



Publish/Subscribe Internetworking

Cooperation with PSIRP and PURSUIT

Jimmy Kjällman, Ericsson NomadicLab

@ Tivit Future Internet pre-conference, 30.5.2012

Projects

› ICT SHOK FI WP3

2008-2012



› EU FP7 PSIRP

2008-2010

Publish/Subscribe Internet Routing Paradigm



› EU FP7 PURSUIT

2010-2013

Publish/Subscribe Internet Technology



Vision

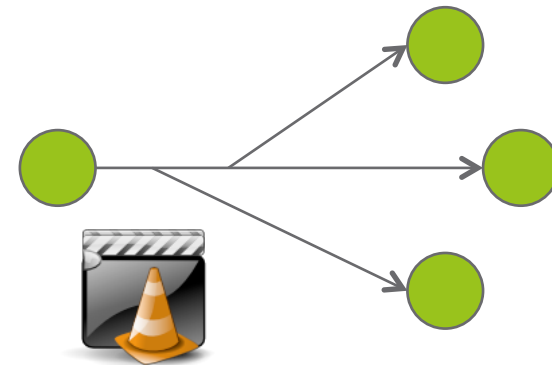
› **Future Internet**

- › Focus on *long-term* research
 - With feedback to short-term work
- › *Clean-slate* approach
- › *Redesigning* the Internet architecture
 - Considering both technical and socio-economic aspects



Vision

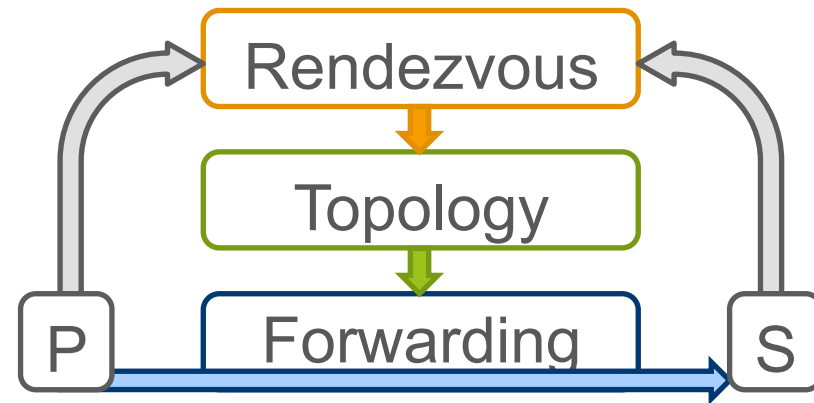
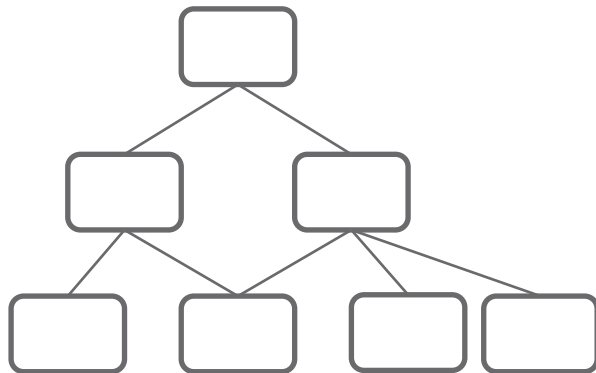
- › **Information-centric**
 - Not host centric
- › **Publish/subscribe**
 - Instead of send/receive
- › **Identify information**
 - No node addresses
- › **Secure and efficient networking**
 - DDoS protection, multicast, ...



Some Key Outcomes

› Architectural concepts

- Scoped information
- Publish/subscribe API
- Rendezvous, Topology Management and Forwarding components
- Strategies, recursive layering



Some Key Outcomes

- › New mechanisms for rendezvous, topology management and forwarding
 - E.g.: Stateless multicast forwarding based on *in-packet Bloom filters*

- › Prototypes for information-centric networking
 - E.g.: *Blackhawk* prototype memory-oriented pub/sub networking

