Publish/Subscribe Internetworking

Prof. Sasu Tarkoma

Department of Computer Science and Engineering
Helsinki University of Technology

11.2.2009

Contents

Introduction

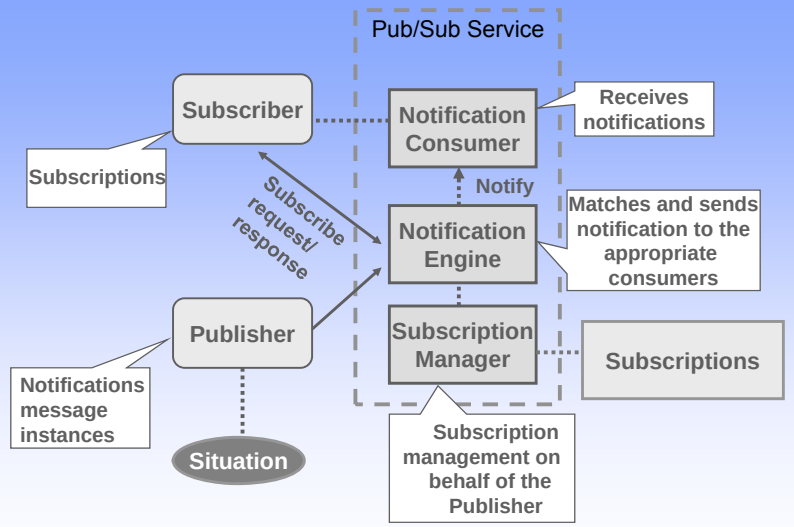
- Information dissemination solutions are needed by many distributed applications
  - Content delivery
    - News, alerts, stock market information, metadata, presence information,…
  - Monitor data (sensors)
  - Control data (actuators, robots,…)
- Motivates research and development of efficient distributed dissemination systems
  - Event-based systems and Publish/Subscribe
  - Reusable building blocks for high-level routing

Event-based Systems and Publish/subscribe

- Event delivery from publishers to subscribers
  - Event is a message with content
  - One-to-many, many-to-many
  - Builds on messaging systems and store-and-forward
- A frequently used communication paradigm
  - Decoupling in space and time
  - Solutions from local operation to wide-area networking
  - Proposed for mobile/pervasive computing
- The event service is a logically centralized service
  - Basic primitives: subscribe, unsubscribe, publish
- Various routing topologies and semantics

Subscriptions

- Subscriptions are described using filters
  - Filter: a stateless Boolean function
    - Defines a subspace of the content space
    - A single event is a point in the content space
  - Selects a subset of published events
  - Expressive interest definition and content-matching
  - Content model is typically typed tuples or XML



Event Systems I

- Traditional MoM systems are message queue based (one-to-one)
- Event systems and publish/subscribe are one-to-many or many-to-many
  - One object monitors another object
  - Reacts to changes in the object
  - Multiple objects can be notified about changes
- Events address problems with synchronous operation and polling
- In distributed environments a logically centralized service mediates events
  - anonymous communication
  - expressive semantics using filtering

Event Systems II

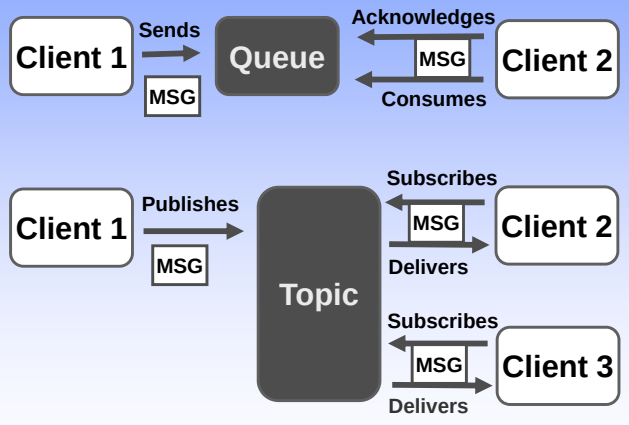
- Push versus Pull
- May be implemented using RPC, unicast, multicast, broadcast,..
- Three main patterns
  - Observer design pattern
    - Used in Java / Jini
  - Notifier architectural pattern
    - Used by many research systems
  - Event channel
    - Used in CORBA Event/Notification Service
- Filtering improves scalability / accuracy
  - Research topic: content-based routing

Tuple Spaces

- Tuple-based model of coordination
- The shared tuple space is global and persistent
- Communication is
  - decoupled in space and time
  - implicit and content-based
- Primitives
  - **In**, atomically read and removes a tuple
  - **Rd**, non-destructive read
  - **Out**, produce a tuple
  - **Eval**, creates a process to evaluate tuples
- Examples: Linda, Lime, JavaSpaces, TSpaces

Java Message Service (JMS)

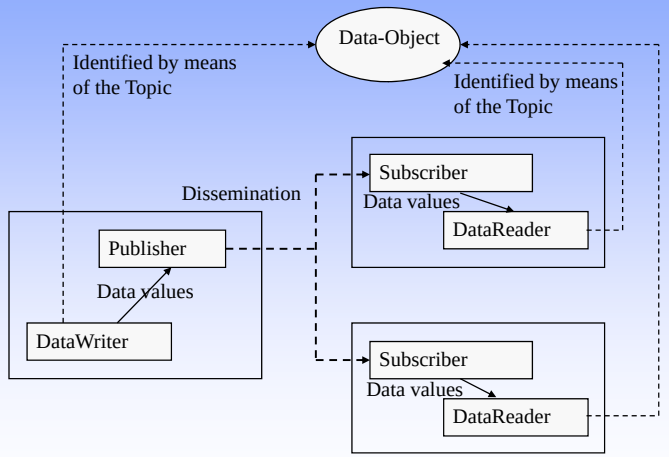
- Asynchronous messaging support for Java
- Point-to-point messaging
  - One-to-one
- Topic-based publish/subscribe
  - SQL for filtering messages at the topic event queue
  - One-to-many
- Message types:
  - Map, Object, Stream, Text, and Bytes
- Durable subscribers
  - Event stored at server if not deliverable
- Transactions with rollback



OMG Distributed Data Service I

- The Data Distribution Service for Real-Time Systems (DDS)
- The specification defines an API for data-centric publish/subscribe communication for distributed real-time systems.
- DDS is a middleware service that provides a global data space that is accessible to all interested applications.
- DDS uses the combination of a Topic object and a key to uniquely identify instances of data-objects.
- Content filtering and QoS negotiation are supported
- DDS is suitable for signal, data, and event propagation.

