

Moving the Control from Senders to Receivers

S-38.4030 Postgraduate Course on Networking
Technology, 4th of December, 2007

Teemu Rinta-aho <teemu@rinta-aho.org>

Contents

- Background
- Publish-Subscribe internetworking
- Prototype
- Conclusions

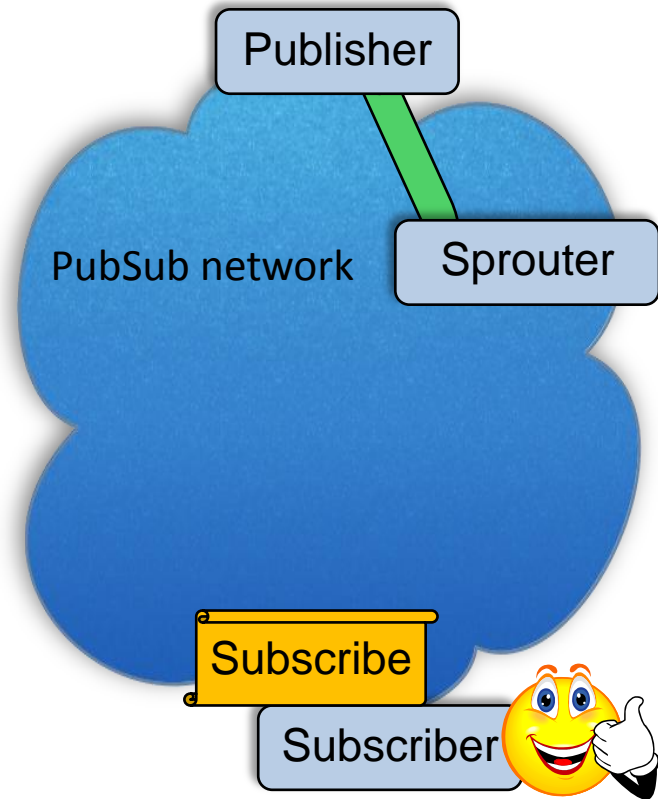
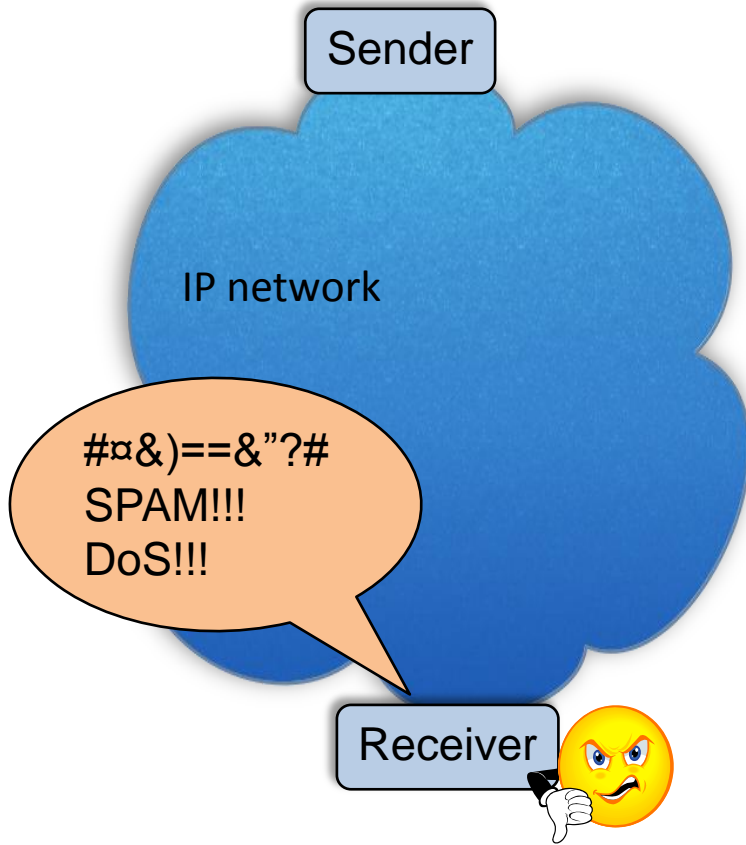
Background

- Most networks of today are built on the *send-receive* model
 - Sender selects the receiver (e.g. IP address)
 - Network helps the sender (routing by dst address)
- One alternative is the *publish-subscribe* (PubSub) model
 - Receiver selects what it wants to receive (data ID)
 - Network helps the receiver (routing by data IDs)
- The research question:
 - *Try to see if it is possible to implement an internetworking architecture on top of the publish-subscribe model instead of the send-receive model*

PubSub Internetworking: Why?

