Future-Net '09

IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS

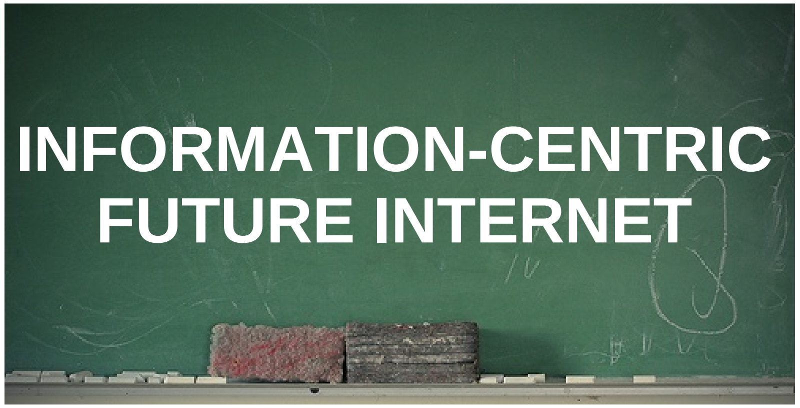
ICC 2009 June 14-18 Dresden, Germany

# Exploring the Pub/Sub Routing & Forwarding Space

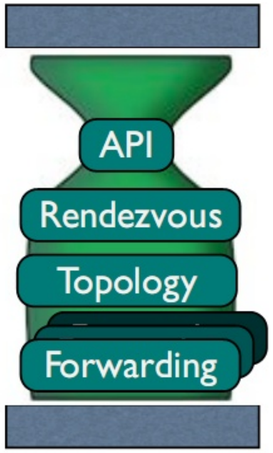
András Zahemszky, András Császár, Pekka Nikander
Ericsson Research
Email: {firstname.lastname}@ericsson.com

Christian Esteve Rothenberg
University of Campinas (UNICAMP)
Email: chesteve@dca.fee.unicamp.br

