Information-centric Internetworking

A Few Insights

Computer Laboratory

Overview

- Background and motivation
- Architectural foundations
- Example: intra-domain forwarding

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

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!

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

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

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
- Provide means for scoping information
  - Allows for forming DAGs (directed acyclic graphs)
- 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

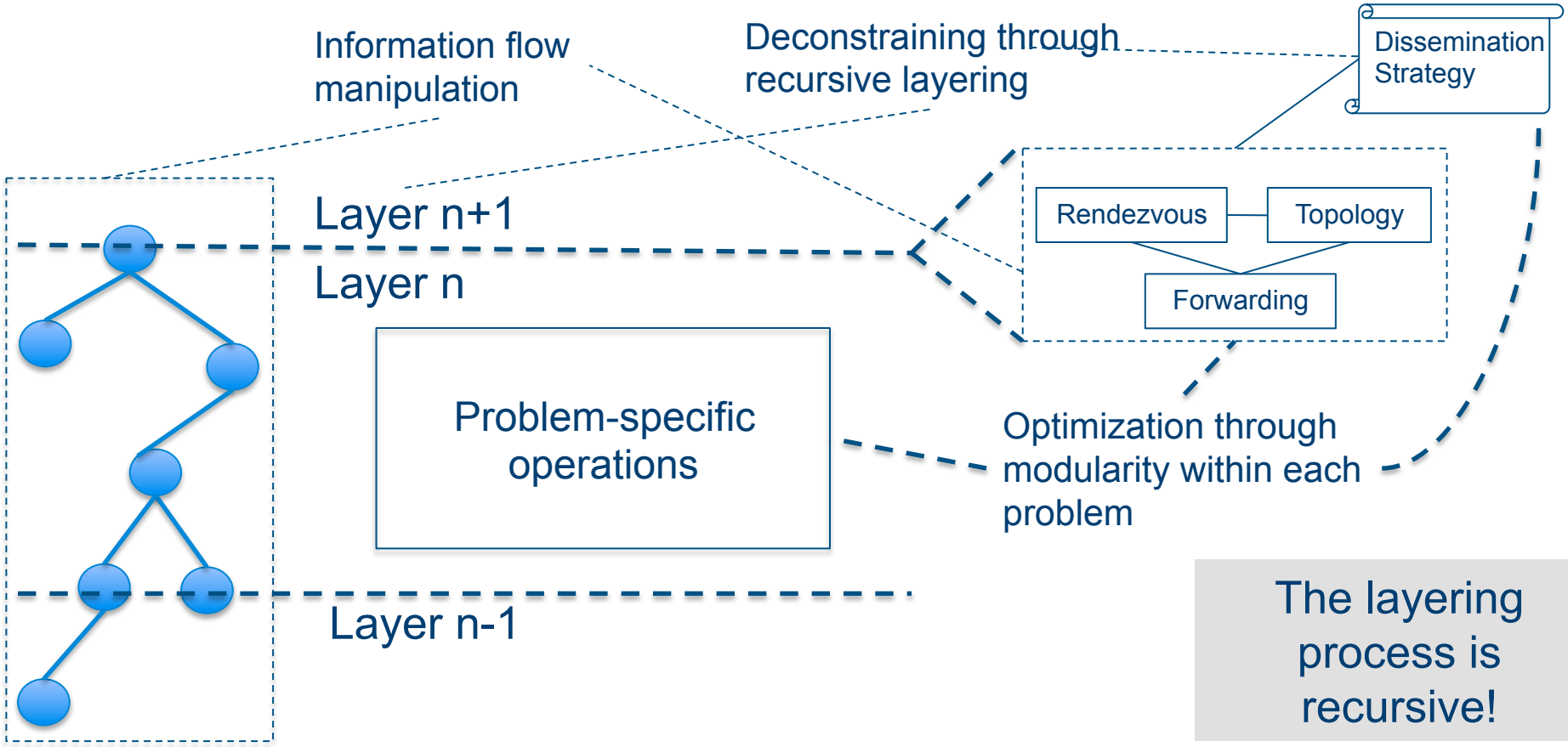
…With An E2E Principle…

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 invariant of the functional model!
- Conflicts are avoided through design and re-design, e.g., via standards procedures!
- Can extend this to runtime reconciliation!

NOTE: I leave it as a thought exercise to relate this to the IP E2E principle!

…And Placed into a Layered Model

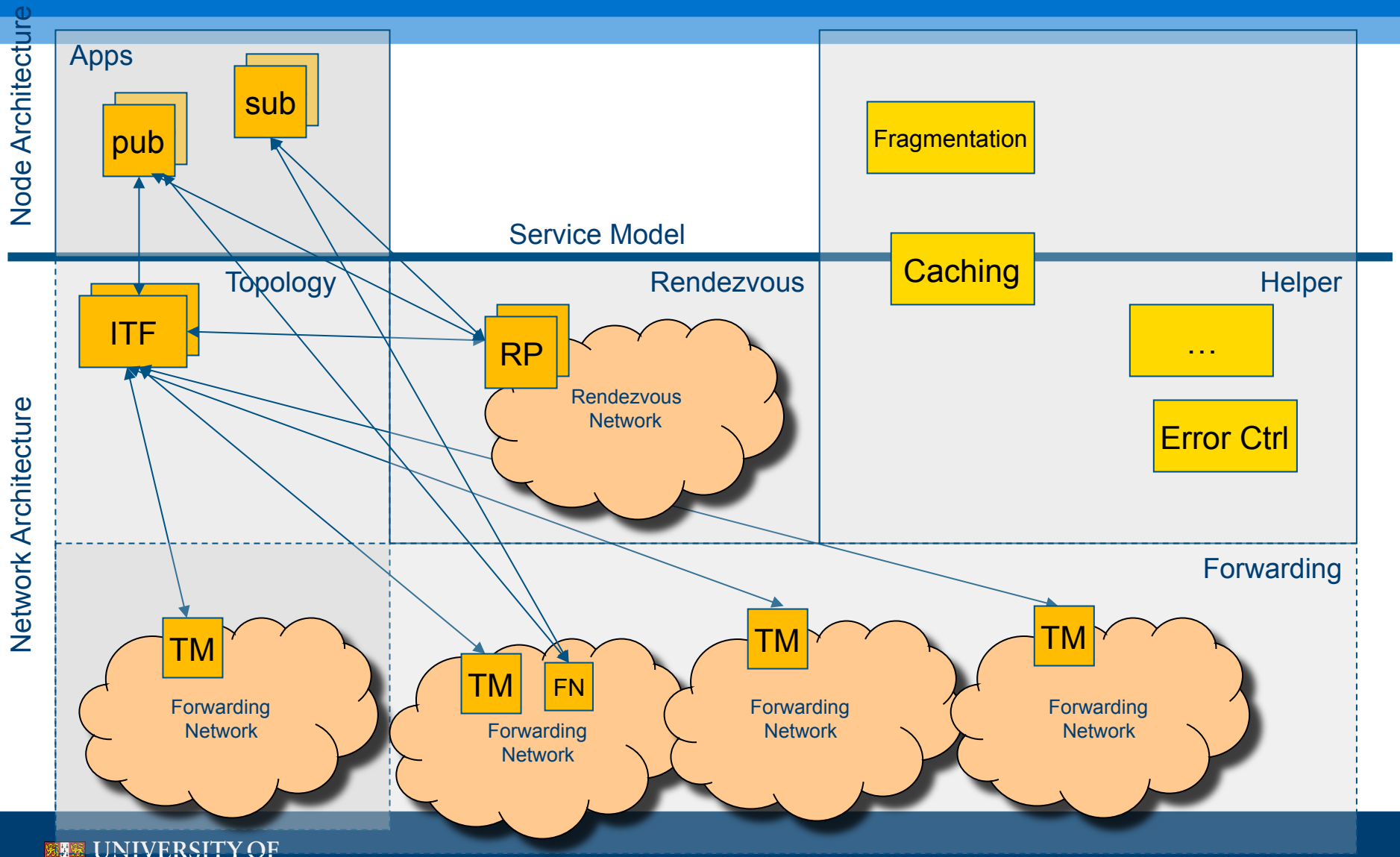


The layering process is recursive!

