstrain constraints**)

(*) REF: CHIANG, M., LOW, S. H., CALDERBANK, A. R., AND DOYLE, J. C. Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures. Proceedings of the IEEE (2007)

…Translated into Design Tenets…

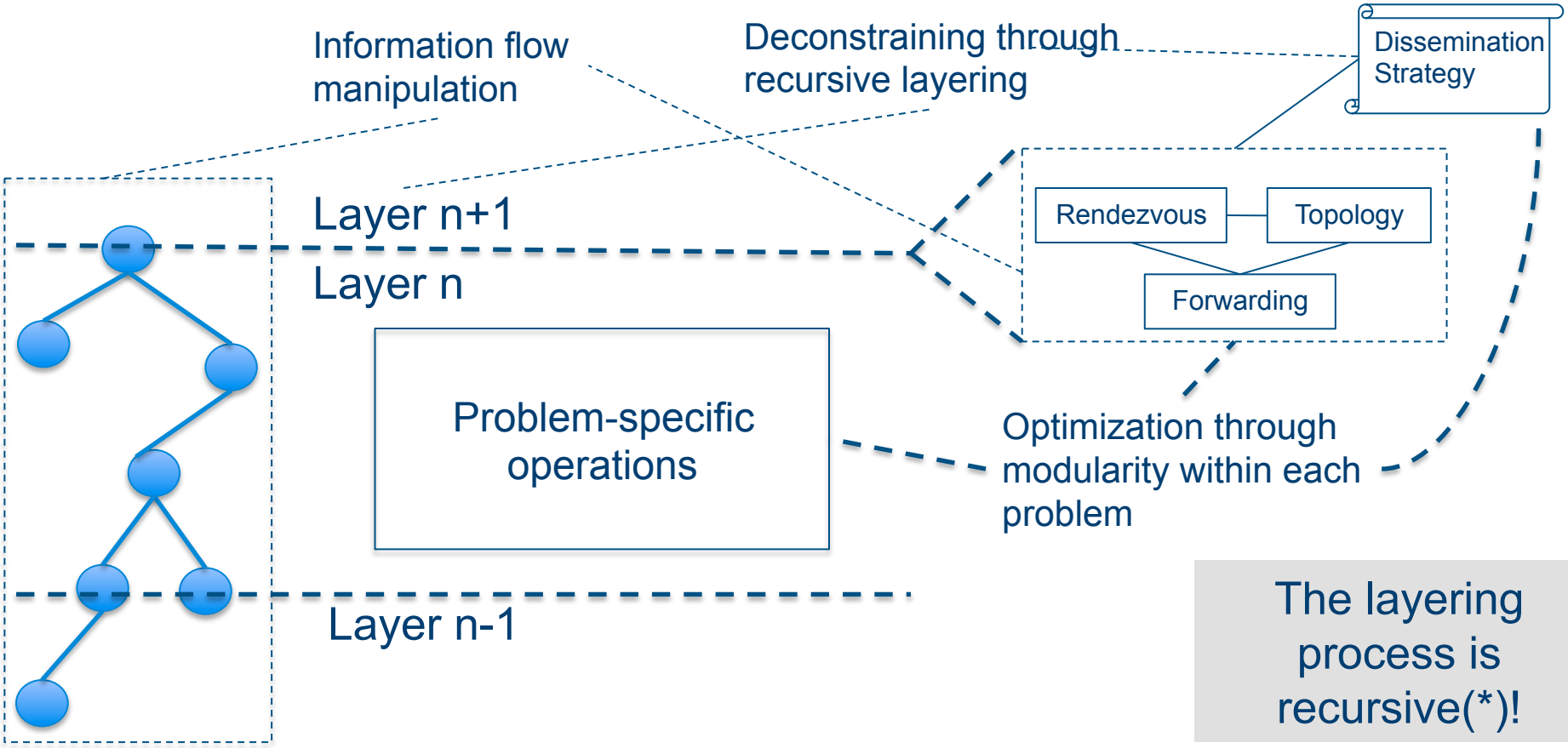
- Provide means for identifying individual information (items)
  - Can be done via labeling or naming (latter requires mapping mechanism)
- Provide means for scoping information
  - Allows for forming DAGs (directed acyclic graphs) of information
- Expose service model
  - Can be pub/sub
- Expose core functions
  - Rendezvous, topology management, and forwarding
- Common dissemination strategy per sub-structure of information
  - Define particulars of functional implementation and information governance

…With An E2E Principle…

The problem in question can be implemented through an assembly of sub-problem solutions, whose individual dissemination strategies are not in conflict with the ones set out by the problem in question.

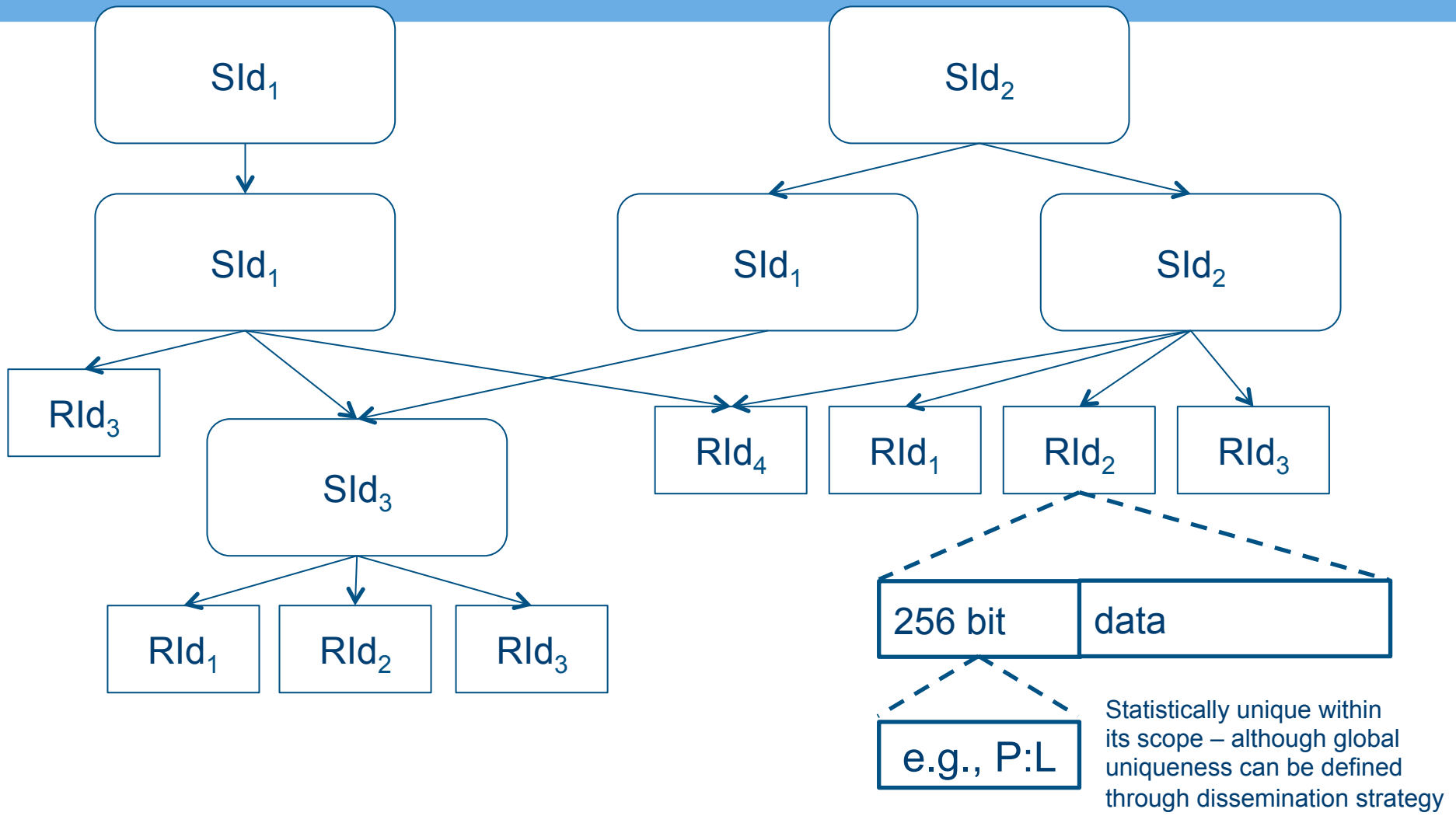
- Hence, problems are assembled to larger solutions by recursively applying the scoping tenet!
- Conflicts are avoided through design and re-design, e.g., via standards procedures!
  - Can extend this to runtime reconciliation!