# Agenda

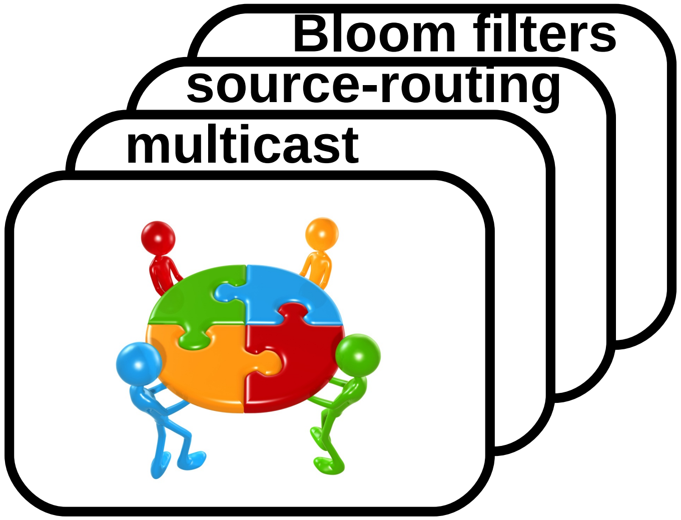


Motivation

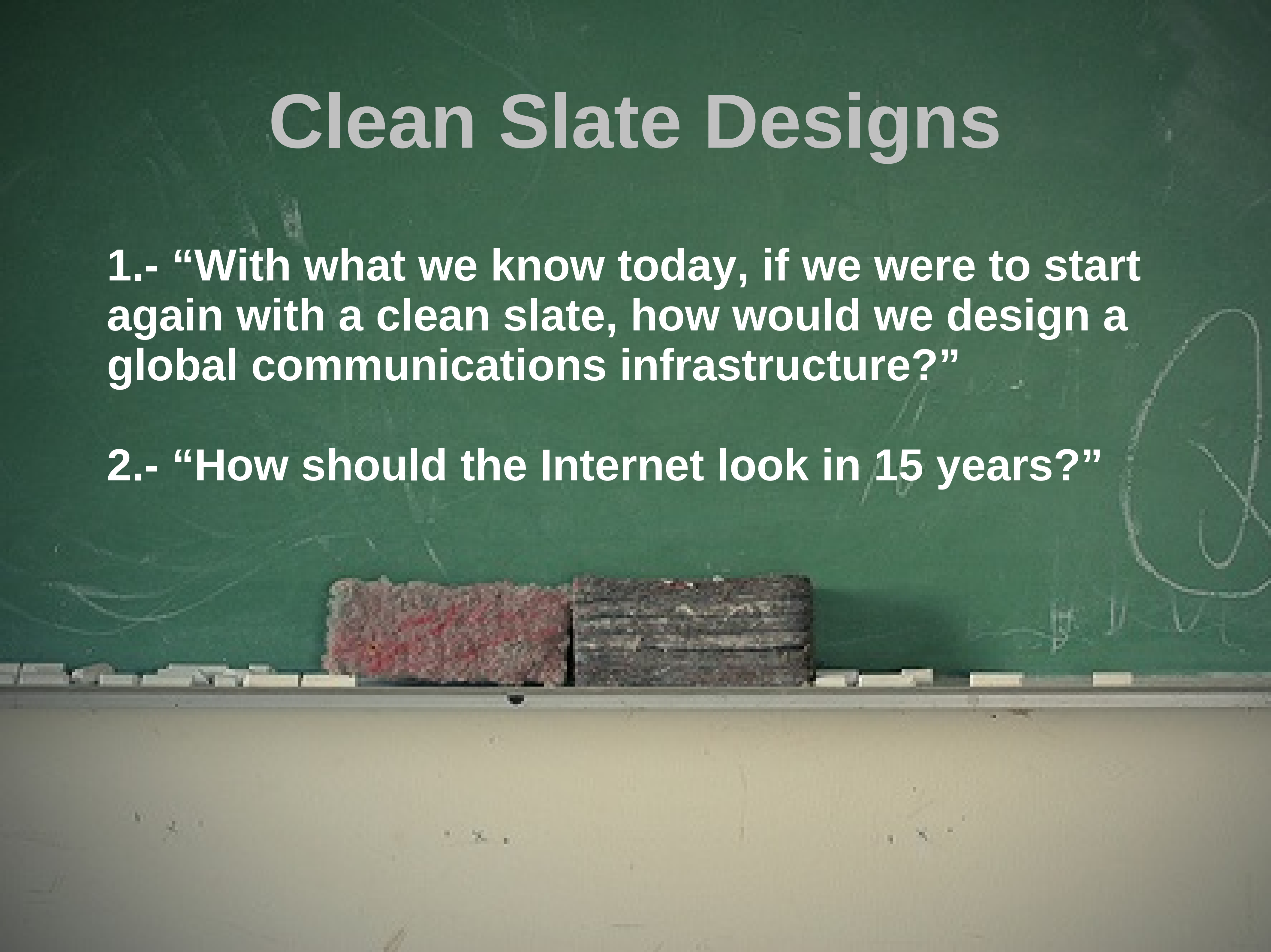


The PSIRP way

Exploring



Pieces of the solution

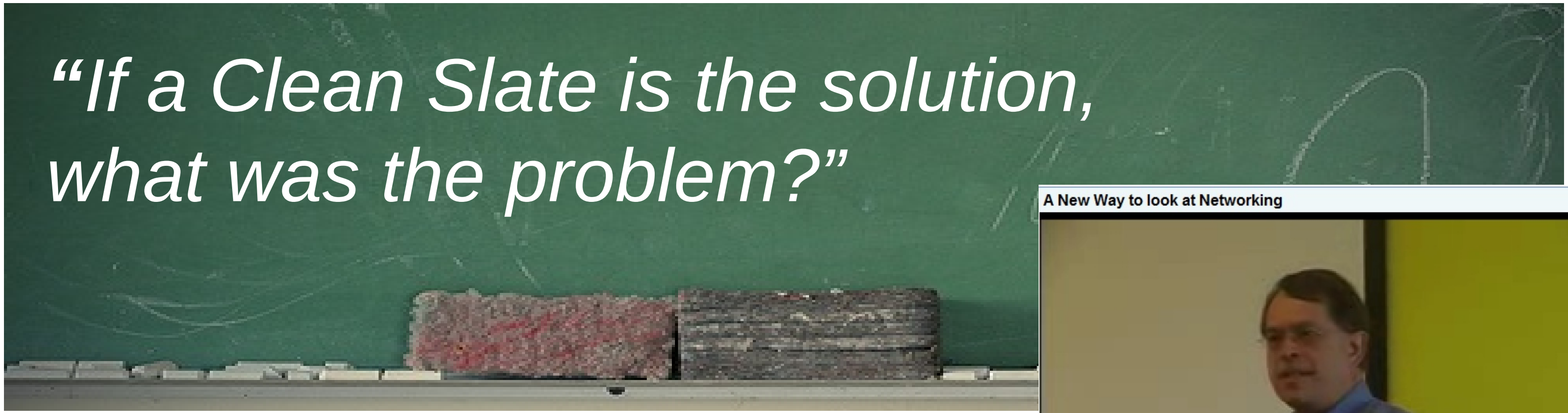


## Clean Slate Designs

1.- **“With what we know today, if we were to start again with a clean slate, how would we design a global communications infrastructure?”**

2.- **“How should the Internet look in 15 years?”**

## Van Jacobson’s waves of networking



*“If a Clean Slate is the solution, what was the problem?”*

**99%** **Internet traffic:**

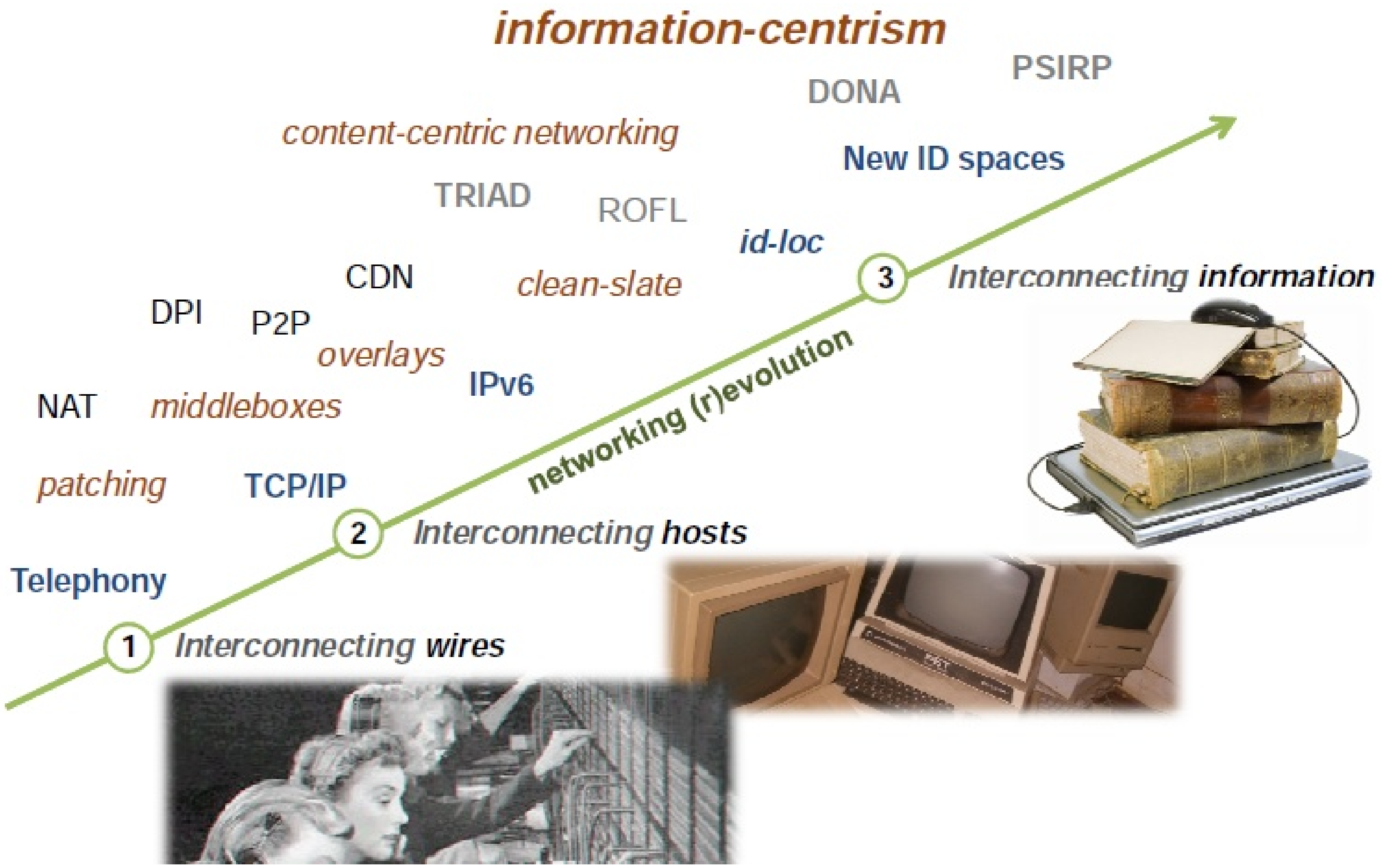
**Named chunks of data (Web, P2P, Video, etc.)**

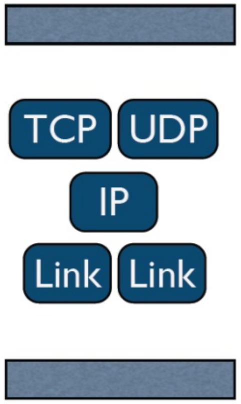
**New problem:** Dissemination of named pieces of data

**Answer:** Content-Centric Networking

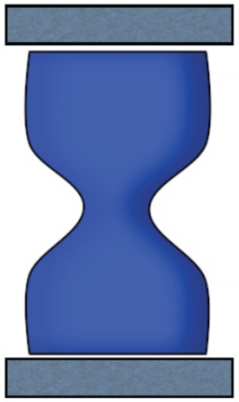
http://video.google.com/videoplay?docid=-6972678839686672840

# Towards information-oriented networking



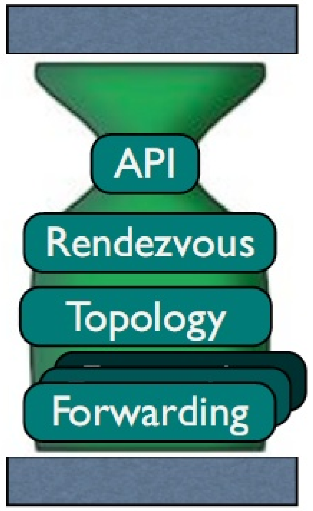


# Information-oriented networking
## - Rethinking fundamentals -

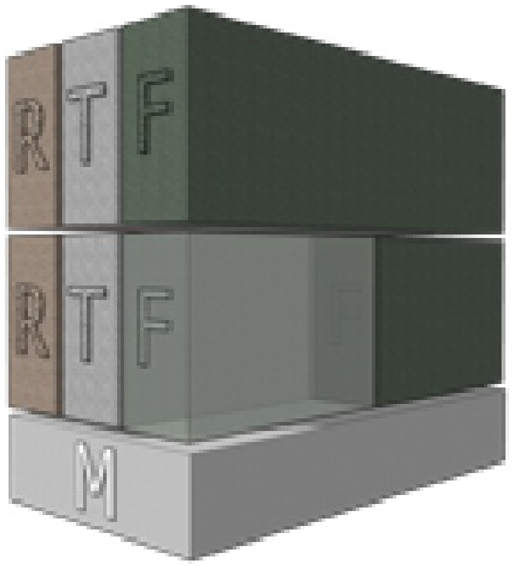


| | | |
|---|---|---|
| Send / Receive | → | Publish / Subscribe |
| Sender-driven | → | Receiver-driven |
| Host names | → | Data names |
| Host reachability | → | Information scoping |
| Channel security | → | Self-certified metadata |
| Unicast | → | Multicast |