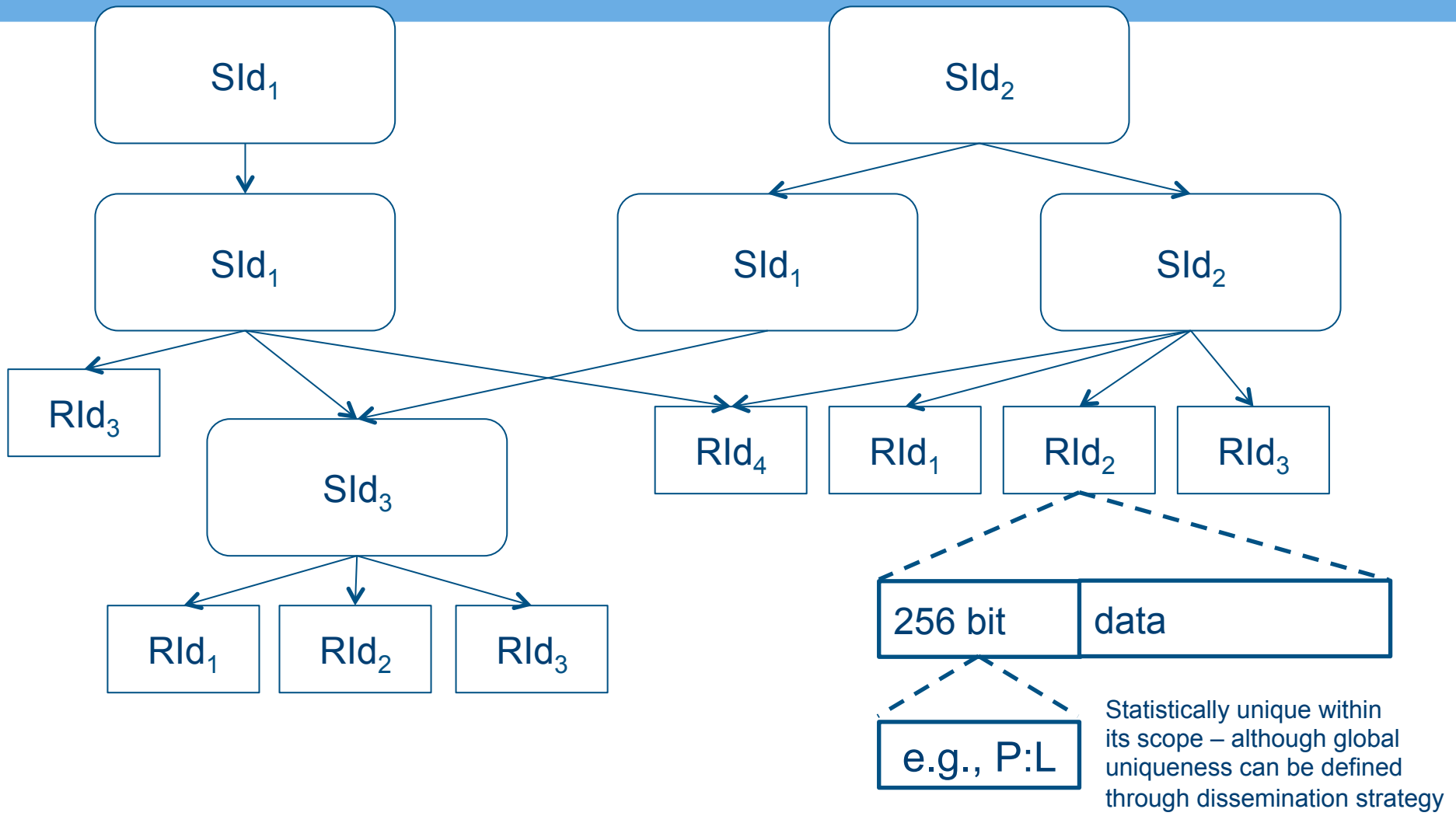
REF: DAY, J. Patterns in Network Architecture - A Return to Fundamentals. Prentice Hall, 2008

…Coming Together in A Global Architecture

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



Operating on Graphs of Information



Information Semantics: Immutable vs. Mutable Items

- Documents
  - 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
  - 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)

