Towards the Future Internet
G. Tselentis et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-539-6-75

Publish/Subscribe for Internet: PSIRP Perspective

Dmitrij LAGUTIN[a,1] , Kari VISALA[a] , Sasu TARKOMA[a]
[a] *Helsinki Institute for Information Technology HIIT / Helsinki University of Technology TKK, Espoo, Finland*

Abstract. Most of the Internet's traffic is data-oriented, while the Internet is based on sending messages to end points. As a result, the efficient multicast is difficult to implement on Internet's scale, and various attacks such as DoS and SPAM are easy to launch. In this paper, we describe a clean slate approach for a publish/subscribe based networking: Publish/Subscribe Internet Routing Paradigm (PSIRP). PSIRP aims to implement publish/subscribe networking without relying on existing networking protocols such as IP. Preliminary results suggest that a clean slate publish/subscribe approach is flexible and scalable to the Internet.

Keywords. Future Internet, publish/subscribe networking, network security

Introduction

The current Internet is a message-oriented system where packets are exchanged between end-hosts. The sender is completely in charge of the communication, and the network simply complies with the wishes of the sender by delivering content to the receiver. Such an approach has several downsides. First, the end-to-end message-oriented architecture is complicated and inflexible in many cases, for example an efficient multicast is practically impossible to implement on the Internet scale. Furthermore, there are plenty of applications for which the current paradigm is awkward and unnecessarily complex. For example, users are often interested in retrieving the actual information, such as a video or a web page, and do not care about its topological location within the network. Finally, since the sender is in complete control of the communication, sending unsolicited e-mail (SPAM) and launching denial-of-service (DoS) attacks is easy.

The *publish/subscribe* (pub/sub) approach aims to solve these issues by building the architecture around the information, instead of network nodes, and by giving the control of the communication to the receivers. Instead of initiating connections to specific nodes, information is published and interested parties may subscribe to it. The publish/subscribe paradigm also aims to achieve a scalable multicast, which greatly increases network's efficiency for content delivery when the same content is requested by multiple subscribers. Since the information is uniquely identified, caching can be done on the network level. The network also makes sure that subscribers only receive the information they are interested in, effectively preventing most of SPAM and DoS attacks.

[1] Corresponding Author.

There have been several publish/subscribe and data-oriented networking approaches, including i3 [1], DONA [2], ROFL [3], and CCN [4]. Other approaches that aim to solve similar problems include content delivery networks (CDN) and peer-to-peer (P2P) networks. However, these technologies are overlay solutions that do not generally try to change the actual network architecture.

The Publish/Subscribe Internet Routing Paradigm (PSIRP) project differs from existing solutions since it aims to build a publish/subscribe based network from scratch, without relying on existing technologies like IP. A clean-slate approach was chosen since the current IP protocol is oriented towards a message passing. While it is possible to create a publish/subscribe networking solution based on top of IP, a clean-slate solution should be sought for the maximum efficiency and flexibility.

PSIRP's background and goals are described in more detail in [5]. This paper goes in more details in the PSIRP's architecture and implementation. The paper is organized as follows. Section 1 briefly covers related work. Section 2 describes the data-centric publish/subscribe architecture in general, and the example realization of such architecture is shown in Section 3. Section 4 discusses the prototype implementation while Section 5 concludes the paper.

1. Related Work

A data-oriented network architecture (DONA) [2] replaces a traditional DNS-based namespace with self-certifying flat labels which are derived from cryptographic public keys. DONA names are expressed as a <P,L> pair, where P is a hash of a principal's public key which owns the data and L is a label. DONA utilizes an IP header extension mechanism to add a DONA header to the IP header, and separate resolution handlers (RHs) are used to resolve <P,L> pairs into topological routes. Under DONA, data is transmitted as triplets, which include the actual data, the principal's public key, and the signature over the whole data using the principal's private key. Therefore a recipient of the data can verify its authenticity after receiving the complete data item.