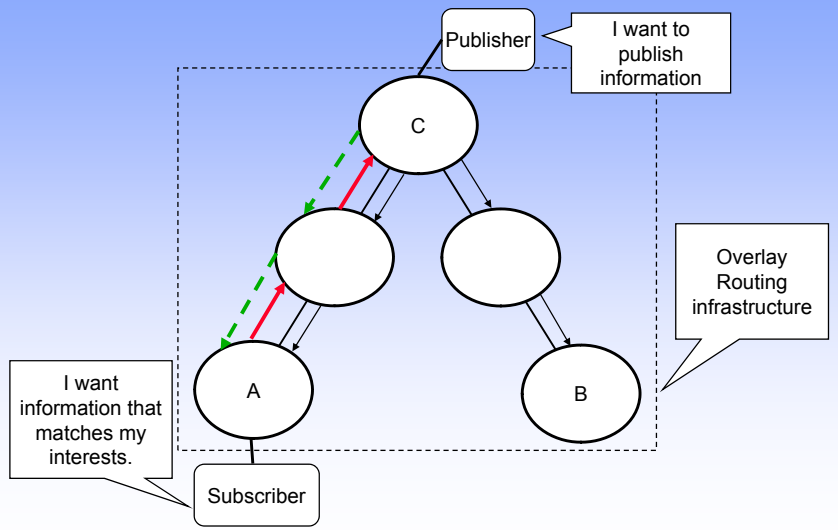
DDS



Content-based Routing

- Filters select events based on the content of event messages
- Can be seen as content-based addressing
- More expressive than topic, subject, or header-based routing
- Research projects and prototype systems
  - Siena, Rebeca, Hermes,..
- Three central design considerations
  - Router topology
  - Interest propagation mechanism
  - Filtering language

Example of Siena-style Content-based Routing



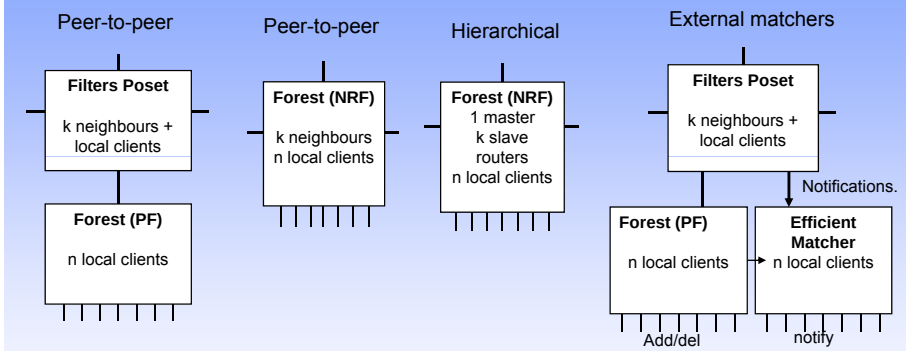
Filters

- A filter is a stateless Boolean function that takes a notification as an argument
- Filter F1 is said to cover filter F2 if and only if all the notifications that are matched by F2 are also matched by F1.
- The covering relation is a partial order (transitive, anti-symmetric)
- If filters are organized in a graph based on the covering relation, several optimizations are possible.
- The **root set** of the graph is the **minimal cover set**

Data Structures for Cover based Routing

- Filters Poset
  - For hierarchical and peer-to-peer routing (acyc/cyc)
  - A directed acyclic graph structure that stores the direct predecessors and successors for each filter.
  - Each filter has a **subscribers** set.
  - Two first levels are used to compute forwarding information (**forwards** sets)
    - Filter S from X
      - All filters from X that are covered by S are removed.
      - Never forward S from X to X or to interfaces where a covering filter has been already sent.
    - Unsubscription may result in uncovered filters to be sent (direct successors of deleted filter)
- Poset-derived forest
  - Data structure with support for fast add/del operations.
  - Forest representation based on covering relation
  - Local clients, hierarchical, and peer-to-peer routing

Routing Blocks



This is a set of generic building blocks for filter cover-based routing.

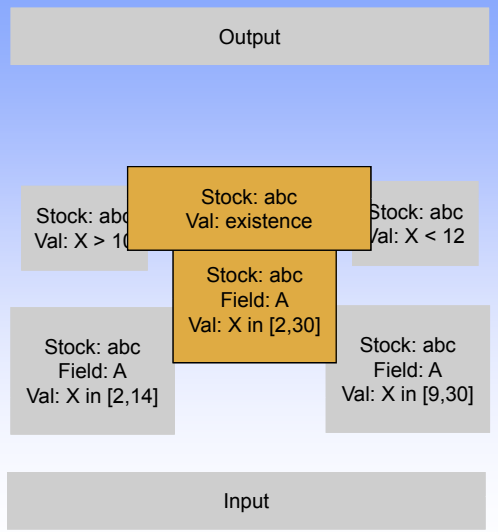
Can be extended with optimizations such as pruning and caching.

Filter Merging

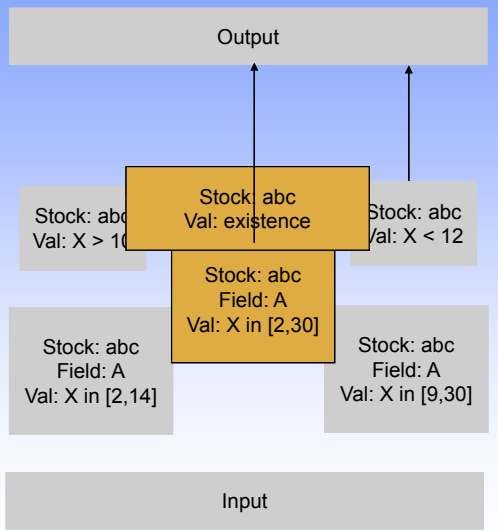
Filter Merging I

- Filter merging is useful, because it allows to further remove redundancy and keep the number of elements minimal.
- There are many ways to perform filter merging.
- We assume that a **merge**(F_1,F_2) procedure exists
  - Returns a single merged filter F_M.
  - F_M covers both F_1 and F_2.
- A merge of two or more filters is called a **merger**.
- A merger is either **perfect** or **imperfect**.
  - A perfect merger does not result in false positives or negatives, whereas an imperfect merger may result in false positives.

Dynamic Filter Merging Example



Dynamic Filter Merging Example



Filter Merging II

- Requirements for filter merging
  - Merging must be transparent for applications and routers.
  - An insert of **x** may result in a new merged node **M** = **merge(x,y)**, but after the delete of **x** the resulting node must cover **y**.
- Formal framework with filter merging rules
  - Keep track of the **components** of a merger and the nodes that are **covered** by the merger
    - Example: **x** and **y** are components of **M**.
  - Does not specify how the selection of merging candidates is done or how possible re-merging after delete is realized
- Filter merging may be applied in different places in the event router.