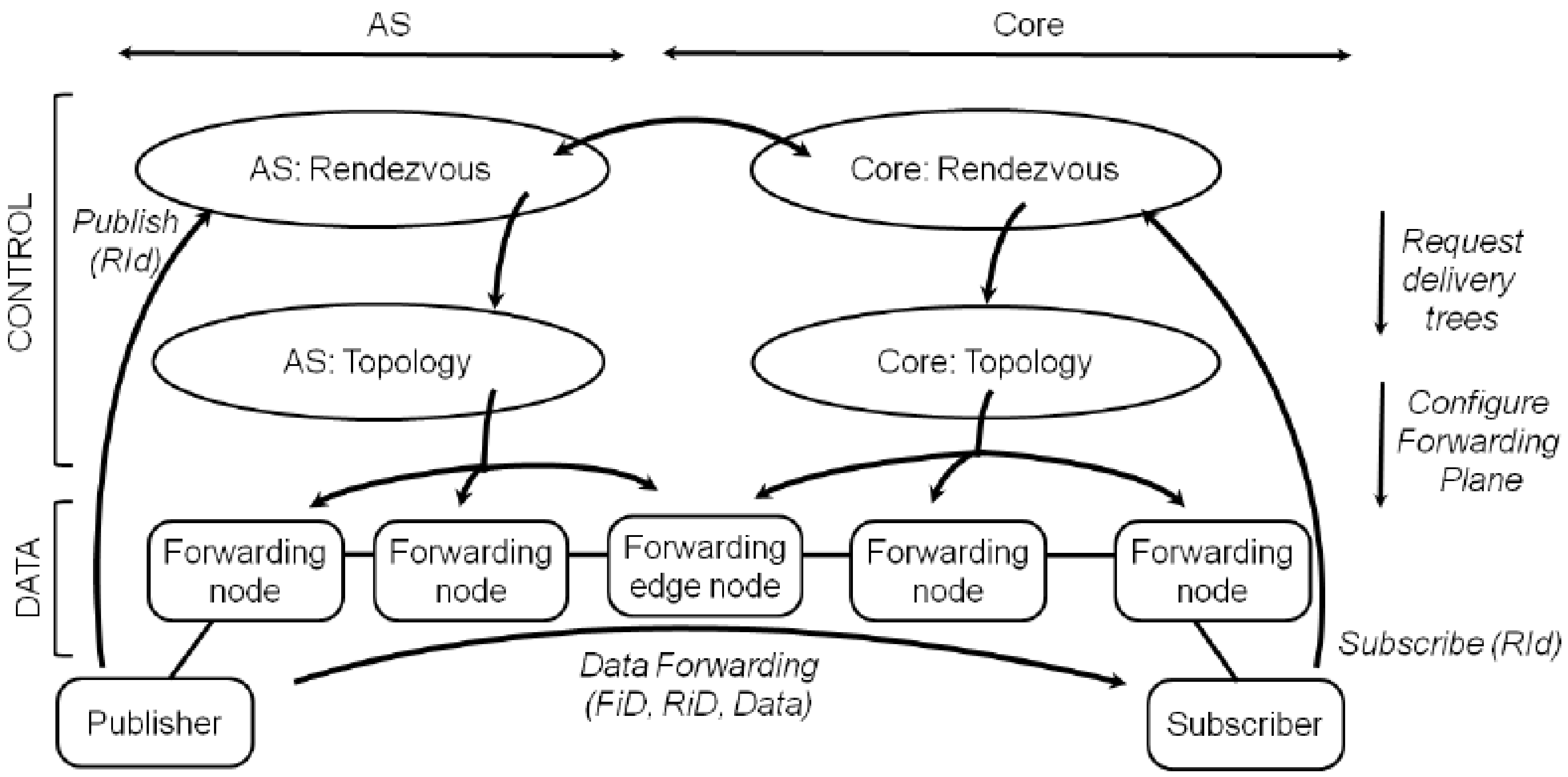
# RTFM Architecture



- **Rendezvous**
  - Matching subscriptions to publications
- **Topology**
  - Creating and maintaining delivery trees used for forwarding publications
- **Forwarding**
  - Data delivery operations. e.g., label switching, fast forwarding
- **and More**
  - Node-to-node link data transfer + e.g., opportunistic caching, collaborative and network coding, lateral error correction etc.

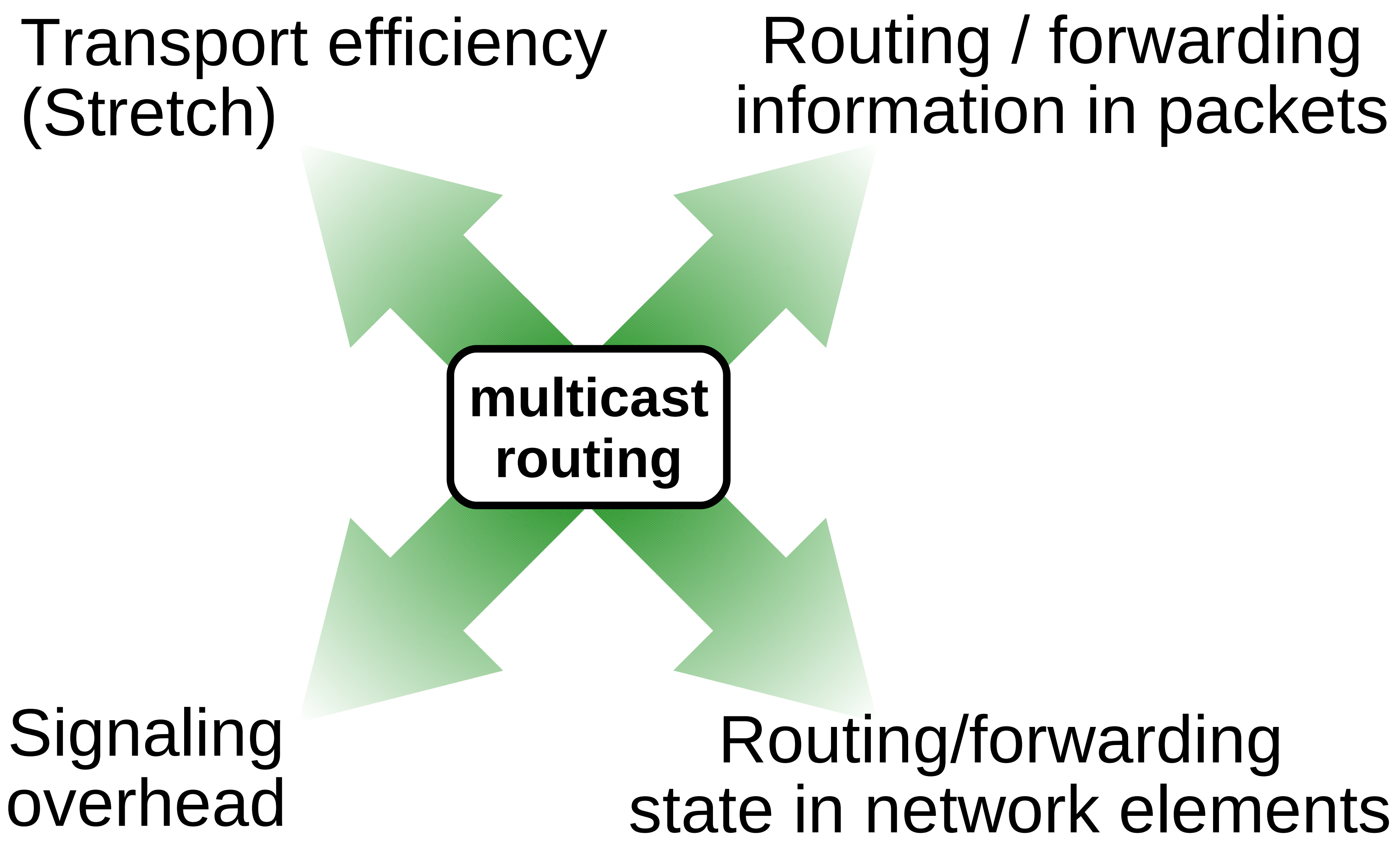


# High level architectural overview
## - Mapping information to delivery trees -



- **Rendezvous identifier (RiD)**:
  - Self-certifying identifier of data
- **Forwarding identifier (FiD)**:
  - Used for fast forwarding