The content-centric networking (CCN) [4] is a data-oriented approach where every packet has an unique human-readable name that is often hierarchical. CCN uses two types of packets. Consumers of data send interest packets to the network, and a nodes possessing the data will reply with the corresponding data packet. Since packets are independently named, a separate interest packet must be sent for each required data packet. The CCN routing in its current form is mostly based on flooding, which is a major downside. Even if such an approach might work inside individual domains, it is not scalable to the inter-domain routing on a global level. Hierarchical human readable names used by CCN to separately identify each packet may lead to a high bandwidth overhead.

The routing in CCN is analogous to prefix-based longest match IP forwarding, except names are used instead of IP addresses. This opens the possibility to use existing routers and infrastructure for CCN, but limits how names can be used as the scalability of such solution relies on the efficient aggregation of the names. Mobility, multihoming, and other identifiers structures than hierarchic may need additional routing techniques. On the other hand, the vision of CCN is to move the “waist” of the network stack higher, on the level of named chunks of content. This means that the system could be extended in multiple directions using the thin waist of the system as a

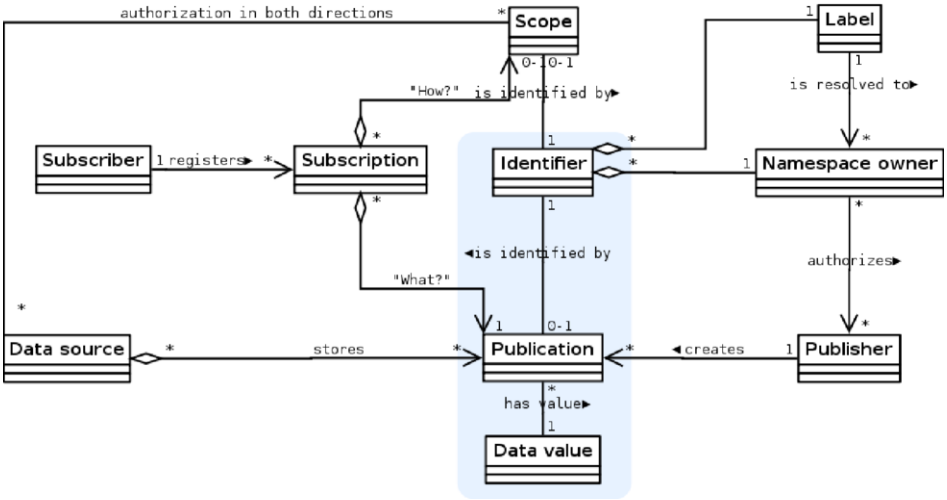

Figure 1: A publication is a persistent association between an identifier and a data value.

glue between extensions and there could be a multitude of technologies used for routing.

2. Data-Centric Publish/Subscribe

In the pub/sub model of communication the sender and the receiver are decoupled in time and space by the publication in the middle. In our case, the publication is a persistent, immutable association between an identifier and the data value of the publication created by the *publisher*. Knowing the identifier, the *subscriber* can retrieve the corresponding data value using a PSIRP network. The basic concepts and their relationships are depicted in Figure 1. We emphasize that this communication primitive is general and may also be used for the real-time communication in addition to the typical existing use case of serving large content files in a fairly static setting.

The immutability of publications enables referential transparency and caching everywhere in the network and shifts the responsibility for synchronization to higher layers. Especially, there is no need to contact any distant data source to verify the freshness of the information, hence the communication can be completely local.

It is also possible to implement a more traditional topic-based pub/sub channel on top of the immutable publications relatively straightforwardly by numbering the events and encoding this information in the identifiers. Because each publication has a single logical publisher or "owner", we can use self-certifying identifiers to bind the content securely to the identifiers. Channels that have multiple publishers can be implemented using an intermediate publisher that "owns" the topic and relays events from multiple sources. Even message-passing can be simulated by subscribing to a special identifier that encodes the identity of the sender and the "publisher" can now subscribe to another publication published by the sender effectively reversing the direction of the communication.

2.1 Identifier types

PSIRP utilizes several types of identifiers. On the high level the information is expressed and structured using *application level identifiers* (Aid) that can have long lifetime, can be fully human-readable, and may name multiple types of objects. These are resolved by some higher level mechanism into *rendezvous identifiers* (Rid), which are understood by the network layer to denote individual publications. The granularity of network level publications may be higher than the data objects used on the application level. Therefore it is possible to efficiently modify only parts of large data structures and communicate only the change while caching the unmodified data.