UNIVERSITY OF CAMBRIDGE

Example of One Core Function

Forwarding with Built-in (Native) Multicast Capability

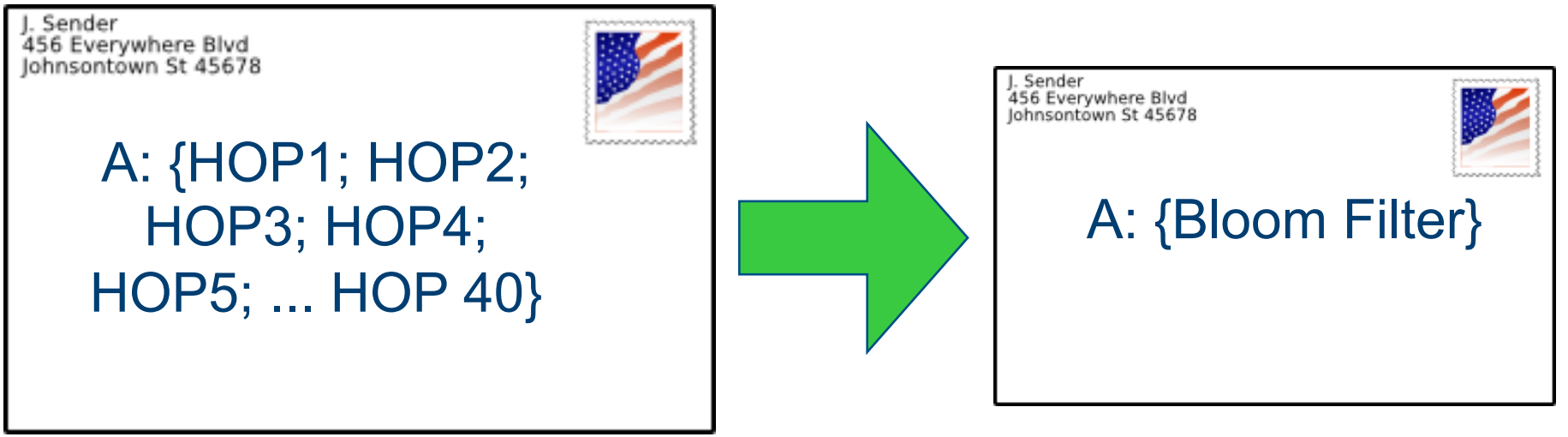
Motivation

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

-> Requires some form of minimal state in each hop

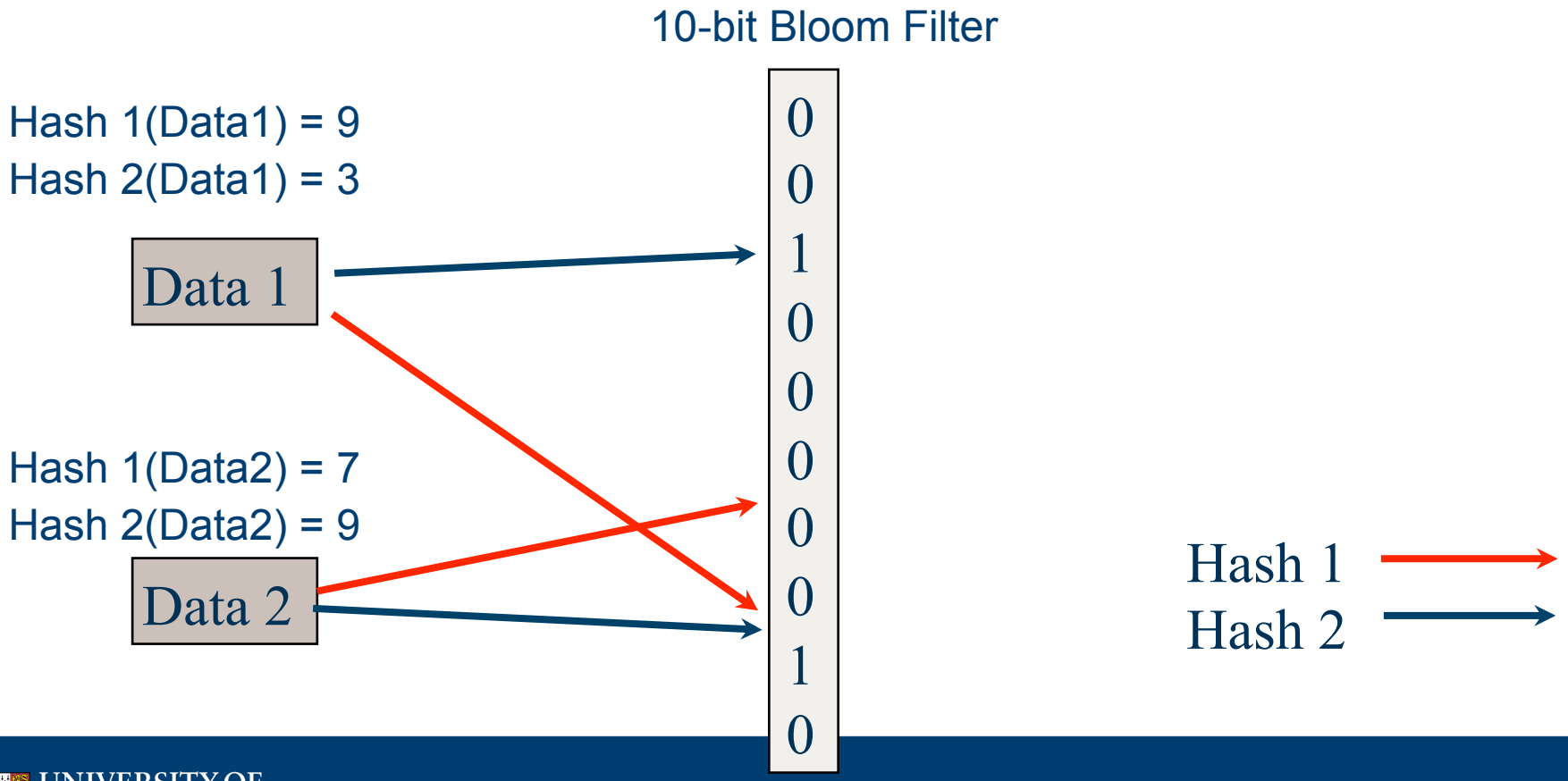
- If forwarding on names, limiting this state is hard/impossible

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



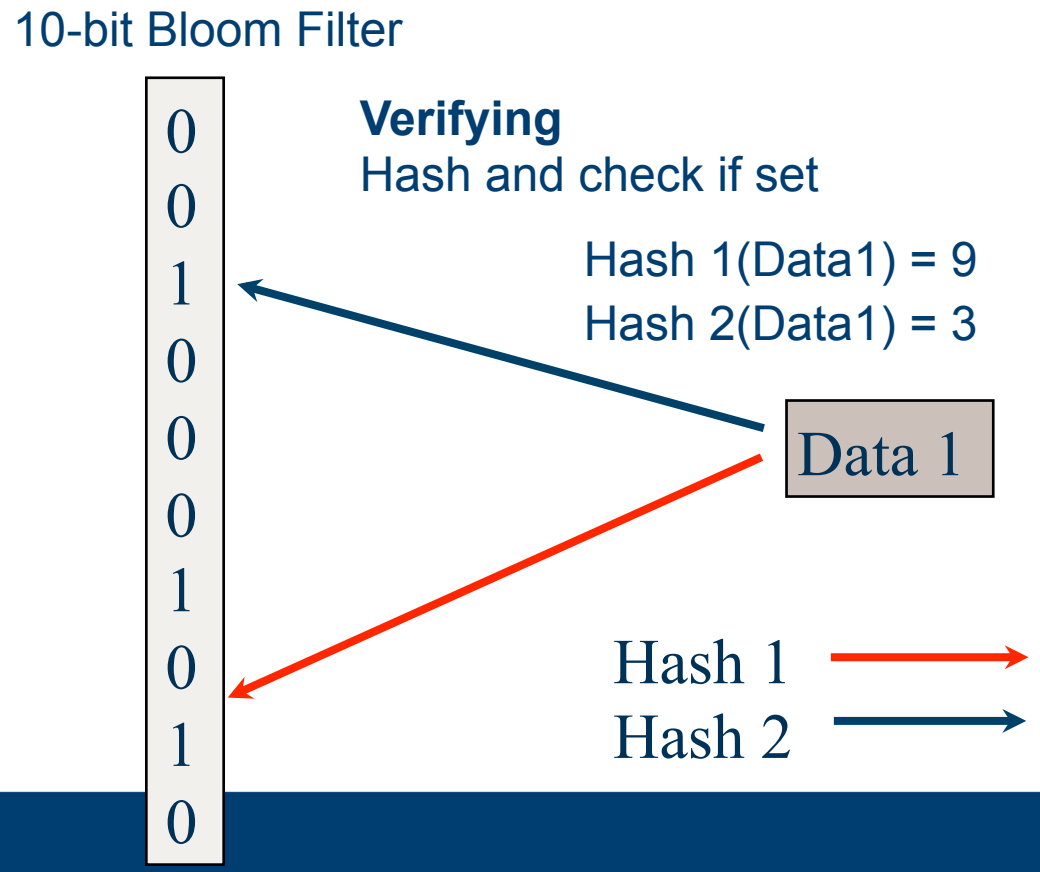
What are Bloom Filters?

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



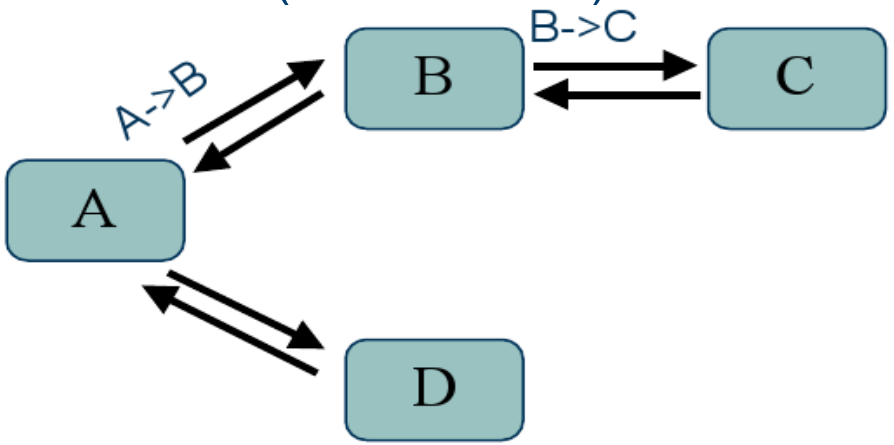
What are Bloom Filters?