Rules for Merging I

$F_1 \sqsupseteq F_2$ => Del(F_2) on data structure, if they are from the same interface.

$F_1 \sqsupseteq M_1$ => Del(M_1)

Del(M_1) => Remove M_1 and reset its covered nodes and components (first rule is applied as well)

Rules for Merging II

$M_1 \sqsupseteq F_1$ => Add F_1 to M_1's covered set

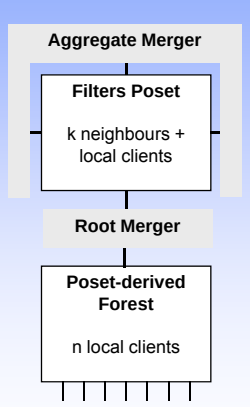
$M_1 \sqsupseteq M_2$ => Add the components and covered nodes of M_2 to M_1 and remove M_2.

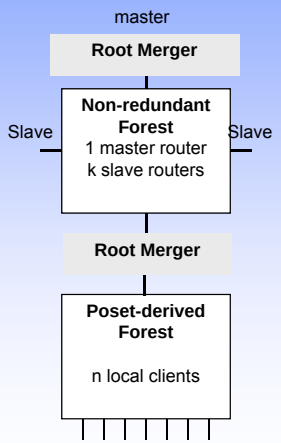
Del(x) and x is a component of a merger M_1 => Remove M_1 from MR, reset M_1's components and covered nodes. This makes them available for re-merging.

Dynamic Merging

- Goal to have a simple and efficient mechanism
  - Optimal merging is a hard problem
  - Should be linear time for practical usage
  - Opportunistic operation
- Keep track of merged elements, elements covered by mergers, and non-covered elements
- Perform merging only on the root set (or 2 first levels)
  - Triggered when the root set changes due to addition or removal of an element
  - Add: given new element, scan root set for merger, then scan for covered elements
  - Del: simply remove element from sets
    - Restore uncovered elements (and non-deleted components of merger) to the root set and attempt to merge them

Merging Blocks

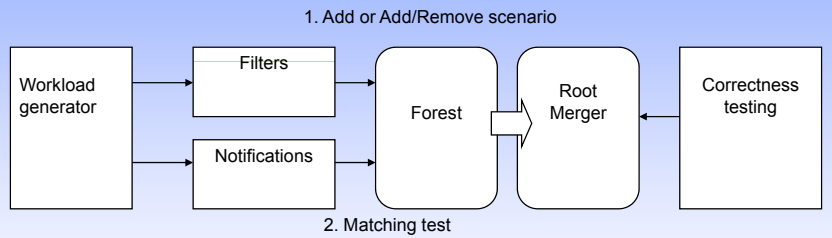




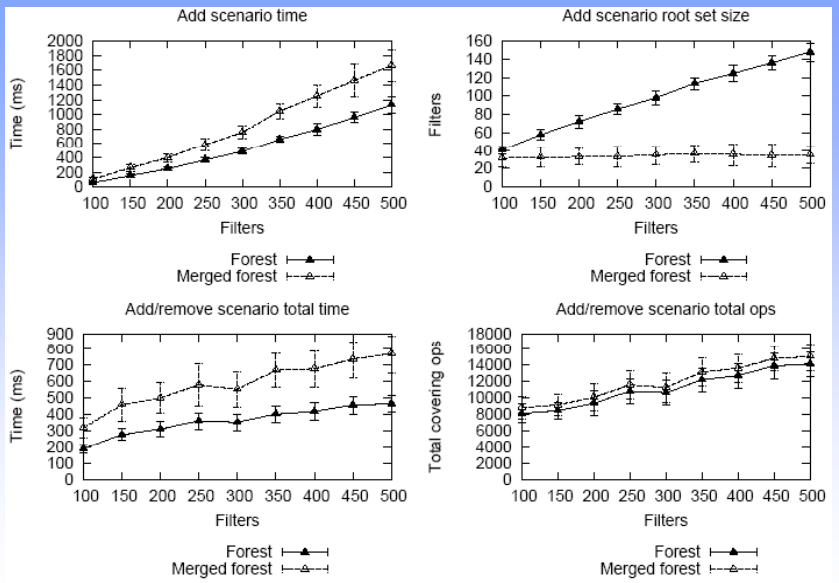
Filter Language and Merging

- The formal framework supports different filter merging mechanisms.
  - Hardcoded rules, merging procedures, rules from ontologies
    - Merging rules may also be approximate (resulting in false positives)
- We use a simple typed tuple model for experiments..
  - Each filter is a set of attribute filters with unique name and type.
  - Relational operators are supported.
    - *Selected from* {<, >,≤ ,≥ ,=, !=, [a, b]} using a uniform distribution.
  - Two filters are mergeable if they have only one differing attribute filter that can be merged.