The architecture introduces the concept of *scope*, which represents a *distribution strategy* for a set of publications published in it. The scope controls among others, access rights of data sources and subscribers, location, reachability, availability, replication, persistence, routing policies, and upstream resources of a publication. Scopes are identified by *scope identifiers* (Sid) that have similar structure to Rids.

The fourth type of identifier, a *forwarding identifier* (Fid), encodes a part of the delivery tree network path and is inserted in the payload packet header by the sender. It also works like a capability, since without a proper Fid, the network will not forward the packet any further. In PSIRP, Fid is based on a 256-bit bloom filter that stores the set of links used for the path segment in the network [9].

2.2 Identifier structure

PSIRP utilizes self-certifying Rids and Sids in order to make the system more secure, and to give ability to nodes to independently verify the authenticity and integrity of information. Similarly as in DONA [2], Rids and Sids are composed of a <P,L> pair, where the P is the public key of the owner of the namespace, and the L is an arbitrary length label that can encode semantical information about the publication. This information is not interpreted by the network because it would violate the end-to-end principle [6]. The payload packets contain only a fixed length hash of such label making the whole Rid and Sid part of the header 256-bit long, but the subscription messages contain the full label information to support dynamic publications, where the data is generated runtime based on the information of the label. While labels can encode human-readable text, the tussle [7] for good names is avoided on the network level by the protected namespaces. For the Aid, we assume that some external mechanism is used to resolve the tussle. For example, the centrally managed DNS system could be used to give human-readable names to our namespaces. Rids themselves cannot be used for long-lived identities as they contain the public key P, which is used as the trust anchor in the signature chain authenticating the publication content. This couples the security implementation to the Rid and makes the publication irrecoverable in case the private key matching to P is leaked.

As can be seen in Figure 1, we have separated conceptually the publisher, which originally creates the publication, from data sources that store and serve the publication for some scope. Often the publisher and the data source are the same entity but there is not necessarily any connection between the publisher and the scopes in which the publication is stored. Both data sources and scopes, which are abstract entities, need to authorize each other. We can say that the abstract scope is implemented by a set of data sources and the rendezvous system. Also, the namespace owner may or may not be the same entity as the publisher. The namespace owner may authorize different publishers to publish only a subset of names inside each namespace.

2.3 Usage of Identifiers

When a subscriber creates a new subscription, she must provide the network with both Sid and Rid. We can think the Rid answering the question “what?” and Sid the question “how?”. For example, the Rid could be *(<current public key of NYT company namespace>, "New York Times, Friday, January 15th, 2010")* and the Sid could be *(<current public key of Google namespace>, ”My family scope”)*. An example of dynamic publication where the identifier can be though as a nullary function could be a label part *“weather (Helsinki, Jan 15 2010)”* that is mapped to content *“sunny”*.

It could be argued that the system would be more general if the subscribers did not need to provide the scope for each subscription, but this creates many kinds of problems because different applications need different network policies. For example, if anyone could advertise globally having some publication, it would open a way to attack the availability of the publication with false advertisements. In general, the scopes are needed because a more abstract interface to the network would hide essential aspects of the communication.

3. PSIRP Architecture

The PSIRP architecture consists of four distinct parts; rendezvous, topology, routing, and forwarding. This section describes a possible realization of the PSIRP architecture.

The rendezvous system acts as a middleman between publishers and subscribers. Basically, it matches the data sources hosting a certain publication with subscribers interested in it in a location-independent way. With the help of the topology function, which manages the physical network topology information, each domain can configure its internal and external routes adapting to error conditions and balancing the network load. The routing function is responsible for building and keeping up the delivery tree for each publication and cache popular content at the branching points inside domains. Finally, the actual publication is delivered to subscribers along the efficient delivery tree by the forwarding function. The detailed overview of the PSIRP architecture is present in Figure 2. A more detailed description of the components can be found in [8].