- › Models for evaluation

In-packet Bloom filters

- › *Line Speed Publish/Subscribe Inter-Networking*

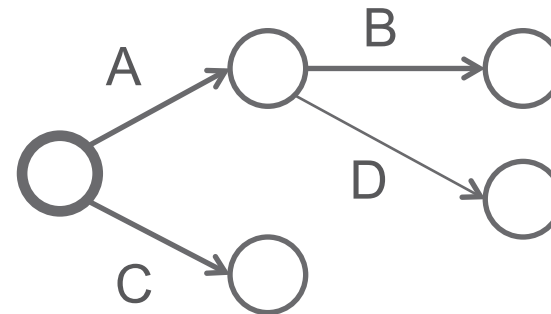
Petri Jokela, András Zahemszky, Christian Esteve, Somaya Arianfar,
and Pekka Nikander,
“LIPSIN: Line speed Publish/Subscribe Inter-Networking”,
ACM SIGCOMM 2009

- › Multicast
- › Source routing
- › Stateless routers
- › Security

In-packet Bloom filters

- › Encode a path from publisher and subscriber into a Bloom filter
- › Network interfaces have *link identifiers* (LIDs)
 - E.g. 256 bits, 5 bits set to 1
- › A path is constructed by ORing LIDs into a Bloom filter

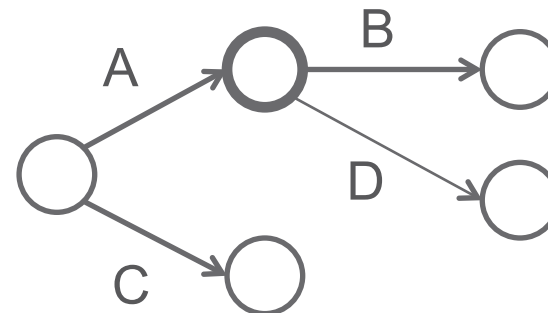
Link A	0000001001
Link B	0010000100
Link C	1000100000
Path	1010101101



In-packet Bloom filters

- › A packet is sent using this Bloom filter instead of an address
- › At each hop, an AND operation is used for checking whether an outgoing link is in the filter

Tree	101010 1 10 1	Forward
Link B	000000 1 00 1	
Tree	1 0 1010110 1	Don't
Link D	0 1 0000000 1	forward



In-packet Bloom filters

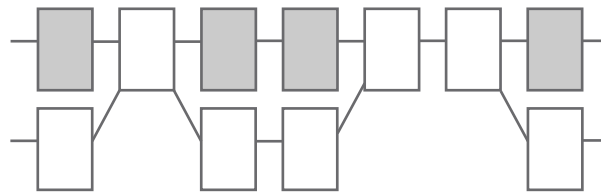
- › Scalability affected by *false positives*
 - A LID might match “accidentally” even if it’s not added
 - Probability of this increases as more bits are set to 1
- › Several scalability improvements have been developed
 - E.g. tree splitting solutions

- › Also solutions to dealing with anomalies have been proposed
- › As well as security improvements, etc.

- › Applications: data centers, MPLS, media streaming, ...

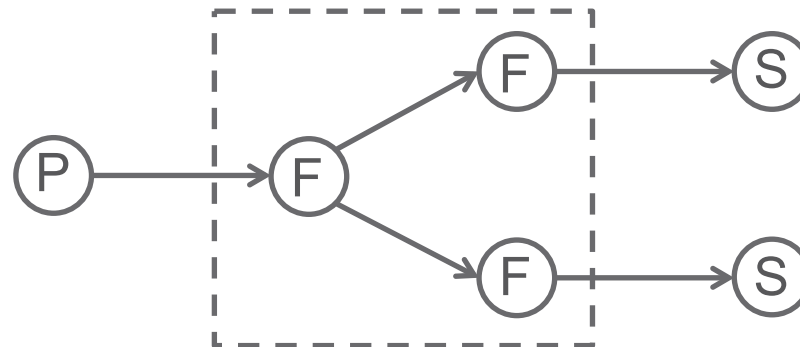
Blackhawk prototype

- › New network stack implementation
 - Integrated to the OS kernel
- › Pub/sub API
 - With events
- › Publishing and subscribing to *memory objects*
 - I.e., data pointers in an application + metadata
 - Directly publish and subscribe to pieces of information
- › Networking with in-packet Bloom filters
- › Support for data *deduplication*, *caching* and pull-based transport



Stateless multicast demo

- › Multicast streaming from publisher to subscribers
- › Based on in-packet Bloom filters
- › No stream-specific state in forwarding nodes
- › Utilizes existing applications





ERICSSON