Root Merger Benchmarks

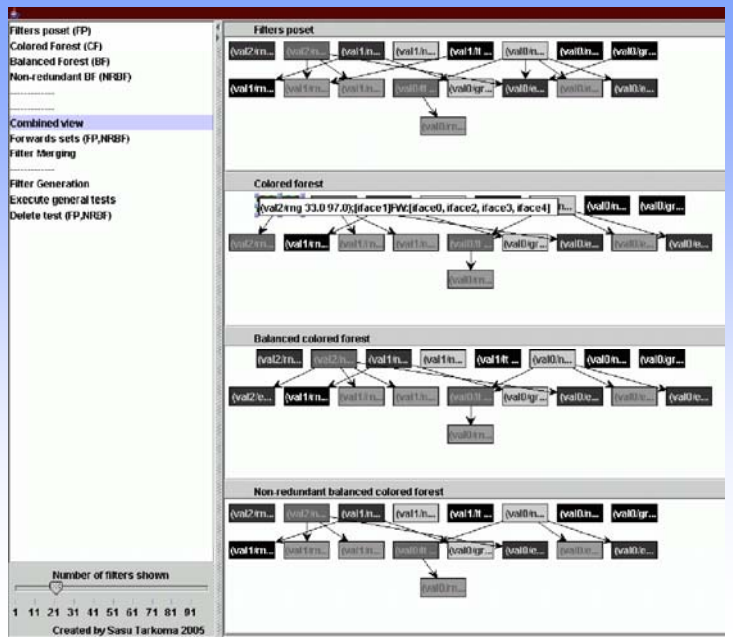


2D Filters



100 filters replaced in Add/Remove

PosetBrowser



Articles on Routing Blocks

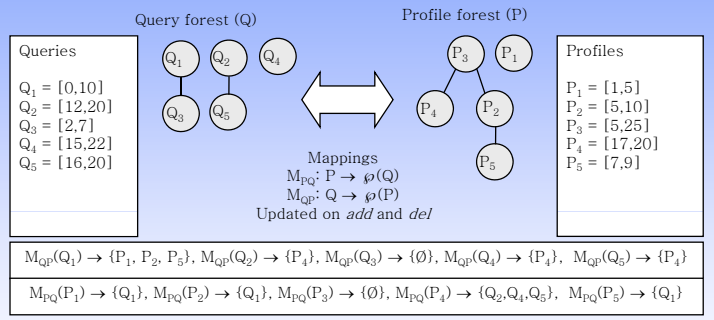
- S. Tarkoma and J. Kangasharju. Optimizing Content-based Routers: Posets and Forests. Distributed Computing 19 (1), September 2006.
- S. Tarkoma and J. Kangasharju. Filter Merging for Efficient Information Dissemination. In 13th International Conference on Cooperative Information Systems, Lecture Notes in Computer Science 3760, Springer-Verlag, October 2005.
- S.Tarkoma. Dynamic Filter Merging for Publish/Subscribe. Accepted as Extended Paper, IEEE WoWMoM 2007.
- S.Tarkoma. Dynamic Filter Stream Detection and Merging for Publish/Subscribe. Elsevier Pervasive and Mobile Computing (Fast Track paper). 2008. Volume 4, Issue 5, Pages 579-788 (October 2008)

Chaining Forests

DoubleForest: Chaining Two Forests

- DoubleForest data structure
  - Combines two poset-derived forests for generic context matching.
  - One forest for queries, one for profiles.
- Support for both subspace matching and temporal profiles.
- Mappings are updated during add/del.
- Optimizations possible using forest structure.
- Experimental results indicate that overlap based matching has more overhead than cover based matching.

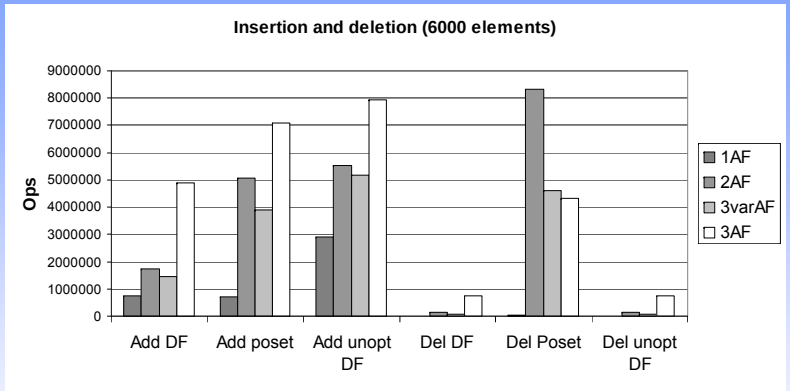
Overview



Chained Forests

- Generalizes to multiple sets, each corresponding to a forest in a chain of forests
- Separates mappings from the forests.
- Improved insertion and lookup cost. Improved or degraded deletion cost and structure size depending on the workload. Sparse bitstrings may be used to alleviate space concerns.
- Nature of optimizations suggest that copes well with self-similar workload.
- Future work investigates how to compact mappings

Results



	DF	Poset	Forest
1AF	1760497	14023	3484
2AF	544265	243826	5998
3varAF	779112	185338	5852
3AF	87026	208447	6000

Size after inserting 6000 elements

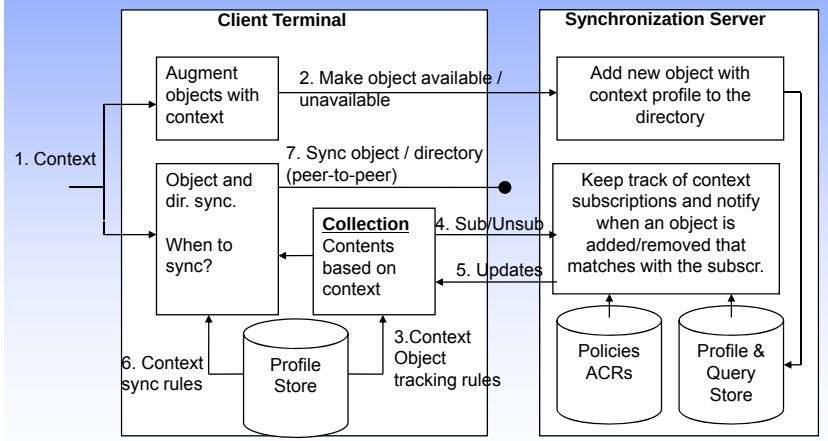
	DF	Poset	Forest
1AF	655	5724	317695
2AF	675	3066	637056
3varAF	667	3200	406348
3AF	719	673	937292

Time (ms) for 30 000 lookups for 3000 elements

Filters and Context

- We propose to represent context using filters
  - Support for points and subspaces (for example ranges)
  - Use filters for both queries and profiles.
    - A query defines a collection and is matched against profiles.
    - A profile describes the context of an object
    - Two semantics: cover and overlap
- A set of user interest → context query → filter → subspace of context space
- A set of context metadata → context profile → filter → subspace of context space

DataSpace Architecture



Articles

- S. Tarkoma, T. Lindholm, J. Kangasharju. Collection and Object Synchronization Based on Context Information. 2nd IEEE/IFIP International Workshop on Mobility Aware Technologies and Applications, 2005.
- S. Tarkoma. TSR: Temporal Subspace Routing for Peer-to-Peer Data Sharing. IEEE Globecom 2006.
- S. Tarkoma. Chained Forests for Fast Subsumption Matching. ACM/IEEE/Usenix DEBS 2007.

Publish/Subscribe Spam Prevention

Pub/Sub Spam

- Typical email spam is unsolicitated bulk mail that clutters inboxes
- SIP spam is envisaged to take many forms
  - IM spam, presence spam, voice spam
- Pub/sub spam has many dimensions
  - Both control and content aspect
  - Different characteristics than email/SIP
    - Different targeting mechanism
      - Filters define end-points and delivery paths instead of explicit addresses
    - Multi-hop networks
    - Many-to-many nature
  - If filters are used for clustering/ranking/recommendations, we can have ”filter engine” spam

Lessons from Email Spam

- Spammers have vast resources at their disposal
  - Zombie machines, cheap labor,..
  - No technique alone seems to be sufficient
  - Obfuscatory techniques such as open relays, spoofing, and zombies make it difficult to track spammers
- Sender authentication is key
  - Well-known solutions include IP-based and crypto-based techniques: SPF, Yahoo’s DomainKeys

Solution Space

- Solution space includes the following
  - Distributed Blackhole Lists, which collect IP addresses of known spammers
  - Crypto-based sender verification
  - Greylisting, in which messages with unseen (IP,sender,receiver) triplets are delayed
  - Rate limiting
  - Proof-of-work techniques
  - Content filtering for outbound and inbound messages
  - Access control using buddy-lists and social networks
    - These can also be used for spamming

