PSIRP
PUBLISH-SUBSCRIBE
INTERNET ROUTING
PARADIGM

PSIRP
Publish/Subscribe Internet Routing Paradigm
FP7-INFSO-IST-216173

DELIVERABLE D4.1

Preliminary Validation Plan and Selection of Tools

Title of Contract	Publish/Subscribe Internet Routing Paradigm
Acronym	PSIRP
Contract Number	FP7-INFSO-IST 216173
Start date of the project	1.1.2008
Duration	30 months, until 30.6.2010
Document Title:	Preliminary Validation Plan and Selection of Tools
Date of preparation	27.06.2008
Author(s)	Trevor Burbridge (BT), Konstantinos Delakouridis (AUEB), Konstantinos Katsaros (AUEB), Leonidas Kazatzopoulos (AUEB), Giannis Marias (AUEB), Pekka Nikander (LMF), George Polyzos (AUEB), Janne Riihijärvi (RWTH), Dirk Trossen (BT), George Xylomenos (AUEB)
Responsible of the deliverable	Janne Riihijärvi, RWTH Aachen University Phone: +49 2407 575 7034 Fax: +49 2407 575 7050 Email: jar@mobnets.rwth-aachen.de
Reviewed by	Dirk Trossen, Arto Karila
Target Dissemination Level:	Public
Status of the Document:	Completed
Version	1.0
Document location	http://www.psirp.org/publications/
Project web site	http://www.psirp.org/

SEVENTH FRAMEWORK PROGRAMME

European Commission
Information Society and Media

Table of Contents

Executive summary

This document outlines the initial plans for validation and evaluation of the PSIRP architecture. Both qualitative and quantitative evaluation are covered. Qualitative validation incorporates especially security-related validation activities as well as socio-economic validation. Quantitative evaluation on the other hand will consist of evaluation of the behaviour and performance of the architecture as whole as well as selected individual components in a number of scenarios of different scales. The document also discusses tools to be used in the activities at some length, and dissemination activities related to extensions and new tools to be developed in the project.

This document has been produced in the context of the PSIRP Project. The PSIRP Project is part of the European Community's Seventh Framework Program for research and is as such funded by the European Commission. All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

1 Introduction

This document outlines the plans for evaluation activities to be carried out in the PSIRP project. Due to the ambitious objectives of the project, targeting the development of an entirely new internetworking architecture, the evaluation of the architecture and related technologies necessarily has to cover a wide ground.

We distinguish on the highest level between two categories of evaluation activities: *qualitative* and *quantitative*. The former of these will focus on validation in terms of security and socio-economic aspects. The latter will consist of evaluation of the behaviour and performance of the architecture as whole as well as selected individual components in a number of scenarios of different scales.

Overall, our goal is to demonstrate that the developed publish-subscribe based internetworking architecture will offer comparable performance to current IP networks for all applications, while a number of applications will benefit from PSIRP technologies in terms of network performance. However, it would also be obviously unrealistic to expect that a complete evaluation of the PSIRP architecture can be carried out within the project lifetime (considering that the TCP/IP stack performance is still being studied in a number of new deployment scenarios). Thus, the activities outlined here should be seen as an important starting point rather than the definitive “final” evaluation of the project outcomes.

The rest of the document is structured as follows. The planned qualitative evaluation activities are described in Chapter 2, followed by an overview of the plans for quantitative validation and performance evaluation in Chapter 3. Our current understanding on the tools to be employed in validation and evaluation activities is discussed in Chapter 4. We also plan to perform dissemination and exploitation activities related to tools and validation, especially towards open source initiatives and testbed activities. Plans for this work are described in Chapter 5 before conclusions are drawn in Chapter 6.

2 Qualitative Architecture Validation

In this section we give an overview of the planned qualitative architecture validation activities. We first focus on security-related issues in Sections 2.1 to 2.3, starting with an overview of the security issues in publish/subscribe architectures followed by an account on the planned validation activities. We then discuss the planned socio-economic validation activities in Section 2.4.

2.1 Overview of security issues in publish/subscribe architectures

To evaluate the PSIRP framework in terms of security we need to identify security services that might be requested from end-users, or provided by the pub/sub service providers. We summarize our security review objectives using three different domains:

- The *end-user domain*. It consists of publishers and subscribers. Publishers and subscribers may not trust each other, and may trust neither the pub/sub network service nor the infrastructure.
- The *pub/sub service provision domain*, consisting of the pub/sub network service providers and the end-users (publishers and subscribers). The provider may not trust publishers and subscribers, and vice-versa.
- The *infrastructure domain*. Its components (cache elements, label switches routers, forwarded nodes, multicast points) may not necessarily trust each other.

2.1.1 Authentication

Two flavours of authentication, namely *end-to-end* and *point-to-point*, are considered in the context of pub/sub systems.

End-to-end authentication applies to the end-user domain. In this scope, if the publisher A and the subscriber B exchange signalling information, messages, notifications or data, both can verify the originator of these communication means. On the other hand, point-to-point authentication applies to the service provision and infrastructure domains. In the former case, end-users should trust and verify providers', services and vice versa. In the latter case, when network elements are logically interconnected, they should establish a mutually authenticated communication channel. End-to-end authentication can be implemented outside of the pub/sub domain (i.e., without involving the providers or the infrastructure), and a major issue here is the scalability of the pursued solution. Traditional PKI might be a solution but it is a question if it scales enough.

In the service provision domain, providers and end-users should have a symbiotic relation. In that sense, strong authentication might be used, such as variations of SSL, Kerberos, HTTPs, etc might be essential, probably using lightweight credentials (e.g., SPKI). Finally, in the infrastructure domain, authentication of network elements can be provided by many means, e.g., using HIP (Host Identity Protocol, [Pek2006]), PLA (Packet Layer Authentication) or the Tesla framework [Per2002].

2.1.2 Integrity

In the end-user domain, the *subscriptions and publications integrity* must be protected from unauthorized tampering, and this should be proven to the peers. In that sense, digital certificates and signatures might ensure integrity in this level. In a more soft approach, trust assessment tools might prove useful for subscribers and publishers to evaluate the trustworthiness of each other to implement functions without malice.

