# Experimentally-driven research in Publish/Subscribe Information-centric Inter-Networking

András Zahemszky[1], Borislava Gajic[2], Christian Esteve Rothenberg[3], Christopher Reason[4], Dirk Trossen[4], Dmitrij Lagutin[5], Janne Tuononen[6] and Konstantinos Katsaros[7]

[1] Ericsson Research, Finland, andras.zahemszky@ericsson.com
[2] RWTH Aachen, Germany, gbo@mobnets.rwth-aachen.de
[3] University of Campinas, Brazil, chesteve@dca.fee.unicamp.br
[4] BT Research, UK, {dirk.trossen,christopher.reason}@bt.com
[5] Helsinki University of Technology, Finland, dmitrij.Lagutin@hiit.fi
[6] Nokia Siemens Networks, Finland, janne.tuononen@nsn.com
[7] Athens University of Economics and Business, Greece, ntinos@aueb.gr

**Abstract.** Testing and evaluating new architectural propositions is a challenge. Given the usual variety of technologies and scales involved in the necessary evaluation, a one-size-fits-all approach does hardly suffice. Instead, a collection of evaluation and experimentation methods must be chosen for a comprehensive testing of the proposed solutions. This paper outlines some of the approaches chosen for an architectural proposition that establishes a publish/subscribe-based internetworking layer for the Future Internet. For that, we outline challenges we identified when turning to experimentation as a means of evaluation. We then present the variety of emulation as well as experimental test bed efforts that attempt to address these challenges. While this is not to be seen as a conclusive summary of experimental research in this space, it is an attempt to summarize our efforts as a work-of-progress for others working the architectural field.

**Keywords:** publish-subscribe, experimental research, NetFPGA, testbed

## 1 Introduction

Future Internet research requires at least three key ingredients to have chances of success. First, a clear *vision* that outlines the direction and sets the goals and requirements of the envisioned global communication infrastructure. Second, *experimentally-driven research* to validate the architectural proposals at scale and under realistic scenarios. Third, understanding the *business* incentives for adoption, which requires socio-economic market evaluations and industry engagement early in the feedback and re-design loops.

The PSIRP (Publish-Subscribe Internetworking Routing Paradigm) project [1] is an EU FP7 funded project that started in January 2008 and aims at covering all three research fronts of a clean-slate design approach that departs from the current host-centric IP inter-networking to an information-centric future Internet. The PSIRP

vision is inspired by the observation that information/content – what a user wants – should have a more central role in future network architectures than it does in today's Internet host-to-host conversation model [2,3,4,5,6]. To this end, architectures based on data-oriented primitives like publish/subscribe [7] and the similar (e.g., get/response, find/register) [8] are well-suited for the unwieldy amounts of named linked data retrieved from the Web and exchanged via overlay networks like P2P and content delivery networks.

The project has already outlined the direction to realize this vision by defining design principles [9] and proposing several design choices towards a novel pub/sub-based Internet-scale architecture [10, 11]. Time has come to accomplish the second key component of clean slate future Internet research: experimental validation and evaluation at scale. From the design phase of the project, prototyping work is one major component in the development of the architecture to provide fast feedback from the practical experiences enabling a fruitful top-down and bottom-up dialogue.

Most researchers have at some point faced questions such as “what is the performance of my new protocol”, “how does my new technology perform in a highly distributed environment” or “how does my pre-commercial code perform under more realistic networking conditions”. When developing clean slate technologies this is no different but arguably more challenging. In the most part, these questions are solved either experimentally or using models, and it is the former of the two which this paper focuses on; “How do I experimentally test and evaluate in a realistic setting?” It is important to note the inclusion of the “in a realistic setting” clause, as testing any multi-domain protocol designed to be run over a potential future Internet architecture over the current Internet will rarely provide irrefutable evidence for its performance.

In order to enable large-scale experimental research with the required levels of flexibility of future Internet architectural proposals, big efforts are undergoing on both sides of the Atlantic in projects such as GENI, FIRE, FEDERICA and OneLab2. Their common denominator is their goal of providing a playground for researchers to validate their visions under "realistic" scenarios. Typical experimental evaluation methods such as emulation, simulation and (experimental and usually local) testbeds, have particular strengths and weaknesses, so an evaluation architecture which combines all three should provide the greatest flexibility while retaining the best features of each. Such a rich evaluation playground is the ultimate goal of our validation efforts.

In this paper, we describe the experimental approach taken by the PSIRP project and the components of the underlying research infrastructure. We account for the experiences gained when working with the selected evaluation tools and implemented prototypes. First, we introduce the background fundamentals of the conceptual architecture and the evaluation challenges (Section 2). Then, we dissect the experimentally-driven visionary research divided by implementation work (Section 3) and the experimental verification efforts (Section 4). Finally, we describe the lessons learned and the ongoing work towards a unified evaluation approach (Section 5).

## 2 Background

The current Internet architecture focuses on communicating entities, largely leaving aside the information to be exchanged among them. However, trends in communication scenarios show that *what* is being exchanged is becoming more important than the *who* is exchanging information. Van Jacobson describes this as moving from *interconnecting machines* to *interconnecting information* [6]. The ambition of the PSIRP project is to investigate major changes to the current IP layer, up to the point of replacing this layer with a new form of inter-networking. To this end, PSIRP undergoes all phases of a clean-slate design project, from state-of-the-art survey over outlining basic design principles and understanding design choices through the definition of conceptual and actual architectures and their implementation. Architectural and technological choices are evaluated from the angles of security, socio-economic and quantitative design constraints. In the following, we briefly outline the underlying conceptual architecture of PSIRP, appreciating that we cannot present the full breadth of the architectural concepts in the given space. Hence, the interested reader is referred to [1] for more information.