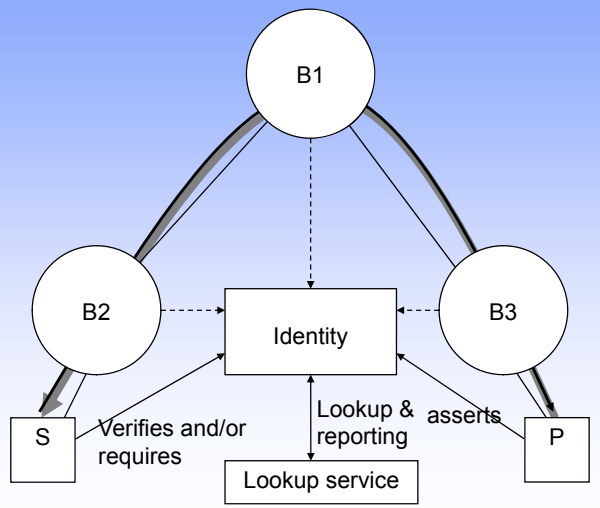
Content-based Spam

- The key issue in pub/sub spam is *event replication*
- A powerful technique for scalable dissemination, but it is also a possible spamming technique
- To ensure wide-scale dissemination of spam, spammers must
  - Estimate filters that are widely subscribed
  - Create content that matches those filters
- In order to find popular filters, spammers may infiltrate the broker network
  - Bogus brokers monitor traffic, possibly drop messages
  - Redundancy is needed to detect and mitigate bogus brokers

Preventive Measures

- The aim of infrastructure-based spam prevention is to incorporate preventive measures into the core routing network
  - Drop unwanted messages as near the source as possible
- Current systems have not explicitly addressed spam prevention
  - Some solutions available in the form of security services.
  - Real workloads and traces needed
- Key aims:
  - Preventing bogus brokers and publishers
  - Preventing black hole queries and advertisements

Identity-based Spam Prevention



Principles for Spam Prevention

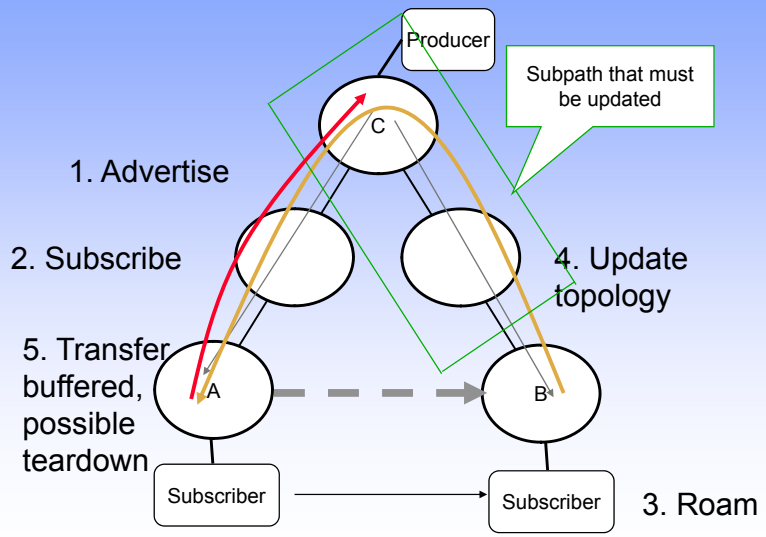
- Publisher and broker authentication to prevent bogus brokers and publishers
- Distributed blacklisting using DNS or a DHT
- Filter set generalization using techniques such as filter covering and merging
- Detecting spam sources that try to systematically cover the content space by flooding
- Preventing black hole advertisements from brokers and black hole subscriptions from clients
- Keeping popular filters secret
- Article: S. Tarkoma. Preventing Spam in Publish/Subscribe. IEEE DEBS 2006 (in conjunction with ICDCS).

Mobility Support

Challenges with Mobility Support

- How to cope with mobile users?
  - Disconnected operation
    - Buffering and queue management
  - Mobile subscribers / publishers
    - Handover protocol for relocating subscriptions and updating the topology
    - Multiple indirection points on the overlay network
    - Covering/merging complicate mobility support
  - General requirements
    - fast convergence of the subscription topology
    - mobility-safety: no false negatives

Example Handover



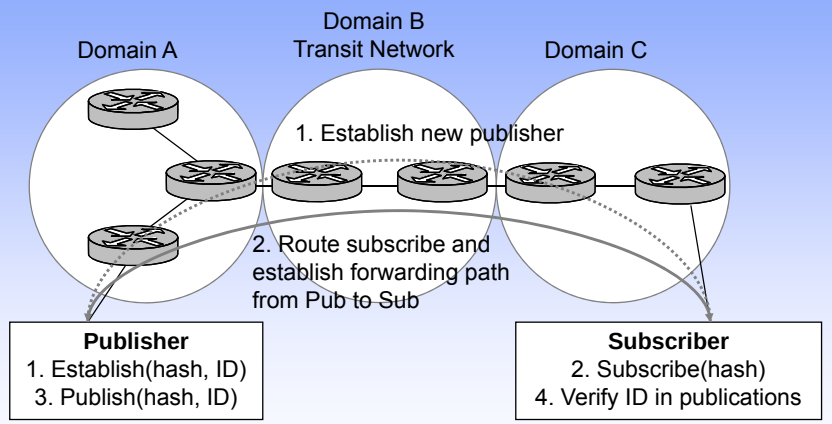
Articles

- S. Tarkoma and J. Kangasharju. On the Cost and Safety of Handoffs in Content-based Routing Systems. Elsevier Computer Networks Journal. 2007. Available at: http://dx.doi.org/10.1016/j.comnet.2006.07.016
- S. Tarkoma and J. Kangasharju. Handover Cost and Mobility-Safety of Content Streams. In Eighth ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, October 2005.
- S. Tarkoma, J. Kangasharju, K. Raatikainen. Client Mobility in Rendezvous-Notify. International Workshop for Distributed Event-Based Systems (DEBS03), in conjunction with the ACM SIGMOD/PODS Conference, San Diego. Available at ACM Digital Library.

