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

DELIVERABLE D4.5

Final Architecture Validation and Performance Evaluation Report

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	33 months, until 30.9.2010
Document Title:	Final architecture validation and Performance Evaluation Report
Date of preparation	07.05.2010
Author(s)	Janne Riihijärvi (RWTH), Dirk Trossen (UCAM), Giannis Marias (AUEB), Trevor Burbridge (BT), Paul Botham (BT), András Zahemszky, Jukka Ylitalo (LMF), Dmitrij Lagutin (AALTO-HIIT), Konstantinos Katsaros, George Xylomenos (AUEB), Jarno Rajahalme (NSNF), Kari Visala (AALTO-HIIT), Mikko Särelä (LMF), Borislava Gajic (RWTH), Christian Esteve (Unicamp), Somaya Arianfar, Pekka Nikander, Teemu Rinta-aho, Jari Keinänen, Kristian Slavov (LMF)
Responsible of the deliverable	Janne Riihijärvi (RWTH) Phone: +49 2407 575 7021 Email: jar@mobnets.rwth-aachen.de
Reviewed by	Dirk Trossen (UCAM), Janne Riihijärvi (RWTH), Paul Botham (BT)
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/

Table of Contents

1 Introduction 4

2 Socio-Economic Validation of PSIRP Architecture 5
2.1 Objectives of Evaluation 5
2.2 Methodology 5
2.3 Rendezvous Evaluation 6
2.3.1 Strawman Architecture 7
2.3.2 Design Characteristics 7
2.3.3 Socio-Economic Outcomes 8
2.3.4 Models 11
2.3.5 Scenarios 14
2.3.6 From Design to Markets 15
2.4 ITF Evaluation 15
2.4.1 Strawman Architecture 15
2.4.2 Design Characteristics 17
2.4.3 Socio-Economic Outcomes 18
2.4.4 Models 21
2.4.5 Scenarios 23
2.5 Status and Future Work 24

3 Security Evaluation 25
3.1 Confidentiality of publications 25
3.2 Integrity and authorization of packets 25
3.3 Availability 25

4 Performance Evaluation of Architectural Solutions 27
4.1 Overlay Approaches and Caching 27
4.1.1 Proposed Architecture 27
4.1.1.1 Deployment 27
4.1.1.2 Multicast 28
4.1.1.3 Caching 29
4.1.1.4 Locality Properties 31
4.1.1.5 Content Fragmentation 31
4.1.2 Performance Evaluation 31
4.1.2.1 Simulation Environment 32
4.1.2.2 Evaluation Framework 32
4.1.2.3 Results 33
4.1.3 Conclusions 37
4.2 Intra-Domain Topology Management 37
4.3 Feasibility evaluation of Multiprotocol Stateless Switching 41
4.3.1 MPSS: Multiprotocol Stateless Switching 41
4.3.2 Efficiency evaluation 42
4.3.2.1 State 42
4.3.2.2 Bandwidth 43
4.3.3 Discussion 45
4.3.4 Flexibility 46
4.3.5 Security 46

4.4 Rendezvous Design 47
4.4.1 Updated Results 47
4.4.2 Conclusions 50
4.5 Deploying a Multi-Site PSIRP Test Network 51
4.5.1 Test Bed Set-up 51
4.5.2 Applications 52
4.5.3 Envisioned Tests 52
4.5.4 Future Usage 53

5 Conclusions 54

References 55

1 Introduction

The PSIRP project is an EU FP7 funded project with a 30 month lifetime. Its ambition is to investigate major changes to the IP layer of the current Internet, finally replacing this layer with a new form of internetworking architecture. In this document we report additional results from the qualitative and quantitative evaluation activities related to the PSIRP architecture, obtained after issuing the initial evaluation deliverable D4.2.

Following our evaluation plan laid out in Deliverable 4.1, evaluation activities have been divided into two categories. Qualitative evaluation focuses on validation in terms of both security and socio-economic aspects, whereas quantitative evaluation consists of evaluation of the behaviour and performance of the architecture as a whole, as well as selected individual components. The basis of the evaluation work consists of the architecture design described in deliverables D2.2, D2.3 and D2.4, and the related updates that will be covered in D2.5. Some of the evaluation activities related to the prototyping work are also covered in this deliverable, and will be further extended through a technical report focussed on performance evaluation of the final prototype as described in D3.5. As in D4.2, the evaluation work described here has not attempted to cover all the aspects of the documents mentioned above, and focus has been placed on key technologies and solution candidates to assist in the architecture design process.

The remainder of the document is structured as follows. In Section 2 an extensive update of the socio-economic validation activities is given, now focussing on applying the methods developed in earlier deliverables to the evaluation of different interdomain mechanisms such as rendezvous and topology formation. In Section 3, the updated security related evaluation of the PSIRP architecture and of the various individual components is discussed in detail, including evaluation of some of the updates to the overall PSIRP architecture to be reported in D2.5. Results from the quantitative evaluation activities are then given in Section 4, including for the different work activities outlines of future work to be reported in dedicated technical reports. Finally, Section 5 concludes the deliverable.

2 Socio-Economic Validation of PSIRP Architecture

2.1 Objectives of Evaluation

The socio-economic validation focuses on the two main inter-domain functions of the PSIRP architecture, namely the rendezvous as well as the inter-domain topology formation function. This is due to their importance in realizing the PSIRP vision of a new inter-domain internetworking architecture. The focus of the evaluation lies on better understanding design choices and characteristics of solutions.

For this, the evaluation devises a methodology that extracts main design characteristics and allows for formulating design strategies that can be evaluated within a set of socio-economic scenarios. The findings regarding these design characteristics are then directly fed into the design for solutions.

2.2 Methodology

We realize that purely technical considerations do not suffice when it comes to the viability of a solution in the marketplace. Hence, the problem of a proper design must be seen under a socio-economic light in the sense that the deployment of a technical solution is nothing less than an enablement (or prohibition) of certain markets.

With that in mind, we assert that if we were to understand the markets that a solution is likely to enable (or prohibit), we would be able to make statements about its wider socio- economic viability. Furthermore, we could devise strategies on a technical and socio-economic level that enable desired market constellations through specific designs. Such strategies are important since they define the necessary efforts on a technical as well as socio-economic level to make a design successful in the market place. These efforts could be reflected in particular technical designs, e.g., through a choice of centralized or distributed technologies, but also in accompanying efforts like required research expenditure, standardization, corporate alliances, marketing, raising awareness at end user and/or policy maker level, and many others. For any adoption of a solution, an understanding of these efforts is essential.

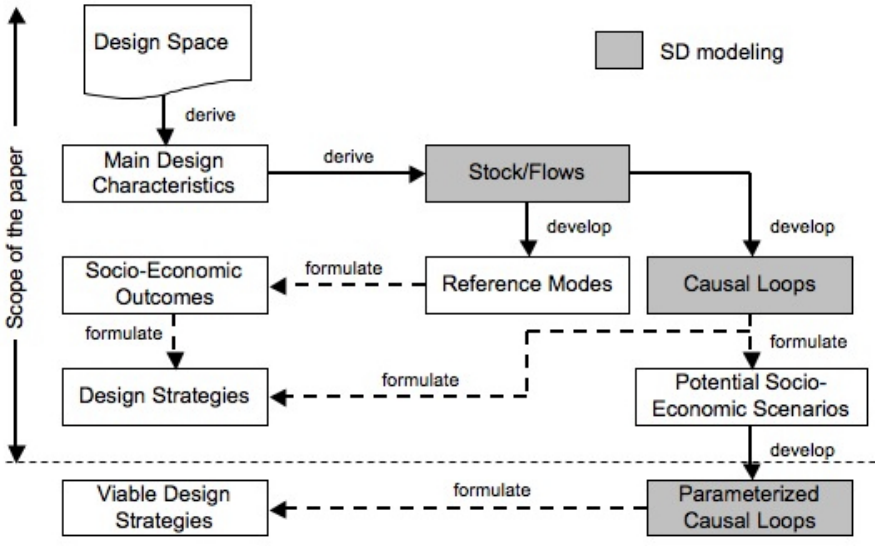

Figure 1: Process to narrow down the design space.

Based on the PSIRP architecture [Tro2009], we intend to make statements about such design strategies within a set of potential socio-economic outcomes that are directly derived from

main structural characteristics of our architecture. Before delving into these considerations, we first outline the methodology to formulate such considerations. This methodology is outlined in Figure 1.

Given the architectural and therefore systems nature of our problem, we choose system dynamics (SD) [Ste2000] as the basis of our methodology – the grey boxes in Figure 1 outline the steps in our methodology that directly involve SD modeling. System dynamics modeling is particularly well attuned to the multi-stakeholder challenge that any design faces, i.e., the process of gathering evidence for causalities that determine the overall system behaviour is well-suited for the "satisficing" process of multi-dimensionally weighing off the different interests of the stakeholders involved. This capturing of causalities is mostly implemented through stakeholder interviews while recording the given, often anecdotal, evidence. SD modeling expresses the causalities as causal loops that influence a set of so-called stocks and flows of the system. For instance, consider the number of chickens as the stock of a (small) system with birth and death rate being its flows (in and out of the system). The causalities influencing the death and birth are captured, and they allow for formulating an analytical model for the time-varying dynamics of the chicken population under a possibly varying set of assumptions for the causalities. The graphical notation of causal loops lends itself to visualizing crucial aspect of the system, possibly even towards the stakeholders involved. The underlying mathematics, i.e., the nested differential equations, is largely hidden from both the SD modeller as well as the stakeholders involved in the process.

Returning to our methodology in Figure 1, we utilize the SD approach to capture main characteristics of our design and their causalities. For that, crucial design characteristics are formulated as SD problems [Ste2000] and eventually represented as stocks and flows in to-be-developed causal loop diagrams – see Section 2.4.1 for the formulation of these problems for our strawman architecture. For each stock and flow model, a so-called reference mode [Ste2000] is developed which represents the expected behaviour of the system under various influences. These reference modes allow for formulating various socio- economic outcomes that are potentially enabled (or prohibited) by particular design choices. A variety of influences are then captured along multiple socio-economic dimensions, ranging from user behaviour over business strategies to regulatory influences. Capturing these influences is usually done through desk research or interviews with crucial stakeholders in the markets assumed to be created, e.g., regulators, incumbents, investors, etc. The causalities of these influences are modelled as causal loop diagrams, resulting in a system dynamics model. Understanding these causalities allows for formulating a variety of design strategies that are geared towards particularly desirable socio-economic scenarios. The expression of design strategies as such is a valuable exercise for the designer as well as the stakeholders involved as they present the potential deployment strategies of various solutions.