# 4-dimensional solution space



# Divide and Conquer

Source routing

Hierarchical aggregation

Install network state only when necessary

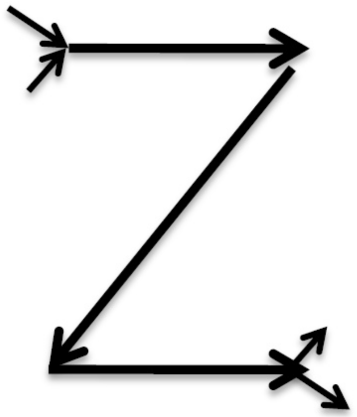
Stepwise approach for delivery tree management

**Transport efficiency**

**Trade-off**

**Scalability**

(non-ideal trees, over-deliveries, min. signalling & forwarding tables)



## zFilters: in-packet Bloom filter encoding of delivery trees

**State** in the *packet headers*

- Each network link has an identity and (a series of) *Link IDs:*
  *LIT: 256 bit vector with just k=5 bit positions set to one*
- Delivery tree by ORing the Link IDs into a fixed-size in-packet Bloom filter (zFilter) representing a *source route*

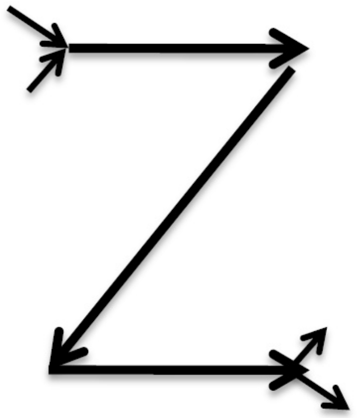
**Basic forwarding operation**

*"Is outbound link A in packet header Z?"*

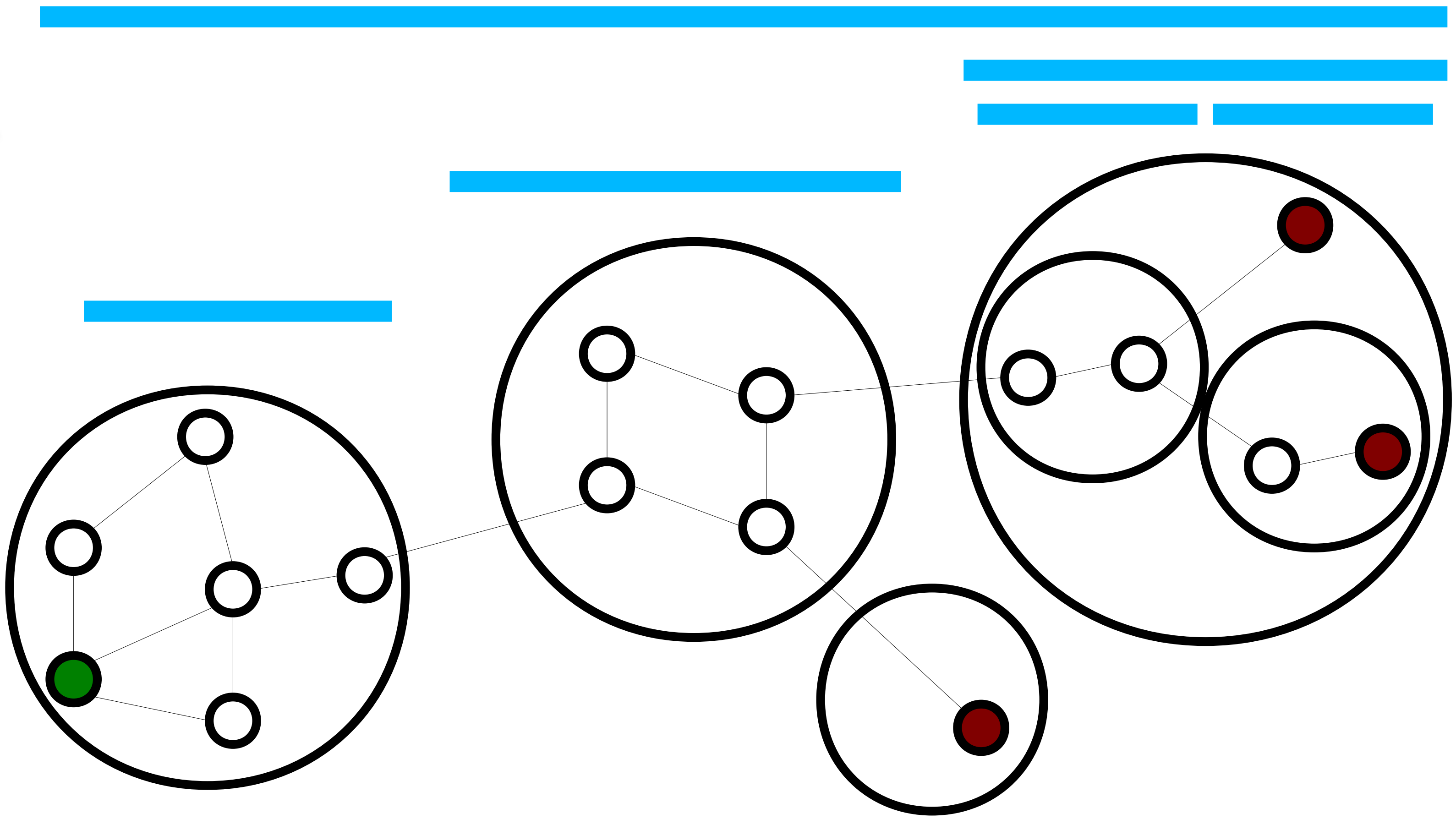
- *Small* forwarding tables (Link ID to neighbors + Virtual Link IDs)
- *Fast* packet forwarding (bitwise AND operations)

**Extensions and details:**

[10] P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, and P. Nikander. LIPSIN: Line speed publish/subscribe inter-networking. In *Proceedings of ACM SIGCOMM'09, Barcelona, Spain*, Aug. 2009.



# Virtual links



**State** in network nodes

- One-to-one, one-to-many, many-to-many, many-to-one forw. structures
- Supporting horizontal and/or hierarchical aggregation
- Less overdeliveries

# Delivery trees in 5 steps

1) Compute an *ideal tree.*

2) Determine the *gaps* between the ideal tree and any existing trees.