- 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)

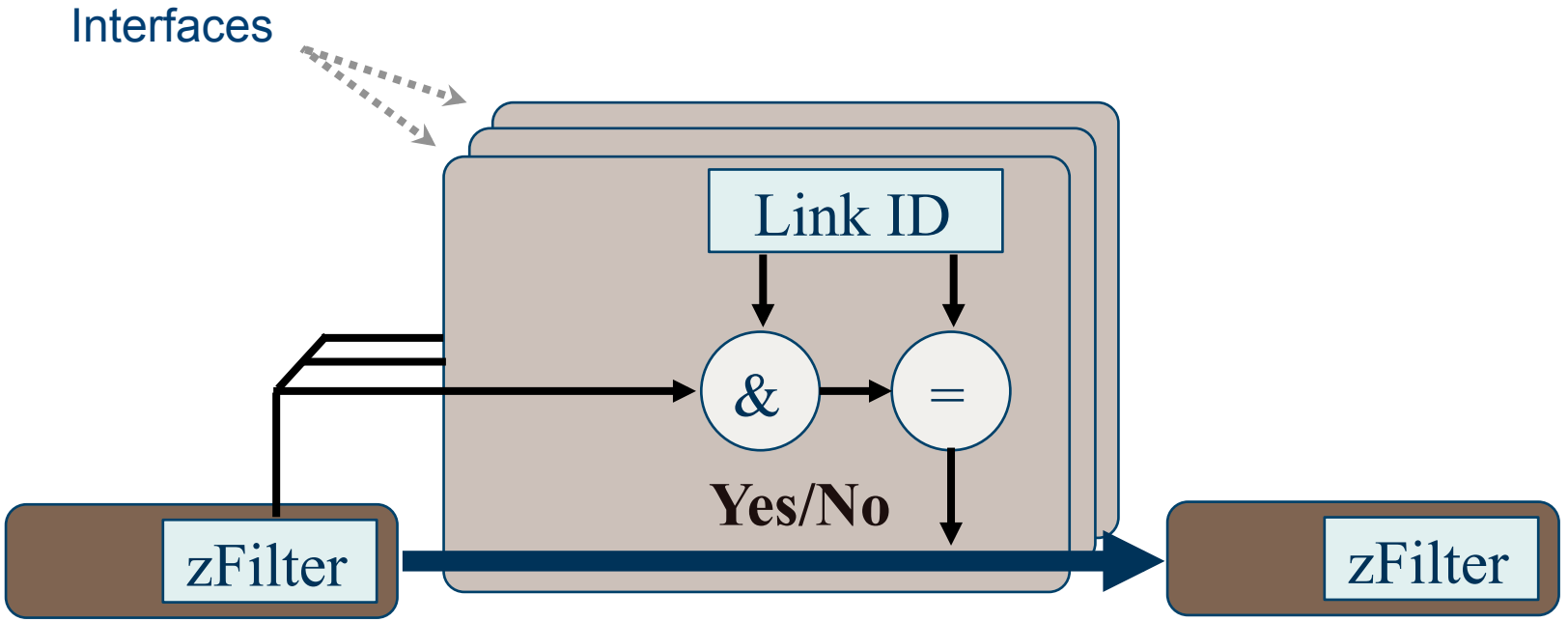
- Line speed forwarding with simplified logic
- Links are (domain-locally) named instead of nodes (LId), therefore there is no equivalent to IP addresses
- Link identifiers are combined in a **bloom filter** (called zFilter) that defines the transit path



- 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



Problem: False Positives in Forwarding

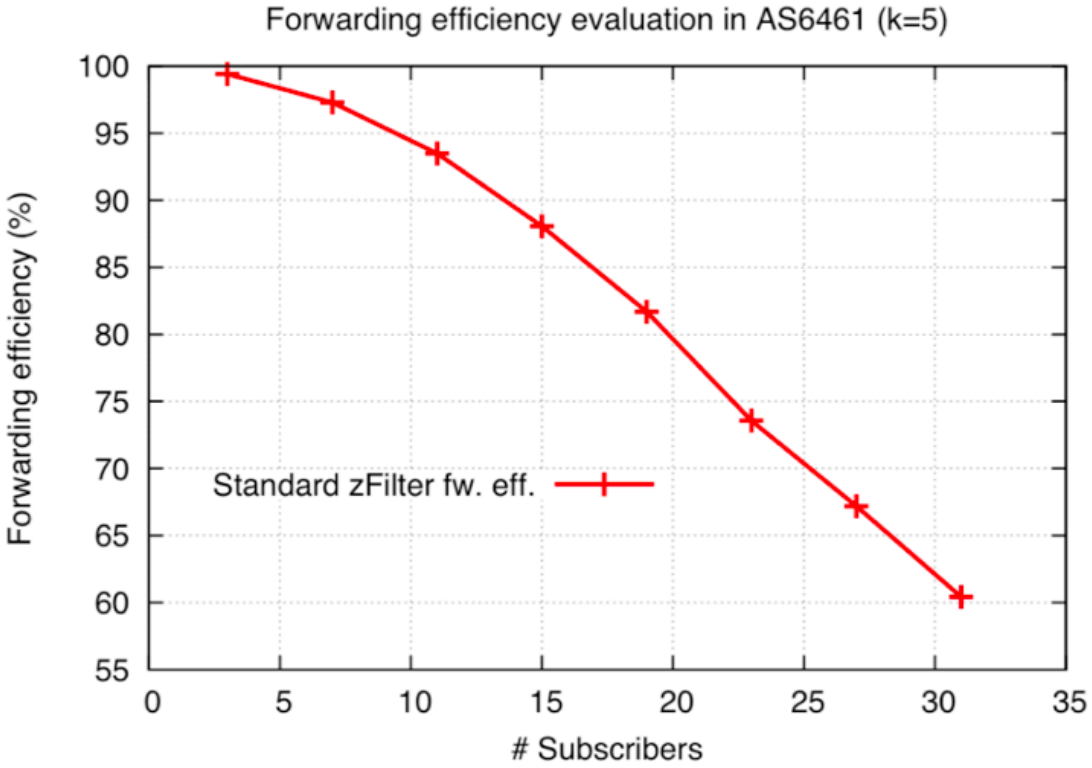
False positives occur when test is positive in a given node despite non-hashed LId (probability for consecutive false positives is multiplicative!)

- 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

Forwarding Efficiency

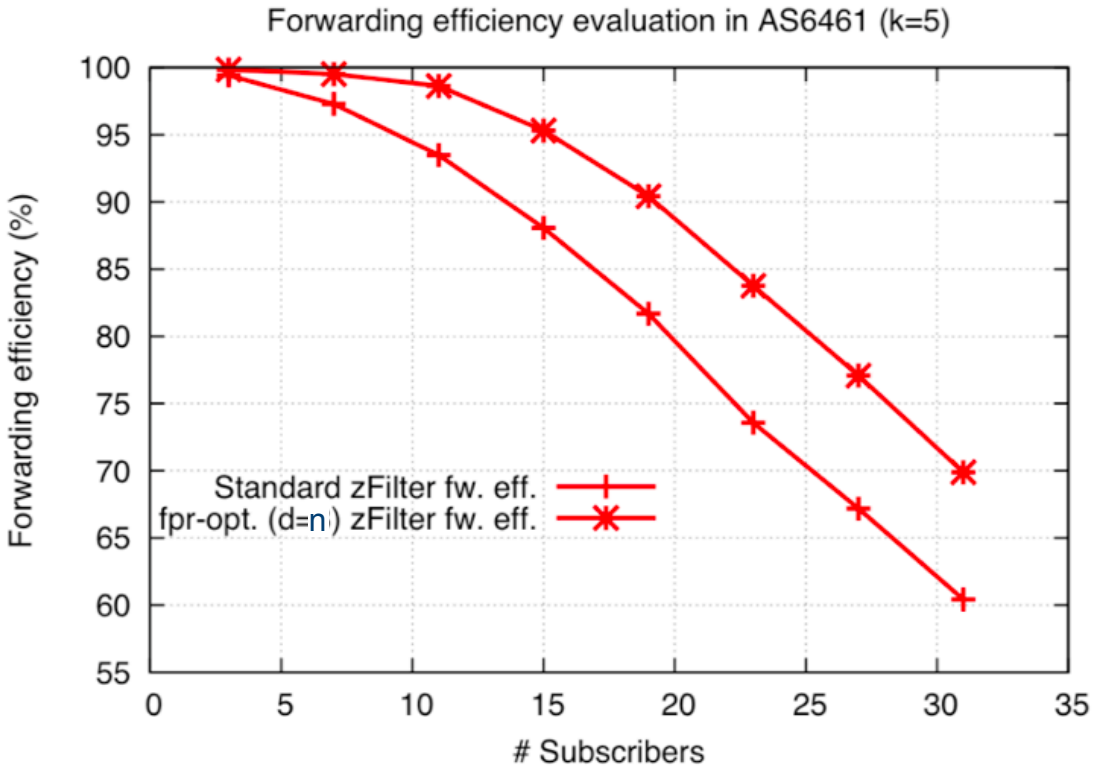
- Simulations with
  - Rocketfuel
  - SNDlib
- Forwarding efficiency with 20 subscribers
  - ~80%