Important, however, is the viability of each of these strategies, i.e., how likely is a particular strategy to succeed? Such statements on viability are achieved through parameterizing the auxiliary variables of the developed SD models, in effect running simulations of the models under a given set of variables. This leads to a set of desirable design choices for a set of viable design strategies. As indicated in Figure 1, we focus on the formulation of design strategies in this section and leave the evaluation of their viability to a later discussion.

2.3 Rendezvous Evaluation

In the following, we present the evaluation for the rendezvous function based on our methodology presented in Section 2. We first outline a strawman architecture that represents the main characteristics of any solution for this problem. We then present the models and scenarios that our evaluation is based on.

2.3.1 Strawman Architecture

Rendezvous or discovery in communication solutions comes in many forms and for many purposes. We argue, however, that the main structure is similar throughout most if not all of these solutions. We present, in the following section, a strawman architecture that encompasses the design characteristics of most rendezvous solutions, and we also outline a few examples of known solutions and related design work to underline the generality of our strawman architecture.

When looking closer at existing rendezvous solutions, we can make the following two major observations regarding their design. Firstly, rendezvous is usually performed in a tiered manner. That is, a request is usually sent to a well-known local entity for resolution (tier 3). If that entity is not able to resolve the request, a local federation (tier 2) is consulted. This local federation usually represents some form of administrative boundary, such as a corporate environment, an administrative network, or similar. If there is no entity in the local federation that is able to resolve the request, it is sent via an interconnection structure to other local federations. The entity performing this interconnection represents tier 1 in the process.

Secondly, tier 3 and 2 entities might choose several next tier entities to forward requests to, in case they cannot be resolved locally, including forwarding to different parents for different requests (e.g., based on some local policy). Such choice does not always exist but might be important to consider for certain solutions. This choice can be implicitly implemented through, e.g., address space management or business arrangements, resulting in publishing relevant information towards a variety of different repositories that later need to be consulted.

Based on these structural observations, we propose the strawman architecture of Figure 2 as a basis for our discussions in this deliverable. In this, the 3-tiered structure is represented as the local rendezvous point (RP) being tier 3 with its local rendezvous network (RENE) being tier 2, which in turn connects to at least one interconnection overlay (IO) as tier 1 in order to eventually deliver the request to the grey RP on the right side of Figure 1.

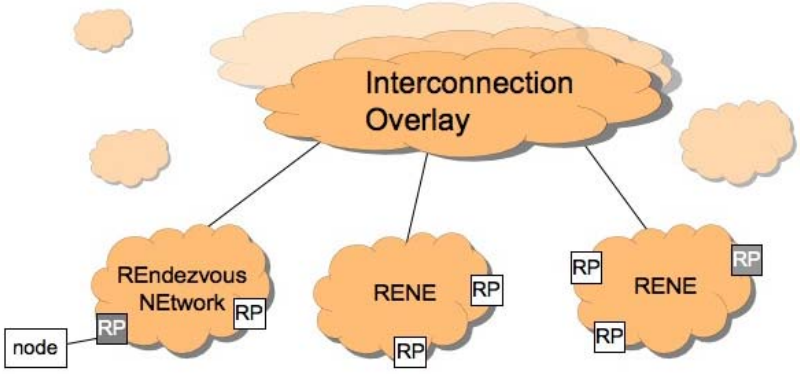

Figure 2: A strawman rendezvous architecture.

The architecture is generic enough to encompass existing rendezvous solutions, and we assert that its fundamental tiered characteristic also applies to future solutions in this space.

2.3.2 Design Characteristics

The crucial step in our design process (see Figure 1) is the input from the design space into the model that helps us deriving design statements on viable choices. This input represents the main design characteristics one intends to focus the discussion on, and it represents the formulation of the problem in the methodology of Figure 1.

Looking closer at the strawman architecture of Figure 2, we can formulate two main characteristics, namely the existence of two types of players, i.e., interconnection overlay and rendezvous network providers, as well as the degree of interconnection between the

rendezvous networks or the different overlays. We believe that these characteristics define the nature of any design choice that implements the strawman architecture of Figure 2. For instance, a higher number of IO providers favours designs with manageable (or low) cost for providing the overlay while solutions with higher costs for overlay provisioning might still be viable in scenarios with a low number of IO providers (as long as the costs can be recovered through viable business models). The number of rendezvous networks could, for instance, give some guidance on required scalability and load balancing for the technical solutions. And the degree of collaboration between these two players defines the required connectiveness between rendezvous networks and interconnection overlay, likely to result in certain routing technologies.

This design focus allows us now to formulate the problems within our methodology of Figure 1, namely:

1. How many interconnection overlay providers will there be?
2. How many rendezvous networks will there be?
3. What is the incentive to interconnect (either within a single or several interconnection overlays)?

While these problems are mainly concerned with design characteristics, a separate set of problems can be formulated with respect to deploying any (particularly large-scale) solution across autonomous systems, such as:

1. How much collaboration will there be between autonomous systems having deployed a rendezvous network and those who have not?
2. How much RENE deployment will there be in terms of autonomous systems having deployed or participating in some RENE?

In the following, we focus on the first set of problems, mainly concerned with the design, leaving out the problem of deployment. We translate these problems directly into a set of stock and flow models from which we derive our socio-economic outcomes as well as the design strategies in the following subsections. In other words, the number of interconnection overlay providers as well as the number of rendezvous network providers, together with the incentive to interconnect (normalized between 0 and 1), represent the stocks in our system dynamics models while the flows are represented by the changes in these stocks. We present the actual models later, focusing on the possible socio-economic outcomes first.

2.3.3 Socio-Economic Outcomes

The possible socio-economic outcomes are derived from the reference modes [Ste2000] for the three stock and flow models that were derived through our formulated problems. Each reference mode describes the possible behaviour of the particular system dynamics model being developed for each of our problems. The different phases of the behaviour as well as the potential final outcomes are depicted in Figures 3, 4, and 5, respectively. It is important to understand that the number of players, depicted along the vertical axis in Figure 3 and 4, only indicates the order of players and is by no means a final number (which in most cases would depend on the case at hand). Also, the exact timeline is left out due to the focus on possible scenarios rather than their actual outcome. A final model and its simulative results will obviously carry appropriate timelines for the simulated scenarios.

The reference modes in Figures 3, 4, and 5 outline the following possible outcomes:

- Monopolization can occur in the space of interconnection as well as in the rendezvous network space. Figure 3 shows the scenario for the former, where after an initial interest for deploying the solution at hand, the initial commercialization and following market consolidation develops towards a monopoly (lower curve in Figure 3) with a single dominant player. Monopolization can also occur in the rendezvous network space, shown in Figure 3. The rendezvous networks can be seen as an instantiation of

certain regions, either geographically or organizationally. This initial regionalization, however, might reverse over time, degrading the rendezvous networks into a single one without any need for interconnection. In this scenario, a direct competition between rendezvous networks and interconnection overlays would occur, removing the need for the (interconnecting) overlay.

- Commoditization occurs when the number of players, both interconnection overlay and rendezvous network providers, will grow. This is likely to lead to a de-valuation of both functions (although regional models of revenue generation are still possible to occur). In this scenario, the establishment of either function, rendezvous networks or interconnection overlays, is likely to be driven by technological developments which make such large-scale deployment possible – we point out other potential drivers for these scenarios later.

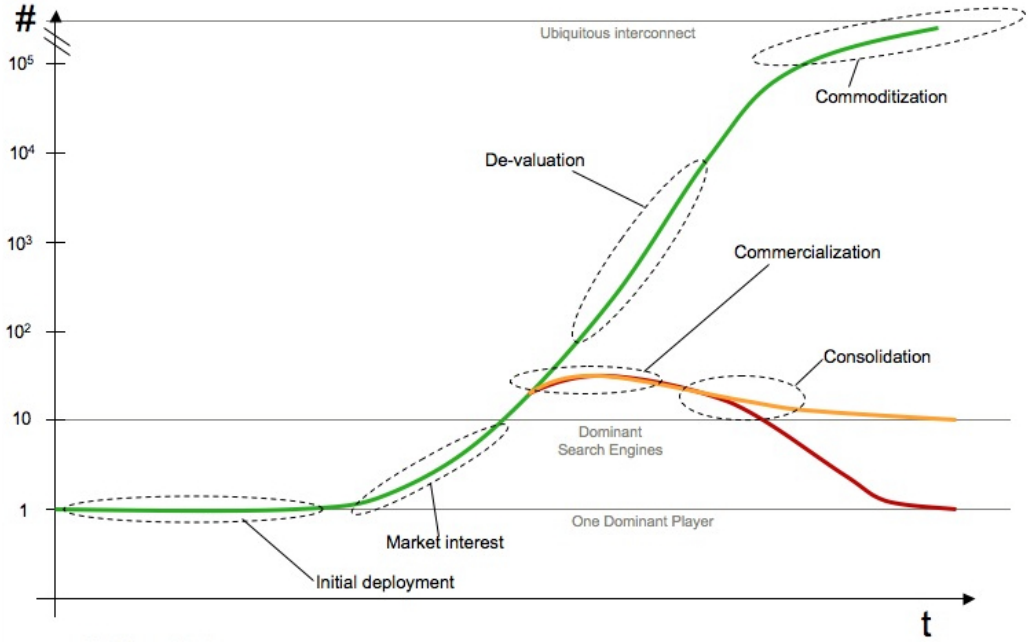

Figure 3: Number of interconnection overlays.

- Stable regions occur for both players when, after an initial market interest, the formation of a stable market occurs with consolidation and a limited number of new market entrants over a sustained period of time. In the case of rendezvous networks, this regionalization is likely to happen along geographic and administrative boundaries although future rendezvous solutions might also enable the formation of rendezvous networks along social network boundaries.