### 2.1 Conceptual Architecture

Based on the design principles outlined in [9], the following information-based architecture relies on basic labeling (cf. *Everything is information*) and grouping of information (cf. *Information is scoped*), while providing a publish-subscribe service model (cf. *Equal control)*. The main objective of the architecture is to provide the required mapping of these concepts onto concrete forwarding relations between endpoints, producing and consuming information. This keeps the network architecture simple, while enabling more complex application-level naming structures, as suggested in [6] and similar work [2,3,4,8].

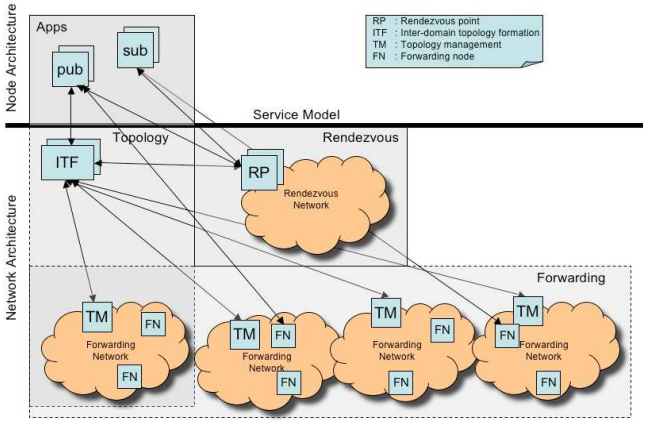


**Fig. 1.** Conceptual Architecture

Figure 1 presents the main architectural components implementing these design concepts. The *pub* and *sub* components at the application level implement applications based on basic publish/subscribe network services, enabling *publications* and *subscriptions* towards information items labeled by RId (Rendezvous ID) within particular scopes, identified by SId (Scope ID) – see more below.

Transactional services, operating in request-reply mode, can easily be supported through a publish/subscribe model [7], with the server subscribing to receive requests. From this basic mode of communication, we can bootstrap internal network operations as well as offer a new information-centric service API, similar to [8]. Such a new communication API replaces in many ways the role of traditional middleware layers since it conflates low-level information discovery as well as location determination of publishers and subscribers into a single network service, therefore largely eliminating the need for such mapping functions to exist on application level. However, there is still a need for mapping application-level information concepts onto the basic concepts provided by our architecture – something being left outside the scope of the network architecture considered here.

The architecture itself consists of three main functions: *rendezvous*, *topology* and *forwarding*. Generally, the *rendezvous* function implements the matching between publishers and subscribers of information items, each identified via a RId. Information items logically reside within at least one scope. Each scope is identified via a SId, which is in turn provided by dedicated rendezvous points (RP). Hence, rendezvous points match the semantic-free information items within the scope they are serving. There is at least one rendezvous point per scope, each of which subscribes to the SId through a global rendezvous system. Upon subscription to an information item in the scope, the request can be routed either to all or to the 'best' rendezvous point, using anycast-like functionality. Furthermore, rendezvous points implement policies associated with the matching, such as access control.

Once the rendezvous point has matched a publication and one or more subscriptions, the forwarding topology is created in negotiation with the inter-domain topology formation (ITF) function. This is based on the publisher and subscriber “locations” on the level of autonomous systems (ASes), the applicable policies and the ITF information that includes peering and transit relationships among ASes. This is similar to BGP or (G)MPLS PCE, but the underlying networks forward information, not (opaque data) packets, i.e., there exists a rich set of policies attached to potentially every information item.

In addition to building inter-domain paths between the forwarding networks to which the publisher and subscribers are attached to, appropriate intra-domain paths need to be constructed. This is done in collaboration with the *topology management* function that resides in every AS. This function is responsible for instructing its local *forwarding nodes* (FNs) to establish paths to local publishers and/or subscribers or to serve as transfer links between ASes. As in the current Internet architecture, this approach does not prescribe any particular intra-domain forwarding mechanism, with the one constraint that the local mechanisms should support ITF compliant policies.

### 2.2. Challenges

The architecture presented in Figure 1 aims at providing an internetworking architecture made for the foreseen scale of a future (information-centric) Internet. Evaluating the concepts for such architecture therefore face particular challenges that comes with that ambition:

- *Scale*: inter-domain functions, such as for rendezvous and forwarding, are built for large scale. This requires experimental methods that can scale to appropriate sizes. Experimentation alone is unlikely to suffice for scaling experiments.
- *Technology Variety*: inter-domain functions such as forwarding are designed to work over a variety of technologies, similar to today's Internet. This, however, requires the availability of such wide variety of technologies when evaluating crucial parameters, such as delay or efficiency.
- *Usage Variety*: it is hard to predict potential usages for any network architecture – the current Internet is the best example for this. Hence, potential user involvement is crucial but also a variety of different usage models for isolated experiments.
- *Economic Variety*: inter-domain functions, such as rendezvous and forwarding, heavily depend on the underlying business relations of ASes in their overall performance. Hence, a proper understanding of various business relations, possibly vastly different from today's peering relations in the Internet, is required to provide insight in the effectiveness of novel inter-domain functions.
- *Platform Variety*: it is obvious that a single platform for testing is hardly achievable given different operation systems, virtualization approaches and simulation/emulation platforms available. The experimental approach must cater to this variety.