  -> 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



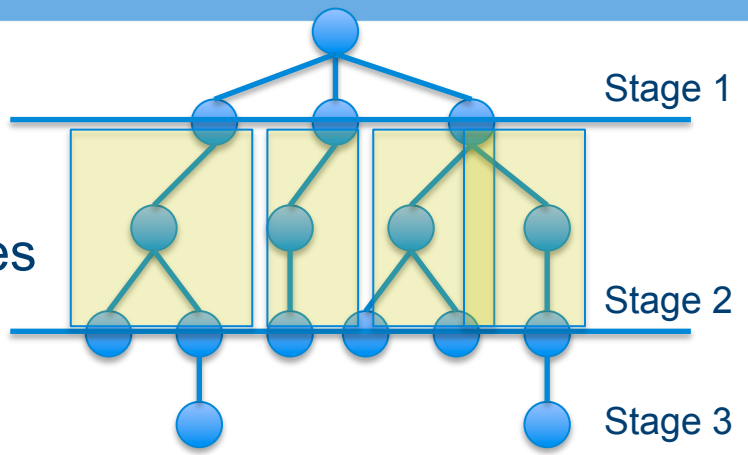
UNIVERSITY OF CAMBRIDGE

From Efficient Forwarding to Scale

Going Beyond LIPSIN

Idea: Multi-stage BF Forwarding

- Divide a delivery tree into stages
  - Generally, each stage has 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

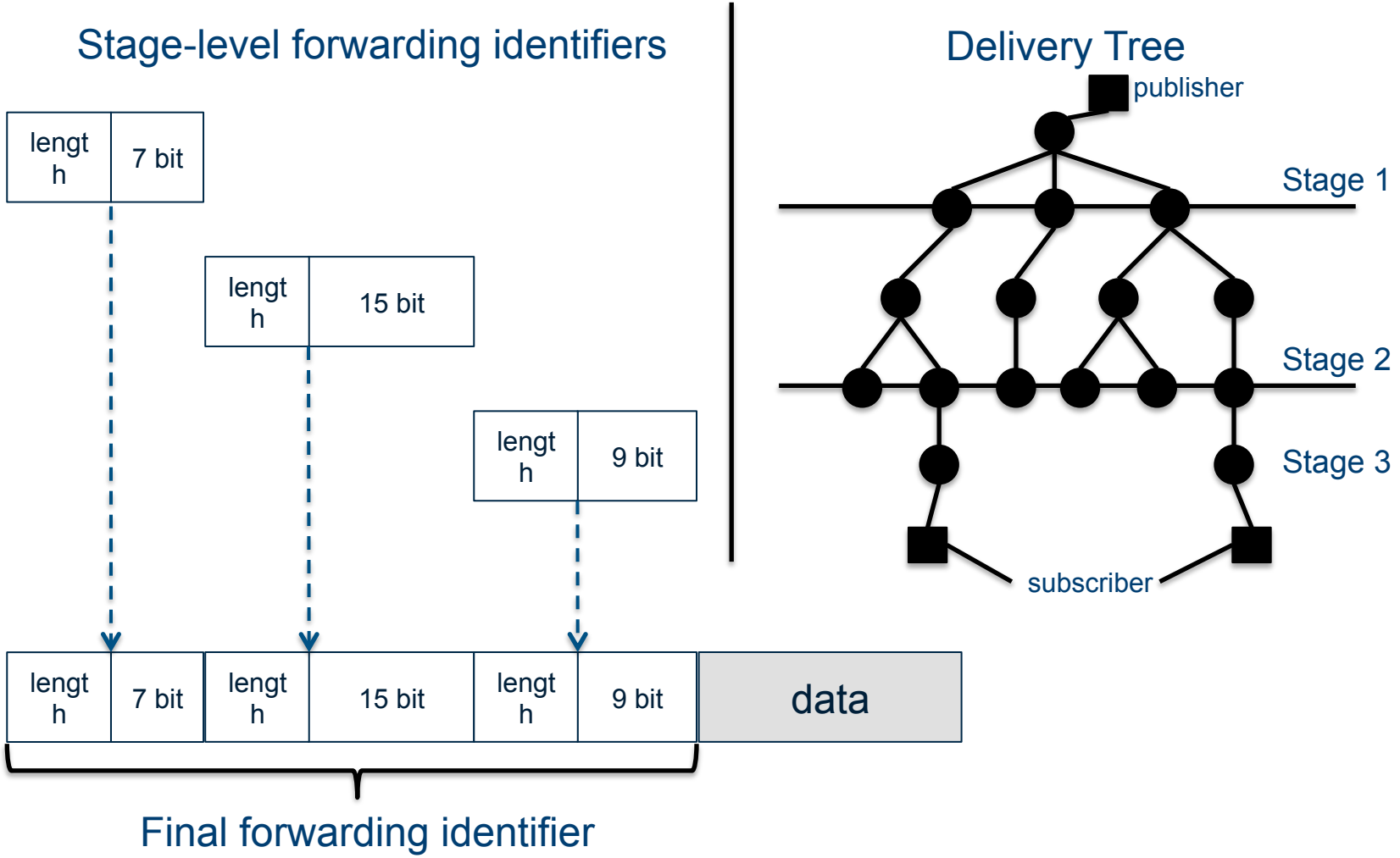


More specifically: Topology Formation

- Calculate tree for given <pub,subs> relation
- For each stage:
  - Define **in_tree** as the set of LIds being in the tree and **out_tree** as the ones not
  - Determine minimal length of BF that can hold **in_tree** with P(false positive)=0 (with the help of **out_tree**)
  - Determine BF through ORing **in_tree** into BF
    - Test if BF would cause false positve (increase, if so)
- Determine overall header through
  - Write length of stageBF through *Elias omega* encoding
  - Write stageBF