The main motivation of separating the rendezvous from forwarding and not using hierarchical names directly in the routing tables is to provide scalability for a large number of flat identifiers while the payload can be routed via the shortest policy compliant path. This also makes the location of data sources irrelevant and they can easily be mobile and multihomed. The main motivation for separating the routing and forwarding functions was that they can be scaled independently.

3.1 Domain structure

We will assume that the network consists of autonomous systems similar to the current Internet with similar bilateral contracts between neighbours defining the policy compliant inter-domain paths. Each domain has at least three types of logical nodes: topology nodes (TN), branching nodes (BN), and forwarding nodes (FN) as can be seen

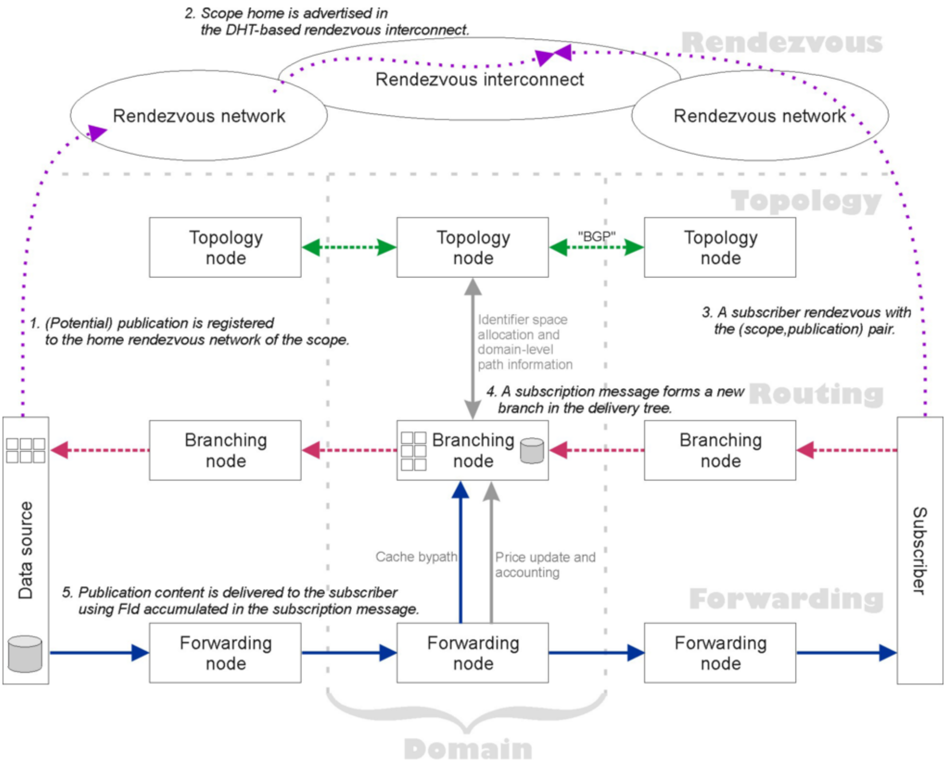
2. Scope home is advertised in the DHT-based rendezvous interconnect.
Rendezvous
Rendezvous interconnect
Rendezvous network
Rendezvous network
Topology
Topology node
Topology node
"BGP"
Topology node
1. (Potential) publication is registered to the home rendezvous network of the scope.
Identifier space allocation and domain-level path information
3. A subscriber rendezvous with the (scope,publication) pair.
4. A subscription message forms a new branch in the delivery tree.
Routing
Branching node
Branching node
Branching node
Data source
Subscriber
Cache bypath
Price update and accounting
Forwarding
5. Publication content is delivered to the subscriber using FId accumulated in the subscription message.
Forwarding node
Forwarding node
Forwarding node
Domain

Figure 2: PSIRP network consists of rendezvous, topology, routing, and forwarding functions.

in Figure 2. Basically, TNs are responsible for managing intra-domain topology, load balance between BNs, and they exchange inter-domain path-vectors similar to BGP for policy-compliant inter-domain routing. TNs relay this information to the BNs of the domain.