In the following, we outline the project's approach to cope with these challenges. It is the ambition of this paper to outline a coherent testing and experimentation approach although its creation is driven by bottom-up testing and evaluation activities and a post-rationalization of these activities in a coherent framework that might aid similar activities in the future that need to address the challenges outlined above. Before doing so, however, we present a brief summary of the implementation work done in order to better understand the chosen evaluation methods.

## 3 Implementation work

The PSIRP project works towards a publish/subscribe solution, where even IP forwarding is re-considered. This creates a need for a prototype with a different structure than existing systems. The prototype development [12] is divided into two separate areas, namely the Lower Layer implementation (LoLI) and the Upper Layer implementation (UpLI). The LoLI is motivated by the need for a new kind of data handling at the end-host (i.e., internal publication management) as well as for forwarding data packets due to the pub/sub architectural approach. The requirements on locating publications and managing the network topology are considered to be "upper layer" tasks. This section goes first through the LoLI, including the FreeBSD-

based blackboard implementation (Blackhawk), the security and PLA-related implementation, and the NetFPGA implementation of zFilter forwarding. Finally, we describe the Rendezvous and Topology functions in the UpLI developments.

### 3.1. Blackhawk: FreeBSD node implementation

The current node architecture [12] implements parts of the PSIRP service model and API [9]. It consists of the following pieces:

- A blackboard that implements a simple memory object model inside a node. Conceptually, the blackboard is the place where data items are stored as publications that can be subscribed to.
- An API for publishing data items to the blackboard, subscribing to them, and getting notifications when new versions of them are published.
- Applications that can communicate with each other via the blackboard. This includes helper applications that implement different network-level functions.

In the *Blackhawk* prototype for FreeBSD, the blackboard is implemented as a kernel module, as shown in Figure 2. It is integrated with the operating system's virtual memory system, i.e., publications in the blackboard correspond to virtual memory objects and pages. The API is implemented as a library that communicates with the kernel module using system calls. It provides functions for creating, publishing, and subscribing to publications. Notifications about publish operations on the blackboard can be acquired via the *kevent* system of FreeBSD. In addition, a file system view to the blackboard is provided as a hierarchy of scopes, publications, publication versions, and memory pages (which all have their own RIds).

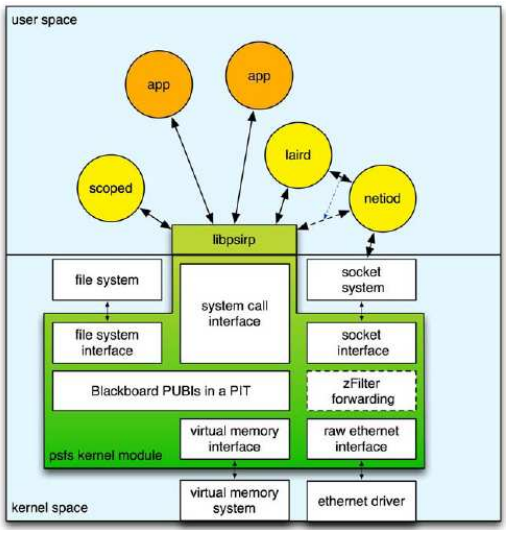


**Fig. 2.** Node Architecture

As mentioned above, user space applications can use pub/sub-based inter-process communication by publishing data to the blackboard, from where other processes can

retrieve the publications by subscribing to them. The component wheel functions (e.g., rendezvous and topology management) are also implemented as applications. However, some parts of the component wheel (e.g., forwarding) may also be implemented in the kernel space to achieve better performance. As an example of helpers, the current version of the Blackhawk prototype features three helpers:

- A scope helper (*scoped*) takes care of instantiating, updating, and re-publishing scope publications, i.e. data items that contain collections of RIds.
- A local-area rendezvous helper (*laird*) extends the blackboard model into the network. It provides local monitoring for publish/subscribe operations, and it advertises the locally published publications to the local-area rendezvous node.
- A network I/O helper (*netiod*) implements packet fragmentation/assembly in addition to forwarding, using sockets for sending and receiving packets over links in the network.

The rendezvous helper communicates with the network I/O module when publication metadata or data needs to be sent into the network. In the reverse direction, the network I/O helper dispatches received metadata to the rendezvous helper.

### 3.2 Security and PLA implementation

Packet Level Authentication (PLA) [13] is a novel method for providing availability at the network layer by using per packet cryptographic signatures. PLA was originally implemented for IP networks; however it does not depend on IP and therefore can also be used with other network layer solutions such as PSIRP. PLA's main aim is to allow nodes on the path to independently verify packets without having separate security associations with the sender, or previous nodes that have handled packets. Any node can verify whether packets has been modified, duplicated or delayed, therefore invalid packets can be dropped immediately, before they reach the destination.