} For all stages

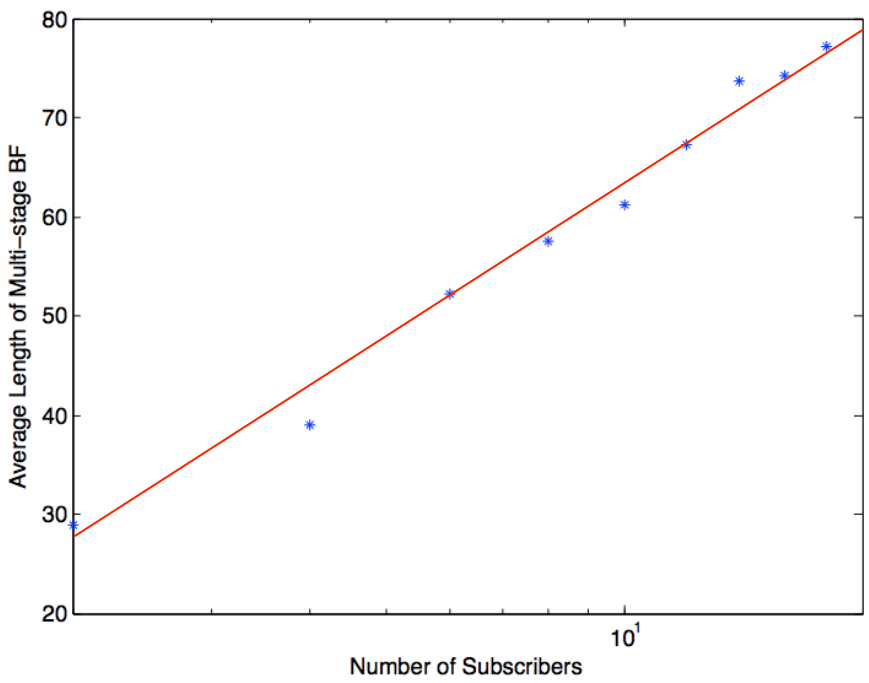
In a Nutshell

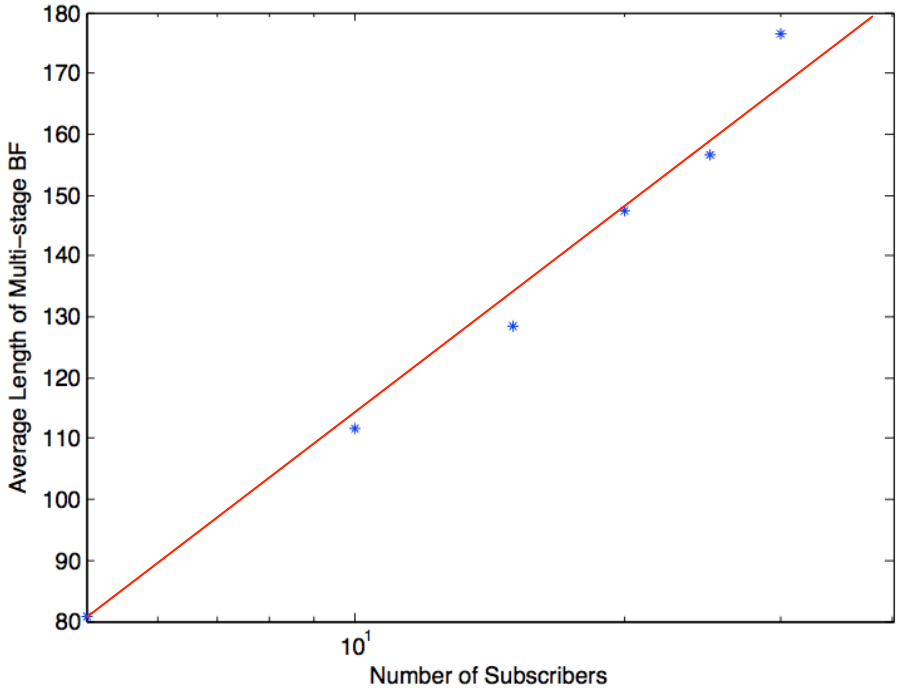


Pros and Cons

- Advantages
  - Arbitrary tree size (limit only when restricting maximum size for variable length header)
  - Tradeoff between false positive rate and header size
    - **Not realized here since P(false positive) is kept zero!**
  - Single hop vs multi-hop stages possible (single hops naturally limit BF anomalies)
  - **Lends itself to inter-domain as well as intra-domain forwarding**
- Disadvantages
  - Higher complexity in forwarding (but only at the stage boundaries)
  - Higher overhead due to variable length (but only for trees that are larger than LIPSIN trees anyways!), but overhead reduces as you traverse the tree