BNs are responsible for routing subscription message flowing from subscribers towards data sources and caching popular content. If there are multiple simultaneous subscribers for the same publication, branching nodes will also become branching points in the delivery tree duplicating the data to all receivers and function as an intermediate congestion control point to support multi-rate multicast congestion control with caching. The intra-domain routing guarantees that the subscription for a certain data source, Rid combination will always go through the same branching node in the domain. Same can also be achieved by domains advertising a set of Fids to reach every branching node and their identifier ranges to the neighboring domains. This is a simpler solution but reveals more information about the domain structure outside and may open new vulnerabilities. Because subscription messages flow different intra-domain path as the the payload path in reverse direction, the architecture naturally supports asymmetric links like satellite connections.

FNs implement a very simple, fast, and cheap forwarding algorithm with practically no routing state using a Bloom filter based forwarding explained in [9]. FNs also periodically send their neighbourhood information and link loads to BNs and TNs. Because BNs are aware of the domain topology, they can append a path segment through the domain to the Fid stored in the subscription message. This way, as the subscription messages are routed through multiple domains, they gather a valid Fid that can be used to send payload data through the same domains in reverse order by the entity that eventually receives the subscription message. Both subscription messages and payload data are routed using best effort semantics and all subscription state stored in the network is soft state. This adheres to the fate-sharing principle and the subscribers are in the end responsible for taking care that they receive all the information they want.

A detailed description of the operation of each type of nodes is outside the scope of this paper but a better description can be found in [8].

3.2 Rendezvous

Rendezvous function is used to locate publications and scopes in the network. The rendezvous system is composed of independent rendezvous networks (RN) that provide their service to scopes that want to store their publication advertisements in the rendezvous network. In most cases the scope owner needs to buy the rendezvous service from some RN which requires that the scope owner trusts the RN. We have not specified the publisher side interface to the RNs as each RN is assumed to implement their own set of features and can be based internally on different technologies. A typical RN that spans multiple domains could run a DONA system, for example.

The rendezvous networks are globally connected using a hierarchical DHT [10] based rendezvous interconnect (RI). Individual RNs advertise their scopes in the RI so that they are globally reachable. The aggregation makes the system more scalable as only scopes are stored in the DHT but has the drawback that scopes are used for optimization in addition to their conceptual role in the system. It is also possible to advertise individual publications and other information in the RI but it is expected to be more expensive than just using RNs. The security of the RI is based on central authorities that authorize all nodes before they can join the RI routing structure. We will describe the security solution in more detail in a paper that we have submitted. A detailed description of the rendezvous architecture can be found in [11].

3.3 Phases of publish/subscribe communication

In Figure 2 the typical phases of the publish/subscribe process are shown. At first, a data source authorized by a scope advertises the set of potential publications it is willing to serve for the given scope in the *home rendezvous network* (HRN) of the scope. In the second step, the HRN advertises the scope in rendezvous interconnect (RI). Next, a subscriber sends a rendezvous request for the publication identified by a <Sid, Rid> pair to the local rendezvous network as shown in step 3. If the (cached) results cannot be found from the local RN of the subscriber, the rendezvous message is passed to the RI that routes it to the rendezvous point of the scope and further to the HRN of the scope. As a result, the subscriber receives a set of data sources and their current network locations that can be used to route the subscription messages toward the data sources on the routing layer. In step 4 the subscription message sent towards the branching point forms a new segment in the delivery tree. If the publication is found in the intermediate cache, it is directly returned to the subscriber. Finally, in step 5 the publication is delivered to the subscriber using the created forwarding path. A reliable communication can be achieved by resubscribing to the missing pieces of the publication.

3.4 Topology and Forwarding

Topology management function is responsible for selecting the intra-domain routes, which are used for delivering publications. Each domain has an own topology management function, and domains exchange information about inter-domain connectivity with each other similarly as BGP protocol.

PSIRP utilizes Bloom filters [12] as forwarding identifiers called zFilters. A Bloom filter is a probabilistic data structure, which allows a simple *AND* operation to be used to test whether the filter element is present in a set. Basically, each network link has an own identifier, and the Bloom filter is constructed by performing *OR* operation over all link identifiers that are located on a wanted path [9]. Bloom filters allow very simple and efficient routers, since the forwarding decision can be made based on a simple *AND* operation without using a large routing table. An interesting property of the zFilter based forwarding is that only the network links have identifiers. Network nodes do not have a network layer identifiers, therefore there is no equivalent to the IP addresses.