…And Placed into a Layered Model



(*) REF: DAY, J. Patterns in Network Architecture - A Return to Fundamentals. Prentice Hall, 2008

Operating on Graphs of Information



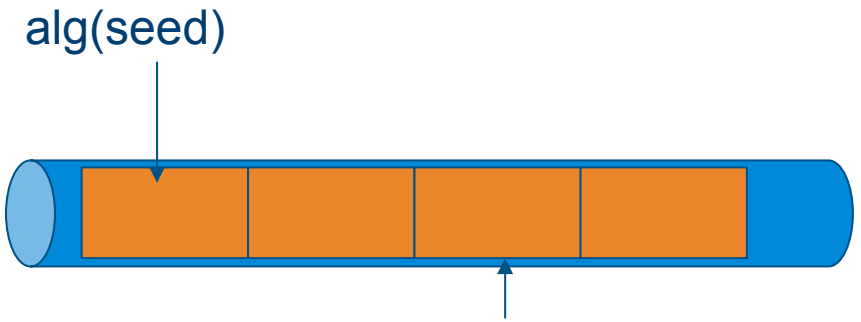
Information Semantics: Immutable vs. Mutable Items

- **Documents (immutable information)**
  - Each RId points to immutable data (e.g., document version)
  - Not well suited for real-time type of traffic
  - Each item is identifiable throughout the network
- **Channel (mutable information)**
  - Each RId points to channel of data (e.g., a video stream), i.e., the data is mutable
  - Well-suited for video type of traffic
  - Problems with caching though (since no individual video segments visible)

Information Semantics: Algorithmic Identification

Idea:

- Use an algorithm to tie together a set of data items
- Allow for data items to be addressed individually through algorithmically generated RIds
- Allow for addressing collection through algorithm (and its seed)
- **Examples**: reliable fragmentation, network coding, layered coding, …

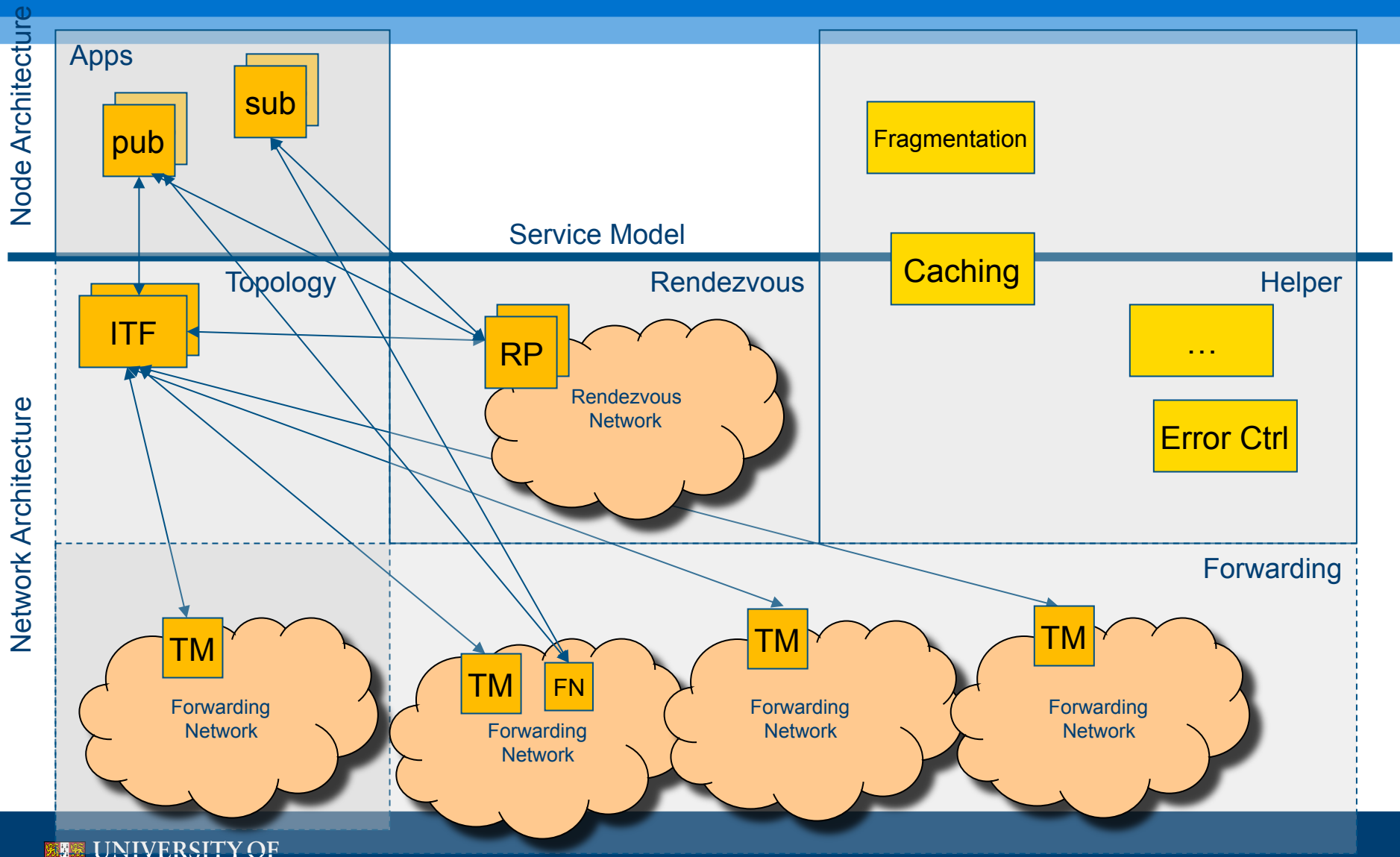


- Access 'channel' via *seed* RId, go to segment via *alg(seed)*
- Publish *alg* as metadata to seed

-> Channel implicitly visible to network, together with individual segment RIds, by virtue of *alg* as implicit channel Id, alg being app-specific

Coming Together in A Global Architecture

RP : Rendezvous point
ITF : Inter-domain topology formation
TM : Topology management
FN : Forwarding node