Header Length

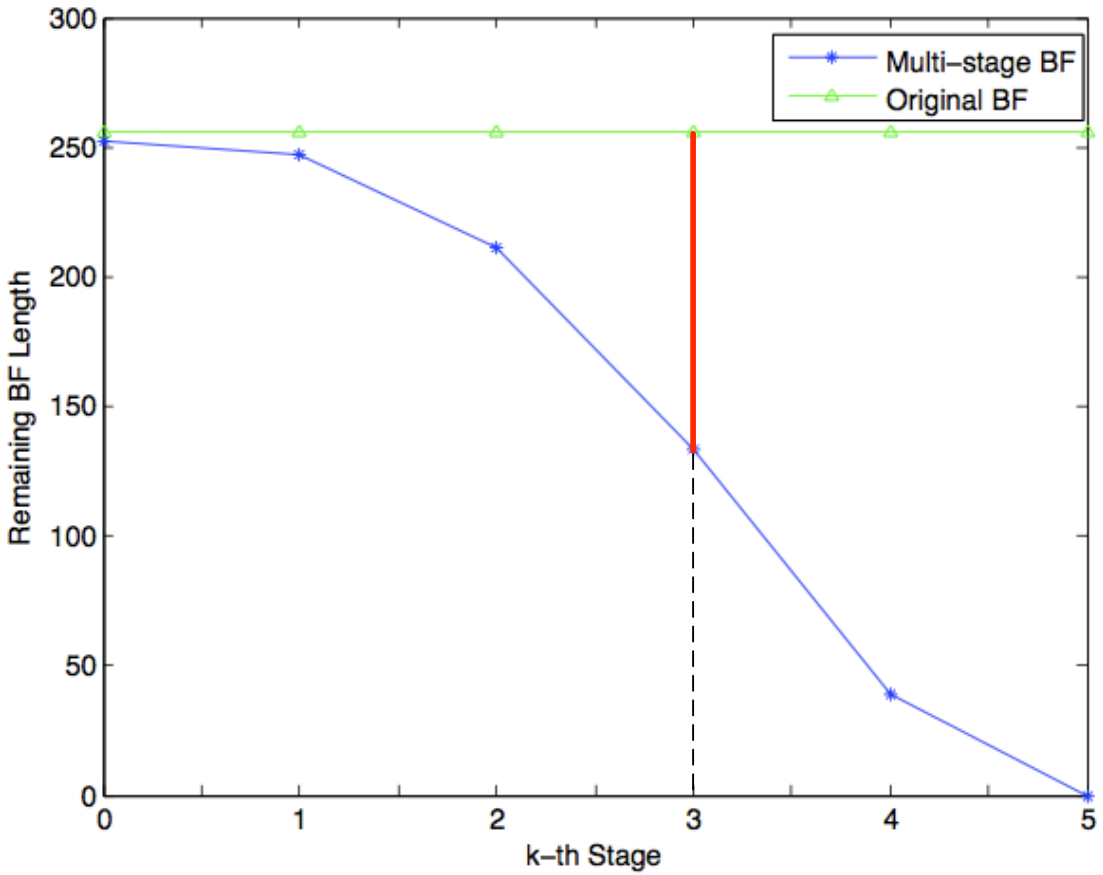




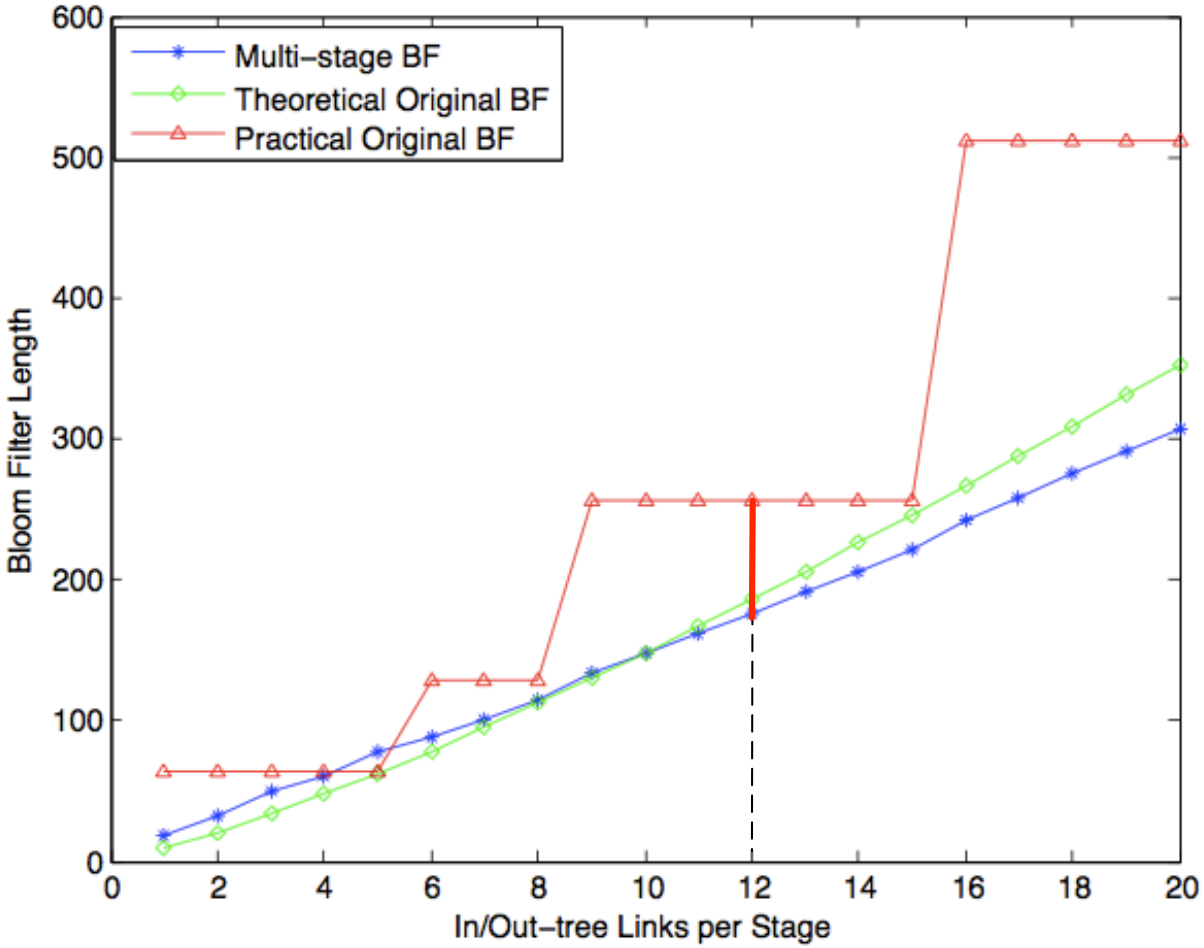
Logarithmic growth of header

- The more subscribers exist, the less likely a full branch needs inclusion!

Shrinking of Header During Transmission



Comparison to LIPSIN (in realistic deployments)