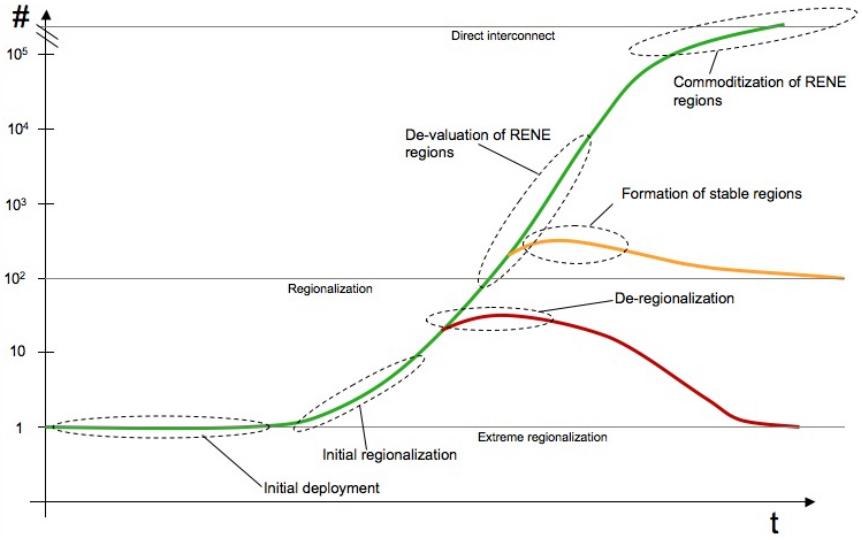

Figure 4: Number of rendezvous networks.

- The aspect of isolation and reachability is captured in Figure 5. A strong incentive to interconnect, here normalized in an interval from 0 to 1, indicates the desire to establish full reachability for the rendezvous functionality. On the contrary, intra-domain solutions, e.g., within single organizations, have little to no desire to interconnect as depicted in the lower two graphs. A likely scenario is depicted by the middle graph that converges towards a fragmented, yet not fully isolated, rendezvous interconnection market. In this case, specific interconnection is achieved by choosing a set of interconnection overlay providers for particular parts of the 'search space'. This scenario depicts, for instance, today's situation of 'controlled information visibility' (or censorship) in some parts of the Internet.

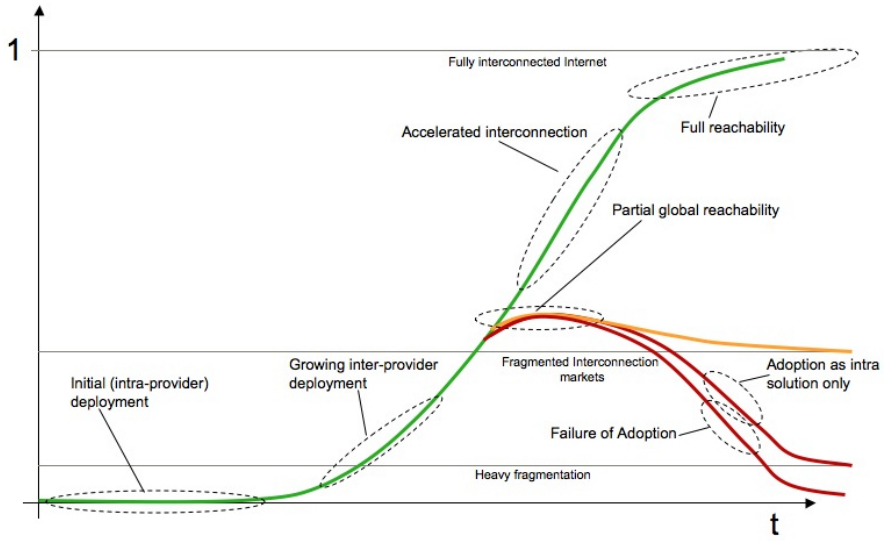

Figure 5: Incentive to interconnect.

It is important to understand that the actual behaviour of any solution for these stock and flow models is more likely to be a combination of the different phases outlined in Figures 3 through 5. For instance, a strong monopoly in the interconnection overlay market is likely to be introduced by a prolonged existence of an oligarchy while such monopoly can easily converge back to an oligarchy or even commoditization case through appropriate socio-economic influences, such as regulation or technological advances. Furthermore, apart from the final

outcome, the reference modes also outline transitory phases, depicted by encircled descriptions in Figures 3 through 5. These are often equally important for outlining strategies for development and deployment rather than only focusing on the final outcome.

2.3.4 Models

In the following, we outline the first iterations of causal loop models that we developed before formulating possible design strategies as an outcome of this first modeling round. As the basis for our causal loops, Figure 6 shows the outcome of the trigger step of our toolkit [Rii2009] applied to our rendezvous example.

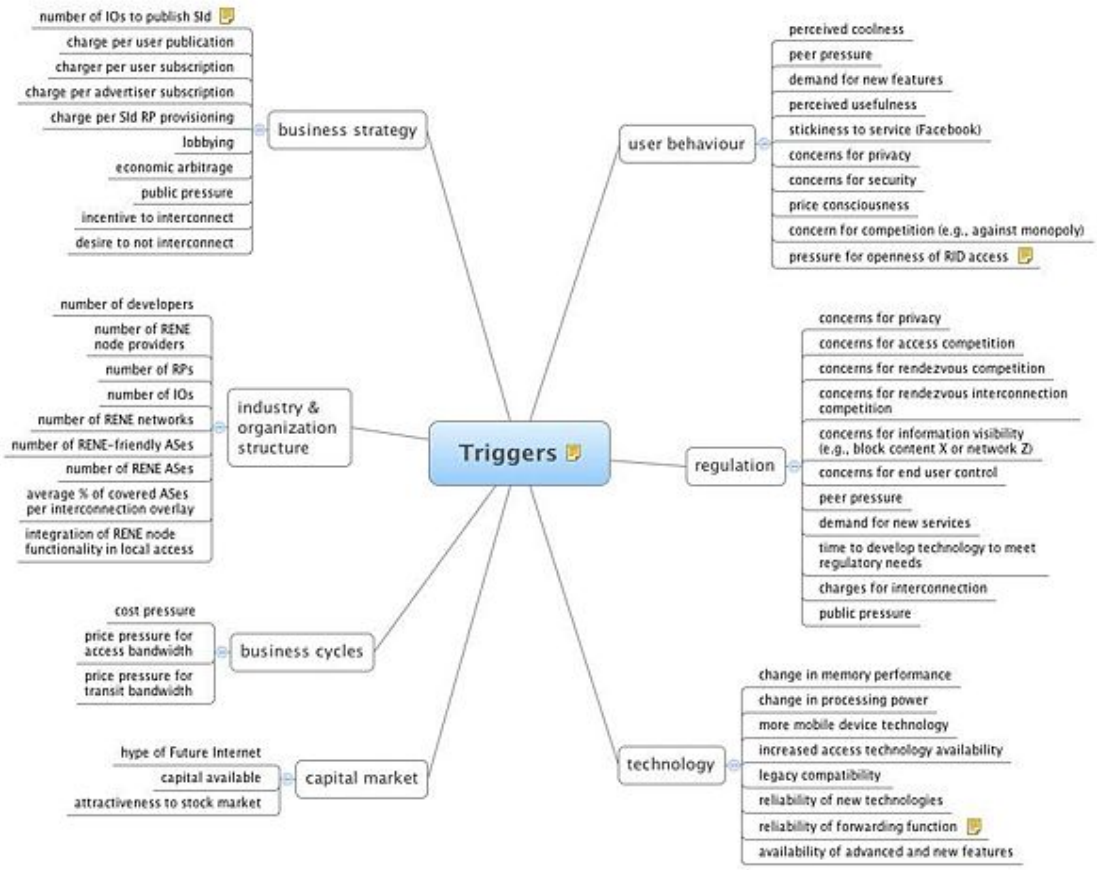

Figure 6: Triggers for system dynamics models.

The triggers are categorized into areas covering user behaviour, business strategy, technology development, regulation, industry and organization structure, business cycles as well as capital market influences on major control points within the overall socio-economic environment.

In order to capture these triggers, we applied the toolkit, as presented in [Rii2009], and recorded interviews with main system designers as well as various stakeholders (the views were captured in a workshop during which various stakeholders (three major ISPs, two major vendors as well as a content provider and a grassroots content creator) provided their insights and opinions on the 'rendezvous space' at large with respect to various issues surrounding user understanding, business strategy, regulatory issues, and technology) in a collection of control points and their influencing triggers. The application of the toolkit proved to be very useful for recording these encounters, explaining the use case to the stakeholders as well as supporting the future work of refining our initial models, which requires re- visiting some of the

evidence given in the first iteration. These identified triggers serve as a basis for identifying the causalities within the developed system dynamics models.

Based on the recorded evidence, we develop the first iteration of system dynamics models for the three stock and flow models. Figure 7 shows the SD model for the first problem, i.e., that of the number of interconnection overlay providers.

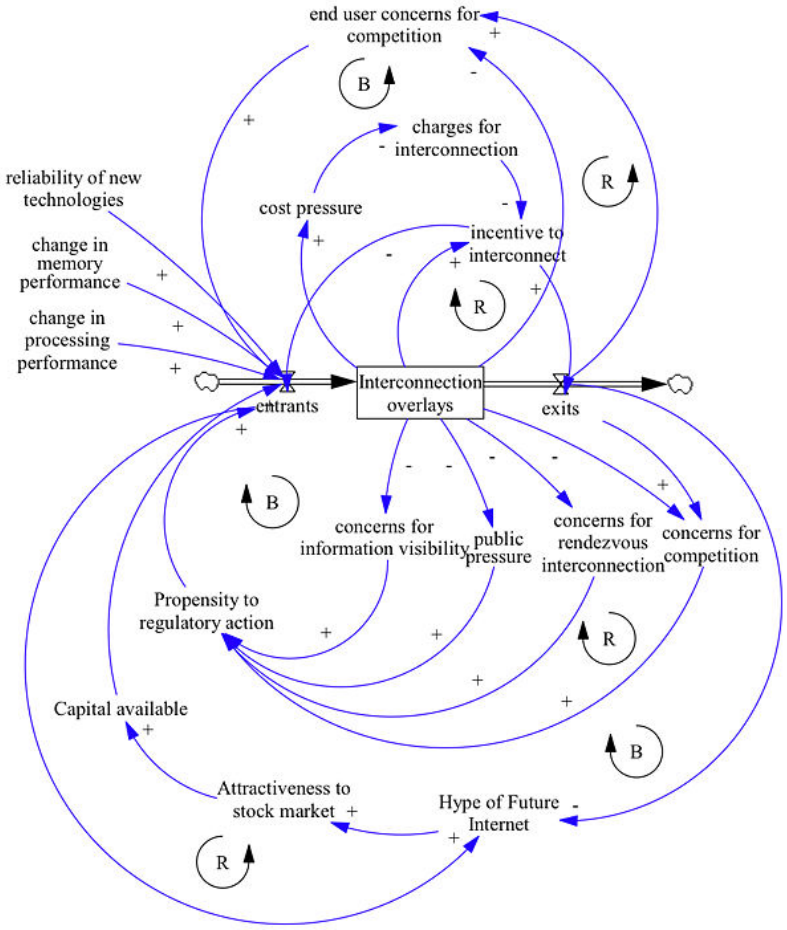

Figure 7: SD model for interconnection overlays.