UNIVERSITY OF CAMBRIDGE

Example of One Core Function

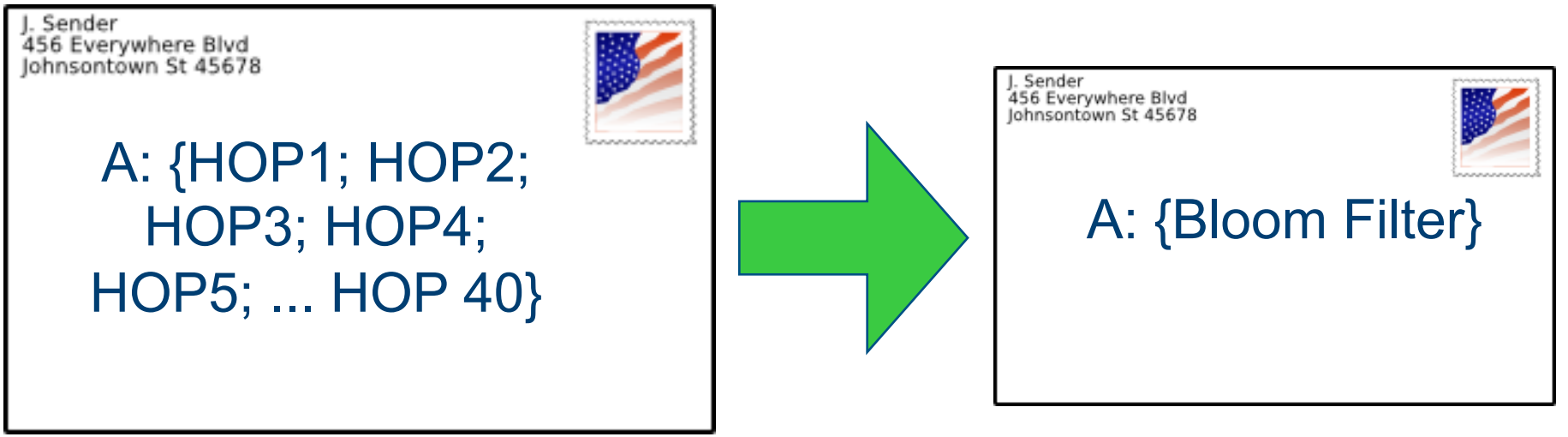
Forwarding with Built-in (Native) Multicast Capability

Motivation

Information is sent along a route of (intra-domain) hops in the Internet

-> Requires some form of minimal state in each hop

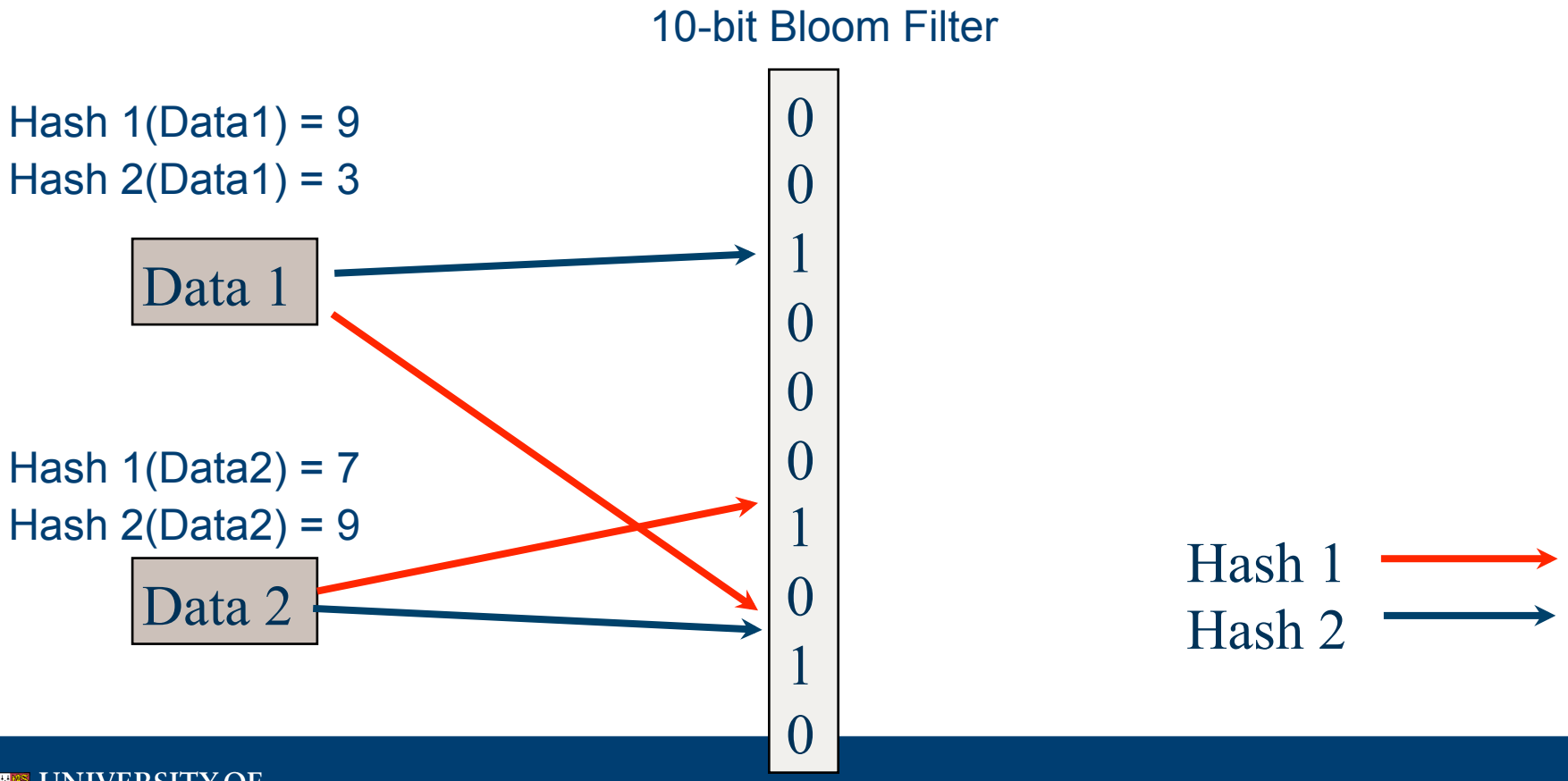
Question: What if we could include the state in the packet?



REF: P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, P. Nikander, LIPSIN: line speed publish/ subscribe inter-networking, ACM SIGCOMM 2009