3.5 Security

In order to avoid inherent security problems that are present in the current Internet, security is an integral part of PSIRP's design in all parts of the system. The basic goal is to provide availability and protect the network from denial-of-service attacks. This applies to the rendezvous system, routing of subscriptions, and the actual payload traffic. Naturally, the security aspect should not compromise the overall system's scalability.

The basic security goal is to prevent unwanted traffic on both rendezvous and forwarding layers, and enforce the principle of pub/sub networking that no data is delivered to subscribers without a valid subscription.

PSIRP utilizes elliptic curve cryptography (ECC) [13], which allows good security with short key and signature sizes. Since a 160-bit ECC key is as strong as a 1024-bit RSA key, the whole public key can be included into Rids and Sids in every packet.

Packet Level Authentication (PLA) [14], which uses per packet cryptographic signatures, provides the integrity, authenticity and accountability on the network layer. PLA is used to secure the rendezvous and subscription traffic, and can optionally used to offer extra security on the forwarding layer. While public key cryptographic operations are expensive, they are by default only used for signalling traffic. With a dedicated hardware acceleration, ECC signature verifications can be performed on wire speed [15].

For the forwarding security zFilters contain some inherent security properties, since link identifiers are not globally known, and without a valid zFilter the attacker can not target the specific victim. To further improve the security a zFormation [16] mechanism can be used. With zFormation, zFilters are valid for a limited amount of time, and are tied to the specific Rid. While zFormation mechanism increases the complexity and requires some state in routers, it increases the security since even if the attacker learns a valid zFilter, it can not launch a denial-of-service attack against the victim.

Finally, the publication content is tied to the publisher's public key through public keys inside rendezvous identifiers and cryptographic signatures, therefore subscribers can verify the integrity and authenticity of the received publication.

4. PSIRP Prototype Implementation and Dissemination of Results

The PSIRP prototype has been implemented for FreeBSD operating system. The prototype consists of parts outlined below.

The software prototype contains basic the publish/subscribe functionality and zFilter based forwarding. The same publish/subscribe paradigm is used for both local and remote publications, therefore applications subscribing to publications do not need to be aware whether the publication resides in a local or remote node. The PSIRP implementation retrieves the requested publication data and passes it to the application. The rendezvous node functionality has been implemented in the prototype as a separate module.

ZFilter forwarding has also been implemented for the NetFPGA development board [17], which is basically a network interface card with four Ethernet ports and a programmable FPGA processor. Hence, simple PSIRP forwarding decisions can be performed on the networking hardware, without intervention from the operating system.

To facilitate testing and application development, a PSIRP plug-in for Firefox web browser has been developed. This plug-in allows the browser to use the PSIRP publish/ subscribe stack, instead of the IP stack, for retrieving the information. The relevant rendezvous and scope identifiers are given through the URL with a *psirp://* prefix.

The code of the PSIRP software prototype, and the NetFPGA forwarding implementation, has been released as open source under GPL and BSD licenses, and is available for download [18]. Also virtual machine images are provided for those who want to effortlessly experiment with the prototype.

To disseminate results, a course has been organized in the Helsinki University of Technology [19] about the publish/subscribe networking and the PSIRP project. During the course students can experiment with the software prototype and develop applications for it.

5. Conclusions and Future Work

This paper described the architecture and implementation of PSIRP, a novel approach to publish/subscribe networking. We believe that publish/subscribe paradigm is a promising one for the future networking and that it offers good flexibility, security and scalability. The clean-slate design of PSIRP allows a more efficient pub/sub solution than current IP networks, which are based on a different paradigm. While such approach is very radical, the preliminary results are encouraging. Our architecture is scalable, efficient and secure, and there exist a working software prototype.