In the pub/sub service provision domain, *integrity of service* means avoidance of service misuse or isolation of malicious faults. A malicious service provider might insert fake subscriptions or publications to attract end-users, and to profit. This is actually a spamming scenario, which might be mitigated by means of authentication, as previously discussed. Service integrity can be also interpreted as availability; this is the state where pub/sub services become available to end-users when requested, or according to the contract, if any. Thus, prevention of denial of service attacks in this level is essential. A DoS attack might appear when several compromised or spoofed subscribers (zombies) request huge amounts of a particular published artefact (e.g., probably a free-of-charge blockbuster chunk) from a particular publisher or service provider, or when the rendezvous service is requested to process unmatched requests. In the latter case, it is foreseen that rendezvous-targeted attacks will demonstrate equivalent significance as the DoS attacks in the current Internet DNS service [Wun2007]. Rate limitation might be useful at the first stage of pub/sub network development, until the actual pattern and signatures of the potential attacks are identified. Pharming might also be deployed when rendezvous entries are poisoned with incorrect data. Additionally, consider the case where the service provider delivers a free-of-charge and unlimited (in size and number) publication facility to its clients [Wal2000]. Such a promotional decision might rapidly increase its profit, e.g., since advertising opportunities are multiplied in its domain, but on the other hand it might subvert the foundations of its service quality. In that sense, access control [Mik2002] and accounting might also be a requirement in this scope. Additionally, computational puzzles and CAPTCHAs might mitigate botnet efficacy.

Finally, when *infrastructure integrity* is examined, the elements that perform any networking function must be uncorrupted, trustworthy, free from deliberate or inadvertent unauthorized manipulation, and resilient to attacks. Pub/Sub networks place much functionality on the infrastructure, such as caching, coding, routing, forwarding, label-switching and multitasking. This plethora of supported functions creates various attack opportunities and extends the vulnerability set. The following paragraphs illustrate some possible threats at the infrastructure level:

- *Cache poisoning* – Exploits the absence of an authentication layer and forces the network element to believe it has received an authentic caching piece, whilst this is incorrect. This can affect the users serviced directly by the compromised cache or its downstream cache peers. Bogus caches could contain malicious content, such as a worms or viruses.
- *Routing service attacks* – Malicious routing attacks target the routing discovery or maintenance phase by not following the specifications of the routing protocols. Examples include routing message flooding, such as hello, route-request and acknowledgement flooding, routing table overflow and routing cache poisoning [HuP2004]. Proactive routing discovers routes before they are actually need, while reactive algorithms create routes on-demand, i.e., only when they are needed. Thus, proactive routing is more vulnerable to routing table overflow attacks. More sophisticated attacks include the Wormhole and Byzantine attacks. In the former case, an attacker records packets at one location in the network and tunnels them to another location, [HuP2002]. In the latter case, a set of compromised intermediate nodes collude to create routing loops, forward packets through non-optimal paths, or selectively drop packets, which results in disruption or degradation of the routing services [Awe2002]
- *Forwarding phase attacks* – Once the route is established, on the fast data path, selfish or malicious entities drop data packets selectively, fabricate data content, or produce packet replay attacks for hijacking. They can also delay forwarding time-sensitive packets, or inject junk packets [WuC2006].
- *Eclipse attack* – A sufficient number of malicious nodes collude, trying to deceive legitimate nodes for accepting malicious ones as trusted, with the goal of dominating a neighbour of the legitimate nodes. This way the attackers mediate

most overlay traffic and effectively “eclipse” correct nodes from each others' view [Cas2002].

- *Sybil attack* – Usually when a system aims to self-protect itself from faulty or malicious actions, it replicates tasks among several remote entities. Each entity is then identified by an identity. However, when a local host has no direct evidence of the remote entities, it is difficult to ensure that specific identities refer to distinct entities. In the Sybil attack a malicious entity is self-presented as multiple identities and undermines the redundancy employed by the system [Dou2002].
- *Amplification* – A type of flooding and DoS attack where an adversary induces delivery of multiple messages to a single entity by injecting a single malicious message [Wun2007]. For instance, a new advertisement may attract many dormant subscriptions or an un-subscription may trigger multiple re-subscriptions to other publications.
- *Resource consumption attack* – Also known as the sleep deprivation attacks. They aim to consume a victim’s resources. The clogging attack is a common type of this category. The target node is requested to verify signatures, or key exchanged during a Diffie-Hellman handshaking, tasks that require significant processing cycles. The threat appears when multiple demands arrive simultaneously on a target node from several compromised peers. Thrashing is a special case of this type of attack. Unlike typical flooding attacks, in thrashing an attacker induces load by abusing repeated state changes that are processing intensive. This can be accomplished using a set of messages that will likely include e.g., unsubscriptions [Wun2007].
- *Message state effect* – Another characteristic of pub/sub systems is that the routing nodes are state-full for performing filtering, as well as event matching. However DoS attacks can take advantage of this fact. For instance it has been observed that DoS attacks that include subscription messages have more severe effects than DoS attacks that use the same amount of publish messages [Wun2007]. This happens because for each new subscription, the routing nodes need to keep state. This shows that there is a need for mechanism that will manage malicious states.

2.1.3 Confidentiality

In the end-user domain, the *subscriptions and the publications confidentiality* is associated with the right of users not to reveal their identities to peers, not to be linked with publications or subscriptions, to remain anonymous when declaring preferences, announcements or provide/consume content. These privacy rules may apply to many pub/sub applications, where publishers do not know and perhaps do not care to know the identity of the subscribers who receive their information, and vice-versa. In terms of end-to-end information confidentiality that is published and contains sensitive parts, publishers and subscribers may wish to keep information secret from the infrastructure, and the providers. This might require an in-band or out-of-band agreement about the function that maintains the information private when transported in the network, and routed between untrusted domains and network elements. The attackers’ goal in this case is to relate subscriptions, publication, and announcements with physical locations and to link users with preferences and actions.

Service layer confidentiality is associated with the end-users’ choice to remain anonymous, and use service provider’s facilities without the risk to reveal their identities. Additionally the content itself should be sufficiently encrypted when delivered to service providers.