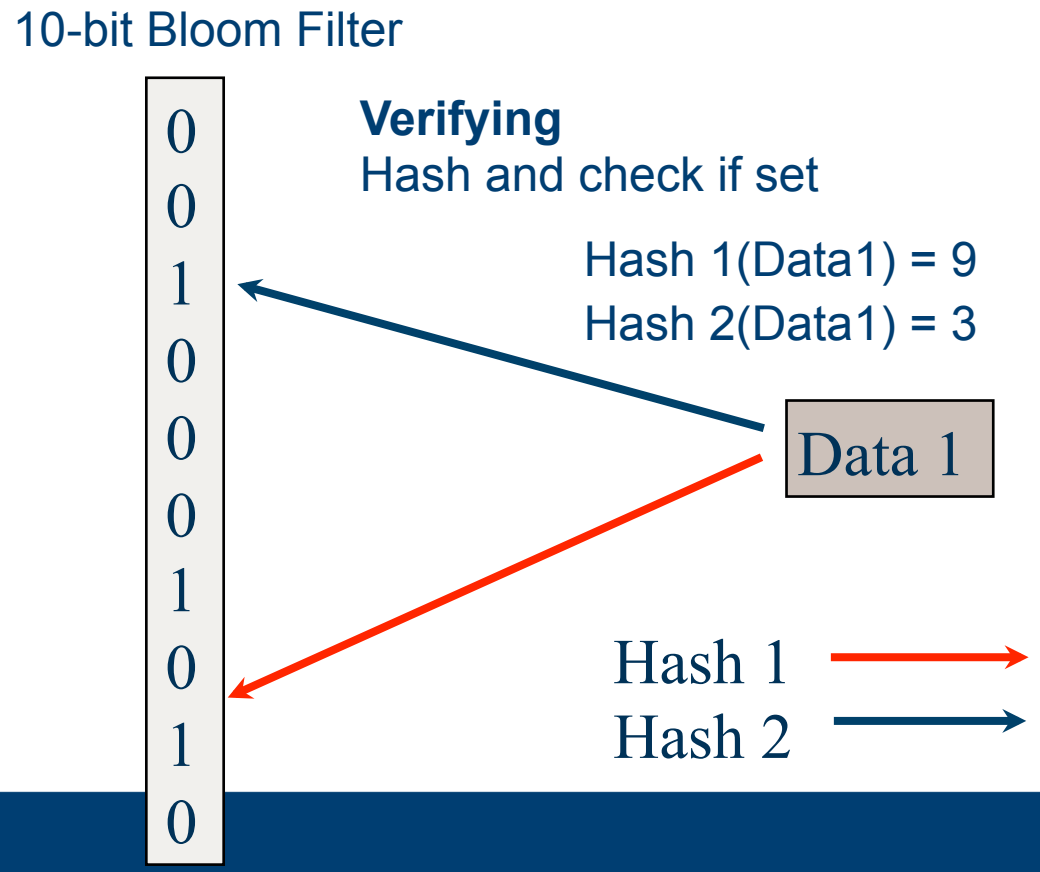
What are Bloom Filters?

- Inserting items
  - Hash the data n times, get index values, and set the bits



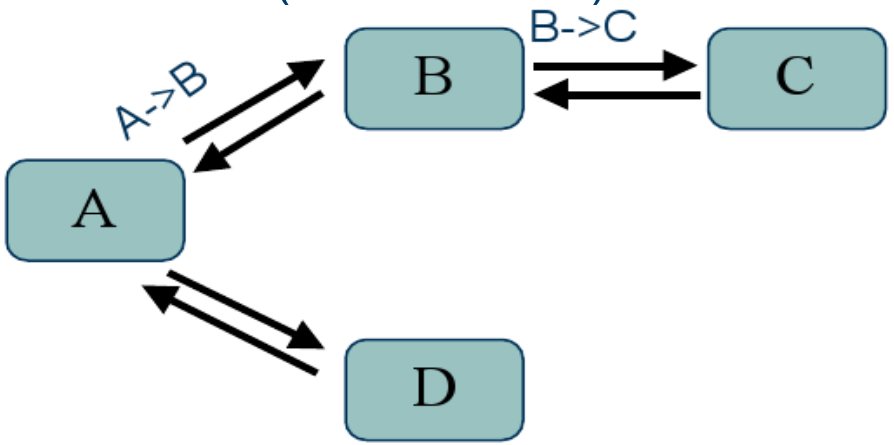
What are Bloom Filters?

- Test if “Data 1” has been inserted in the BF
  - All corresponding bits are set => positive response!



Idea: Line Speed Publish/Subscribe Inter-Network (LIPSIN)

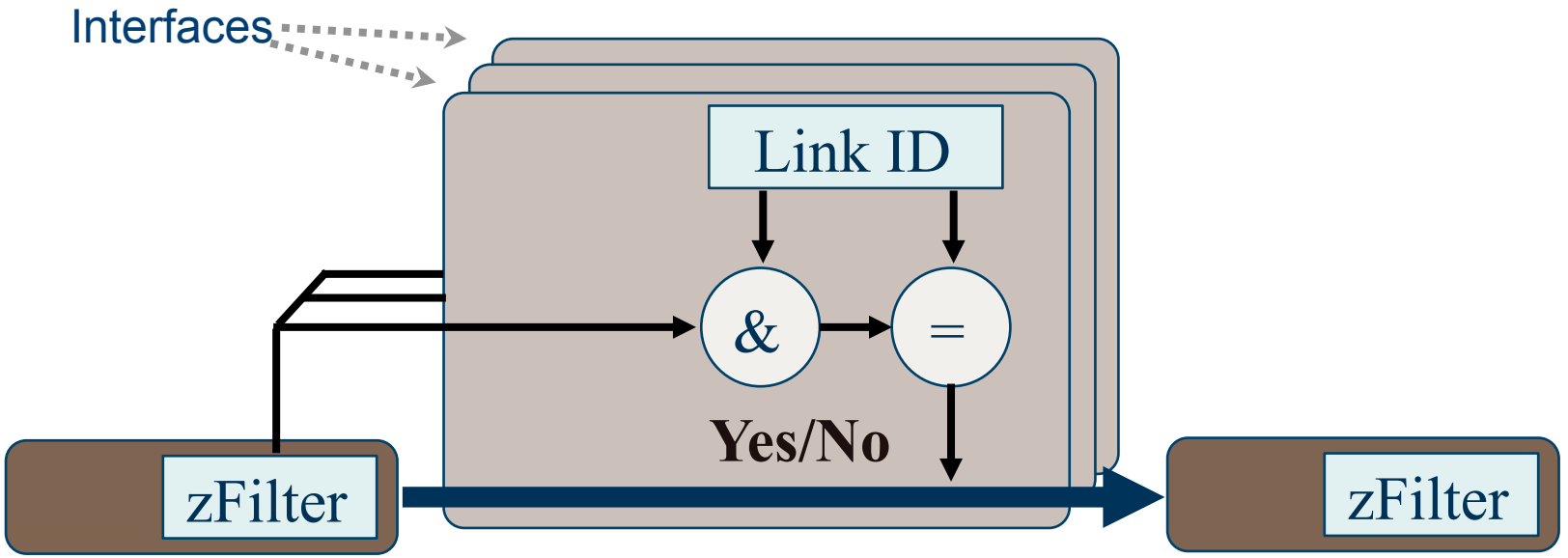
- Line speed forwarding with simplified logic
- Links are (domain-locally) named (**LId**) instead of nodes, therefore there is no equivalent to IP addresses
- Link identifiers are combined in a **bloom filter** (called zFilter) that defines the transit path



- **Advantages**
  - Very fast forwarding
  - No need for routing tables
  - Native multicast support

Forwarding Decision

- Forwarding decision based on binary AND and CMP
  - zFilter in the packet matched with all outgoing Link IDs
  - Multicasting: zFilter contains more than one outgoing links
- **Only state required in forwarders are the own LIds!**