PLA works by adding a security header including the sender’s cryptographic identity, certificate from a trusted third party, timestamp, sequence number and the cryptographic signature. The timestamp and sequence number offer protection against replay attacks, while the signature protects the packet's integrity and offers accountability. PLA uses elliptic curve cryptography (ECC) since it offers a good security with compact key sizes. While public key signatures are computationally intensive, they can scale to high speed networks and low power devices as long as dedicated hardware is used for accelerating signature calculations [14]. Preliminary simulation results have shown that a 90nm dedicated ASIC would be able to perform almost one million signature verifications per second, such performance would be enough verify 5Gbps of average traffic.

In PSIRP, PLA is used mostly used to secure control messages (publish/subscribe), and can be optionally be used for securing all traffic. In our system, the most important security properties of control messages are integrity protection and authentication. For example, has the packet been modified? Does the publisher have a permission from the scope to publish in certain SId:RId? PLA functionality has been implemented as a separate library [15], which is used by the PSIRP networking daemon to add and verify PLA headers.

### 3.3 NetFPGA forwarding implementation

Forwarding nodes implement the multicast source-routing mechanism described in [15] based on an in-packet Bloom filter referred to as *zFilter*. The mechanism allows for compactly representing a delivery tree within the limited header space of a packet. Basically, a routable Bloom filter is formed by ORing the Bloom masks of the network links of a delivery tree. Forwarding decisions are based on simple logical AND operations between the zFilter and the forwarding node Link ID table. As with all Bloom filter based approaches, false positives of such AND operations can occur, leading to false deliveries along the AS links. Hence, determining a rate for such false positives is a typical performance evaluation objective. Mechanisms to minimize such false positives have been proposed in [16], such as the introduction of virtual link identifiers, which combine certain paths/trees into a virtual (single) link, 'thinning' out the Bloom filter space and therefore reducing the potential for false positives.

The zFilter forwarding algorithm has been implemented on NetFPGA [17], a flexible and open hardware platform for research and classroom experimentation in terms of networking and traffic processing. The implementation [18] was based on the Stanford reference switch implementation, which was modified to create a simple zFilter switch. According to early measurements, the efficiency of the NetFGPA-based zFilter forwarding is very good and requires about 3-5μs per hop, which represents a lower latency than the reference IP router implementation.

Our experiences with NetFPGA as the prototyping platform are overall positive. It is a solid development platform that is available in a complete package and it is suitable for clean-slate developments requiring line-speed operations.

### 3.4 Rendezvous

The PSIRP rendezvous architecture, defined in [9, 27], is a composition of modular rendezvous networks that are interconnected to form a globally reachable inter-domain rendezvous system. The rendezvous networks are formed by rendezvous nodes (RNs) that are organized as a policy controlled inter-domain hierarchy. Each RN may host multiple rendezvous points (RPs) that are logical meeting places in the pub/sub system for a certain <SId, RId> pair i.e. for each pair there exists at least one RP in the rendezvous system. In many cases the same RP is shared by the publications in the same scope. In the event of rendezvous, RP initiates creation of the forwarding path creation in the topology function that, when finished, enables transmission of data between the publisher and the subscriber.

The rendezvous function is implemented in two separate instances; the local rendezvous helper, described in the Blackhawk prototype, that handles rendezvous in local node and in small-scale local area networks, and the rendezvous node, which implements the rendezvous network. Large and geographically dispersed Autonomous systems (ASes) may be covered by multiple rendezvous networks, but in the typical case a rendezvous network would be a collection of rendezvous nodes from cooperating ASes. Therefore, in the same way as peering between two rendezvous nodes is supported, the current rendezvous network implementation can be seen also as a simple version inter-domain rendezvous system.

Figure 3 illustrates the architecture for the existing two versions of the rendezvous node implementation: the UDP forwarding based standalone version and the new Blackhawk integrated version. The architecture includes a stub version of the topology function that cooperates with the rendezvous function in the pub/sub system. Both versions implement the same functions to establish and operate pub/sub rendezvous networks.

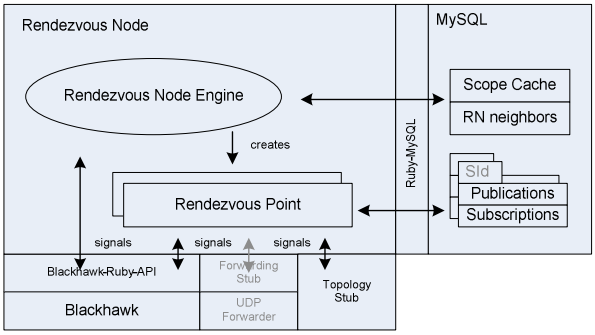


**Fig. 3.** Rendezvous node implementation architecture

### 3.5 Topology

Following the administrative division of the current Internet, we distinguish between intra-domain and inter-domain topology management mechanisms as two functionally separated units retaining a strong interconnection between their structural pieces. The main role of intra-domain topology management is the discovery of topology information, using it as an input for computing necessary network states, and sending updated forwarding information to the relevant nodes. The inter-domain topology formation functions on the domain level, in addition to discovery, is responsible for configuring and maintaining inter-domain topology states for creating forwarding paths based on various policy compliance requirements.

Our current Python based implementation of topology management is divided into two modules: client and server, which can simultaneously coexist on each node. The client module runs on each forwarding node and is mainly responsible for discovering local connectivity information, whereas the server module collects these local information pieces and structures them together to form a picture of the overall network topology within the domain of operation. The server module is also responsible for computing the optimal forwarding paths and publishing that information towards forwarding nodes. Additionally, each forwarding node runs a link state helper module which maintains the table of “known” links along with link related available information, e.g., throughput, and delay. Information about relevant link properties can be provided by low level helper functions, which collect physical data about the link.