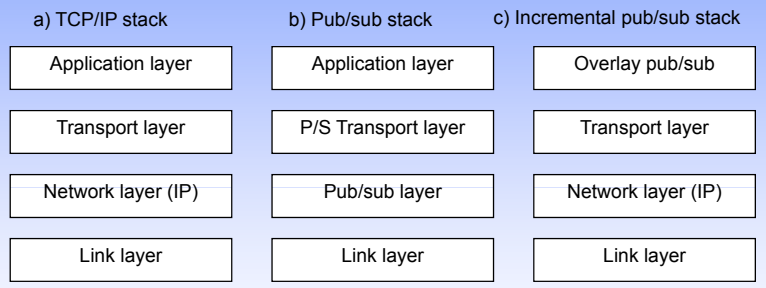
Towards Pub/Sub Internetworking

Rethinking Naming, Trust, and Primitives

- Current Internet architecture is sender-oriented
  - Unwanted traffic
  - Connectivity problems
- We propose a future Internet that gives more control to the receiver
  - **Publish/Subscribe** model
  - Only end-points described using interests and local links
  - Basic routing using flat self-certifying labels
  - **Data-centric** routing, forwarding, rendezvous
- Two approaches:
  - **Incremental** with overlay networks
  - **Radical clean-slate** approach with a new networking stack: **Internet without IP**
- Related work: UIP, FARA, TRIAD, Nimrod, i3, DOA, ROFL, DONA, AIP



Looking at Layers



Summary

- Publish/subscribe supports many-to-many information networking
- Construction of routing systems using reusable building blocks, namely posets and forests
- Recent research on
  - Spam prevention
  - Pushing pub/sub down the stack
  - Efficient matching
- Live demos on the web
  - http://www.tml.tkk.fi/~starkoma/fc/

Publish-Subscribe Internet Routing Paradigm

PSIRP

Prof. Sasu Tarkoma
Helsinki Institute for Information Technology
Helsinki University of Technology

Contents

Observation

Fundamentals of the Internet

- Collaboration
  - Reflected in forwarding and routing
- Cooperation
  - Reflected in trust among participants
- Endpoint-centric services (mail, FTP, even web)
  - Reflected in E2E principle

⇒ **IP, full end-to-end reachability**

VS.

Reality in the Internet Today

- Phishing, spam, viruses
  - There is no trust any more!
- Current economics favor senders
  - Receivers are forced to carry the cost of unwanted traffic
- Information-centric services
  - Do endpoints really matter?
  - Endpoint-centric services move towards information retrieval through, e.g., CDNs

⇒ **IP with middleboxes & significant decline in trust in the Internet**

Hypothesis: Clean-Slate Design Required

- What stood at the beginning
  - Collaboration
  - Cooperation
  - Endpoint-centric services

  does not seem enough
- What about:
  - Trust?
  - Information centrism?
  - Legitimacy of E2E?
  - Role of overlays?

Clean-slate design…

- Question ALL fundamentals
- Challenge our thinking
- Take nothing for granted, including industry structures
- Clear vision

…with late binding (to reality)

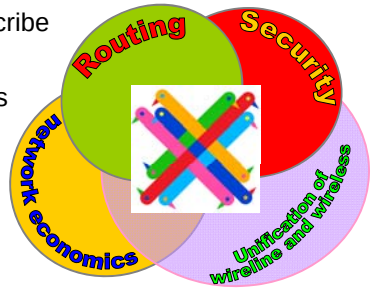
- Consider migration and evolvability in separate work items
  - How to get our design into real deployments, e.g., overlay vs. IP replacement?
- Consider necessary evolution of industry (and regulatory) structures
  - How do industries need to evolve in certain scenarios?

PSIRP PUBLISH-SUBSCRIBE INTERNET ROUTING PARADIGM

Vision

Envision a system that dynamically adapts to evolving concerns and needs of their participating users

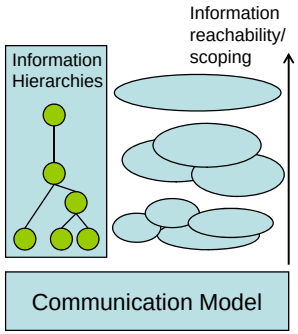
- Publish–subscribe based internetworking architecture restores the balance of network economics incentives between the sender and the receiver
- Recursive use of publish-subscribe paradigm enables dynamic change of roles between actors



PSIRP PUBLISH-SUBSCRIBE INTERNET ROUTING PARADIGM

Main PSIRP design principles

- Information is multi-hierarchically organised
  - Higher-level information semantics are constructed in the form of directed acyclic graphs (DAGs), starting with semantic free forwarding labels towards higher level concepts (e.g., ontologies).
- Information scoping
  - Mechanisms are provided that allow for limiting the reachability of information to the parties having access to the particular mechanism that implements the scoping.
- Scoped information neutrality
  - Within each scope of information, data is only forwarded based on the given (scoped) identifier.
- The architecture is receiver-driven
  - No entity receive data unless it has agreed to receive the data beforehand, through appropriate signalling methods.



PSIRP PUBLISH-SUBSCRIBE INTERNET ROUTING PARADIGM

Potential Impact of our Work

User

- Relevant Information at your fingertips
  - Wherever, from whoever, through whatever access, on whatever device
- More natural form of communication
  - Emulates sensing, processing, actuation
- Ability to avoid information overload
  - Tackle attention scarcity problem
- Increased security & privacy
  - Only relevant information gathered & provided to user

Industry

- Entry of new players, e.g., information brokers, information processing providers
- Content providers likely to become more powerful
- New technology means potential for new business
- Increase in (information-centric) communication needs will increase need in solutions
- Enable cross-value chain scenarios
  - retail, health, …

PSIRP PUBLISH-SUBSCRIBE INTERNET ROUTING PARADIGM

Project Objectives

- Specify, implement and test an internetworked pubsub architecture
  - follow **clean-slate design** approach
- Perform qualitative and quantitative evaluation
  - Security and socio-economics important!
  - Migration and incentive scenarios important (e.g., overlay)!
- The results will be widely published
  - Open source code for the Future Internet
  - Targets specifically SMEs opportunities in Future Internet
- Engage with FI community
  - Cooperate with FIRE (Onelab2) to test on large scale
  - Engage openly through public Wikis

Project Overview

Project Coordinator
Arto Karila
Helsinki University of Technology, HIIT
Tel: +358 50 384 1549
Fax: +358 9 694 9768
Email:arto.karila@hiit.fi

Partners:
- Helsinki University of Technology
  Helsinki Institute for Information Technology (FI)
- RWTH Aachen University (DE)
- British Telecommunications Plc (GB)
- Oy L M Ericsson Ab (FI)
- Nokia Siemens Networks Oy (FI)
- Institute for Parallel Processing of the Bulgarian Academy of Science (BG)
- Athens University of Economics and Business (GR)
- Ericsson Magyarorszag Kommunikacios Rendszerek K.F.T. (HU)

Duration: January 2008 – June 2010
Total Cost: €4.1m
EC Contribution: €2.5m
Contract Number: INFSO-ICT-216173

WP1 Management (TKK-HIIT)

WP2 Architecture Design (TKK-HIIT)