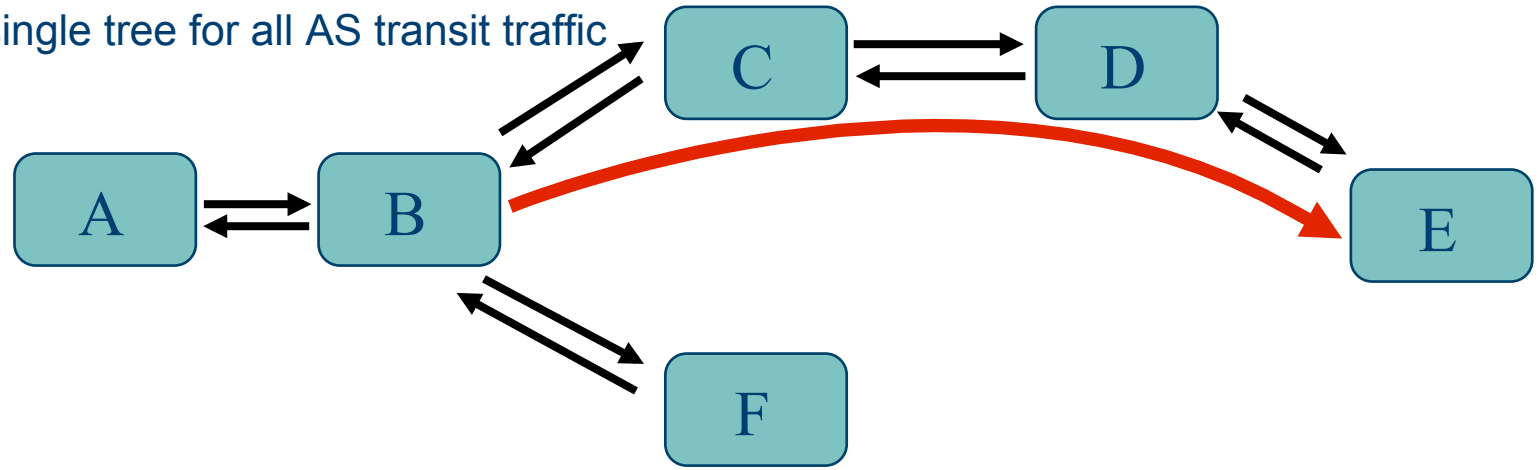
Problem: False Positives in Forwarding

False positives occur when test is positive in a given node despite non-hashed LId (probability for consecutive false positives is multiplicative!)

- Increase with number of links in a domain (since more data is hashed into constant length Bloom filter)
- Two immediate solutions:
  - Use Link Identity Tags: tag a single link with N names instead of one, then pick resulting Bloom filter with lowest false positive probability
  - Virtual trees: fold “popular” sub-trees into single virtual link, i.e., decrease number of LIds to be used

Virtual trees

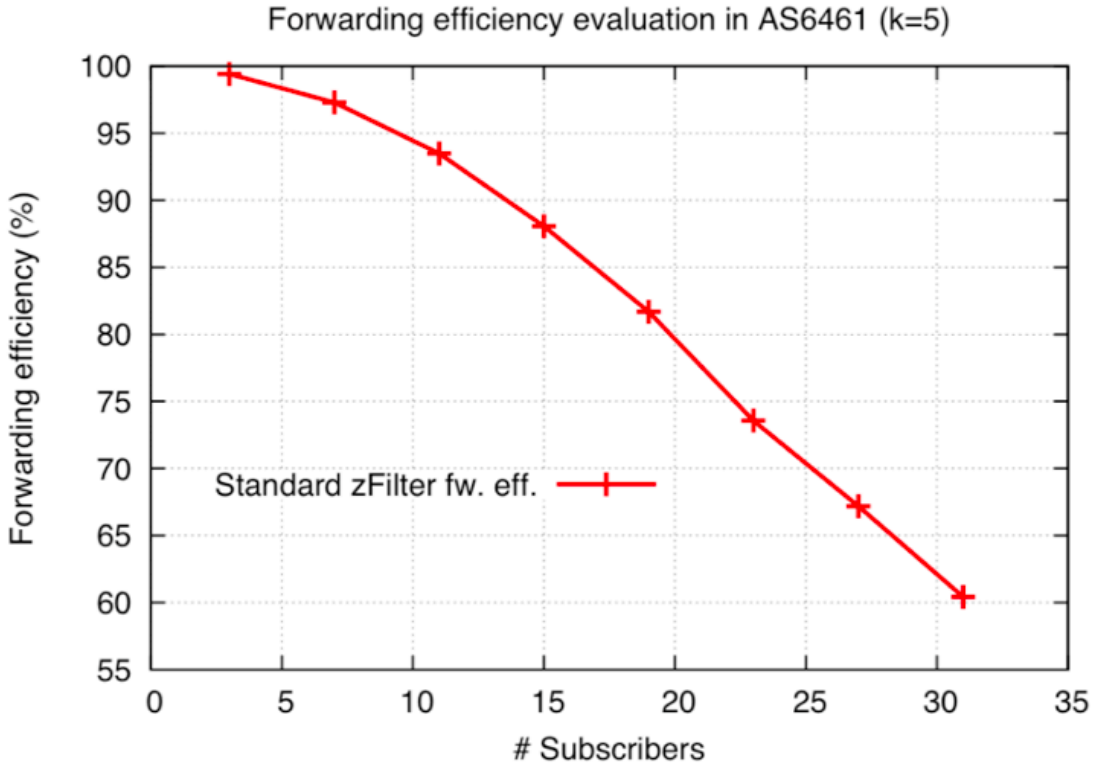
- Popular paths can be merged into virtual trees
  - A single Link ID for the tree
  - PRO: Increase scalability due to decreased false positives, which could be combined with lower-layer optical techniques
  - CON: Additional state in the forwarding nodes
- A virtual tree is not bound to a certain publication
  - E.g. a single tree for all AS transit traffic



Forwarding Efficiency

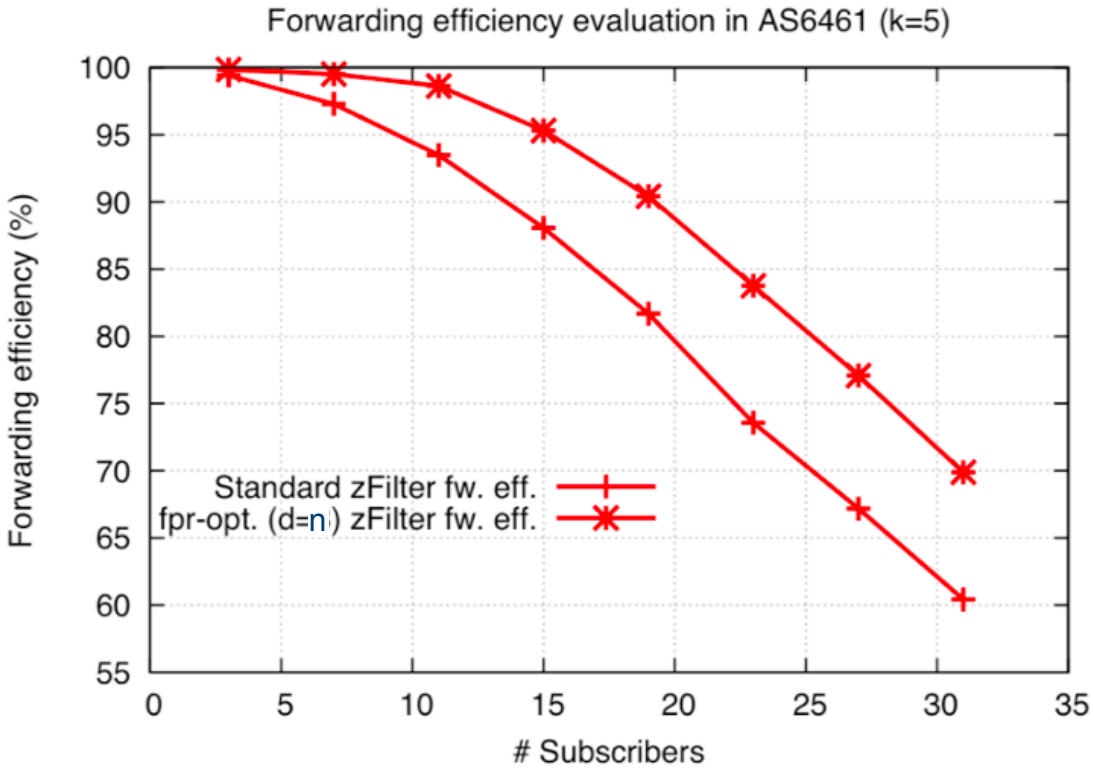
- Simulations with
  - Rocketfuel
  - SNDlib
- Forwarding efficiency with 20 subscribers
  - ~80%