In order to facilitate the exchange of required data different scopes are defined for exchanging particular information, e.g., a scope for exchanging the existence of information coming from each particular node, a scope for distributing and collecting

information representing the set of neighboring nodes, a scope for dissemination and collection of link data messages. Using the publish/subscribe paradigm, forwarding nodes and topology servers receive and update the required information ("Hello", "LSA" messages). The topology manager implementation also performs simple forwarding tree and zFilter creation using shortest path tree calculation .

## 4 Addressing the Evaluation Challenges

With the understanding of how we implemented the architectural foundations that we outlined in Section 2 in a component architecture that we presented in Section 3, we can now move on with addressing the evaluation challenges presented in Section 2.2. For this, we give specific examples from evaluation tasks within PSIRP targeting the identified challenges.

### 4.1 Simulation and Emulation: Addressing Scale and Variety of Technologies

For rapid evaluation of different networking solutions in terms of packet-level performance, simulations are the standard approach. In particular, inter-domain solutions, such as the outlined rendezvous and inter-domain topology formation (see Section 2.1), are targeted in simulative evaluations. But also our developed forwarding solutions are candidates for simulations, in particular when coupled with emulation methods. In the following, we outline the used technologies for these tasks.

#### 4.1.1 NS-3

Simulation is a common means to achieve scale of evaluation, not requiring direct equipment to be handy for evaluation. Given our scale challenge, it is natural to resort to simulations as a central method to address this challenge. Examples of architectural solutions that are simulatively evaluated are the rendezvous and forwarding solutions. Our natural choice for performance analysis of parts of the PSIRP architecture is ns-3 [18], as it offers a clean simulator architecture, easy extendibility, and features for network emulation. Evaluation work with ns-3 included network coding solutions, as well as the performance of the multicast forwarding mechanisms based on zFilters. On-going work includes exploitation of network emulation functions and integration with the prototype implementation (Blackhawk).

Ns-3's easy extendibility is very attractive for projects designing clean-slate architectures, as new protocols can be installed into any desired level of the networking stack. Following this design philosophy, we extended the simulator to support the zFilter-based forwarding. Primarily written in C++, in ns-3 new mechanisms can be added intuitively via the techniques of class inheritance and new class creation. The implemented forwarding layer supports the major design elements presented in [15], including the optimization of using multiple parallel forwarding tables, various loop prevention techniques, fast reroute mechanisms and virtual links. With the usage of the simulator, we showed that zFilter forwarding is feasible, and

supports unicast comfortably, as well as sparse mode multicast communication up to topologies of the size of metropolitan area networks.

The integration of ns-3 simulations with real world traffic is possible via two modes. *Virtualization* allows real hosts to communicate via simulated networks, while *network emulation* allows simulated nodes to exchange information through real links, i.e. a networking testbed. So far, an early inter-operation test has been carried out between a simplified forwarding simulator and the first iteration of the BSD prototype, while future work includes further integration into a PSIRP testbed (cf. Section 5).

#### 4.1.2 Network Emulation

Emulation allows for inserting a "sense of reality" into a simulative framework by emulating part of the real-world in combination with developed solutions, e.g., running our real-world forwarding implementation (see Section 3.3) in an emulated Ethernet network of a larger size than we achieve with a testbed setup.

For tests and evaluation activities involving a larger number of nodes in a controlled environment we use the network emulation testbed at RWTH. The testbed consists of powerful servers equipped with multi-core CPUs and four Gigabit Ethernet network interfaces each. All the servers are connected to a high-performance switch allowing for different network topologies to be set up for the experiments. Control traffic is sent over a dedicated network interface, over a different switch, so as not to interfere with measurements. Each of the servers is capable of hosting a large number of virtual machines functioning either as communication endpoints, or nodes normally associated with the network infrastructure such as forwarding nodes, rendezvous servers or topology management nodes.

Experiments with even larger number of nodes can be realized by combining the use of VMs with network emulation and tap techniques provided by ns-3 network simulator. In such a setup individual VM instances can emulate complete forwarding infrastructures within individual domains, while other VMs connected to those emulated forwarding domains can act as traffic sources or as nodes offering rendezvous or topology management services. On the other hand, each VM can be connected with simulated networks via ns-3 tap device, acting as a regular PSIRP node. Therefore we can implement the case of PSIRP traffic originating on VM instances running the developed prototype implementations and traversing over large scale simulated networks.

#### 4.1.3 OMNet++ and OverSim

A medium term alternative to a native implementation of PSIRP, operating directly on top of the network hardware, is an overlay implementation on top of IP. The overlay work has so far focused on network support for scalable multicast, the main enabler for providing the entire PSIRP functionality as an overlay solution. In particular, a solution has been designed to operate on top of the Pastry DHT based content routing scheme [19] and the Scribe overlay multicast scheme [20]. Special attention is paid to the incremental deployment process of the overlay architecture as

well as to the potential benefits of in-network caching. In addition, the concept of hierarchical DHTs has been explored with the purpose of making routing conform to the policy-compliant interconnection of networks on the Internet.

For the work on DHT based overlays, the OverSim [21] platform has been used, which is an overlay network simulation framework for the OMNeT++ simulation environment. The OverSim framework provides implementations of several overlay schemes and applications, including Chord and Pastry, as well as overlay multicast schemes, such as Scribe.