Infrastructure confidentiality means that the network elements maintain security handles to protect from eavesdropping caching entries, routing tables, and multicast parties. Additionally infrastructure confidentiality is associated with those mechanisms that protect the network traffic from analysis and monitoring, for instance using dummy traffic or Privacy Enhancement Overlays (PET) such as onion routing and mist. Hop-by-hop confidentiality support might introduce overhead, but under some circumstances it would be useful (e.g., route advertisements). When PET overlays are used, they contribute to the end-user privacy only if they are able to prevent blending and timing attacks that might reveal the identities of subscribers and publishers.

2.1.4 Availability

In the *end-user domain*, there is probably no availability requirement. Availability is already discussed within the scope of service integrity. *Service availability* means that the publication, notification, announcement, subscription, registration and rendezvous facilities are available when requested. As previously mentioned, several service integrity threats affect availability. Vulnerabilities in this scope are mainly exploited by DoS attacks. Sophisticated DoS attacks are camouflaged as routine flooding circumstances, but their aggregation is the actual threat. Finally, *infrastructure availability* means that the elements should be always available and robust enough to provide routing, caching, coding, multicasting and other lower layer functions. To achieve service and infrastructure availability, (D)DoS mitigation is essential, and this is a twofold objective. The (D)DoS attack when identified should be spread in a minimum network span, or, otherwise it should be populated with a minimum harsh risk. It is widely recognized that high availability protocols, redundant network architectures and system design without single points of failure ensure availability and robustness.

Finally, in a broad sense, spamming might be considered as an end-user domain availability threat. As shown in [Tar2006], although pub/sub architectures are less vulnerable to spam messages than email, this threat might actually exist. Spam messages can be classified in two categories; inbound and outbound. Different techniques should be applied to fight spam messages for each category. Spam may also exist in bogus brokers, which can be used as black boxes that insert spam messages while dropping all legitimate messages or as normal brokers which monitor network traffic in order to learn users' preferences and later on insert more effective spam messages. One key issue in pub/sub architectures is event replication; an event can be replicated to neighbour routers as long as it matches their filters. In case of poor filter design, a spammer may construct a single message that will flood the network.

2.2 Evaluating Security Mechanisms and Concepts of the PSIRP Architecture

The primary objective of the security task is to identify vulnerabilities and threats to the evolving PSIRP architecture. To achieve this, the evaluation team will work in collaboration with the architecture design to identify the security goals. This task will then attempt to identify obstructions to those goals, along with identifying further potential goals that have not been adequately expressed. These security goals are typically expressed in terms of Confidentiality, Integrity and Availability (CIA), along with Authenticity and Accountability.

The security goals of the system, together with the architecture and break-down component designs will be used to analyze threats to the system. Such architecture and component designs are often expressed as Data Flow Diagrams (DFDs) and UML diagrams to aid security analysis. These diagrams detail the actors, processing and data storage functions in the system, along with defining trust boundaries between these functions.

We will employ misuse cases during the threat analysis phase, aided by various threat categorisations such as the CIA(A) triad, Paker's Hexad (extended from the CIA triad to include utility, possession, and authenticity, [Par2002]) and Microsoft's STRIDE. The first two models are expressed in a language of security goals, so threats must be considered which counter such high level goals. STRIDE[1] uses DFDs and UML to uncover Security Design Flaws (mainly Spoofing, Tampering, Repudiation, Information disclosure, Denial-of-Service, Elevation of privilege). These models are used as frameworks to encourage the inventiveness

[1] Recommended by the Open Web Application Security Project (OWASP)

and completeness of the misuse cases. The misuse cases describe actions that should not be possible in the system, and in security terms start from the definition of attackers attempting to perform illegitimate goals. Individual threats can be further explored using threat trees that capture the detailed composite or alternative steps required to perform an attack.

This task will not spend significant effort on the risk analysis for each threat. Risk can be defined by analyzing the likelihood of any attack, along with the impact on the system. Any attempt to perform a detailed ranking of threats requires an understanding of the deployment of the system, along with the users and capabilities of the attackers. Such ranking is often subjective and particularly difficult for general networks and middleware (where we do not know the exact business models, users and applications). We will therefore use a simple ranking system (e.g. critical, moderate, important, and low as suggested by the Microsoft Security Response Center Security Bulletin Severity Rating System).

The threats and vulnerabilities in the architecture and component designs that are identified will be fed back to the architecture team to develop mitigation strategies and formalize the security goals.

To analyze misuse cases, input will be collected from [Ale2002], [Ale2003] and [Kye2002]. To model security threats the goal-oriented framework for generating and resolving obstacles in [Lam2004] might be useful. Sequences of Misuse Cases could contribute to effective test planning of the PSIRP architecture and system, including:

- Specific Failure Modes,
- Security Threats, and
- Exception-Handling Scenarios.

2.3 Modelling and Analysis of Cryptographic Protocols for PSIRP

To provide authenticity, integrity and confidentiality – privacy services, and to some extent availability and access control, cryptographic techniques are recommended as a powerful tool set. Cryptography relies on symmetric keys, public/private key pairs, and it is combined with hashing functions when integrity and authentication should be provided in the end-user and pub/sub service domains.

To support PSIRP security functionality (CIA or Paker’s Hexad extensions), cryptographic mechanisms and protocols should be designed, modelled and evaluated. The majority of work in the area of cryptographic protocol design and modelling has been based on the two-party communication model, with a Dolev-Yao [Dol1983] intruder. Such a model might be insufficient for pub/sub networks, where no identity is used for the active parties. Furthermore, the set of interesting security problems goes beyond the standard user-oriented examples, such as authentication, key distribution, and secure file transfer; in addition to those, we need to consider service-level and infrastructure security topics, such as group communication and secure multicasting, denial of service prevention, and the overall security of the network service and infrastructure itself. A recent paper [Pek2008] surveys existing work in the area of modelling and analysis of cryptographic protocols.