  -> suited for MAN-size multicast groups



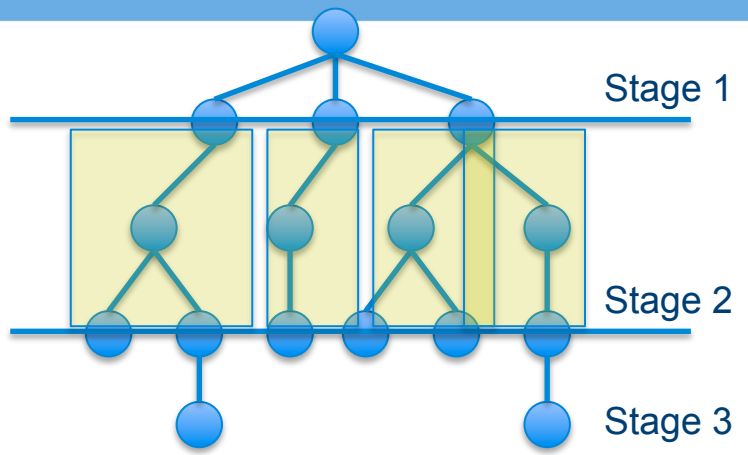
Forwarding Efficiency

- Simulations with
  - Rocketfuel
  - SNDlib
- Forwarding efficiency with 20 subscribers
  - ~80%
  - Can be optimized to 88% using extended mechanisms



Extending Scalability: Multi-stage BF Forwarding (1)

- Divide a delivery tree into stages
  - Each stage has its own individual trees
  - Operation performed at topology manager



- Provide single BF forwarding identifier per stage

- Concatenate all stage into variable size header

- Perform BF-based forwarding at each stage
- Remove appropriate BF after each stage (reduces overhead ratio along the tree)

Extending Scalability: Multi-stage BF Forwarding (2)

- Advantages
  - Arbitrary tree size (limit only when restricting size for variable length header)
  - Tradeoff between false positive rate and header size possible
  - **Lends itself to inter-domain as well as intra-domain forwarding**
- Disadvantages
  - Higher complexity in forwarding (but only at the stage boundaries)
  - Possible higher overhead due to variable length (although 'engineering' length of LIPSIN header nullifies any possible advantage)

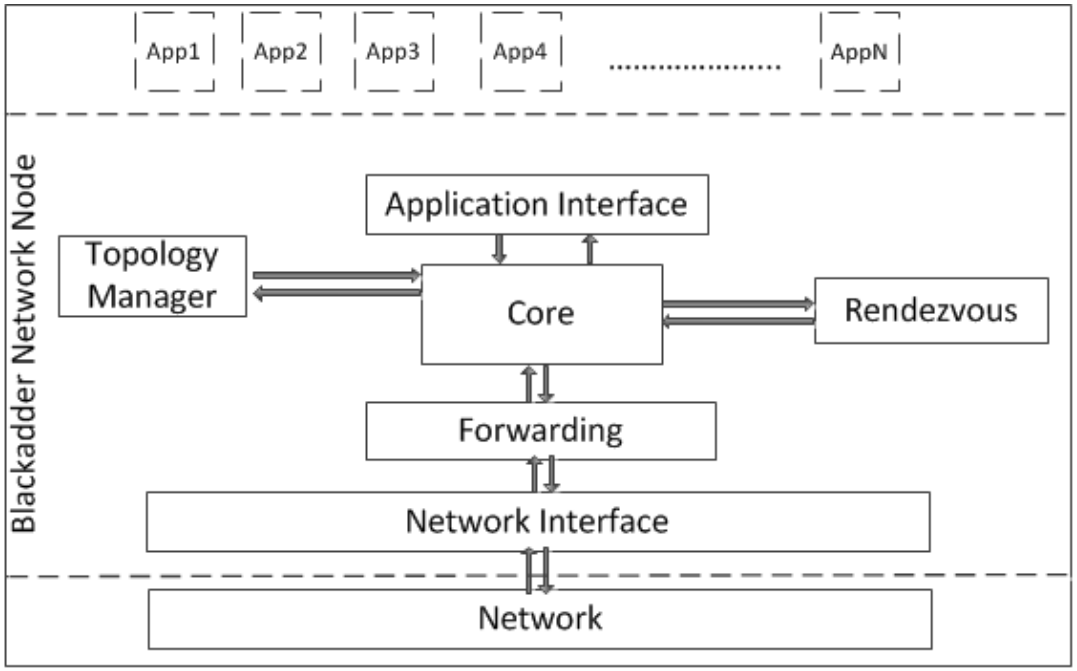
REF: J. Tapolcai, A. Gulyas, Z. Heszberger, J. Biro, P. Babarczi, D. Trossen, "Stateless Multi-Stage Dissemination of Information: Source Routing Revisited", to appear in Globecom, 2012

UNIVERSITY OF CAMBRIDGE

Prototype, Deployment & Some Results

Making it work and run - where have we gotten to?