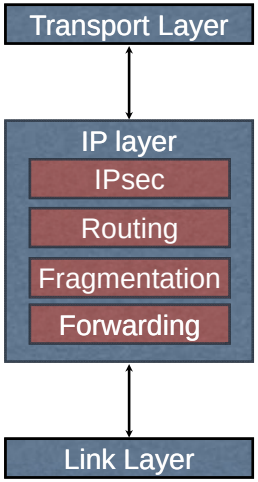
WP3 Implementation, Prototyping & Testing (LMF)

WP4 Validation and Tools (BT)

WP5 Dissemination and Exploitation (NSNF)

Project website: **www.psirp.org**

Current State



Observations

End-to-end reachability is broken

Unwanted traffic is a problem

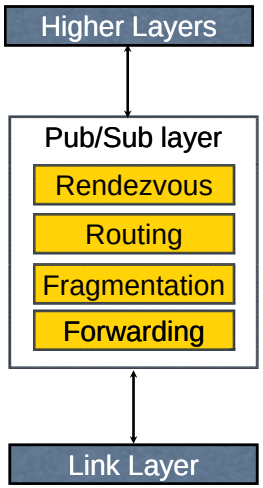
Mobility and multi-homing are challenging

Multicast is difficult (does not scale)

Security is difficult

Not optimal fit for broadcast and all-optical networking

Where we are going



Observations

No topological addresses, only labels

Security enhanced using self-certification

End-to-end reachability, control in the network

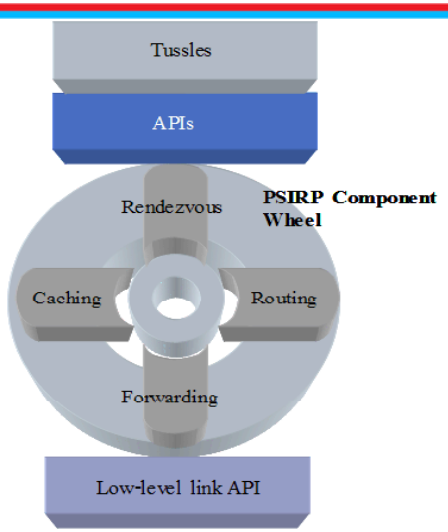
Natural support for multicast, it is the norm

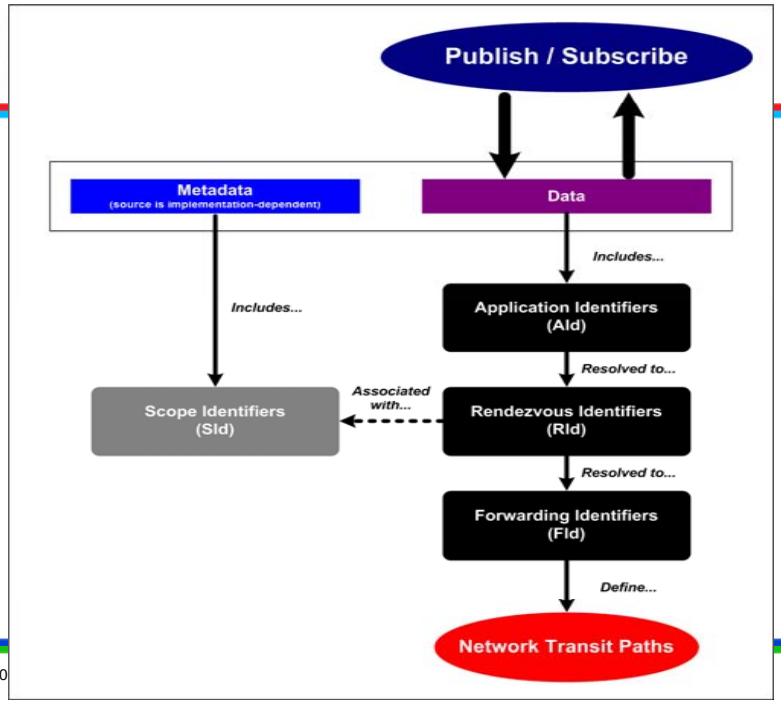
Support for broadcast and all-optical label-switching technologies

Dynamic state is introduced into the network

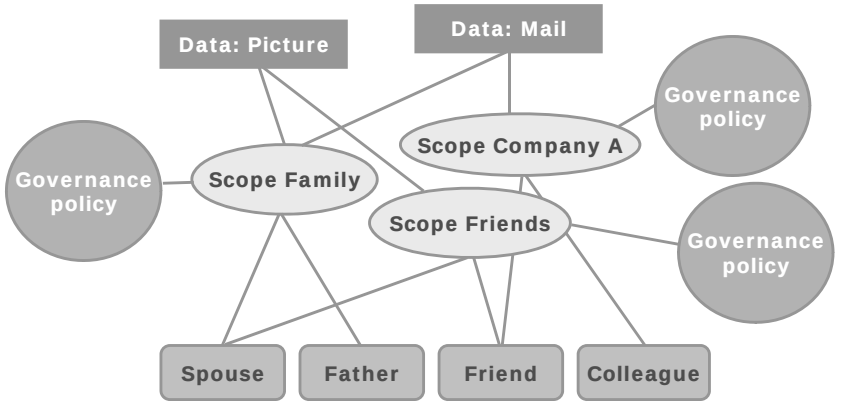
How do we make it scale?

Component Wheel

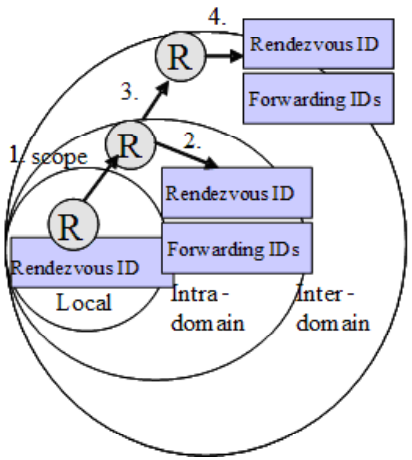




Scopes



Rendezvous and Forwarding



Packet Level Authentication (PLA)

- We assume that per packet public key cryptography operations are feasible in Internet's scale because of new digital signature algorithms and advances in semiconductor technology
- PLA is a novel solution for protecting the network infrastructure against various attacks (e.g., DoS) by providing availability
- The network should be able to fulfill its basic goal: to deliver valid packets of valid users in reliable and timely manner in all situations