As mentioned before, the number of interconnection overlays is reflected as the stock of the model with entrants and exits as respective flows into and out of the flow, respectively. The model captures five main areas of causalities:

1. The development and reliability of new technologies and in particular the change in performance with respect to memory and processing power is depicted as exogenous factors feeding into our model (far left side of Figure 7).
2. The regulatory causalities are reflected in a nested loop of four causalities right under the stock symbol. The concern for information visibility as well as the concern for competition are main issues in this space.
3. Market causalities are reflected at the very bottom of the model through a simple hype-capital loop, feeding from the exit into the entrant flow of the model.
4. Business strategy causalities largely revolve around cost pressures, leading to changes in interconnection charges and incentive to interconnect. This is reflected right on top of the stock symbol.

5. Last but not least, end user concerns are depicted at the very top of the model with concerns around competition being of main importance.

The causalities for the second characteristic, namely the number of rendezvous networks, are depicted in Figure 8.

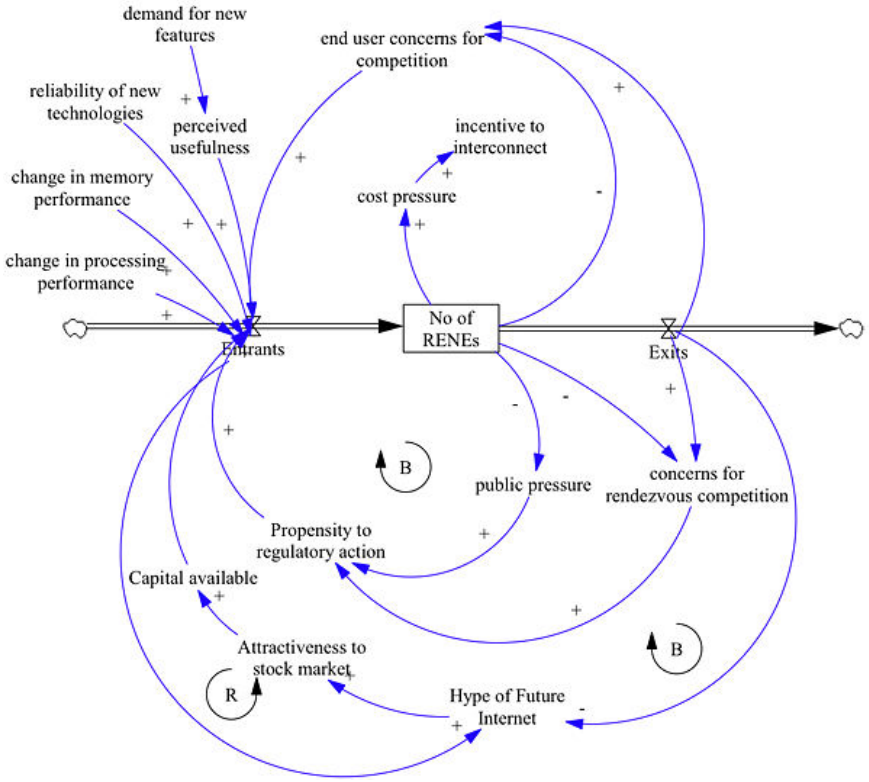

Figure 8: SD model for rendezvous networks.

In many ways, the model in Figure 8 is similar to the one in Figure 7 but with subtle differences:

- The demand for new features (and therefore the perceived usefulness of the rendezvous service) drives the entrance into the market of rendezvous networks as an additional exogenous factor – the rendezvous network providers being user-facing in contrast to the interconnection overlay providers.
- A growing number of RENEs is seen as increasing the incentive to interconnect, driven by cost pressures and the pressure to provide full reachability.
- The regulatory concerns are largely centred around competition issues, less involving issues on information visibility (i.e., the separation of the search space) and interconnection competition.

Last but not least, the causalities for the third characteristic, the incentive to interconnect RENEs through an interconnection overlay or to interconnect several interconnection overlays themselves, are shown in Figure 9. It can be seen that the regulatory part of the model is similar to the one in Figure 7 (the number of IOs). This is not surprising since the number of IO providers directly relate to the problem of interconnection incentive.

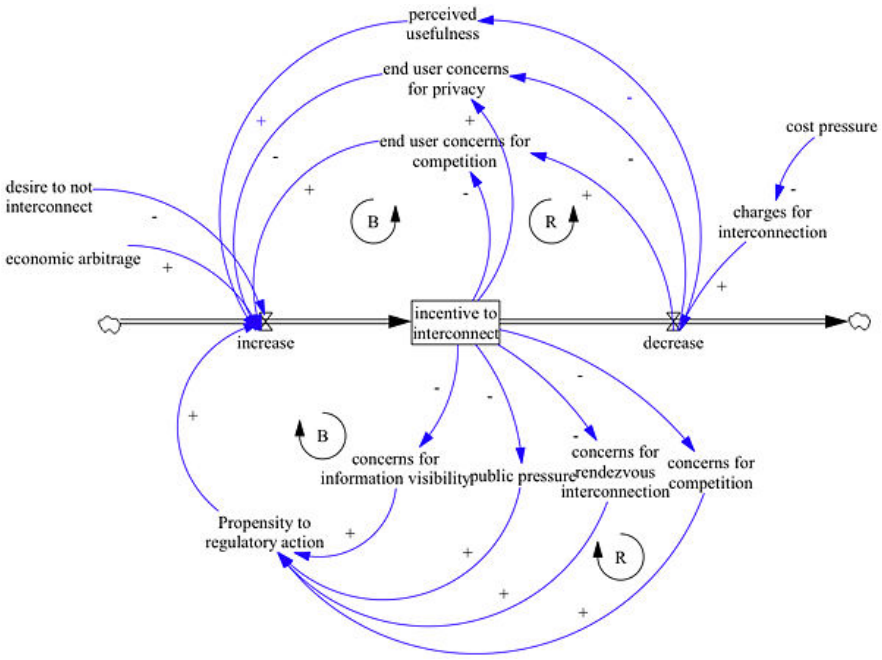

Figure 9: SD model for the incentive to interconnect.

A main difference comes through the captured end user concern for privacy, i.e., there is a causality that expresses the desire to isolate certain parts of the rendezvous network (either within RENEs or interconnection overlays) for privacy reasons. These can be driven by consumer or organizational concerns. This indicates the necessity to be able to isolate parts of the search space. The desire to not interconnect captures a similar isolation aspect, in this case not specifically driven by a privacy concern but rather driven by, e.g., ideological barriers or differences rather than mere privacy concerns. Furthermore, the perceived usefulness of interconnection, e.g., in terms of reachability, drives the incentive to interconnect.

2.3.5 Scenarios

The evaluation of the main design characteristics, based on our models presented in Section 3.4, is driven by a selection of scenarios, each of which defines a particular range of parameterization of our models. In the following, we outline these scenarios as well their main drivers from a stakeholder perspective before outlining the main parameterization of the models resulting from these drivers. The individual evaluation results are then tied back to the reference modes of Section 3.4, giving us some evaluation of the likely socio-economic outcomes for each scenario.

Scenario 1: Privacy Loss Backlash

This scenario considers a backlash against the almost ubiquitous loss of privacy through the use of digital communication with current trends in this direction outlined in [Pew2009]. Several stakeholders are driving this backlash, namely end users (through perceived and actual loss of privacy of crucial data), legislators (through public pressure to tackle largely ungoverned information exchange between major corporations and resulting loss of sensitive data), corporations (through increased public pressure to put in place proper privacy protection mechanisms) as well as stock markets (through negatively reacting towards public pressure).

Scenario 2: Anti-Monopoly Movement

This scenario assumes an increasing movement against various monopolies, resulting, e.g., in increasing number of grass root movements of various forms, as for instance seen in the wireless access space with a number of community schemes [Pew2010]. The stakeholders

driving this trend are end users (through public campaigns), legislators (through increase public pressure and the need to dismantle international monopolies), regional powers (not accepting monopolies imposed by other regional powers) and corporations not being successful in establishing themselves as monopolies.

Scenario 3: Social Serendipity

Driven by the increase in social networking technologies, this scenario assumes a movement to communication modes that are largely driven by social serendipity, i.e., they are based on established social rules, such as friendship, local community, etc. The scenario investigates the possible impact that such trends, as observed in [Pew2010] could have on the nature of the underlying communication environment, here the rendezvous solution. End users are seen as the main stakeholders driving this trend besides corporations attempting to benefit from it as a route to market.

Scenario 4: Regional Power Struggles

Regional power struggles can already be observed in the current Internet in many areas, e.g., the name as well as address space management [ISO1999]. Also the setting of standards for key technologies is often a sign of regional power struggles. This scenario investigates the potential impact of such power struggles on the structure of the rendezvous market. Stakeholders driving this scenario are end users (through perceived superiority of regional values), legislators (through setting policies for strengthening local structures in disadvantage of global ones), corporations (attempting to benefit from such struggles) as well as stock markets (speculating on the outcomes of such regional struggles).

Scenario 5: Copyright War

Digital rights management and its impact on digital communication is already visible in today's Internet, e.g., through various so-called three-strike policies [Pew2010]. The focus on information in our architecture makes an arms race of copyright methods (and the ones trying to circumvent these methods) a likely scenario. Advances in application or network layer technologies to protect individual information items [Tro2009] additionally drive this scenario. Stakeholders involved in driving this scenario are end users (both, valuing protecting their individual content while being prosecuted for sharing other information), legislator (caught in the arms race to react to demands from content industry while preserving the ability to freely exchange information) and corporations (benefiting from new protection schemes).

2.3.6 From Design to Markets

Another focus in our socio-economic evaluation is given by trying to understand the markets being created by the designs that a PSIRP architecture would put in place. For this, we slightly change the methodology presented in Figure 1 in that focus is given towards the socio-economic outcomes, e.g., monopolization of functions or regionalization of markets. The general approach, utilizing SD modeling as the underlying analysis tool, remains the same.

This focus allows for making statements about potential strategies to strive towards particular market outcomes or to prevent exactly these. While the project takes a neutral stand on the particular outcomes, such work can be seen as beneficial for input to corporate or regulatory strategies.