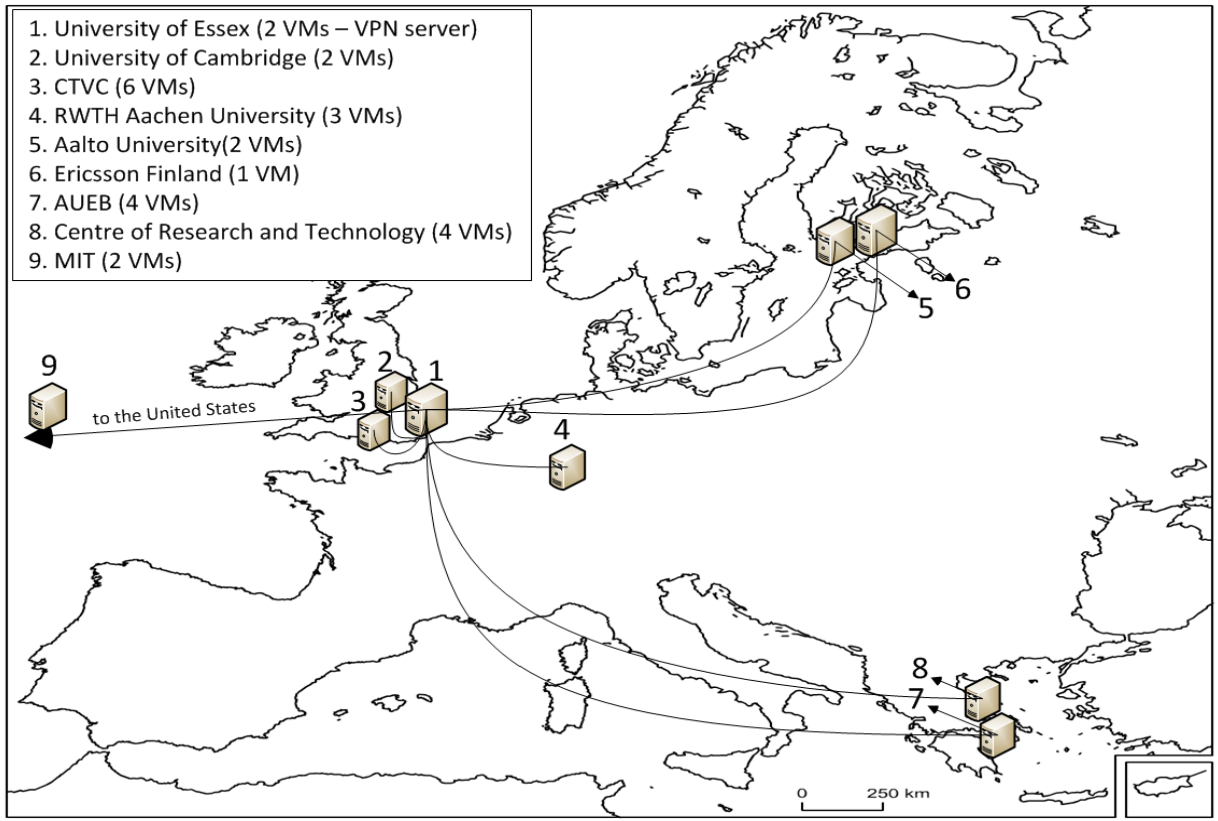
Our Prototype: Blackadder



- Implements design tenets
- Based on **Click** router platform (*)
  - Easy user/kernel space support
  - Easy porting onto other OSes
  - Easy plugging into ns-3
- Available at https://github.com/fp7-pursuit/blackadder
- Domain-local throughput reaches 1GB/s

(*) REF: E. Kohler, R. Morris, B. Chen, J. Jannotti, F. Kaashoek. The click modular router. ACM Trans. Comput. Syst. 18, 3 (August 2000), 263-297.

Our Test Beds

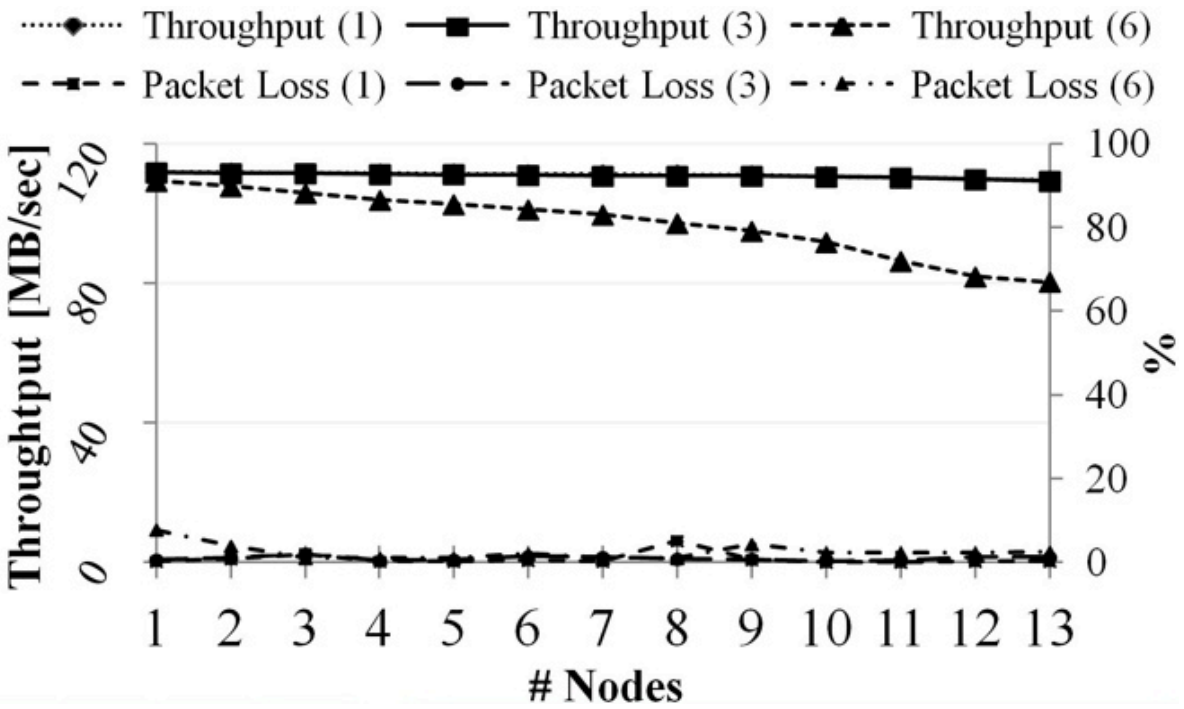


- 9 international sites
  - 26 machines with +40 on-demand ones
  - tunneled via openVPN with configurable topologies

Also available:

- Dedicated 1GB/s test bed with 15 nodes
- Planetlab (>100 nodes)
- Emulated topologies via ns-3

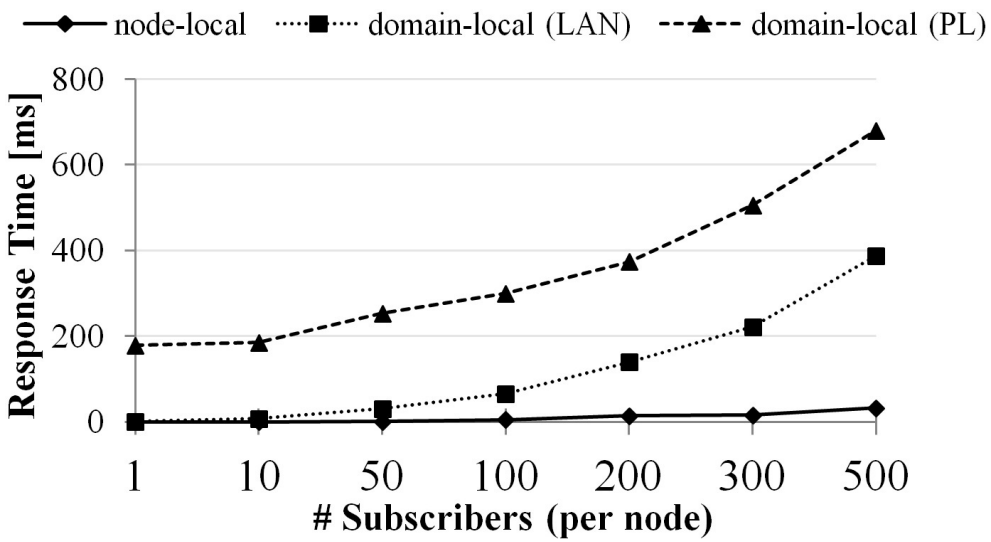
Experimental Evaluation: Fast Path



Forwarding efficiency

- 15 in a chain
- Multicasting (when nodes is sub)
- ~line speed even when 3 subs per node for 13 nodes
- Degradation when 6 pubs and more due to local copies

Experimental Evaluation: Slow Path



100.000 adverts under single scope

- Subscribers subscribe to random item, wait until receive it and reiterate (500 times)

-> worst case for slow path (ignores any possible optimizations due to domain-local rendezvous or mutable semantics)