Many tasks remain to be done. The PSIRP prototype needs to be further developed, tested and packaged in a user-friendly manner. We also plan to further evaluate our rendezvous solution and investigate efficient caching schemes for pub/sub networks. Furthermore, this work concentrated mostly on the network layer architecture. Additionally, there is a need for higher layer solutions based on the PSIRP architecture. These include a considerations for constructing application identifiers and publish/subscribe optimized applications.

Since the amount of mobile devices will continue to increase, a good mobility support is essential for a future network architecture. The network should support seamless handovers and multihoming, allowing users to use multiple network interfaces in parallel. The lack of IP protocol and data-oriented nature of PSIRP makes implementing mobility much easier since communication is not tied to the user's location within the network.

Migrating to the new network technologies is a very long and slow process, which occurs gradually. We must study deployment issues further and create a migration plan for PSIRP. The IP compatible overlay version of PSIRP should also be developed to allow a gradual deployment.

References

[1] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, Internet Indirection Infrastructure, *Proceedings of SIGCCOM 2002,* August 2002.

[2] T. Koponen *et al*., A Data-Oriented (and Beyond) Network Architecture, *Proceedings of SIGCOMM 2007*, Kyoto, Japan, August 2007.

[3] M. Caesar *et al.*, ROFL: Routing on Flat Label, *Proceedings of SIGCOMM 2006,* Pisa, Italy, September 2006.

[4] V. Jacobson *et al.,* Networking Named Content, *Proceedings of ACM CoNEXT 2009,* Rome, Italy, December 2009.

[5] S. Tarkoma, M. Ain, and K. Visala, The Publish/Subscribe Internet Routing Paradigm (PSIRP): Designing the Future Internet Arhictecture, *Towards the Future*, pp. 102–111, 2008

[6] J. Saltzer, D. Reed, and D. Clark, End-to-end arguments in system design, ACM Transactions on Computer Systems, volume 2, issue 4, pp. 277-288, November 1984.

[7] D. Clark *et al.* Tussle in Cyberspace: Defining Tomorrow's Internet, IEEE/ACM Transactions on Networking, volume 13, issue 3, pp. 462-475, 2005.

[8] K. Visala, D. Lagutin, and S. Tarkoma, LANES: An Inter-Domain Data-Oriented Routing Architecture, *Proceedings of ReArch'09*, December 2009.

[9] P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, and P. Nikander, LIPSIN: Line Speed Publish/Subscribe Inter-Networking, *Proceedings of SIGCOMM 2009*, Barcelona, Spain, August 2009.

[10] P. Ganesan, K. Gummadi, H. Garcia-Molina, Canon in G Major: Designing DHTs with Hierarchical Structure. in ICDCS'04, 2004, pp. 263-272, 2004.

[11] Rajahalme, J. *et al.,* PSIRP technical report TR09-003: Inter-Domain Rendezvous Service Architecture, December 2009 [online]. Available at: TODO

[12] B. H. Bloom, Space/time Trade-offs in Hash Coding with Allowable Errors, *ACM Communications, v*olume 13, issue 7, pp. 422-426, 1970.

[13] Miller, V. S., Use of elliptic curves in cryptography, *Proceedings of CRYPTO '85: The Advances in Cryptology*, August 1985.

[14] D. Lagutin. Redesigning Internet - The packet level authentication architecture. Licentiate's thesis, Helsinki University of Technology, Finland, June 2008.

[15] J. Forsten, K. Järvinen, and J. Skyttä. Packet Level Authentication: Hardware Subtask Final Report. Technical report [online]. Available from: http://www.tcs.hut.fi/Software/PLA/new/doc/PLA_HW_final_report.pdf .

[16] C. Esteve *et al*., Self-routing Denial-of-Service Resistant Capabilities using In-packet Bloom Filters, *Proceedings of European Conference on Computer Network Defence (EC2ND)*, Milan, Italy, November 2009.

[17] NetFPGA [online]. Available at: http://www.netfpga.org/ .

[18] PSIRP, Source code of the prototype [online]. Available at: http://www.psirp.org/downloads .

[19] T-110.6120 Special Course in Data Communication Software: Publish/Subscribe Internetworking. Helsinki University of Technology [online]. Available at: https://noppa.tkk.fi/noppa/kurssi/t-110.6120/ .