2.4 ITF Evaluation

2.4.1 Strawman Architecture

In this section, we discuss the problem of inter-domain topology formation (ITF), our assumptions and the design requirements driving technical considerations regarding possible solutions. It is assumed the reader is already familiar with the wider definition of the PSIRP architecture, as discussed in [Tro2009]. In this context, the topology layer works with the rendezvous and forwarding functions in the routing phase of communication to:

- Help build routing information of the forwarding nodes and forwarding paths according to policies set by operators and users
- Store policies and network topology information
- Manage edge routers between domains that prevent policy violations and protect domain internals

We assume a PSIRP network will be divided into autonomous systems (domains) controlled (as in the Internet today) by a mix of competing, commercial operators seeking profit and communities like universities and governments that may have other goals. Domain level connectivity is largely determined by the relationships between these organizations, the needs of their customers, geographical, historical and political considerations, and only indirectly guided by the technology used. The starting point for our work has been the current Internet but it is expected that the change in underlying network paradigm will affect the evolution of the domain level topology.

Previous work [Tro2009] has identified the following list as essential properties of the inter-domain topology forwarding function, providing a basis for subsequent design choices:

- The topology layer should allow operators to flexibly control routing policies of packets going through their domain
- It should be possible for customers to define per-RId specific policies, which are then taken into account in overall topology formation
- A PSIRP solution should consider costs and policies of both publishers and subscribers when building forwarding trees (in contrast to the current Internet where only sender side control of the routing exists)
- The topology layer should have enough expressive power to enable complex policies and business relationships between ASes, not relying on assumptions such as a fixed set of Tier-1 operators and strictly hierarchical AS topology (e.g. multi homing and RId-specific partial transit should be easily possible)
- Operators of domains should be able to keep their intra-domain topology hidden, only being required to expose minimal information towards the process of topology formation
- Inter-domain topology formation should not unnecessarily limit implementation and management of intra-domain topologies; there can be multiple different implementations inside domains, all compatible with the inter-domain topology formation
- Incremental deployment on top of the current Internet AS topology should be feasible
- The topology layer should automatically and quickly adapt to changes in network topology and efficiently use available routing resources, in accordance with constraining policies
- Topology formation should take into account the fact that large domains are linked at multiple geographically dispersed PoPs (even if they have only a single logical business association) and by leaking some intra-domain information the routes could be further optimized between the domains
- Topology formation should allow for potentially different policies between the same domains, depending on their point of interconnection

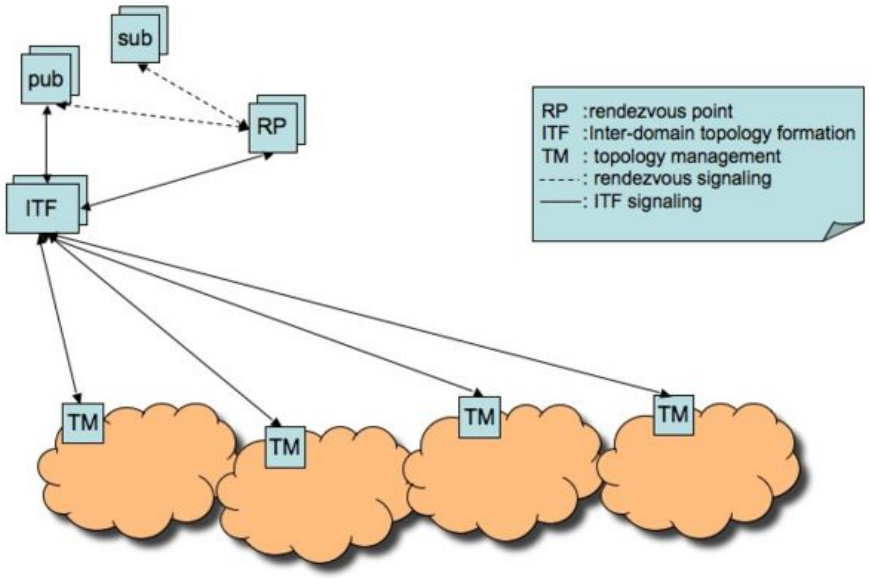

Figure 10: Conceptual components for Inter-Domain Topology Formation.

The corresponding conceptual component architecture relating to inter-domain topology formation has been detailed previously in [Tro2009]. A topology management function is assumed to exist within each autonomous system (domain). This function implements the local topology management and communicates the relevant peering information to the ITF function.

Publishers and subscribers come together in the rendezvous process within the rendezvous point representing the particular SId in which the information items (labelled via an RId) are located. The arrows in Figure 10 show the relations of these components and are not meant to illustrate the exact message and information exchange between them. However, dashed arrows indicate relations stemming from the rendezvous process while solid arrows show topology formation relations.

2.4.2 Design Characteristics

In general, we seek to guide upstream architectural decisions to better take into account economic forces [Ber2001] further downstream, particularly with regard to encouraging technology deployment. In practice, this involves recognising the broad range of likely socio-economic requirements and ensuring the technical capability to best meet these needs.

The ITF supplies inter-domain topologies, ultimately facilitating distribution of content via “quality” routes (for publishers and subscribers who are willing to pay for better than Internet best-effort transport services). Significant efficiencies can potentially be achieved via economies of scale, increasing ITF incentives to grow.

The above discussion has served to define:

- The inter-domain topology formation problem
- Accompanying assumptions
- Design requirements to be satisfied by the ITF function
- Conceptual component architecture for ITF function implementation

Additionally, previous work [Tro2009] has outlined a series of initial considerations to guide ITF design choices, addressing issues such as:

- Role of the ITF component (efficiency and modularity)

- Inter-domain topology information (granularity)
- Publish/subscribe approach to inter-domain topology formation
- Creating a (policy-driven) peering topology market
- Control of the formation process (which parties make decisions)
- Fault tolerance and multipath routing
- Anycast

Given these, we now seek to understand the impact of PSIRP ITF design choices (made to encourage technology deployment) with regard to both:

- Evaluation of which design choices are possible under general (evolving) socio-economic conditions
- Specific security-related considerations, based upon analysis of the information needs, policies and preferences of the parties involved

There are a variety of potential design choices, involving initiation of the topology formation process by either the publisher, rendezvous point (serving the retrieval request) or local ISP to which the publisher is connected. This variety stems from addressing the problem of exposing the necessary (topology) information to the different parties involved. In other words, the fundamental concerns are "who initiates communication" and "how to accommodate sensitivities regarding exposure of information" (e.g. ISP routing topology). Each of these aspects is considered in more detail below.

In the present ITF context, the specific stocks selected as most important are:

- Number of ITF providers
- Importance of RP views when choosing topology
- Importance of local ISP views when choosing topology
- Importance of publisher views when choosing topology

Number of ITF providers is an obvious measure of market size, while the remaining "importance-related" stocks reflect the balance to be struck in accommodating the different motivations and policies of RP, ISP and information publisher.

2.4.3 Socio-Economic Outcomes

As in the rendezvous case, we develop "reference modes" of the ITF system as time series graphs of key variables, showing their likely behaviour in typical scenarios (e.g. observed historically or expected in future). This also serves to capture mental models and suggest appropriate model structure, helping to:

- Identify important variables
- Establish the likely time scale (duration) of interest
- Highlight relevant behaviour the model must mimic (e.g. oscillation, overshoot and collapse, S-shaped growth etc.) in a particular regime

In our ITF case, we have identified four stocks as the most important variables, while ten years is here suggested as a likely order-of-magnitude estimate for time scale (based largely on historical experience of Internet peering evolution). The general behaviour each stock might reasonably follow is postulated as a traditional "S-curve", involving a relatively slow start-up period, followed by rapid growth, terminating in a stable "plateau" at a level determined by overall success of the technology.

Considering first the number of ITF providers, the market might be expected to evolve in various ways over time, subject to regulatory constraints. In particular, fragmentation into "regions" may occur, these defined according to some general notion, such as:

- Geography to help optimise network resource usage
- Local peering, conditioned by business relationships, multiple providers cooperating to provide extended coverage
- Resilience requirements (e.g. financial services provision) implying direct fragmentation by market

The reference mode for the number of ITF providers stock is shown in Figure 11 below. The degree of success envisaged ranges from an optimal scenario involving very strong take-up to relative failure where Internet users/providers largely ignore PSIRP and rely mainly on BGP-type interconnect.

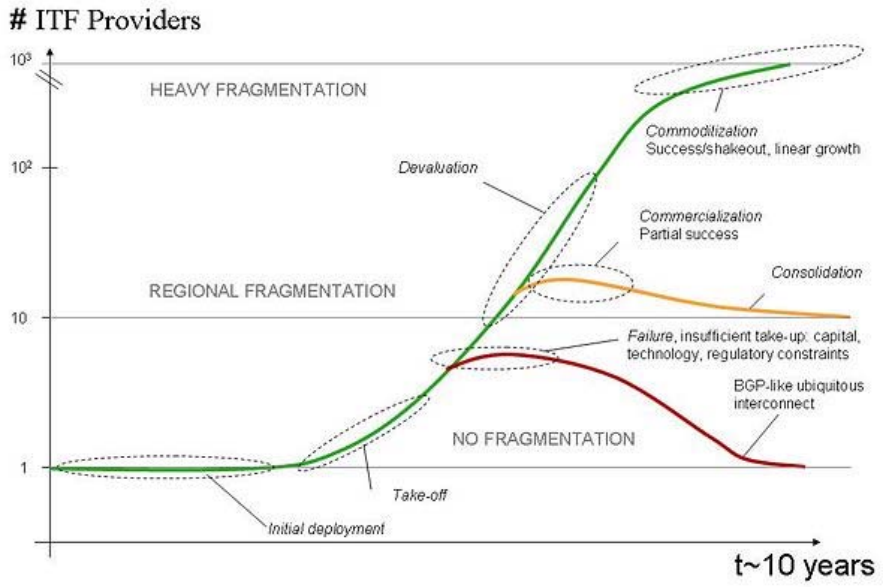

Figure 11: # ITF providers: reference mode.

With regard to the interplay between rendezvous provider (RP), ISP and publisher wishes when choosing topology, different views will dominate depending on the prevailing market and regulatory conditions (e.g. balance between legal requirements on transit/peering and packet information-content):

- Publishers will normally be motivated by a desire for link differentiation (e.g. QoS, access regulatory compliance etc.) and may well tend to mistrust ISP behaviour, perhaps based on negative experiences historically when the peering market has been heavily biased towards providers
- An ISP will be similarly motivated by link differentiation (as the agent legally responsible for transit/peering regulatory compliance) and may well bias any decisions strongly towards its own interests (e.g. protecting network infrastructure vs. publishers, topology hiding vs. competitors and favouring cached content nearer to itself for resource optimisation)
- A RP will desire link differentiation with respect to information-related regulatory compliance (as the agent legally responsible), seeking "best" long-term compromises

to any "tussle" between RP, publisher and ISP views, probably favouring cached content nearer itself rather than individual ISPs

The functional behaviour (S-curve) of RP, ISP and publisher importance-related stocks will probably be very similar (in terms of their evolution over time), so Figures 12 and 13 below concentrate on comparing local ISP and RP reference modes, as a typical example.