Node-local

- No net delays
- No TM
- 20ms for 500 processes

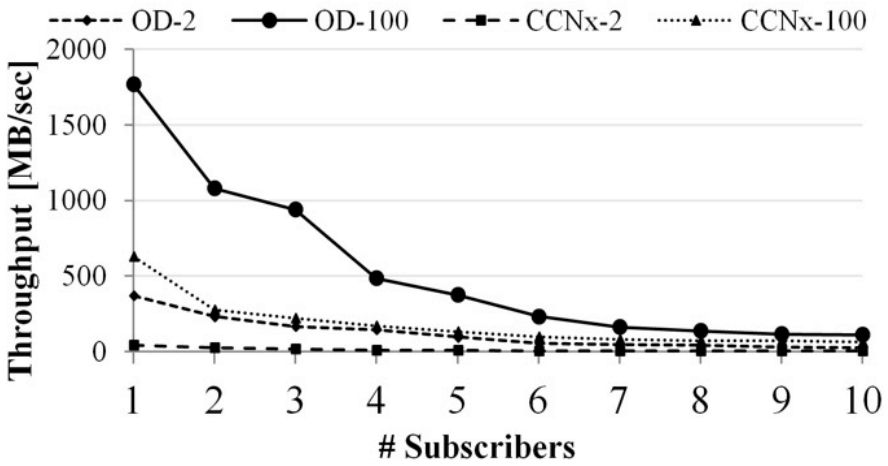
Domain-local (Gbit LAN)

- Centralized TM
- ~400 ms for 500 processes per node (7000 subscribers)

Domain-local (PlanetLab)

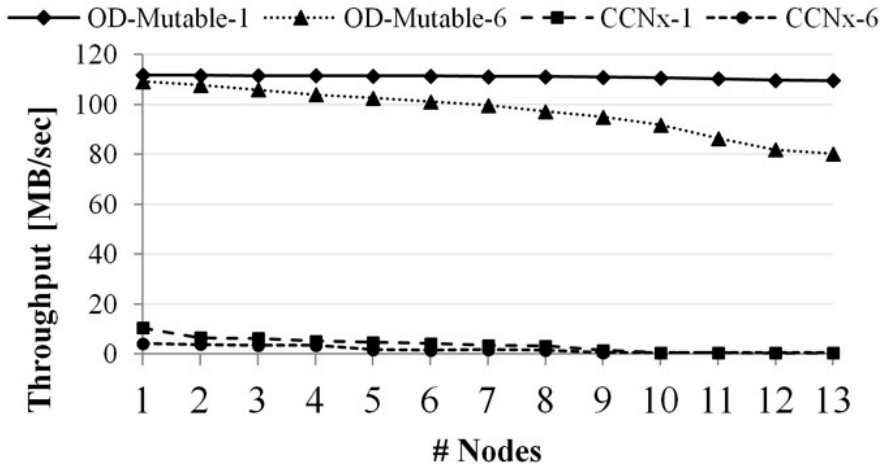
- Large delays
- ~200ms for 1 sub per node (73 in total)
- ~680ms for 36,500 subs

Comparison with CCN(x)



Node-local (payload size: 2 & 100KB)

- CCNx application expresses interest for 10000 items (/content/segmentNumber)
- CCNx replays all data from the local content store (to avoid signing penalty)



Domain-local (13 nodes in 1GB/s)

- Realize simple window-based flow control
- CCNx replays all data from the first hop cache (to avoid signing penalty)
  - Throughput falls to 170KB/sec if signing each packet on the fly!

Significantly Larger Throughput!

What is the Take-Away Here?

- Information-centric networking is **NOT** about disseminating information *because the Internet is no good at it!*

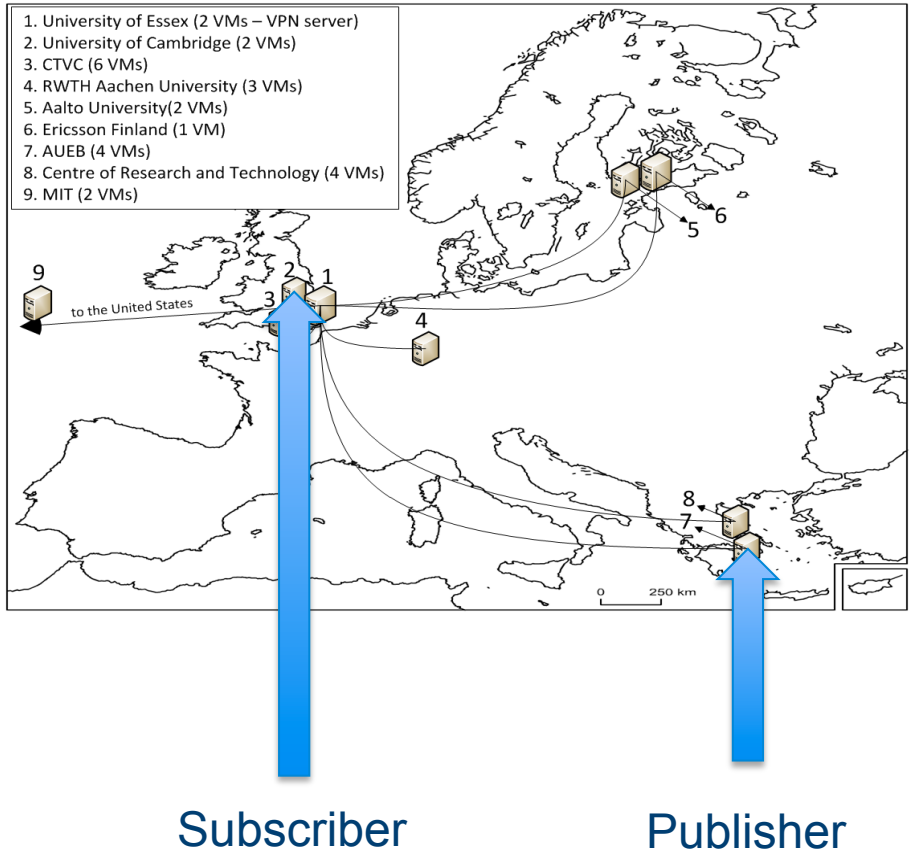
- Information-centric networking is about utilizing the entire design space provided by information (& storage) as well as computation
  - Aided by technological developments that made computation (and storage) ubiquitously available

- To get there, we need to re-think how we design/build systems
  - We have first results, working prototypes, and a growing test bed

What is the Take-Away Here?

- Main principles of our work
  - Center around **information**
  - Allow for **flexible** graph-based information structures with **semantic-free** labels as identifiers
  - Clearly **separate** functions for *finding* information, *creating* appropriate delivery relations, *delivering*
- Node design for making it all work
  - **Running & performing!**
- Test beds for making it all real

Demo: Simple Video Delivery Scenario



- Information scope represents category of video
  - Scope contains particular videos as information items
  - Mutable semantics (video is channel here)
- Publisher sits in Athens (Greece)
- My laptop is the subscriber, joining test bed as Cambridge node
- **All this over the public Internet!**

My Closing Words (for now)

- Please come to http://www.fp7-pursuit.eu to understand more!
- Please download Blackadder from https://github.com/fp7-pursuit/blackadder to play with it!
- Play with our activity recording app AIRS at https://play.google.com/store/apps/developer?id=Dirk+Trossen&hl=en_GB
- Please send me an email at dirk.trossen@cl.cam.ac.uk with any questions or suggestions!