2.3.1 Adversary model

The standard attacker model in cryptographic protocol design and analysis is that of Dolev and Yao [Dol1983], often enriched with the correspondence assertions by Woo and Lam [Woo1993]. The Dolev-Yao model assumes two honest parties that are able to exchange messages through a powerful adversary that is able to intercept, eavesdrop, and inject arbitrary messages. In the pub/sub model, communication is expected to be one way data transfer rather than two way transactions. This requires two distinct channels. Moreover, given that in a more realistic model the attackers will typically compromise only part of the infrastructure, instead of having complete control, a richer attacker model is needed. Additionally, due to the multicast nature of the pub/sub paradigm, multicast security services are essential, and, thus, [Dol1983] and [Woo1993] models should be extended. Attention will be given to probabilistic or micro-economic adversary models, such as Meadows' model for analysing resource-exhausting denial of service [Mea2001] or micro-economics flavoured models [But2002].

2.3.2 Spi calculus

Process algebras, such as Spi calculus [Aba1998], and especially Patternmatching Spi-calculus [Haa2004], seem to be capable of modelling PSIRP, including multicast communication and explicitly named messages. However, in order to derive useful and interesting results, one may want to reconsider adversary capabilities. That is, instead of assuming a Dolev-Yao type all-capable intruder, one may want to model an intruder that is capable to subscribe to (eavesdrop) any messages and message sequences (publications) that it knows about, but has limited capabilities of eavesdropping messages whose names they do not know or publishing messages on message sequences that they do not know about.

2.3.3 Strand spaces

Like Spi calculus, strand spaces [Tha1999] appear capable for basic modelling. For example, multicast is naturally modelled, requiring no extensions. However, as in the case of Spi calculus, an open question is how to model the network and the intruder in order to derive interesting results. One approach might be to continue using the basic intruder model, but add new strands that model the pub/sub nature of the network.

2.3.4 Information-theoretic models

Currently, it is an open issue how the more information theoretic models, such as the one underlying Huima's tools [Hui1999] or developments thereof (e.g., [Mil2001]), could be applied to pub/sub.

2.4 Socio-economic Validation

The socio-economic validation aims at a qualitative evaluation of the architecture, i.e., an evaluation of its validity in particular social and economic scenarios. It is important to note that not all evaluation based on *economic methods*, such as game theory, falls under this category. Evaluation of particular technologies, such as new flow control mechanisms, that apply economic theories does not fall under this category of evaluation but rather under quantitative evaluation.

2.4.1 Scope of Analysis

With this in mind, *the plan for the socio-economic evaluation is to apply methods, such as system dynamics, to evaluate different architectural deployment settings from the angle of creating sustainable value chains*. In other words, the work intends to construct possible value chains or value networks as enabled by our architectural and technological design choices, and evaluate the sustainability of these value chains under different *trigger scenarios*. These triggers can span different dimensions, ranging from end user behaviour (e.g., adoption of technology) or technology (availability of a particular new technology) over corporate strategy (e.g., Merger & Acquisition strategies) to regulatory triggers (e.g., enforcement of particular privacy regulations in future environments). We intend to cover a wide range of potential trigger scenarios, creating a set of potential deployment and therefore value chain scenarios for our architectural solutions.

The difficulty in this type of evaluation, apart from creating the actual trigger scenarios and underlying models (such as system dynamics models) is the correct parameterization of the models for simulating potential outcomes regarding the sustainability of the particular deployment scenario. We intend to work extensively with externally available data in the different areas and describe the particular integration of the data into the model. This will allow future evaluations, e.g., after more realistic or at least different data will have become available, in order to re-run the evaluations we will have performed in this task.

2.4.2 Methodology

The methodology we intend to use is an extension of the methods developed in the Communications Futures Program at MIT. In this research program, value chain dynamics have been studied for some three years prior to the PSIRP project. As part of this work, a methodology has been developed that allows for constructing the deployment (and therefore business) designs that are envisioned for our task.

This methodology leads to a collection of so-called *control point constellations*, representing potential business models on the one hand and technical deployment solutions on the other hand. These constellations are then evaluated under the abovementioned trigger scenarios, leading to an element of quantitative simulation using methods such as system dynamics or game theory. This simulation, albeit quantitative, leads to a qualitative statement of the sustainability of the investigated deployment scenarios when being run under given scenarios (expressed in particular variables of the simulation).

The methodology can be used for different angles with respect to the statements regarding sustainability. These angles can represent, for instance, a particular view of a player within the value chain. It can also evaluate the health of the overall value chain, e.g., through the degree of competition enabled by particular solutions. Work, however, is still required to clearly formulate these particular evaluation angles as part of the overall simulation setting.

2.4.3 Expected Results

Generally speaking, the expected results of this task are the analysis regarding the viability of our proposed architectural choices and technology solutions.

Viability however can be expressed from many different viewpoints (or angles as formulated above), such as corporate viability (i.e., viability from the viewpoint of a single economic player), economic viability (from the viewpoint of a healthy overall value chain), societal viability (from the viewpoint of clearly expressed regulatory goals) or consumer viability (from the viewpoint of fulfilling actual end user needs) and many more. It is expected that the constructed models will allow for turning the model towards these particular viewpoints by choosing an appropriate parameterization of the underlying model.

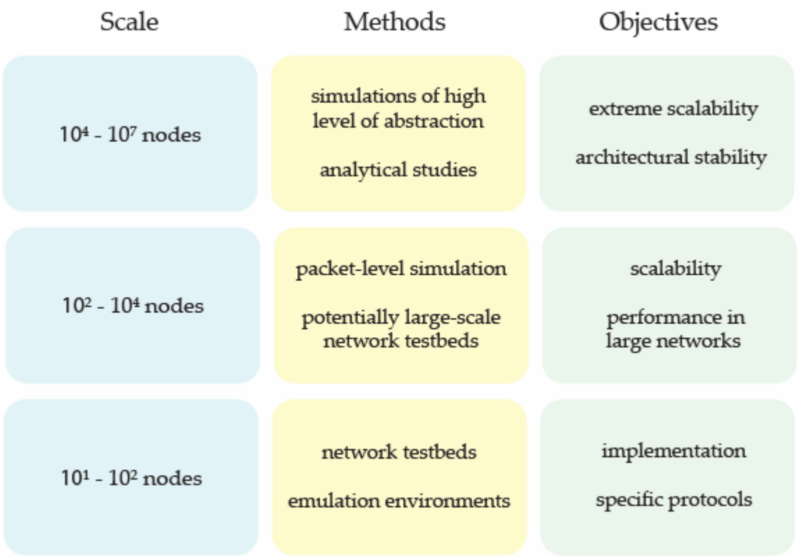
These different viewpoints are expected to serve as an input to judge, e.g., investment of particular players in the projected value chains or regulatory actions required for sustaining a particular balance in others. In other words, the expected results specifically but also the methodology in general is expected to serve as a *strategic decision tool* that will help evaluating the deployment of particular (architectural and technological) choices made by the project. This evaluation is expected to be done by investors, such as the industrial partners in PSIRP, but also potentially regulatory bodies and other stakeholders in this process of deploying a potentially important part of the Future Internet.