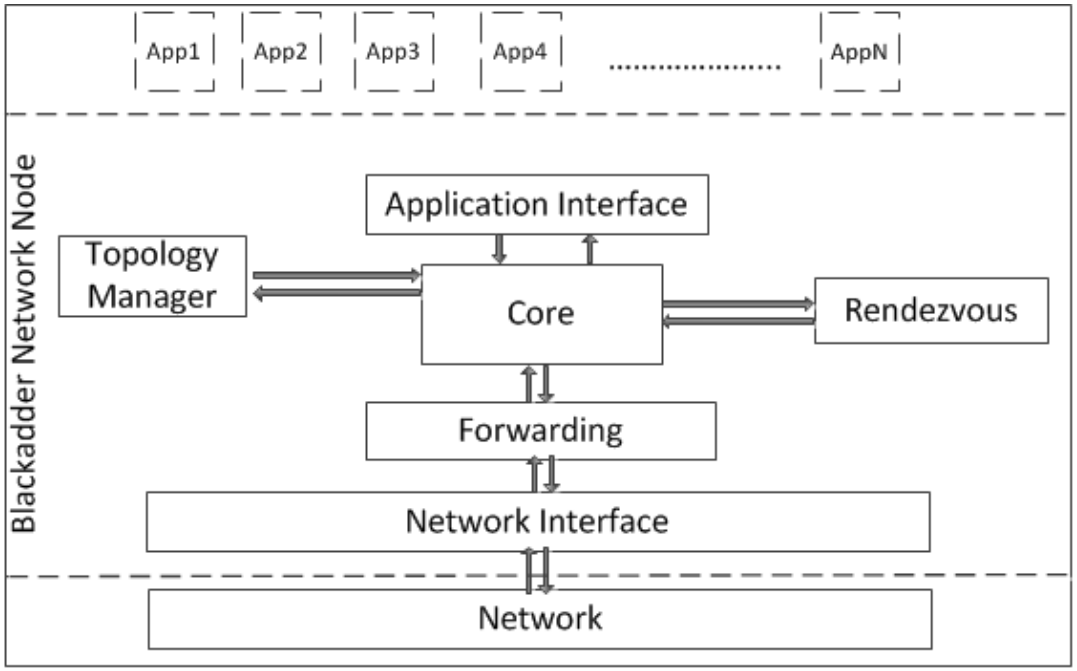


UNIVERSITY OF CAMBRIDGE

Prototype, Deployment & Some Results

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

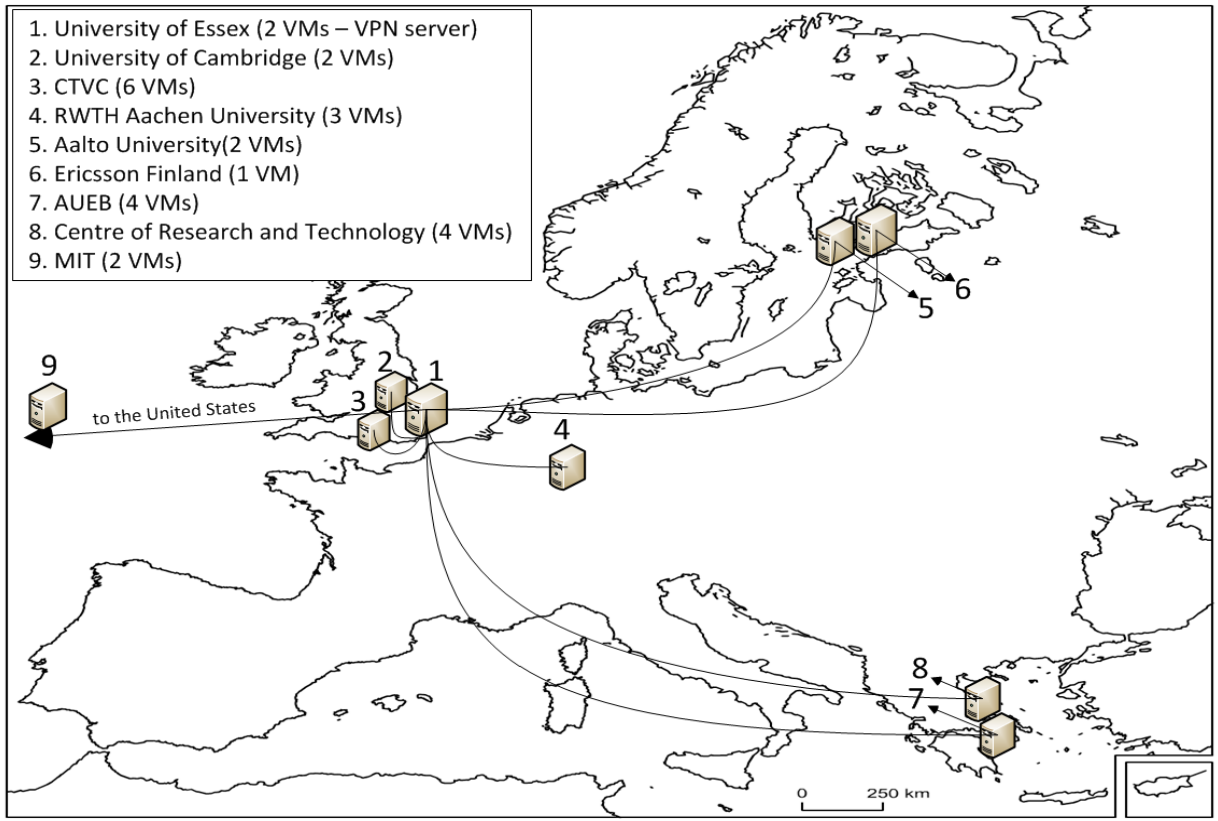
Our Prototype: Blackadder



- 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

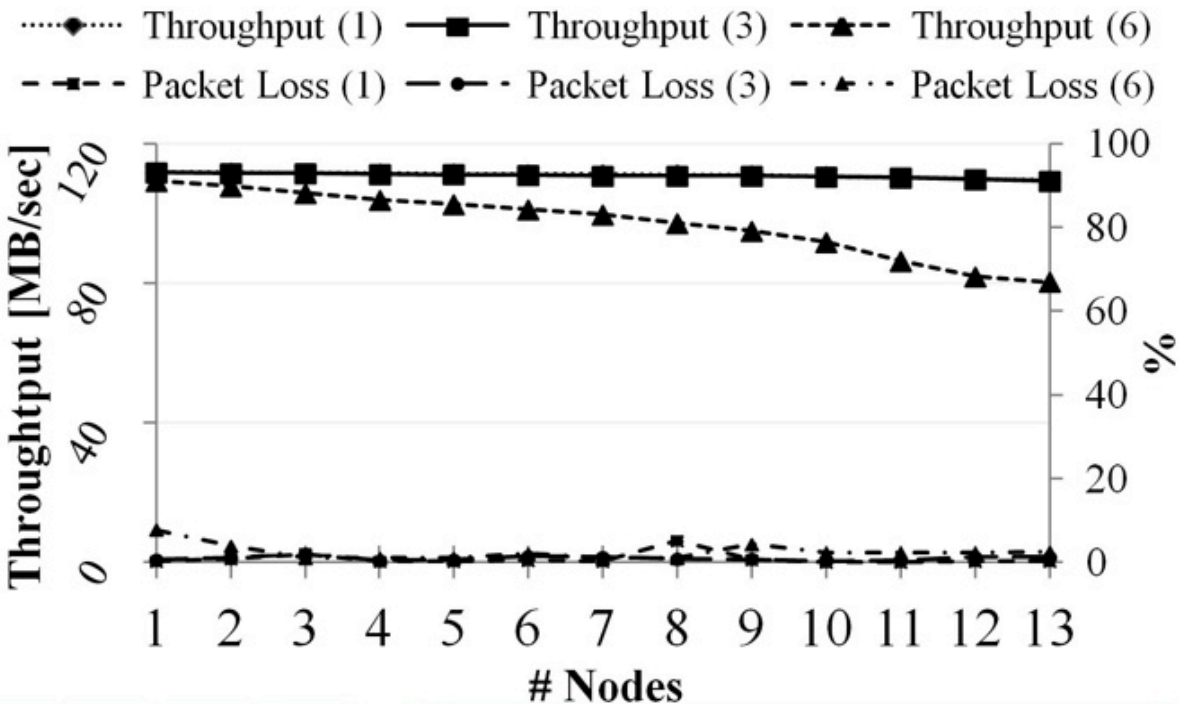


- 9 international sites
  - 26 machines with +40 on-demand ones
  - tunneled via openVPN with configurable topologies

Also available:

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

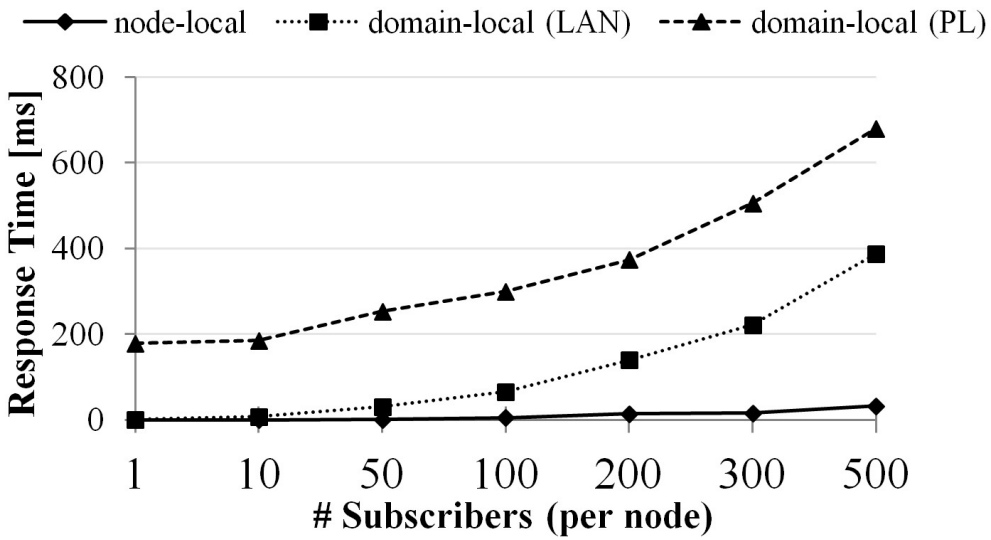
Experimental Evaluation: Fast Path



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

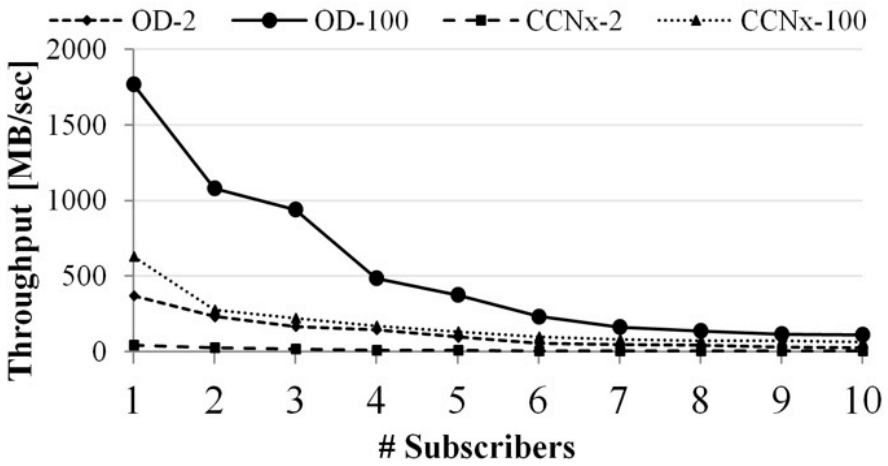
Domain-local (Gbit LAN)

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

Domain-local (PlanetLab)

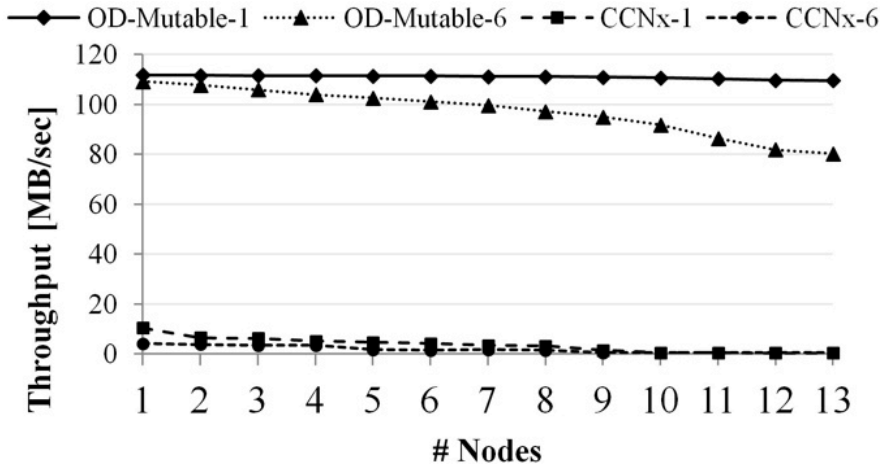
- Large delays
- ~200ms for 1 sub per node (73 in total)
- ~680ms for 36,500 subs

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!

Conclusions

- Changing the internetworking architecture surely is ambitious!
  - But there's a growing case being made in the community!
    - CCN, PSIRP, PURSUIT, recent pubs (e.g., ACM CCR 04/2010)
- A sound architectural model is crucial
  - Tenets, functional model, E2E assembly
- There is lots of technology providing solutions
  - LIPSIN, DHT, caching based on locality/social relation, swarming, …
    - Some of it seems to fit better in an information-centric model
- Pieces are being put together as we speak
  - Work on deployment strategies and socio-economic evaluation
  - PURSUIT test bed between 8 international sites with working prototype

More Information

- Websites
  - http://www.psirp.org (the start of this work)
  - http://www.fp7-pursuit.org (the continuation of this work)
  - http://www.named-data.net/ (successor of CCN)
- Papers
  - ACM CCR 04/2010, SIGCOMM 2009 (LIPSIN), CONEXT 2009 (CCN), and many more on http://www.psirp.org
- Contact: dirk.trossen@cl.cam.ac.uk (for questions or student projects)