- Micro-economics: Prevents DDoS very effectively
 - sender does have incentive to send, always
 - receiver does not necessarily have incentive to receive
 - current networks help the sender
 - network forwards whatever senders send
 - “rendezvous” takes place at the receiver, with the receiver’s resources
- Fundamentals: How could the network help receiver?
 - by allowing the receiver to select what to receive
- Architecture: Unifies unicast and multicast from the beginning
 - unicast becomes a 1-recipient multicast
 - makes radio and wireline more similar
- Applications: More natural to many applications
 - content delivery networks

IP vs. PubSub Internetworking



Architectural Components

- Identifiers
- Primitives
- Publication metadata
- Compensation mechanisms
- Authentication mechanisms
- Rendezvous, routing and forwarding

Identifiers

- End-points are not identified, only data
 - Publisher may have an ID
 - Not bound to a location
- Publication ID
 - Private, a.k.a. "The Private Key of the Publication"
 - E.g. a hash over the data+a public key+...
- Subscription ID
 - Public, a.k.a. "The Public Key of the Publication"
 - E.g. a hash of the Publication ID

Primitives

- publish
 - Publish data and associated metadata
 - E.g. Publish a file or a stream
- subscribe
 - Subscribe to a publication
 - Breaks down to *publishing* a subscription

Publication Metadata

- Data needed to handle a publication
 - Not application data
 - Contains e.g.
 - Publication ID
 - Subscription ID
 - Scope
 - Related compensation mechanism