It is clear that the range of viewpoints that can be covered in our evaluation is likely to be limited, foremost through the availability of appropriate data. We intend however to work with corporate but also regulatory partners in order to 'prime' the models appropriately and possibly derive useful results in various viewpoints of evaluation.

3 Overview of Quantitative Validation and Performance Evaluation

The objective of the quantitative evaluation activities is to ascertain the performance of individual technological components as well as the architecture as a whole in terms of a variety of metrics in a number of different foreseen deployment scenarios. The metrics to be considered will include classical performance metrics, such as the quality of service as perceived by the applications, network overhead, stability, and scaling of state stored in different system components. However, we will also explore additional metrics, such as techniques for quantifying the complexity of the overall architecture, to be used in the quantitative work.

In terms of evaluation techniques and methods, we foresee three levels of abstraction being employed throughout the process as depicted in Figure 1. In the lowest abstraction level the prototype implementations produced in WP3 will be used directly either in physical network testbeds or in virtualization environments (or even both simultaneously). This approach allows the most detailed studies of the implementations and individual protocols as well as the calibration of simulations at a higher level of abstraction. The scale foreseen for these evaluation activities will be on the order of some tens or hundreds of nodes.



Scale	Methods	Objectives
10^4 - 10^7 nodes	simulations of high level of abstraction analytical studies	extreme scalability architectural stability
10^2 - 10^4 nodes	packet-level simulation potentially large-scale network testbeds	scalability performance in large networks
10^1 - 10^2 nodes	network testbeds emulation environments	implementation specific protocols

Figure 1: Scope of quantitative evaluation in PSIRP.

Selected parts of the architecture will be further evaluated using packet-level network simulation as a tool (again possibly integrated with emulation testbeds). Some level of realism is lost, but much larger scales in terms of network sizes become feasible. We expect to simulate networks of at least some thousands or tens of thousands of nodes in this level of abstraction.

Since the project architecture must scale to deployments of Internet scale, we also plan to conduct simulation studies on very large scales to indentify, for example, *hotspots* in terms of scalability of signalling and state. Packet level simulations are no longer applicable in this abstraction level, and the approach adopted will resemble more Monte Carlo studies of distributed processes on graphs. The scale targeted will be at least on the order of some millions of nodes per simulation instance, and larger if possible. Also techniques from population dynamics, fluid-flow modes of network traffic, evolutionary game theory and other such techniques will be used when found to be appropriate in order to study the behaviour of PSIRP solutions in the large network limit.

All of these activities require as inputs scenario definitions as well as some choices of network parameters (such as technologies used, user behaviour, network topology, and models for application behaviour). Prototyping scenarios will be used as a baseline, but we will also place strong emphasis on evaluating against foreseen technology roadmaps and forecasts of network evolution. Also new research work is foreseen in terms of network modelling since many of the architectural choices made in the project tackle issues rarely dealt with in network simulation. For example, there is a clear need for effective domain-level network models. Results of the qualitative evaluation activities will also play a major role here, as different migration scenarios must be studied to create confidence in the viability of migrating towards the publish/subscribe type of internetworking.

4 Tool selection and development

The quantitative evaluation of the PSIRP architecture will rely on a set of primary tools that will allow us to simulate and emulate the architecture, depending on the evaluation scale and goals, as well as on a set of secondary tools that will support the simulation and emulation tasks, such as topology generators. The following selection of potential tools for use in the quantitative evaluation guiding is based on three principles;

- **Openness**: Rather than producing a commercial product, the project targets the development of a first cut towards a publish/subscribe architecture that will be open to extensions by others. As a result, the tools used during development but also during evaluation, need to be openly available to anyone in the network research community. Fortunately, numerous open source tools are available for both tasks, therefore the focus of our tool selection process will be on these tools.
- **Extensibility**: The development of a radically new architecture is very likely to create the need for developing new evaluation tools that may better capture its novel nature. Rather than developing and maintaining a large set of disparate tools, we would prefer to build upon proven existing tools and then share the results with the network research community. Again, open source tools are ideal for this purpose since both the original platform and our extensions will be easy to make publicly available.
- **Familiarity**: Considering that the learning curve of many evaluation tools is very steep and that the duration of the project is limited, it is reasonable for the project team to show preference to tools with which it is already familiar to some extent. The use of tools that the project team has no experience with will only make sense if the benefits from their use greatly outweigh their learning curves.

4.1 Packet level simulators

The network simulator version 2 (ns-2) is an open source discrete event simulator specifically targeted at networking research. While the core development of ns-2 is supported by US based organizations (DARPA, NSF and ACIRI), many external projects have contributed large amounts of code to ns-2 that have become an integral part of later releases. Ns-2 is probably the most widespread open source simulation platform in the networking research community, therefore extensions made to ns-2 can be disseminated to a very wide audience. Ns-2 originally offered strong support for TCP and congestion management, as well as unicast and multicast routing, which it later complemented with good support for wireless and mobile networking. The simulator is written in C++ but most compiled C++ simulation objects are paralleled by Object TCL interpreted objects, with the Object TCL interpreter itself being embedded into the ns-2 executable, thus allowing simulations to be controlled in great detail by interpreted Object TCL code.