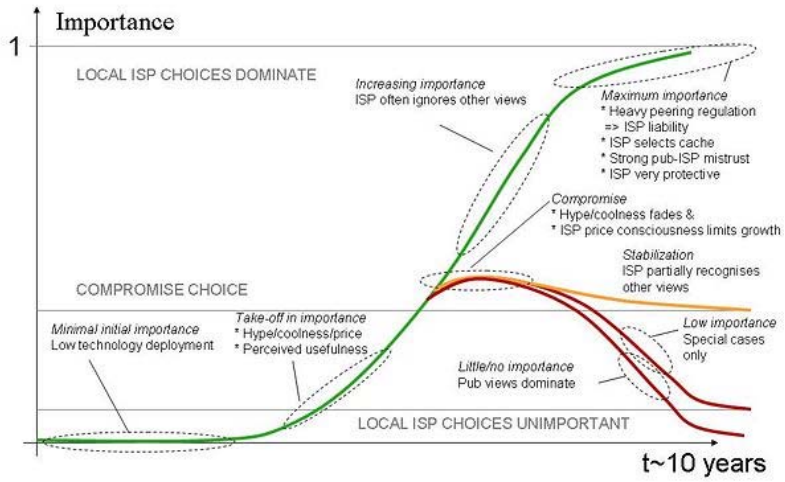

Figure 12: Importance of local ISP choice: reference mode.

Importance of ISP choice is heavily affected by trends in regulation and the degree to which initial enthusiasm for the service (in terms of hype/coolness etc.) wanes in the face of price pressures. There is a strong ISP tendency to be more biased towards its own assets (information-hiding). Importance of RP choice is most sensitive to information-related regulation and will normally adopt a more conciliatory posture to reconcile interests of the various parties involved.

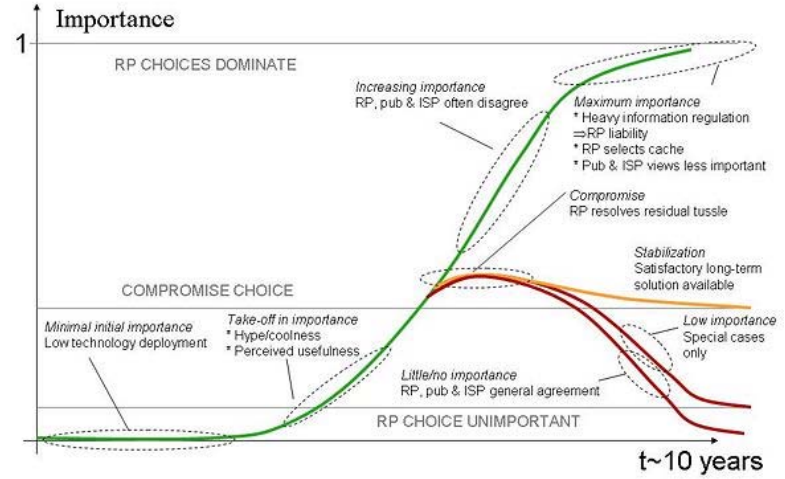

Figure 13: Importance of RP choice: reference mode.

2.4.4 Models

Just as for rendezvous, the corresponding triggers determining likely ITF behaviour must be identified. These are shown in Figure 14 below, arranged into various categories (technology, user behaviour, regulation etc.) reflecting their role in topology formation. On the technology side, advances in performance and availability of new features must be considered alongside legacy compatibility and reliability requirements. Regulatory pressures might range from traditional controls over transit/peering competition to new concerns regarding information visibility at the individual packet content level. Apart from obvious price issues, user perception of coolness/usefulness will be crucial for stimulating demand, while business/industry sensitivity to exposure of information (topology hiding) must again be recognised.

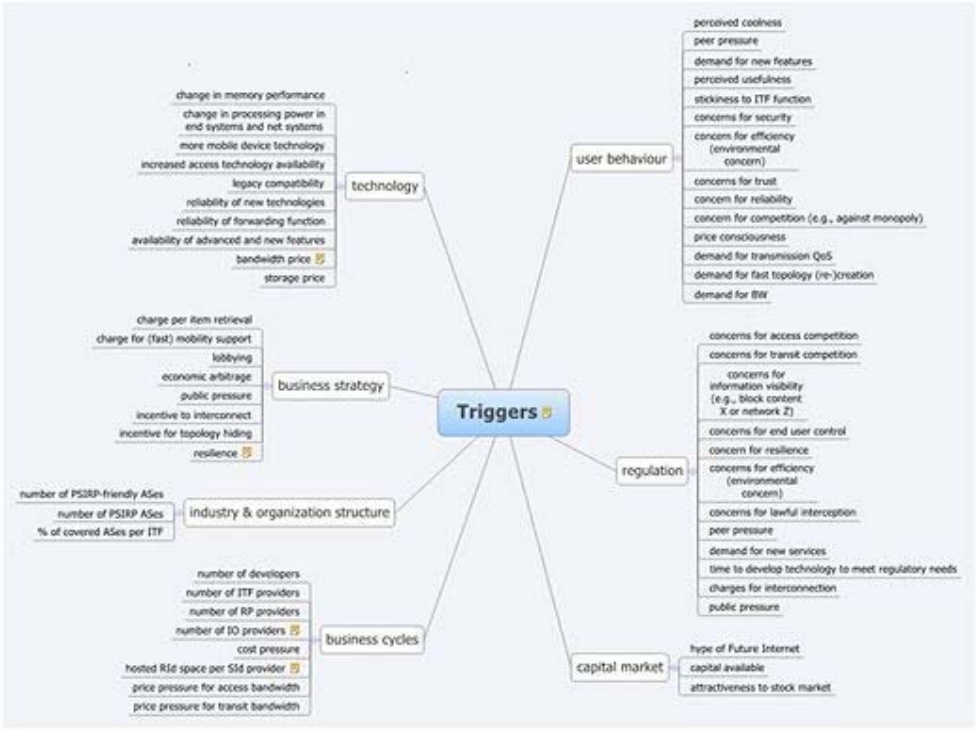

Figure 14: Triggers for Inter-Domain Topology Formation.

Within System Dynamics, control relationships amongst the various components may be captured in a “stocks and flows” diagram, as shown in Figure 15 below, where the ITF market is driven by the various triggers:

1. RP choice importance
2. regulation (information visibility)
3. ISP choice importance
4. regulation (transit competition)
5. publisher/subscriber choice importance
6. regulation (access competition)
7. incentive for topology hiding
8. user concerns (e.g. anti-monopoly)

9. industry concerns (time to develop technology to meet regulatory needs)
10. demand for BW etc./concerns for trust
11. charge per item retrieval
12. capital available
13. hype
14. perceived usefulness

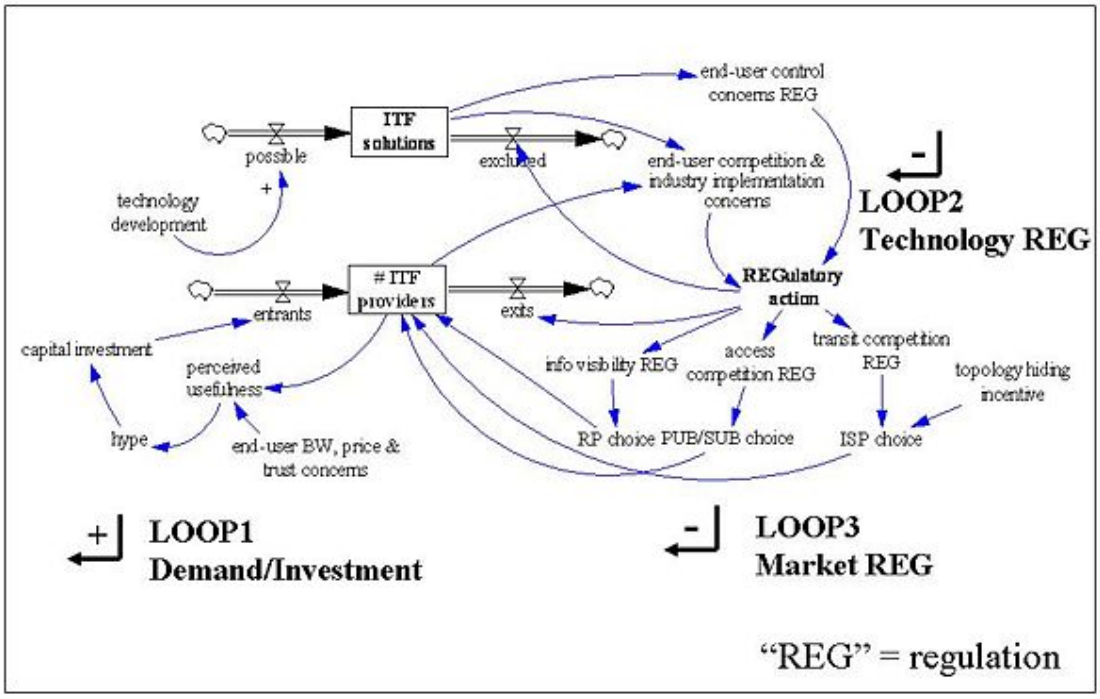

Figure 15: Stocks and flows for Inter-Domain Topology Formation.

Arrows represent influences of triggers on stocks/flows and on each other. Some triggers are essentially treated as inputs (e.g. items 12-14 related to business cycles, largely reflecting external economic conditions). However, there are also three feedback loops evident in the diagram (LOOP1, LOOP2, LOOP3):

1. Supply vs. demand between ITF providers and users, generating positive feedback
2. Technical solutions supporting ITF providers but limited by regulatory controls, generating negative feedback
3. Regulatory concerns (items 2, 4, 6), tending to restrain market growth and also influencing importance of RP, ISP and publisher/subscriber topology choices (items 1, 3, 5), generating negative feedback

Focusing particularly on security/privacy aspects (LOOP3) in Figure 16, the crucial issue will be compatibility of PSIRP design choices with the relative importance of RP, ISP and publisher/subscriber topology choices.