Due to the popularity of P2P content distribution applications, BitTorrent was chosen as the main application model for benchmarking. In order to be able to perform a comparison study between our overlay multicast based BitTorrent alternative [22] and the regular BitTorrent of today's Internet, the BitTorrent suite of protocols for OMNeT++ and a churn generator module for OverSim based on an analysis of real BitTorrent traces was created [23].

### 4.2 Application Innovation Process: Addressing Usage Variety

Running component and architectural evaluation according to identified performance parameters is crucial. But the real test for any solution is that of being applicable to a certain (often large) set of real-life applications. It was recognized early in the planning phase that our efforts would not be able to address the potential usage variety for a Future Internet. Hence, an application innovation process was established that would attract developers to the new platform for trying out novel usages of the platform. This process is facilitated by the open source release of major node and network components, allowing for developing applications on an open platform with existing network technology like Ethernet.

One straightforward example of such usage is the development of a plug-in for Firefox, which provides mechanisms for users to subscribe to publications using the PSIRP protocol through their web-browser. The plug-in intercepts all PSIRP protocol calls in the address bar or in a link embedded in a webpage (e.g. psirp://), passing the PSIRP parameters (SId:RId) to the XPCOM component. This component interacts with the PSIRP library by subscribing to the publication identified by the SId and RId pair and, after retrieving it, the component saves it as a local file. Finally, the plug-in opens the fetched publication and displays it in the web-browser. Currently, the retrieved publication is saved as a local file, which is later opened by the web browser. As a future improvement, publications will be displayed directly in the web browser without requiring copies to be saved locally.

Other applications are currently explored in collaboration with external partners. But it is obvious that this challenge is a difficult and time-consuming one to be addressed.

### 4.3 Testbed Infrastructures: Addressing Scale and Variety of Technologies

Experimental testing of the technology solutions developed in PSIRP is crucial for evaluating the viability of the overall proposition of the project, namely to develop a

viable alternative to the current IP paradigm. For this to happen, the implementation work is integrated into a single coherent prototype (see Section 3) [24]. As a result, not only is the architecture work converging, but also the various implementation efforts on these architectural components are starting to converge into a coherent and running system. In particular, core components like the node architecture, forwarding and rendezvous node (see Section 2.1 and 3.2) are coming together, enabling the progression towards a first networked setup of a PSIRP network. Although crucial components, such as the ITF function (see Section 2.1) are still missing from this coherent prototype, the foundation has been set to perform experimental testing in testbed infrastructures. A crucial step in this testing is the extension of a limited laboratory prototype towards a fully networked test network that operates based on the central components of the architecture.

A first step in this direction is the establishment of localized test network at the BT and University of Essex facilities in the UK. These facilities are based on a heterogeneous network infrastructure that was built under the recently finished UK TSB (Technology Strategy Board) funded project HIPnet. It provides a variety of access technologies in the wireless and wireline domain, e.g., WiMax, WiFi, and all-optical fixed infrastructure. The infrastructure spans the local campus at Essex University, located at the edge of Colchester (UK). The university's facilities hold about 2500 students in their dorms, with access to their infrastructure. The wireless coverage has recently, in June 2009, been extended to full campus coverage with a single SSID. The WiMax coverage spans most of the campus, using a rotating antenna. The physical connectivity on the optical level extends to Cambridge University as well as to the BT facilities at Adastral Park (UK). This variety of access technologies is currently utilized for a fully networked PSIRP testbed. Given the largely Ethernet basis of the infrastructure, this is relatively easy for the wired part. Specifically, there are currently three types of machines being installed for a simple PSIRP network setup. The two end nodes are publisher and subscriber, respectively, currently running the latest release of the PSIRP node architecture (Blackhawk). These will be available at BT premises, at Essex as well as Cambridge University, enabling a variety of test cases through dedicated test applications running on these nodes. The third type is a forwarding node, utilizing the current NetFPGA implementation [16] as well as the FreeBSD-based forwarding engine.

Such setup will not only enable demonstrations but also provide a testing ground for the implementation itself. For instance, real network load performance experiments can be conducted for evaluating (a) end node architecture performance and (b) forwarding node performance. In addition, the setup will be utilized for extensions at the technological level. One such extension is the development of a topology management module, which will demonstrate the applicability of the PSIRP information concepts for optimizing resource utilization on the optical level. For this, we will utilize the existing optical infrastructure in the testbed in partnership with Essex University. Furthermore, the integrative demonstrator will be used as the basis for a UK-funded project between Essex University and Cambridge University in the area of lifestyle management [25]. This project targets novel services in the user-centric health area through self-monitoring and information processing. This is an area where we expect novel input from an underlying information-centric architecture like PSIRP.

While the localized experimental facilities allow for testing components and parts of the architecture under a variety of access and network technologies as well as in possibly diverse application settings, the issue of required scale still remains for crucial, in particular inter-domain, functions. For this reason, PSIRP is collaborating with the European Onelab2 efforts to establish a direct experimental platform support for architectural propositions as pursued by PSIRP. The immediate result of this collaboration is the connection of the localized testbed to the Planetlab Europe facilities, enabling the ability of experimentally testing technologies over the current Internet. In addition, direct connections to the experimental FEDERICA platform are currently explored to enable large-scale testing of PSIRP technologies that directly operate on Ethernet level, such as the forwarding solution [15].