There exists an emulation facility that allows traffic to flow into ns-2 from a real network and vice versa, but it is not portable to all the platforms where ns-2 is available. While ns-2 has some support for running large simulations, including hierarchical routing and session level simulations where complex paths are abstracted as links, it is not really appropriate for very large general purpose simulations. There is no support for parallel or distributed simulation, and therefore the scale of the simulations is limited by the single most capable machine available to any given researcher. This may be the most important limitation of ns-2 for the goals of the PSIRP project. On the other hand, many members of the PSIRP team have considerable experience with ns-2, which includes writing entirely new modules for it.

Even though ns-2 is gradually evolving with new releases coming out once or twice per year, an entirely new version of the simulator, ns-3, is under development. Ns-3 is actually an entirely new simulator that will be as far as possible, but not fully, compatible with ns-2, in the sense that the large library of existing simulation modules will be reused as much as possible, but some code is likely to require modifications. Probably the most important departure of ns-3 from ns-2 is the specific targeting of parallel and distributed platforms for simulation, something that will require existing code to be modified but will greatly improve the ability of ns-3 to simulate very large networks. Other goals of ns-3 include allowing real network code that is publicly available to be used inside the simulator, mirroring to a larger extent real networking code, by introducing the APIs available in real systems and removing the Object TCL code that is not necessary in order to allow the developers to concentrate on the core C++ code.

While ns-3 is very promising, it is still not even at the alpha test stage, meaning that the code is frequently modified and even the API has not been frozen yet. As a result, it is not reasonable at this stage to port existing ns-2 code to ns-3; therefore the practical use of ns-3 is still quite limited. However, the development may be sufficiently advanced during the lifetime of the project to allow us to exploit its advanced features, in particular parallel and distributed simulation, for our evaluation work. While ns-3 is similar to ns-2, meaning that the experience of project members with ns-2 will be useful for ns-3, the changes in the internal structure of the simulator mean that time will be needed in order to learn its new features, especially those related to parallel and distributed simulation.

Another open source simulation platform with which many project members have experience, albeit less than with ns-2 on the average, is OMNeT++, also written in C++. OMNeT++ in itself is not an entire network simulation platform but only the base on which network simulation modules can be implemented. These modules are separately maintained and distributed, which makes for a more distributed developer community than with ns-2 but also for less consistency between the different parts of a complete simulation and introduces the possibility of running into unexpected errors as the different modules of a simulation evolve independently.

The modules available for OMNeT++ are not as many as for ns-2, but as more developers turn to OMNeT++ the availability of useful modules has grown. Of particular interest to the PSIRP project is the availability of a package (Oversim) for simulating content based routing based on distributed hash tables which is actively maintained and widely used. The simulator supports parallel and distributed simulation with the multiple instances communicating via MPI, as well as support for network emulation via interfaces with real networks and the ability to use real networking code inside the simulator. It should be noted that while OMNeT++ does not have the following and credibility of ns-2 in the networking research community, it actually offers now many of the features promised by ns-3 for the future.

4.2 Emulation platforms & testbeds

In order to simulate large networks in a single machine, simulators necessarily offer an abstract model of a real network that hopefully captures the essential features of reality. Unfortunately, the complexity of modern simulators makes it hard to determine whether a simulator is indeed a reasonably accurate model of reality, especially when considering a novel network architecture as the one targeted by PSIRP. For this reason, experiments will need to be performed in real networks in order to completely assess our design and uncover any problems that cannot be detected by simulation alone. Fortunately, recent advances in networking research testbeds and virtualization architectures will offer the project the opportunity to test its ideas on medium to large scale networks.

Regarding testbeds, the *PlanetLab* initiative [Pll2008] was the first to provide a large-scale shared experimental facility to researchers around the world based on each partner site contributing a set of machines that formed part of the PlanetLab virtual network. The European part of the global PlanetLab network is *PlanetLab Europe* which is interconnected with the US based part of PlanetLab hosted out of Princeton University. The *OneLab2* FP7 project (starting on 1st of September 2008) aims to bring new partners into PlanetLab Europe, especially those with cutting-edge network environments. Since some partners of the project are already affiliated with OneLab2, it seems to be an ideal platform on which to test the architecture that will be developed by PSIRP in a realistic network setting. While the number of nodes participating in any given OneLab2 experiment cannot be large in the long term, since many experiments will be taking place in parallel, the diversity of systems and locations that are connected to OneLab2 greatly exceeds what can be offered by the PSIRP partners themselves.

Regarding virtualization, the high performance computing facilities available to each partner, including clusters of multicore machines, can be exploited to instantiate very large numbers of virtual machines running real networking code and communicating over both simulated and real network links. In the spirit of relying on open source tools, the most likely candidate virtualization platform is Xen, upon which it is easy to run FreeBSD, the platform of choice for the lower layers of the PSIRP protocol architecture. Of particular interest with respect to our project is the construction of a large network emulation environment at one of the project partners that will include a large array of computers on a real network running Xen to multiply the number of virtual machines visible to the project team. It is also envisioned that small or large clusters of machines running Xen and the PSIRP code at each partner site will be interconnected to construct a large distributed emulation platform including both local and wide area network links. This platform will not be as realistic as that offered by OneLab2, but it will be able to support experiments of a much larger scale, providing increased flexibility in the design of experiments.

4.3 Topology generators

Since the PSIRP project aims to design an architecture that will operate on the scale of the Internet and beyond, it is imperative that any simulations will consider realistic network topologies. Since the topology of the present Internet is a result of economics rather than network constraints, it is reasonable to expect that it will not significantly change even if an entirely new networking paradigm is introduced, including the one advocated by PSIRP. Therefore, the project will need to rely on Internet-like topology generation tools to create realistic topologies at different scales for input into the simulation platforms. We can roughly split the available topology generator tools in two categories: standalone and embeddable, which will be discussed separately in the following section.