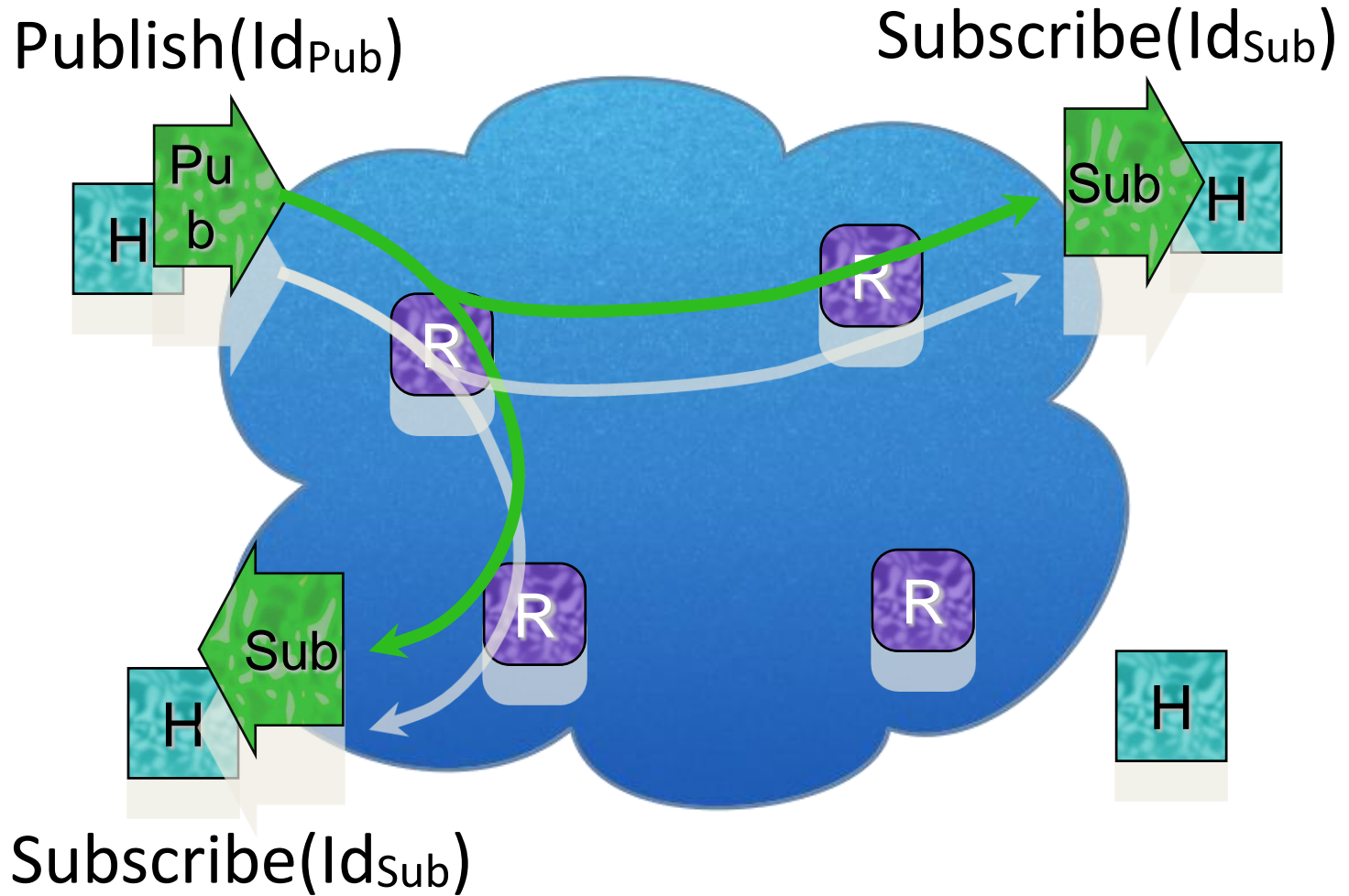
Compensation Mechanisms

- Needed to build a new marketplace where publishing and subscribing have a price
- In the core of the network, not a per-application solution
- Mechanism may vary from basic authentication (home WLAN) to business agreements (between ASes)
- Effective method to reduce the SPAM and DDoS problems?

Rendezvous, routing & forwarding

- Rendezvous
 - How subscription and publication are matched?
 - If IDs are flat, then maybe a DHT solution
- Routing
 - Based on multicast delivery trees that are pre-built
- Forwarding
 - Configured by routing

Functional model

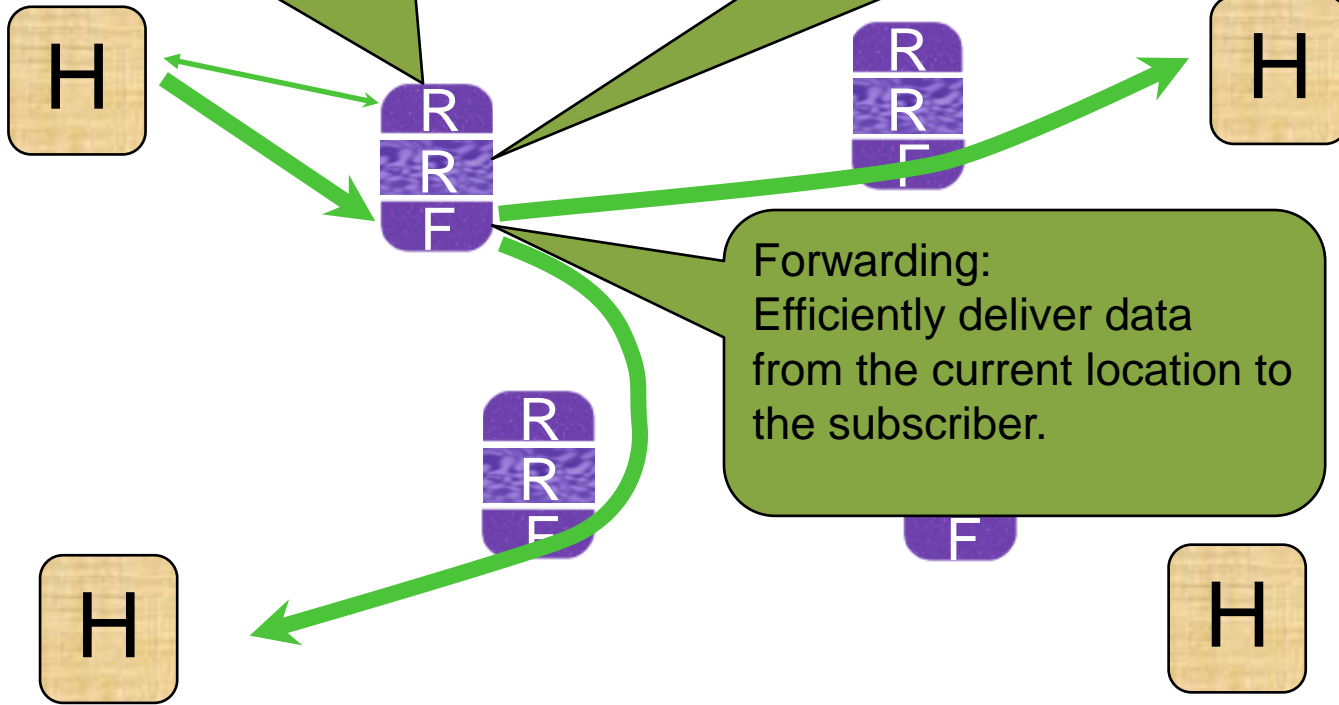


Three-layer architecture

Rendezvous:
Maintain publication
information, find the right
publication when
subscribed

Routing:
Make routing decisions,
how to build a route from
the publications location
to the subscriber

Forwarding:
Efficiently deliver data
from the current location
to the subscriber.