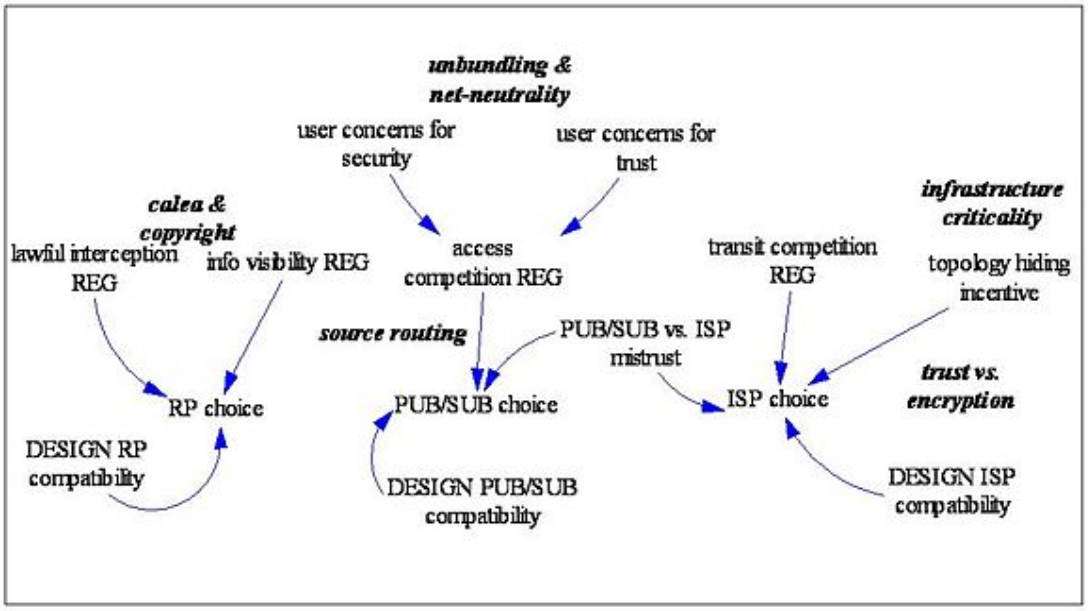

Figure 16: Design choice compatibility for Inter-Domain Topology Formation.

The stock/flow description thus exposes the dynamic structure of the ITF system. It forms a natural prelude to System Dynamics simulation, where the goal is to explore in more detail the consequences of ITF design choices for technology uptake in the wider socio-economic environment.

2.4.5 Scenarios

In this section, we consolidate the previous discussion and develop pointers to plausible Internet evolution scenarios, with a view to studying the consequences for ITF architectural design choices.

Scenario 1: Finance

In view of recent global economic upheavals, fluctuations in the wider economy (e.g. telecom bubbles vs. global recession) could obviously strongly affect PSIRP viability/success; this is particularly true for a new technology seeking to establish a foothold against incumbent competitors. We envisage a scenario where capital investment available changes markedly over our 10-year timescale, expecting a differential impact on PSIRP take-up relative to more conventional network offerings. This also serves to calibrate the model in a relatively simple context.

Scenario 2: Technology

Both memory and processing limits are primary technological causes for concern [Mey2007]; historically, Moore's Law tended to ensure technology scaled at rates surpassing the growth rate of information but the Law (arguably) does not apply to building high-end routers; growth in resources available to any one router could eventually slow down and may even stop, while network demand continues to grow. It seems intuitive that, as a new entrant, PSIRP will be particularly vulnerable to any sustained technological slow-down. In this relatively simple scenario, we propose to investigate PSIRP sensitivity to such fluctuations.

Scenario 3: Routing

There is a general regulatory trend away from legalistic requirements [Hui2002] towards more co-operative regimes with shared/devolved responsibility; previous modelling work has tended to reflect this, focusing on how telecommunications regulation must become more flexible in

the face of potentially disruptive technology change; failure to do so will inevitably compromise the delicate balance between regulatory control and innovation.

From a provider viewpoint, we also note the trend towards partial transit, paid peering and multi-homing which could significantly affect ITF design choices; similarly, net neutrality remains a hot topic with potentially significant implications for network access regulation.

With regard to interdomain routing and connectivity, PSIRP essentially offers high-quality (VPN-like) routes at (premium) price. This scenario will investigate the trade-off between likely performance (e.g. delay) improvement and price, to better understand the market for such PSIRP services and implications for ITF design decisions.

Scenario 4: Privacy

The dramatic increase in computing power, bandwidth and storage capacity has radically increased the ability of organisations to collect, store and process personal data. This is a potential cause for concern [Bro2009]. On the one hand, new technologies like ubiquitous computing, surveillance technologies, biometrics, behavioural advertising, or social networking provide a hitherto unknown capability for eroding privacy. On the other hand, general social and political fears of terrorism or organised crime may drive both public and private authorities to make use of these possibilities. Overall, these developments are generally thought to pose a serious challenge to existing privacy laws and principles. Cybercrime remains a major issue for policymakers and law enforcement agencies. Besides problems with fraud, key concerns include malware, spam and cyberwar attacks.

As an obvious privacy-related example, we consider eHealth and telemedicine applications. While the medical and economic benefits of integrated health information systems may be substantial, the usual public policy concerns associated with large-scale information systems apply. Given the extraordinary sensitivity of personal health data, special attention must be given to issues of privacy and IT security. A key challenge will be to make the best possible use of eHealth technologies for the benefit of the patient while complying with local privacy and security regulations.

This scenario will explore the tradeoffs between the importance of the various user choices and consequences for PSIRP ITF design decisions, such as the balance between source routing and topology hiding (emphasising importance of PUB/SUB and ISP choice, respectively).

2.5 Status and Future Work

Currently, the evaluation work has resulted in appropriate models being developed for the main design characteristics, the development of scenarios and the determination of appropriate parameterization of crucial auxiliary variables within the models. Results regarding design strategies and markets being created have not yet been produced at the time of writing this deliverable. However, results are being prepared for publication towards the end of the project for both the ITF and rendezvous function.

These results, however, will only be the beginning of our understanding how different inter-domain designs will be affected by certain socio-economic conditions. Our future work will concentrate on continuing to refine the causalities discovered so far. For this, we will continue to work with various stakeholders in order to better understand the various impacts and their proper parameterization. We will also focus on investigating the applicability of historical evidence for such relatively novel area like information-centric networking.

Ultimately, the desire of such socio-economic evaluation, as outlined in our methodology of Figure 1, is to better understand and evaluate various design strategies on a more rigid and formal level. For this, however, more work is needed that goes beyond the current project efforts.

3 Security Evaluation

One of the important design goals of PSIRP publish/subscribe paradigm is to follow the Trust to Trust (T2T) principle. According to this, functions should be implemented at points of the network that can be considered trustworthy from the user's (of the function) perspective. An elementary differentiation of the PSIRP information-centric internetworking from other approaches and the current Internet architecture is that security and privacy countermeasures can be built-in within networking, forwarding, topology management and other fundamental procedures. Thus, evaluating the trustworthiness of functions and their placement within the architecture is easier, and fully enables choice based on the evaluation of trustworthiness.

PSIRP provides the following security and privacy services over publications, subscriptions and scopes.

3.1 Confidentiality of publications

By introducing information networks (or scopes) access to publications is provided via knowledge of Scope ID (SId) and Information element ID (references as Rendezvous identifier, RId). The Sid, RId should be known in-advance by trusted publishers (to grand publication permission in SId) and subscribers (to request information from this scope). The rendezvous system is responsible to perform access control, along with checks that the publisher is authorized to send to the RId and SId in question and subscriber is permitted to receive this publication. So, publishers and subscribers should receive scope and rendezvous identifiers form the scope owner to publish or request information elements in the scope. Such an access schemes enables private or public scopes to be created. The usage of Algorithmic Identifiers (or AIds) for producing SIds and RIds introduces the concepts of information networks and information collections within the networks, respectively. Using private keys (seeds) during the construction of such information-element graphs, an entity (publisher or subscriber) is prohibited from revealing the graph structures and the information semantics (its actual Sid/RId) without prior knowledge of the seed.

3.2 Integrity and authorization of packets

Packet authorization is implemented via Packet Level Authentication (PLA) [Lag08]. PLA protects the whole packet payload with a cryptographic signature. In PLA is not necessary to verify the packet's cryptographic signature at every step. A data packet structure contains two sets of signatures and public keys. The signature of the scope owner authorizes the other public key associated with the RId to publish content in the scope. The second signature proves that the content was sent by the authentic RId owner. By default, PLA uses implicit certificate mechanisms where the publisher's public key for verification is derived from the TTP certificate that is attached to every packet. To reduce bandwidth overhead produced by signatures and public key information, PLA uses elliptic curve cryptography (ECC). PLA is a re-active mechanism. It applies and drops bogus packets that have been already injected on the forwarding topology and probably traverse some links towards the node that performs verification functions. One the other hand, when zFilters are used for the dynamic topology formation procedure they prevent bogus packets injection within every node, without introducing overhead information [Jok09].

3.3 Availability

From the security point of view, availability is equivalent to DDoS prevention or mitigation. In the rendezvous, publishers and subscribers plane, preventing DDoS is associated with minimizing the probability of unauthorized entities to publish information element or subscribe to content. The usage of AIds, SIds, Rids prevents attackers from introducing complex search queries (subscriber view) or bogus content (publisher view) into private (or in other words non-

public) information networks. This is because subscription or publication requires pre-knowledge of these IDs, or their relation. Additionally on the forwarding plane, PLA introduce verification of packets, whilst zFitler prevents off-path attackers from sending data towards a delivery tree even if they know both the RId. Thus, a DDoS resistant service is support on the forwarding plane. For the Inter-connected Rendezvous Systems, the usage of Crescendo (a Canon-version of Chord) is envisioned. In [Del2.4] several DDoS, poisoned data and spoofing attack scenarios have been analysed, using botnet as attack initiators. The countermeasures that have been proposed are either built-in in the PSIRP model, or inherit DHT's security methods.

PSIRP provides strong in-built security functionality. The overall concept of building trustworthy functions is achieved, since PLA, z-filtering, Algorithmic identification as access control via trusted rendezvous points and interconnection using hierarchical DHT provides security features by design.

One drawback of the overall architecture deals with Public Key Certificates (PKC). PKCs are assumed for publishers, subscribers or even domains and scopes. Nevertheless, building core networking functions based on Scope and Publishers PKCs need circumspection. For instance what will happen to a PLA-protected packet in transit if a scope or publisher's certificate is suddenly revoked? This packet will never be authenticated. Normally it will be discarded. The next packets of the same flow will also be discarded, since the scope or the publisher have updated the certificate status to invalid, but several packets in transit were signed with the invalid private keys. Thus, several requested (by the subscriber) packet chunks might never reach the destination. These set of dropped and undelivered packets is of more importance when scope certificate is revoked, since the undelivered content will be larger in amount and value.