In the standalone category, the graph library produces an output file with a description of a graph that can then be imported to another tool, such as a simulator. One of the oldest and most popular standalone topology generators is the *Georgia Tech Internet Topology Module* (GT-ITM) which creates hierarchical network topologies similar to the structure of the Internet in the mid 90s. In particular, GT-ITM offers the transit-stub topology model in which a number of (transit) routing domains form a backbone that interconnects (stub) routing domains which only forward traffic originating/terminating therein. In this model, a graph is first generated with each node representing an entire transit domain. Each node is then replaced by a new graph, which represents the backbone structure of the transit domain. Stub domain graphs are then generated and connected to each transit domain node. Additional edges may be placed between transit and stub domain nodes and/or between stub nodes belonging to different domains. The selection of edges inside a routing domain is provided by a variety of alternative models ranging from pure random to several Euclidian distance aware models. In the produced topologies, the separation between transit and stub domains is enforced with the definition of appropriate edge weights. These weights ensure that, by applying traditional shortest-path algorithms, as those implemented by current routing protocols, the resulting routing reflects the hierarchical nature of the network e.g. two nodes in the same domain are connected by a path containing routers residing only in that domain, a path connecting two nodes in different stub domains goes through one or more transit domain. GT-ITM interfaces directly with ns-2, being part of the comprehensive ns-2 all-in-one distribution, while project partners are currently working to interface it with OMNeT++. It should be noted that the interface is one way and offline, in the sense that GT-ITM generates a topology that is imported into the simulator, but in general it cannot be controlled by the simulator.

Another standalone option is BRITE, which aims to generate topologies that accurately reflect the actual Internet topology in terms of the hierarchical relations between nodes and their degree distributions. BRITE can create many different network models for smaller or larger networks, it allows models from GT-ITM or other systems to be imported for further processing and provides export filters for interfacing with simulation tools such as ns-2 and OMNeT++. Unfortunately BRITE is no longer actively being maintained, and some of its features do not function properly, but, to some extend it has proved useful as a means of interfacing the output from other tools with the simulators under consideration. More specifically, BRITE is being used as an intermediate step in the translation of GT-ITM topologies into OMNeT++'s topology description language (NED). BRITE's GT-ITM transit-stub topology parser has been extended to preserve the hierarchical network structure, which was until now flattened, and Omnet++'s BRITE-to-NED conversion patch has been extended in order to support this hierarchical structure and at the same time to be interpretable by the Oversim package, which requires a specific NED description structure for the underlying network topology.

The consortium has also substantial expertise on modelling problems related to wireless networking, and we seek to integrate the related in-house tools to the selected simulation environments and then release the key topology generation components to the networking research community under an open source license.

4.4 Graph libraries

In the embeddable category, graph libraries become part of a larger program that instructs them to create graphs and then either operate on them or pass them to the program for its own use. One such graph library is *igraph*, which includes many algorithms to solve graph oriented problems such as routing and calculate graph properties such as connectedness, without the need to implement such algorithms into the simulator. The igraph package interfaces with C, Python and Ruby, but it is actually written in ANSI C. Another option is

LEMON, a C++ library for creating and manipulating graphs, also offering algorithms to solve common graph problems. Some project partners already have experience with LEMON. A third option is the Boost graph library which is actually part of the Boost framework of reusable C++ components, one of which is the graph library.

The main potential role of such graph libraries in the project at present is to enable the quantitative evaluation work in the highest level of abstraction. By viewing different networking processes as operations on a labelled graph, abstracting away technology-specific details, very large scale networks can be simulated.

4.5 Tools for value chain analysis

As outlined in Section 2, we intend to base our evaluation on the previous work on value chain dynamics at MIT. The methodology is currently not implemented as a set of software elements (which is not unusual for this type of evaluation). The actual simulation however is currently performed as system dynamics simulations, for which any available tool in this area can be used.

Our intention is to make the evaluation methodology available as a public set of material in order for any party to participate in the particular evaluation of our solution through priming the simulations with their own (potentially often confidential) data. We further intend to use publicly available simulation tools for, e.g., system dynamics, to be added to our evaluation package.

It is expected that simulation data often cannot be made available due to the confidentiality of the underlying data sets. While we intend to follow the open approach as much as possible, we do acknowledge that such restrictions are sometimes necessary and will likely lead to better results than rejecting the particular data sets. Hence, data sets are intended to be made available, if possible, and in agreement with the owner of the particular data sets.

4.6 Discussion and Other tools

The project will try to use similar underlying facilities in the simulation and testbed platforms in order to employ as far as possible the same protocol and demo application code in both cases. For example, if a distributed hash table approach is taken for some aspects of routing, an attempt will be made to use the same DHT in the form of a real implementation in OneLab2 and in the form of a simulation module in ns-2/3 or OMNeT++. At a higher level, if a multicast facility built on top of a DHT based routing substrate is deemed to be appropriate for the project, we will prefer to use a scheme available both as a real implementation and as a simulation module. After a preliminary examination of the available possibilities, we have identified Pastry (for DHT routing) and SCRIBE (for DHT multicast) as possible options for use in both a real network setting, using their publicly available implementations from their creators, and in OMNeT++ via the Oversim package. Such an integration of implementation code and simulation code is already a part of the project workplan.

It is clear that additional utilities will also be required for, e.g., workload generation and statistical analysis of the results. Unfortunately many of the domain-specific tools developed for simulation and analysis of IP-networks are not directly applicable for work in evaluating publish/subscribe network architectures due to fundamental changes in the nature of traffic patterns. We envisage limited new development efforts being needed to create such tools or algorithms for them. However, similar basic principles as outlined for software selection above will be applied, and any extensions will be targeted towards widely used open-source platforms whenever possible. Also, other tools used in the evaluation activities related to, for example, game theoretic analysis or use of fluid models will be selected and exploited according to these principles.

5 Dissemination Related to Tools

The emphasis on open platforms and testbed initiatives has also been made to create and improve dissemination and exploitation opportunities for the project. We plan to release substantial amounts of code from our simulation implementations and developed tools under an open source license using similar principles as for code releases for our prototypes developed in WP3. This plan extends also to implementations related to specific simulation components (such as network coding) that will be used to evaluate the merits of particular technologies even if they are in the end found unsuitable for inclusion to the overall architecture. Making simulation implementations publicly available is also important in creating confidence in results reported in research articles, to enable repetitions of experiments by other groups, and to enable others to build on our evaluation work.