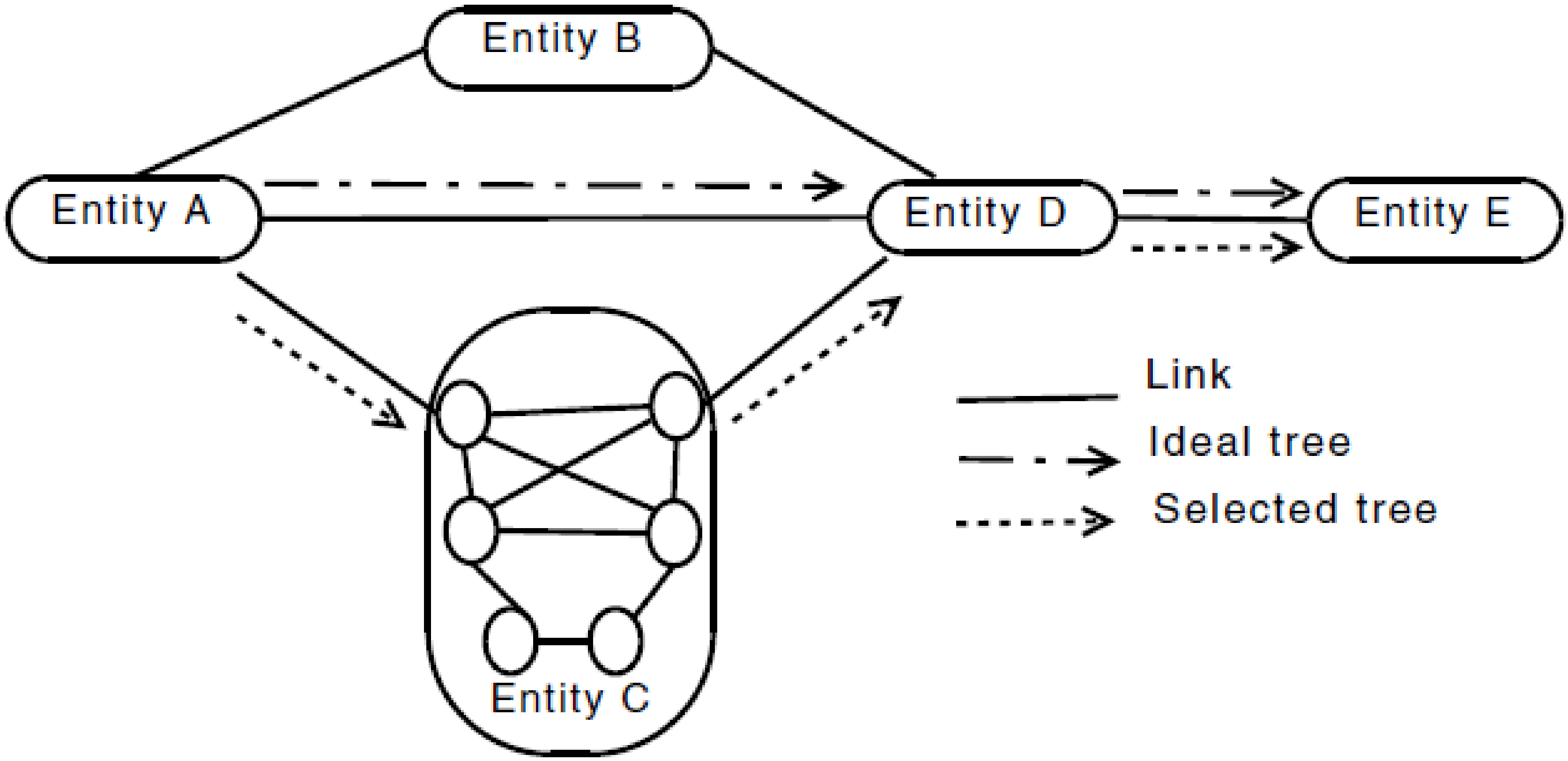
3) Select *tree-creation* strategies or *gap-filling* strategy for each gap.

4) *Compute* the needed *changes* according to the strategies.

5) *Apply* the changes to the network.

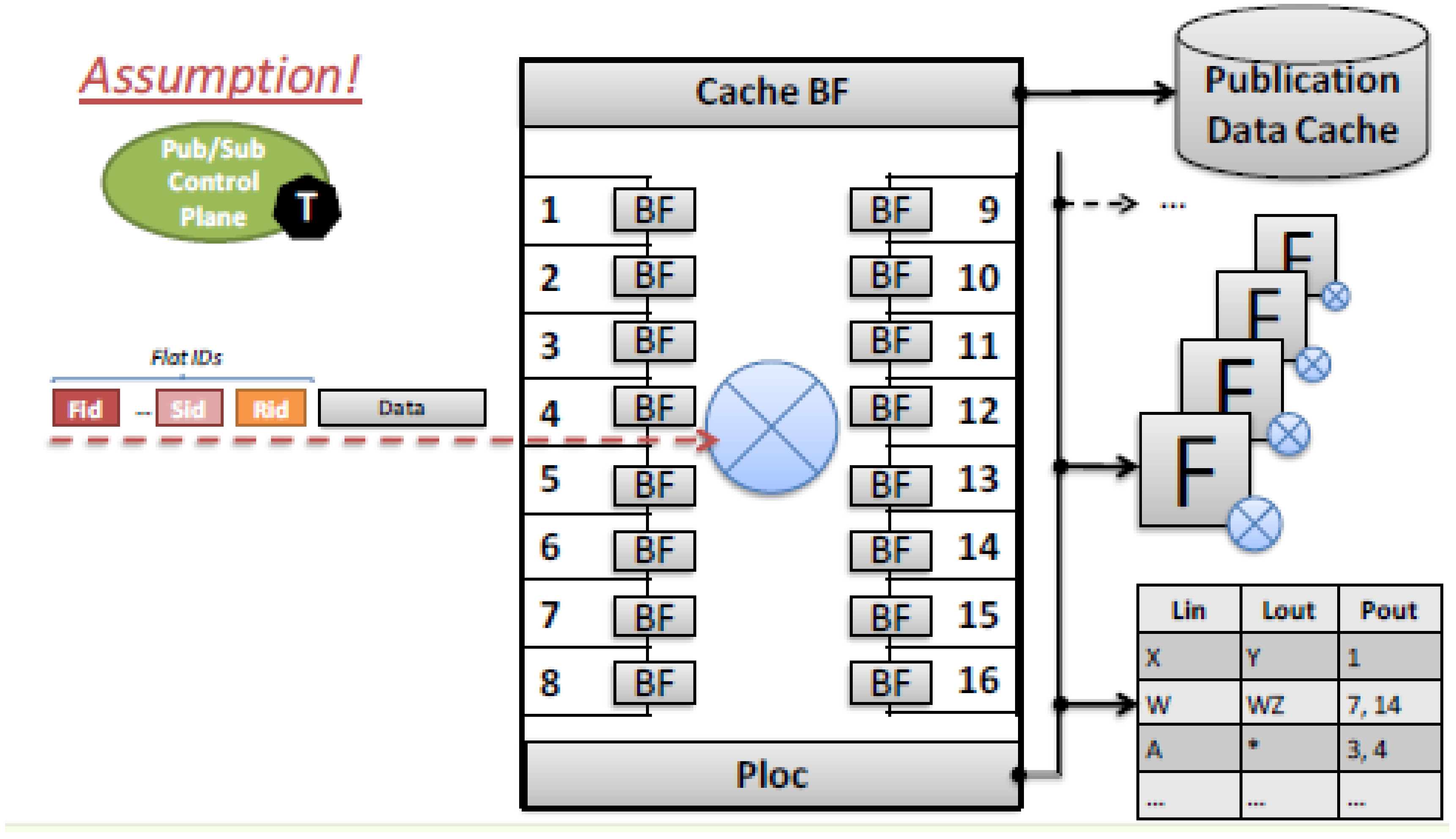
(Image Credit: Scott Maxwell / http://www.lumaxart.com / CC By-SA 2.0)

# Example



- Hierarchical aggregation
  - AS confederations, ASes, intra-domain areas, routers
- Selecting a *good enough* tree
  - Strict requirement: containing all the subscribers

# SPSwitch: Approximate fast stateful edge switching



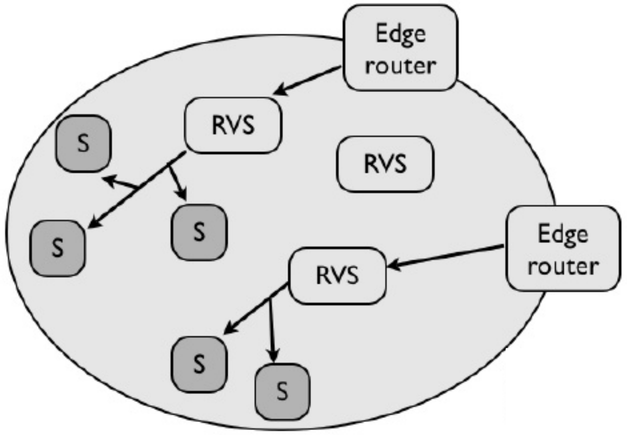
| Lin | Lout | Pout |
|---|---|---|
| X | Y | 1 |
| W | WZ | 7, 14 |
| A | * | 3, 4 |
| ... | ... | ... |

# Challenges and future work

## Inter-domain routing and forwarding

Avoid the mapping problem:

- Between intra-AS trees and inter-AS trees no one-to-one mapping exist
- *Do we really need rendezvous identifier-based matching for label swapping?*
- Hints for future directions:
  - Information scopes
  - Non-routable link identifiers for mapping

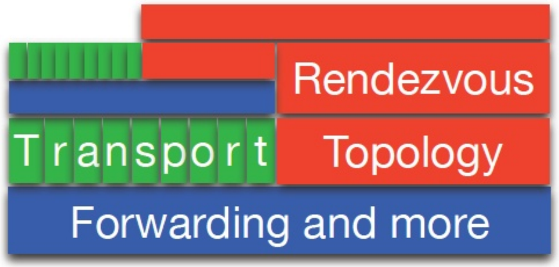


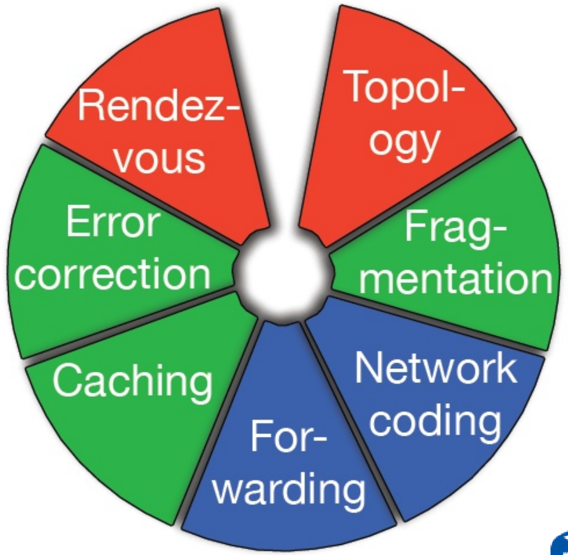
Topology functions:

- performance implications
- delay
- inter-operation between Topology Managers

# Prototype implementation

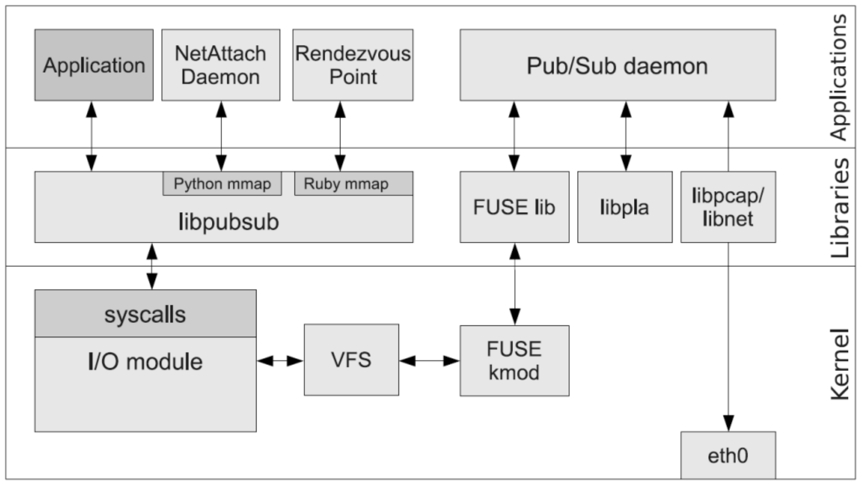
## RTFM architecture



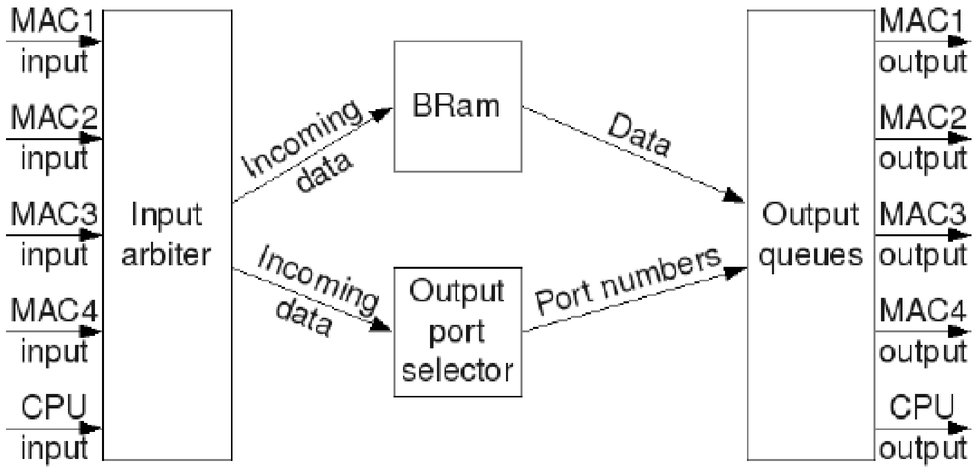


## Component Wheel

FreeBSD®



NetFPGA



Source: EU FP7 PSIRP Project, http://psirp.org

PSIRP
PUBLISH-SUBSCRIBE INTERNET ROUTING PARADIGM

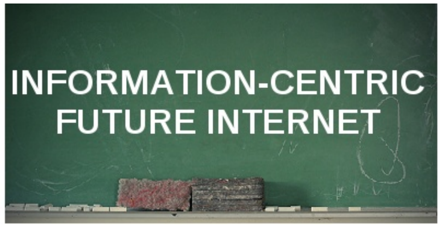
# Take Aways

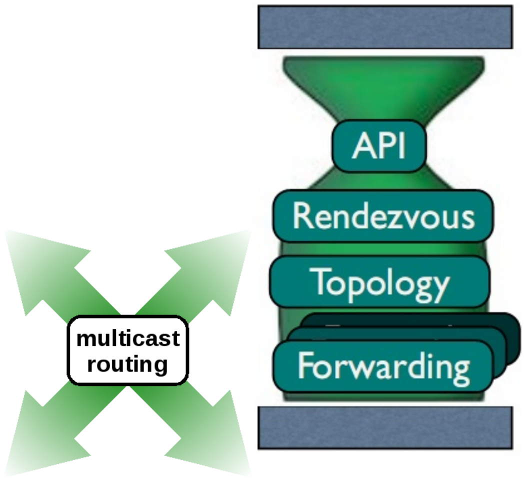
We are building an *information-centric* network based on the *publish / subscribe* paradigm

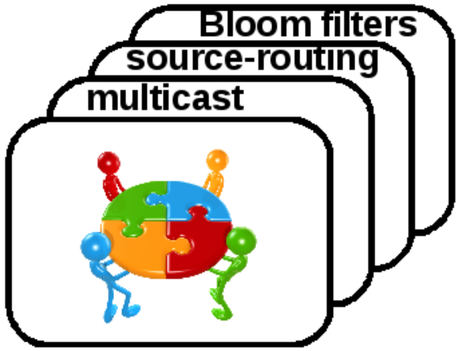
We are re-thinking the forwarding plane with *native multicast* departing from host-centric designs