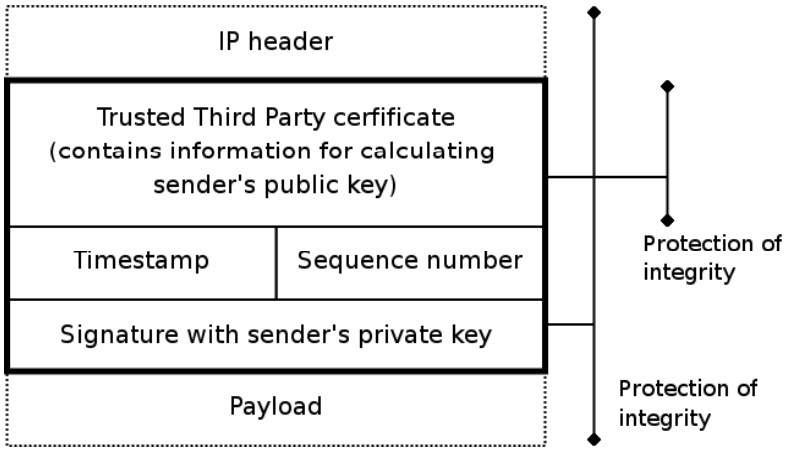
PLA continued

- The main aim of PLA is to make it possible for any node to verify authenticity of every packet without having previously established trust relation with the sender of the packet
  - Malicious packets can be detected and discarded quickly before they can cause damage or consume resources in the rest of the network
  - Good analogy for PLA is a paper currency: anyone can verify the authenticity of the bill by using built-in security measures like watermark and hologram, there is no need to contact the bank that has issued the bill

PLA continued

- PLA accomplishes its goals by using public key digital signature techniques. PLA adds an own header to the packet using standard header extension technique
  - The PLA header contains all necessary information for detecting modified, duplicated and delayed packets
  - PLA complements existing security solutions instead of replacing them. PLA can work together with other security solutions such as Host Identity Protocol (HIP) and IPSec
- Initial PLA implementation has been built on top of IPv6, however PLA is not dependent on the network layer protocol used and it can be also be positioned on top of layer 2 protocols

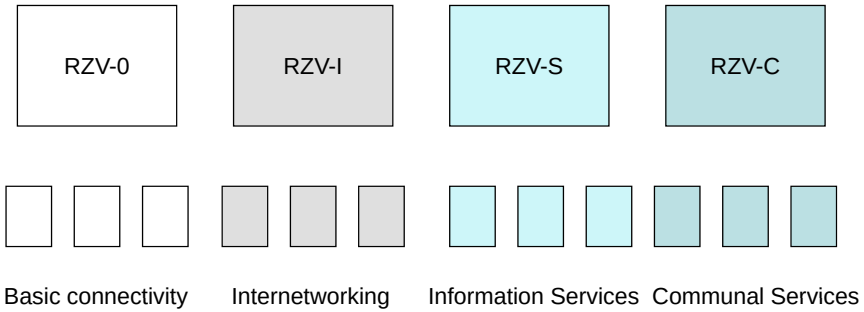
PLA Header



PLA Performance

- With the help of dedicated hardware acceleration, per packet public key cryptography is scalable to high speed core networks and mobile devices
  - Simulation results show that an FPGA based accelerator developed for PLA is capable of performing 166,000 verifications per second
  - Transferring the design into a 90nm ASIC using Altera's Hardcopy technology would improve performance to 850,000 verifications per second with power consumption of 26µJ per verification
  - Such performance would be enough to verify 50Gbps of traffic with jumbo frames (60kbits of payload per frame)

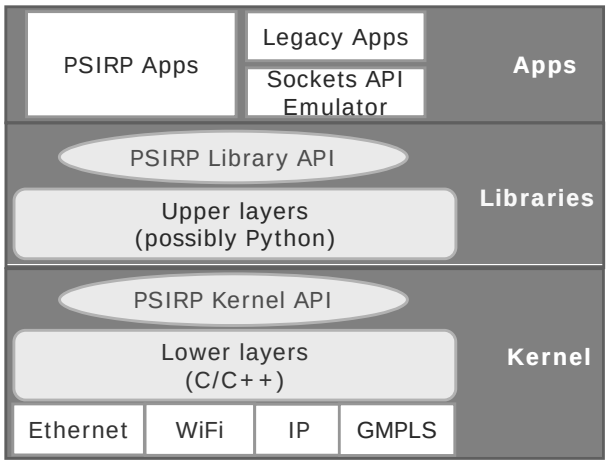
Many Faces of Rendezvous



Rendezvous

- The network is defined in terms of domains and their interconnections
  - Interconnections between domains include upstream, transit, downstream
- Rendezvous is the central primitive
  - Rendezvous on multiple layers
  - Builds forwarding paths
- We utilize the notion of completeness to optimize processing and mobility updates
  - Complete / incomplete dissemination structures between rendezvous points
  - A structure is complete when the operation (sub, adv) has been processed by all elements that should process it → typically partial in a global network
  - Completeness can be used for network diagnostics

Implementation



Dissemination

- Deliverables and Technical Reports
  - State of the Art Report and Technical Requirements (D2.1)
  - Conceptual Architecture (D2.2)
  - Prototype Platform and Applications Plan and Definition (D3.1)
  - Preliminary Validation Plan and Selection of Tools (D4.1)
  - Dissemination and Exploitation Plan (D5.2)
  - From Design for Tussle to Tussle Networking: PSIRP Vision and Use Cases (TR08-0001)
  - LIPSIN: Line Speed Publish/Subscribe Inter-Networking (TR09-0001)
- Publications
  - RTFM: Publish/Subscribe Internetworking Architecture. Mikko Särelä, Teemu Rinta-aho, Sasu Tarkoma. Mobile ICT Summit 2008.
  - Towards Understanding Pure Publish/Subscribe Cryptographic Protocols. Nikander, Pekka, Marias, Giannis F. Cambridge Security Protocols Workshop (SPW 2008).
  - Black Boxed Rendezvous Based Networking. Sasu Tarkoma, Dirk Trossen, Mikko Särelä. Mobiarch 2008 — The 3rd ACM International Workshop on Mobility in the Evolving Internet Architecture.
  - Xylomenos G, Katsaros K and Kemerlis V. Peer Assisted Content Distribution over Router Assisted Overlay Multicast. Euro-NF Future Internet Architecture workshop, November 20-21, 2008.
  - Rajahalme J, Särelä M, Nikander P and Tarkoma S. Incentive-Compatible Caching and Peering in Data-Oriented Networks. Re-Arch'08,

Conclusions

- We outlined a information centric network architecture
  - **Publish** and **subscribe** are the basic primitives making **multicast** the norm
  - Receiver driven (subscriber has control)
  - **Rendezvous** as the primitive to connect publishers and subscribers across domains on multiple levels
    - Mapping to forwarding structures
  - **Scoping** to group data into manageable sets
  - Architecture work is iterative
  - Implementation and evaluation are on-going activities