In addition to software releases we plan to interact with the research community beyond the project in dissemination and liaison activities related to testbeds and emulation environments. Many of the initiatives active at present have been implicitly targeting IP-based networks, and offer little support for evaluating alternative internetworking paradigms. PSIRP has direct contacts to, for example, OneLab2, which will be targeted for such an exchange of views and new requirements for network testbeds.

6 Conclusions

In this document we have outlined and discussed the main issues and approaches for qualitative and quantitative evaluation of the PSIRP architecture together with the associated tools. The foundations of our validation approach are firmly rooted on the experience of the project partners in similar activities for classical internetworking architectures and publish/subscribe overlays.

However, we also foresee substantial extension of the state-of-the-art in validation due to the unique characteristics of the project. In line with the openness of the project, we plan to make our evaluation activities as transparent as possible. This includes both the use of well-established tools whenever possible as well as the distribution of developed dedicated evaluation and validation tools and technologies in an open manner. Workpackage 4 will collaborate closely with other technical workpackages throughout its lifetime, providing feedback to architecture development and implementation. We also plan to have substantial liaison and dissemination activities centred on the validation work, which will be coordinated with the project-wide dissemination and exploitation activities housed in WP5.

7 References

[Nik2008] P. Nikander. G. F. Marias, “Towards Understanding Pure Publish/Subscribe Cryptographic Protocols”, Sixteenth International Workshop on Security Protocol Cambridge, England, April 2008

[Hui1999] A. Huima, “Efficient Infinite-State Analysis of Security Protocols”, Workshop on Formal Methods and Security Protocols (1999).

[Mil2001] J Millen, and V. Shmatikov, “Constraint solving for bounded-process cryptographic protocol analysis”, In 8th ACM Conf. on Computer and Communication Security, pp 166-175, 2001

[Tha1999] F.J. Thayer Fábrega, J.C. Herzog, and J.D. Guttman. “Strand Spaces: Proving Security Protocols Correct” Journal of Computer Security (1999) vol. 7 pp. 191–230

[Aba1998] M Abadi and A.D. Gordon, “A Calculus for Cryptographic Protocols — The Spi Calculus” Research report (1998) SRC 149, p. 110.

[Haa2004] C. Haack C, and A. Jeffrey, “Pattern-matching Spi-calculus”, FAST’04 (173) pp. 193–205, 2004.

[Woo1993] T.Y.C. Woo and S.S. Lam. “A Semantic Model for Authentication Protocols”. IEEE Security and Privacy. 1993.

[Mea2001] C. Meadows, “A formal framework and evaluation method for network denial of service”, In Proceedings of the 12th IEEE Computer Security Foundations Workshop. IEEE Computer Society Press, June 1999.9(1):47–74, 2001

[But2002] L. Buttyán, J.P.and Hubaux. “A Formal Analysis of Syverson’s Rational Exchange Protocol”, IEEE Computer Security Foundations Workshop (2002)

[Dol1983] D. Dolev and AC Yao, “On the security of public-key protocols” IEEE Transactions on Information Theory, 2(29):198–208, March 1983.

[Par2002] Parker, Donn B. “"Toward a New Framework for Information Security", in The Computer Security Handbook, 4th ed., edited by Seymour Bosworth and M. E. , New York, NY: John Wiley & Sons, 2002

[Cas2002] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach. “Secure routing for structured peer-to-peer overlay networks.” In Proceedings of USENIX Operating System Design and Implementation(OSDI), Boston, MA, Dec. 2002

[Tar2006] Sasu Tarkoma Preventing Spam in Publish/Subscribe, Distributed Computing Systems Workshops, 2006

[Dou2002] Douceur, J.: The Sybil attack. In: Proc. of IPTPS, pp. 251{260 (2002)

[HuP2002] Y. Hu, A Perrig, and D. Johnson, Packet Leashes: A Defense Against Wormhole Attacks inWireless Ad Hoc Networks. Proc. of IEEE INFORCOM, 2002.

[Wun2007] A. Wun, A. Cheung, H.A. Jacobsen, “A taxonomy for denial of service attacks in content-based publish/subscribe systems”, Proceedings of the 2007 international conference on Distributed event-based systems, 2007.

[Mik2002] Z. Miklos, “Towards an Access Control Mechanism for Wide-area Publish/Subscribe Systems”, In Proc. of the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW’02)

[Wal2000] M. Waldman, A. Rubin, L. Cranor. “Publius, A robust, tamper-evident, censorship-resistant web publishing system”. In the proceedings of the 9th USENIX Security Symposium. August, 2000.

[Pek2006] RFC4423, “Host Identity Protocol (HIP) Architecture”, R. Moskowitz, P. Nikander, May 2006

[Per2002] A. Perrig, R. Canetti, J. D. Tygar, D. Song,, “The TESLA Broadcast Authentication Protocol” In CryptoBytes, 5:2, Summer/Fall 2002, pp. 2-13

[HuP2004] Y. Hu and A. Perrig, A Survey of Secure Wireless Ad Hoc Routing. IEEE Security & Privacy, pp. 28-39, 2004.

[WuC2006] B. Wu, J. Chen, J. Wu, M. Cardei, “A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks”, Wireless/Mobile Network Security, Y. Xiao, X. Shen, and D.-Z. Du (Eds.) 2006 Springer

[Awe2002] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, “An On-demand Secure Routing Protocol Resilient to Byzantine Failures”. Proceedings of the ACM Workshop on Wireless Security, pp. 21-30, 2002.

[Ale2003] Alexander, Ian, Use/Misuse Case Analysis Elicits Non-Functional Requirements, Computing & Control Engineering Journal, Vol 14, 1, pp 40-45, February 2003

[Ale2002] Alexander, Ian and Thomas Zink, Systems Engineering with Use Cases, Computing & Control Engineering Journal, Vol 13, 6, pp 289-297, December 2002

[Kye2002] Soon-Kyeong Kim and David CarringtonIntegrating Use-Case Analysis and Task Analysis for Interactive Systems”, Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC’02)

[Lam2004] Axel van Lamsweerde, “Elaborating Security Requirements by Construction of Intentional Anti-Models”, in Proceedings of the 26th International Conference on Software Engineering (ICSE’04)

[Pll2008] PlanetLab initiative, http://www.planet-lab.org (2008)