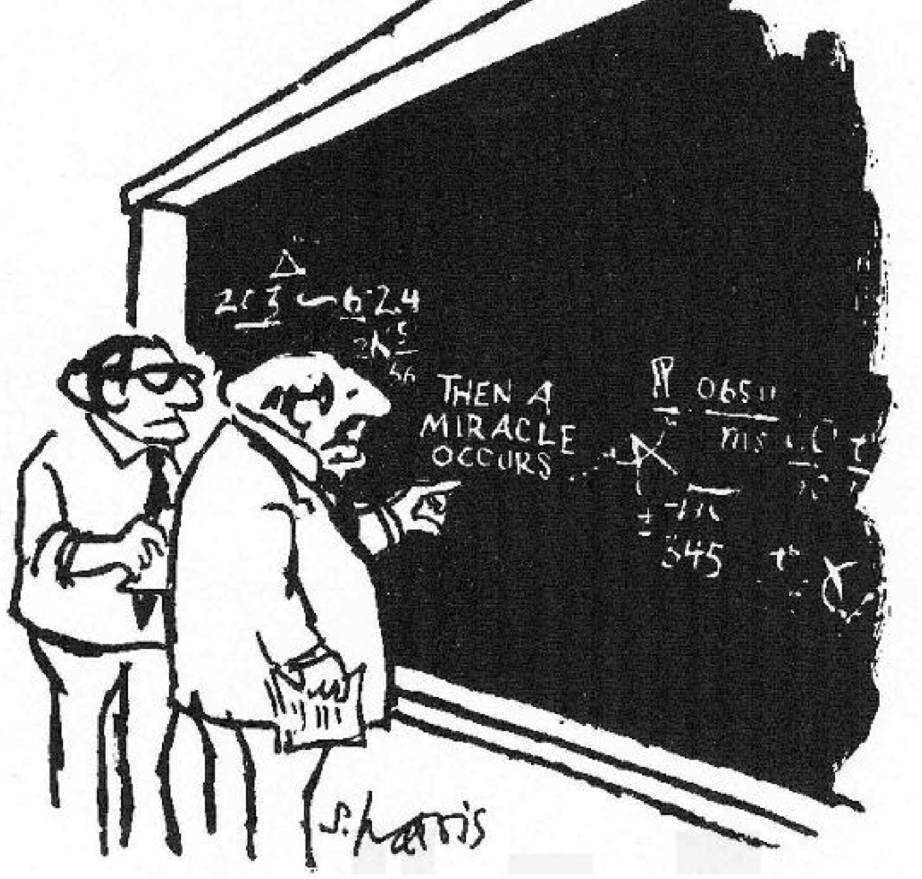
To meet the *scalability* requirements, we explore the trade-off between *transport efficiency* and network state via
1) *Bloom-filter-based* forwarding decisions
2) approximate *delivery trees*
3) hierarchical/horizontal *division*

We have a flexible design for routing & forwarding, with component enablers allowing:
*stateless* and *stateful* operations
*balance state* : packet *headers* <·> netw. *nodes*





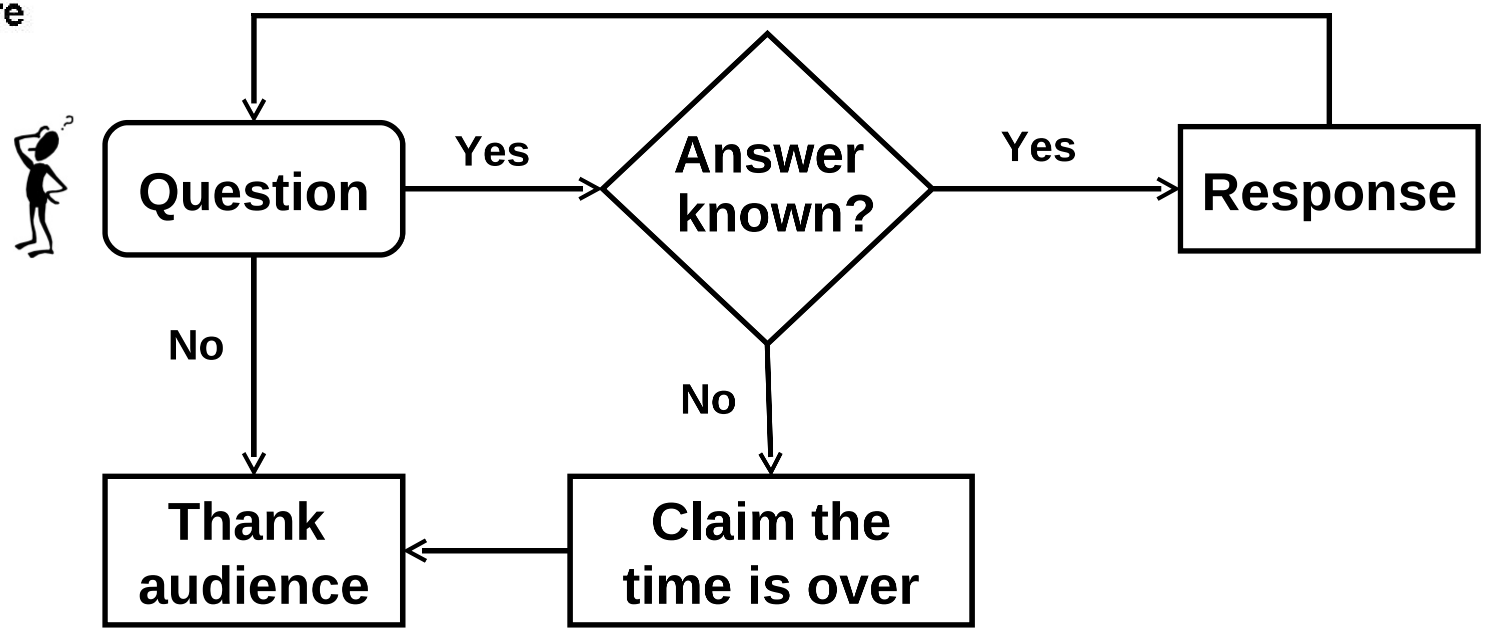




"I think you should be more explicit here in step two"

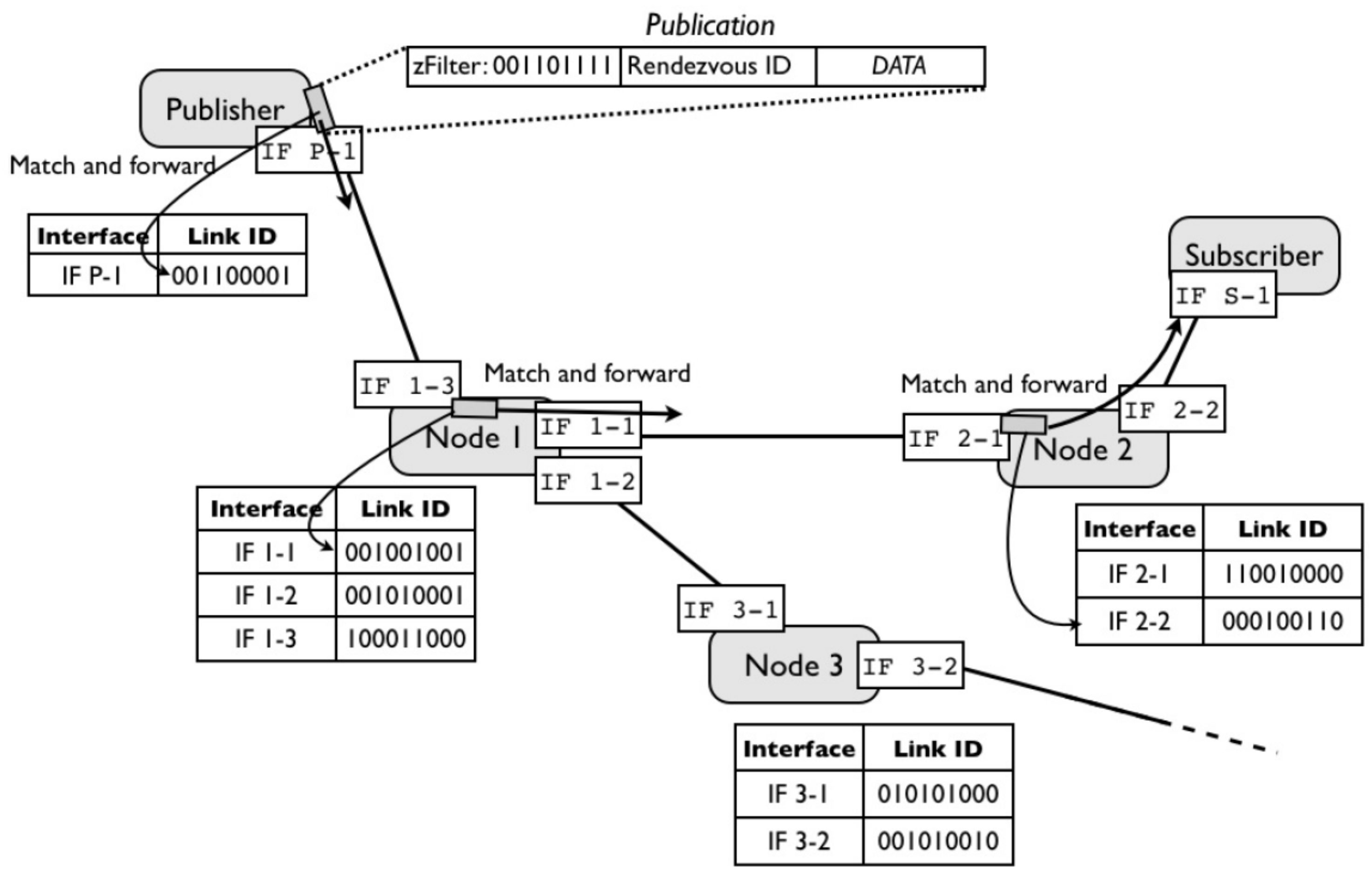
questions?