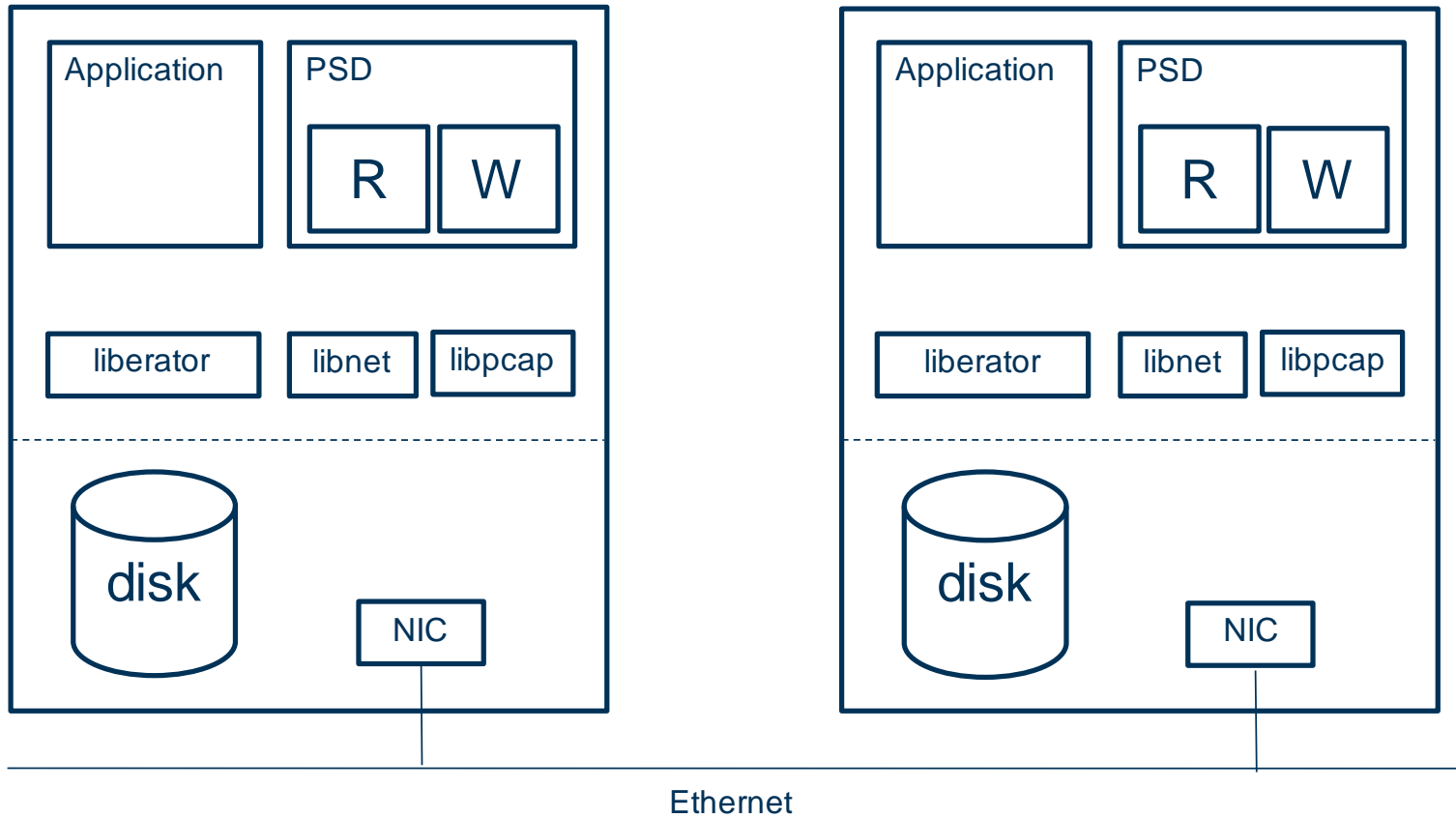
Prototype

- A prototype implementing a publish-subscribe type of communication interface between applications
- Implemented completely in Linux userspace
- Everything above link layer implemented "from scratch"
- Stack internally using pubsub-type approach
 - No "vertical stack": applications and network managers using the same "blackboard"
- Currently running over Ethernet
 - Practical to implement
 - Ethernet addresses are ignored
 - Using Ethernet as a broadcast channel

Prototype (2)

- Currently implemented
 - Publishing and subscribing of static files
 - Simple rendezvous, routing and forwarding
 - Fragmentation support
- Future
 - Compensation mechanisms
 - Inter-domain RRF
 - Support for all types of applications (stream,...)
 - Unifying file system and networks

Prototype Architecture



Conclusions

- PubSub vs. send-receive
 - Huge change in *thinking* regarding networking
- PubSub internetworking architecture
 - First ideas
 - 1st prototype up and running
- PSIRP EU project starting in 2008
 - **P**ublish-**S**ubscribe Internet **R**outing **P**aradigm
 - 8 partners, 2.5 y, 335 MM, 2.6 M€ EU contribution
 - Everything from link layer to application layer
- The work has just begun...
 - More open questions than answers

Questions?
Thank you!