## 5. Lessons Learned: The Attempt of a Coherent Approach

Section 4 highlighted the approaches being taken [26] for evaluating the variety of technologies and solutions being developed in PSIRP, according to the presented concept architecture. It is not surprising that the mix of simulations, emulations and testbeds of various kinds have been explored, given the variety of challenges to be addressed by the evaluation. But when looking closer at the variety of evaluation techniques, as presented in Section 4, one can recognize certain patterns that help formulating principles for a coherent approach that enables evaluating large-scale architectures like the one envisioned in Section 2.1. These principles for a coherent approach, directly addressing our challenges laid out in Section 2.2, are as follows:

- *Enable scale*: The scale challenge is most prominent in an evaluation of this kind and any approach must achieve this. While this is seemingly obvious, given our challenges, the approach must achieve this scale while preserving aspects like locality and component integration. We implement this principle by combining simulative and emulative elements.
- *Enable component-level testing*: Any solution development goes hand in hand with component development of the envisioned architecture. A comprehensive evaluation approach must enable component-level testing while being integrated into a scalable testing environment. This is achieved in our approach through the usage of emulation methods.
- *Enable locality*: Large-scale architectures do not live off inter-domain components and technologies only. Intra-domain solutions, such as for high-speed forwarding are part of this larger picture and need to be evaluated. Such testing requires often localized availability of components for the ability to quickly reconfigure and manipulate the test environment, while still being integrated into a large-scale framework of evaluation. Furthermore, locality enables a certain set of user experiments that are often not possible in global environments, such as sensing or local content scenarios. Hence, local test networks are crucial in a coherent approach.
- *Enable inter-domain operation*: The workings of protocols and technologies across multiple technological and administrative domains, lead to an often very different set of problems than a mere intra-domain operation. Hence,

enabling such inter-domain operation, directly addressing the variety challenges of Section 2.2, is crucial in a coherent approach.

Taking into account these principles and our lessons learned from the evaluation, as presented in Section 4, we can formulate our attempt for a unified approach in Figure 4. This approach combines all evaluation environments: simulation (ns-3), emulation (in isolated environments) as well as testbeds like global solutions (PlanetLab) and a small general purpose testbed (at Essex University) to provide the best of all worlds while enabling a larger evaluation environment to be built than using just one environment alone. As a consequence, the localized testbed would enable the creation of large, high bandwidth tier-one equivalent ASes, with a direct connection to the raw Ethernet network provided by FEDERICA through the UK NREN.

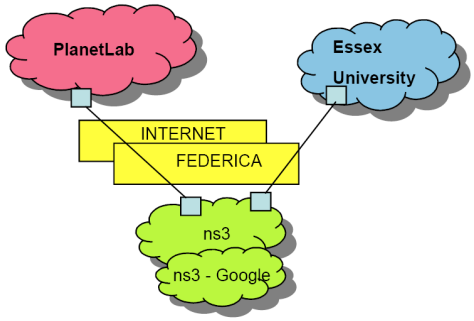


**Fig. 4.** Proposed environment for testing the PSIRP prototype

The ns-3 simulator(s), connected either directly to FEDERICA or to the Internet would allow the creation of a large number of smaller ASes, which could then be used to generate traffic to be injected into the PlanetLab and Essex University testbed. The PlanetLab environment, connected over the Internet, could be used to evaluate the performance of technologies over the current internet and its associated protocols, and would also be a prime candidate for creating a number of smaller ASes (depending on the available node bandwidth) with real world traffic properties (e.g., delays, link failures. etc). Combinations of tests, such as real-world experiments conducted at the local network, e.g., through applications that students would use, and background Internet-type traffic from PlanetLab, can be conducted with this approach.

The integration of an all-Ethernet testbed through FEDERICA and the localized testbed would provide the ability for evaluating native inter-domain technologies which would not run over IP. It also gives the experimenter the flexibility to choose the network environment with the most appropriate underlying physical topology to partially (or fully) match the overlay topology. The integration of the Planetlab environment, however, allows for amending the experiments with overlaid solutions, e.g., global rendezvous alternatives, in which an overlay execution suffices.

The above solution would also provide a large amount of flexibility for creating network topologies, as all three environments provide tools for this purpose. Ns-3 by design allows any network topology. PlanetLab has now been federated with VINI to enable layer 2 overlay creation and work is underway for a pure PlanetLab topology manager [27]. FEDERICA provides researchers the ability to specify a topology map

containing V-nodes, virtual IP routers and virtual links. Early consultations with FEDERICA have determined that, for instance, forwarding techniques of PSIRP could easily be experimented with within FEDERICA without (logical) topology constraint. This eventually targets the last remaining challenge of Section 2.2, namely the economic variety. Testing inter-domain technologies, like global rendezvous or topology formation (see Section 2.1) over a platform like Planetlab forces the current business relations of ASes, represented through their underlying peering relations, on the tests. Approaches like VINI federation or FEDERICA integration would allow for creating inter-domain topologies that are significantly different from today's peering relations. The structure of such peering relations, however, is left for socio-economic considerations rather than experimental verification. But it is the proposed combined approach in Figure 4 that would facilitate the latter.