Thank you!



## BACKUP

# Forwarding on Bloomed link identifiers



| Interface | Link ID |
|---|---|
| IF P-1 | 001100001 |

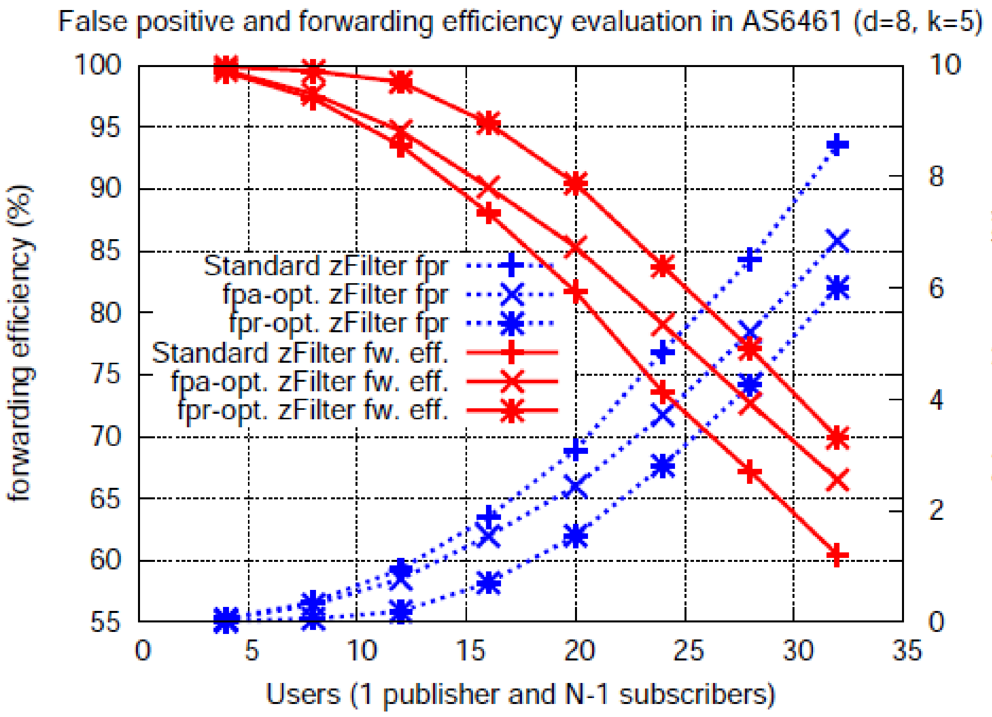
| Interface | Link ID |
|---|---|
| IF 1-1 | 001001001 |
| IF 1-2 | 001010001 |
| IF 1-3 | 100011000 |

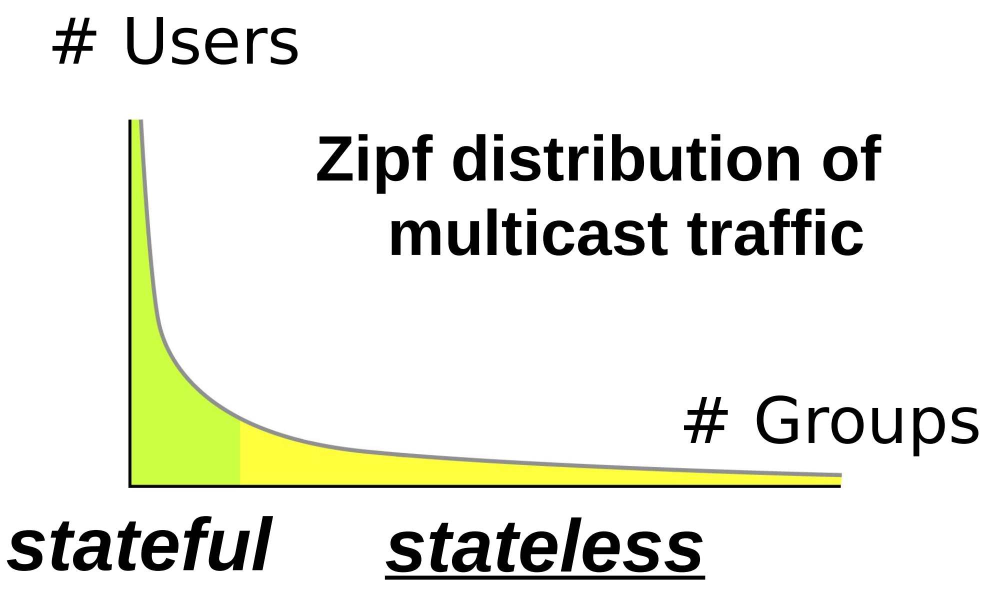
| Interface | Link ID |
|---|---|
| IF 2-1 | 110010000 |
| IF 2-2 | 000100110 |

| Interface | Link ID |
|---|---|
| IF 3-1 | 010101000 |
| IF 3-2 | 001010010 |

# Practical results

- Stateless multicast with 256-bit zFilters (35 links -> 20 subscribers)
- Enough for sparse multicast in typical WAN

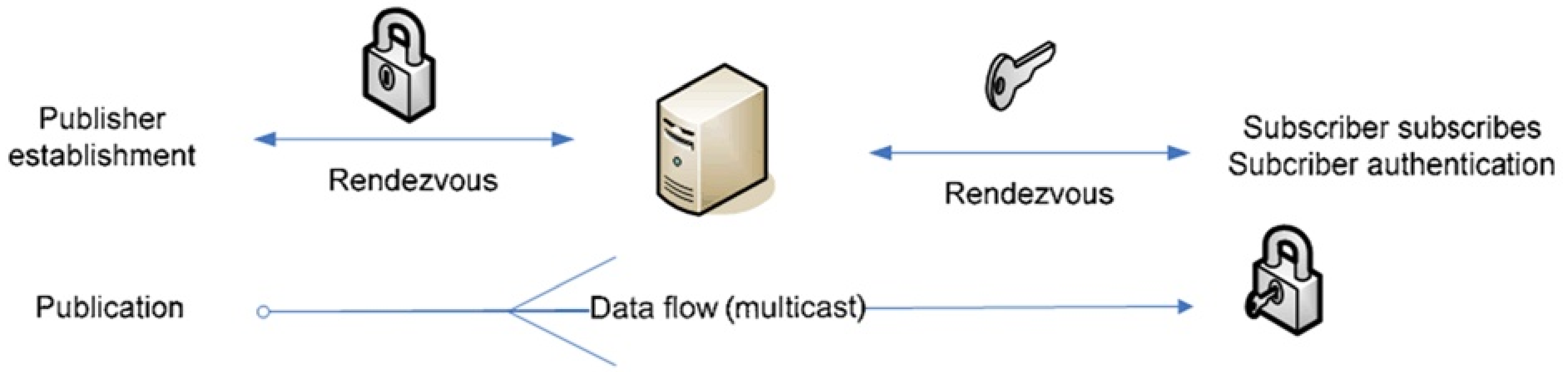




## EU FP7 PSIRP Project

Redesign the Internet architecture from the pub/sub point of view, taking nothing (not even IP) for granted:

- *Take information to the center of attention*
- *Remove the location-identity split that plagues current networks*
- *Innovative multicasting & caching features to optimize performance & efficiency*
- *Security as a native core component of the architecture*



Source: EU FP7 PSIRP Project, http://psirp.org