## 6. Conclusion

Experimental verification of new architectural approaches for the Future Internet is a difficult and challenging undertaking. Scale and variety on levels of technology, usage and economics places a burden on any evaluation task within an architectural effort. In this paper, we outlined the challenges that were faced by an exemplary architecture effort in the area of information-centric networking. While our approaches to implementation and experimentation are based on the particular efforts, many of the lessons learned easily apply to other, similarly ambitious, efforts currently being undertaken. Our combined approach of simulation, emulation and localized as well as global testbeds allows for addressing the various challenges. However, it is well recognized that more work is required for a coherent testing and experimentation approach for Future Internet solutions.

## Acknowledgments

This paper outlines the combined efforts of the PSIRP project in various areas of design, implementation and evaluation. We therefore acknowledge the contributions of a large number of individuals in these areas. This paper would have not been possible without their work.

## References

1. EU FP7 Publish-Subscribe Internetworking Routing Paradigm (PSIRP), http://psirp.org
2. M. Gritter and D. R. Cheriton. "An architecture for content routing support in the internet". In Usenix Symposium on Internet Technologies and Systems (USITS), 2001.
3. H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, I. Stoica, and M. Walfish. "A Layered Naming Architecture for the Internet". In SIGCOMM, 2004.
4. T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," ACM SIGCOMM Computer Communication Review, vol. 37, issue 4, pages 181-192, October 2007.

5. C. Esteve, F. L. Verdi, and M. F. Magalhães. “Towards a new generation of information-oriented internetworking architectures”. In ReArch08 workshop at CoNEXT, 2008.
6. Van Jacobson D. K. Smetters Nick Briggs Michael Plass Paul Stewart, James D. Thornton and Rebecca Braynard, "Networking Named Content", In ACM CoNEXT 2009.
7. P. T. Eugster, P. A. Felber, R. Guerraoui, A.-M. Kermarrec,”The many faces of publish/subscribe”, ACM Comput. Surv., 35(2):114–131, 2003
8. M. Demmer, K. Fall, T. Koponen, S. Shenker., “Towards a modern communications API”, 6th ACM SIGCOMM Workshop on Hot Topics in Networks (HotNetsVI), 2007
9. Ain (ed), “Architecture Definition, Component Descriptions and Requirements”, PSIRP deliverable D2.3, February 2009
10. M. Särelä, T. Rinta-aho, and S. Tarkoma, “RTFM: Publish/Subscribe Internetworking Architecture”. ICT-MobileSummit 2008 Conference Proceedings, 2008
11. S. Tarkoma, D. Trossen, and M. Särelä, “Black Boxes: Making Ends Meet in Data Driven Networking”. MobiArch’08, August 22, 2008, Seattle, Washington, USA. pp 67-72.
12. P. Jokela (ed), "Progress Report and Evaluation of Implemented Upper and Lower Layer Function", PSIRP deliverable D3.3, June 2009
13. D. Lagutin. “Redesigning Internet - The Packet Level Authentication architecture”. Licentiate`s thesis, Helsinki University of Technology, June 2008.
14. J. Forsten, K. Järvinen, and J. Skyttä. “Packet Level Authentication: Hardware subtask final report”. Technical report, Helsinki University of Technology, 2008, [online] available at: http://www.tcs.hut.fi/Software/PLA/new/doc/PLA_HW_final_report.pdf
15. P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, P. Nikander, “LIPSIN: Line speed Publish/Subscribe Inter-Networking”, Proc. of ACM SIGCOMM, 2009
16. J. W. Lockwood and et al.. *NetFPGA – an open platform for gigabit-rate network switching and routing*. In MSE ’07, 2007.
17. J. Keinänen, P. Jokela, K. Slavov, "Implementing zFilter based forwarding node on a NetFPGA", Proc. of NetFPGA Developers Workshop, August 2009
18. Henderson, T.M., Lacage, M., Riley, G.F.: “Network Simulations with the ns-3 Simulator”. In: SIGCOMM demonstration (2008)
19. A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems". IFIP/ACM Middleware, Nov 2001.
20. M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, “SCRIBE: A large-scale and decentralized application-level multicast infrastructure”, IEEE Journal on Selected Areas in Communications, vol. 20, no. 8, pp. 100–110, 2002.
21. I. Baumgart, B. Heep, S. Krause, ¨OverSim: A Flexible Overlay Network Simulation Framework, Proceedings of 10th IEEE Global Internet Symposium, May 2007
22 G. Xylomenos, K. Katsaros, and V. P. Kemerlis. “*Peer assisted content distribution over router assisted overlay multicast”*. 1st Euro-NF workshop on Future Internet Architecture, Paris, France, Nov. 2008.
23. K. Katsaros, V. Kemerlis, C. Stais, and G. Xylomenos. *A BitTorrent module for the OMNeT++ simulator*. In MASCOTS, 2009.
24. D.Trossen (ed), “Integration and Demonstration Plan”, PSIRP deliverable D3.4, June 2009
25. "PAL: Personal and Social Communication Services for Health and Lifestyle Monitoring", available at http://pal.dalore.net, August 2009
26. J. Riihijarvi (ed.), "Description of validation and simulation tools in PSIRP context", PSIRP deliverable D4.4, July 2009
27. J. Lischka (ed), ¨Routing-in-a-slice platform extension requirement¨, OneLab2 Deliverable D7.3, October 2008
28. J. Rajahalme, M. Särelä, P. Nikander, S. Tarkoma, "Incentive-Compatible Caching and Peering in Data-oriented Networks", Re-Arch´08, Dec 2008.