Moreover such an environment will create disputes between TTPs (or scope / rendezvous network operators). Private and public bodies might consider of providing new pub/sub services. Cross-certification should be easy, since trust should be smoothly and mutually established between peer providers (eg., in the Inter-domain forwarding). In its simplest form, cross-certification requires the signing of the public key of one peer provider using the private key of the other peer. Normally, this is a trivial process. The difficulty comes with the different certificate policies and practices the might apply per provider. These practices are not always aligned and equivalent. For instance, it is not always acceptable to use the 512-bits RSA key of one provider to sign the 1024-bits public key of the peer provider. Concluding, constructing end-to-end forwarding trees it is important to take into account the trust relationships and links between the nodes (or domains), beyond any technical, economical or other optimization criterion.

Additionally, little attention has been given to the mechanism that will enable subscribers to search of publications' RIds. This mechanism however should be secured and distributed so that will to protect the network from DoS. Complex and synchronised queries that exhaust rendezvous capacity, the absence of limitations on accepted queries per subscriber-publisher pair (within a scope), and highly consuming subscribers that require large amount of data and then de-attach form the network once they receive few bytes of the content. Such scenarios might threat the rendezvous system normal operation, or overload the forwarding plane. Finally, a mechanism might be needed to prevent bogus publications even in public scopes and rendezvous points. Such a mechanism can be based on information ranking [Fot10], but it is still an open question how to rank publishers that introduce spam or malicious content and prevent them from polluting the pub/sub public network.

4 Performance Evaluation of Architectural Solutions

In this section we provide results from the quantitative evaluation activities carried out in the project over the past year, that is, since deliverable D4.2 was issued. We first provide an update on the performance evaluation of the latest developments in the overlay PSIRP variant in Section 4.1. We then discuss the performance of the current intra-domain topology management prototype in Section 4.2, followed by an account on recent developments in the forwarding domain in Section 4.3. Finally, we give an update on the performance evaluation of the rendezvous architecture in Section 4.4, and an outline of the ongoing testbed development work in Section 4.5.

4.1 Overlay Approaches and Caching

Extending the functionality of the overlay PSIRP variant, we have designed MultiCache, an overlay architecture that aims at taking advantage of information-awareness to improve the utilization of network resources via resource sharing. To this end, network operators deploy and control proxy overlay routers that enable the joint provision of multicast and caching, targeting both synchronous and asynchronous requests. End hosts interact with the infrastructure by simply providing flat, location independent identifiers for the desired content, without engaging in the process of locating an end host providing the data. Inside the network, the Scribe overlay multicast scheme [Cas2002] is employed to transport the content from its origin in a publish/subscribe fashion, thus serving synchronous requests (e.g., flash crowds) and feeding in-network shared caches. By taking advantage of the locality awareness of the established Pastry routing substrate [Row2001], anycast queries based on the already established overlay multicast forwarding state are later used to locate nearby caches that can serve asynchronous requests by unicasting the cached content.

4.1.1 Proposed Architecture

4.1.1.1 Deployment

MultiCache functionality is deployed in an overlay fashion inside access networks. This entails the deployment of additional infrastructure in the form of Overlay Access Routers (OARs), possibly collocated with regular access routers. OARs provide the following functionality:

- They participate in the overlay routing and forwarding substrate, enabling the use of overlay multicast. This entails the maintenance of Pastry routing information [Row2001], as well as the functionalities of Scribe and MultiCache, as described below.
- They act as proxies of end-hosts in the overlay, i.e., an end-host establishes a control connection to an available OAR designated during network attachment. The selected OAR (proxy OAR) may be collocated with the access router of the end-host or it may be located several hops away, subject to the density of OAR deployment. The role of the proxy OAR is to act as the interface of the end-host to the overlay, possibly aggregating data requests from multiple attached end-hosts.
- They cache content destined to their attached end hosts. As a result, the same content is cached at multiple locations in the network, i.e., at all leafs of established overlay multicast trees.
- They provide cached content to other OARs via unicast.

The deployment of overlay functionality inside access networks serves several important goals. First, the overlay character of the architecture facilitates the deployment process, as it does not require the replacement of existing infrastructure, while it allows the unobstructed operation of established services and applications. By deploying MultiCache inside access networks, content is cached close to the clients [Tew1999], facilitating the discovery of caches

in the clients' networking vicinity and therefore enabling the localization of traffic. Finally, as discussed in [Raj2008], placing caches close to the end points of the network avoids incentive incompatibilities regarding inter-domain relationships. A simple deployment example is given in Figure 17 below, with OARs being collocated with the corresponding access routers.

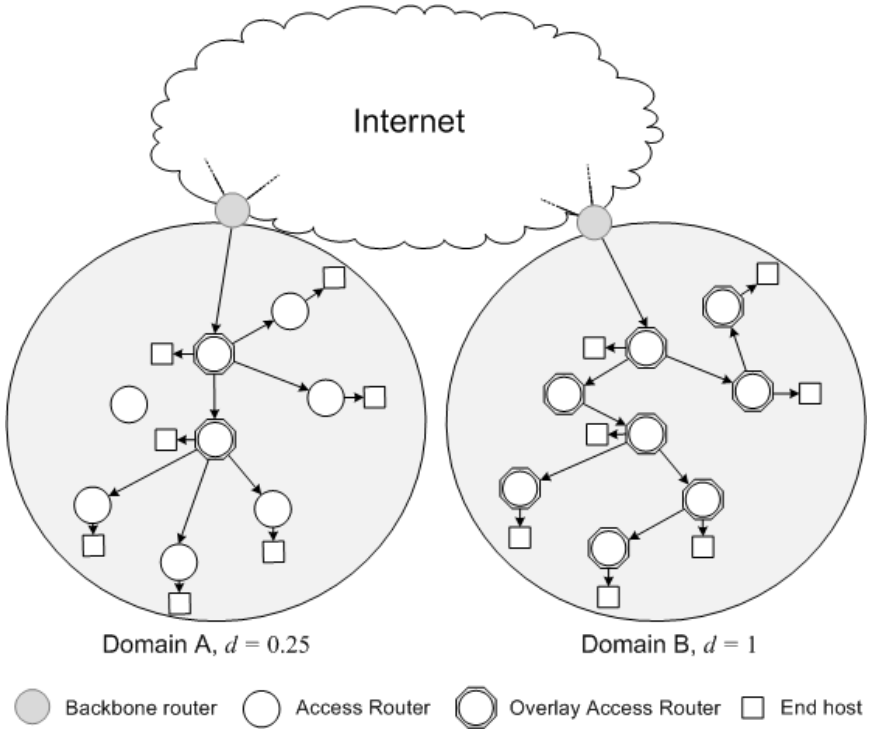

Figure 17: MultiCache deployment example.

4.1.1.2 Multicast

Multicast forwarding takes place among OARs driven by end-host requests, i.e. after end-hosts issue requests for desired data objects to their proxy OARs via the established control connections. These requests may be translated to corresponding Scribe JOIN messages, depending on the current state of the proxy OAR with respect to the indicated data item. The joining process deviates slightly from regular Scribe in that JOIN messages are extended to further carry the IP address, the listening port number, the credentials of the initial issuer of the JOIN message (i.e. the proxy OAR; note that in cases of multi-overlay hop paths, the proxy OAR is not the node that eventually delivers the JOIN message to an already joined node) and the 32-bit Autonomous System (AS) number [IAN2009] of the proxy OAR's AS. This extra information is used during cache searching and provisioning, as explained in the next subsection.

During the joining process, OARs establish TCP connections with their children for the reliable delivery of the requested data. When a JOIN message eventually reaches the Rendezvous (RV) point, the content provider will be solicited to deliver the data which will then start traversing the tree created via the already established TCP connections. It is assumed that the content provider has already created the respective group, and therefore has contacted the RV point. Due to the asynchronous character of request arrivals, this process may result in partial data availability at the leafs of the multicast tree at the end of the multicast session. However, the caching mechanism ensures that these partial feeds will be able to complete later.

A simple example of these operations is given in Figure 18. The first two subfigures depict progressive snapshots of a simple Scribe tree. The arrays below each OAR denote the availability of the content. OAR 1 first joins a Scribe multicast tree via OAR 2, followed by

OARs 3 and then 5 that join during the multicast session via OAR 4. At the end of the multicast session (end of Step 3), OARs 3 and 5 have received part only of the multicasted content, i.e., they are missing blocks 0 to 3 and 0 to 6 respectively.

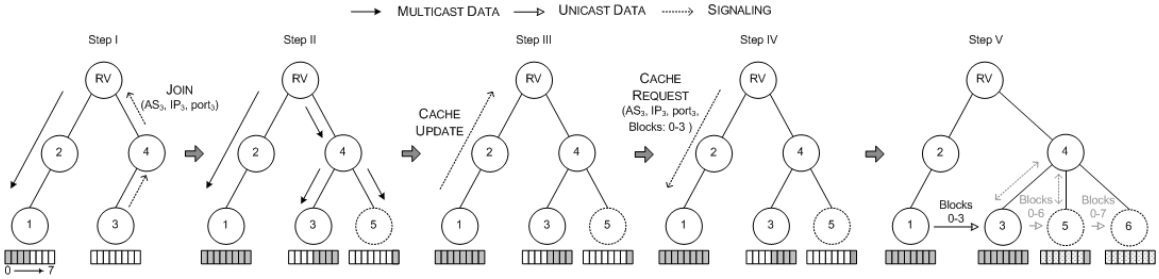

Figure 18: MultiCache example.

4.1.1.3 Caching

In MultiCache, caches are located at proxy OARs, i.e., at the leaves of multicast trees. The same content may be cached at multiple network locations close to the clients. This facilitates the discovery of caches in the networking vicinity of a requesting OAR and enables the localization of traffic. It is also noted that placing caches at the edge of the network avoids incentive incompatibilities regarding inter-AS relationships as discussed in [Raj2008].

Cache discovery

In order to locate an available cache, MultiCache uses the already established overlay multicast forwarding state. The OARs cache the forwarding state established during tree creation even after the end of a multicast transmission. As caches are created, CACHE UPDATE messages are issued by leaf OARs towards the RV of the multicast tree. The purpose of these messages is to notify ancestors about the availability of cached items downstream and allow their discovery upon cache requests. Note that caches may be fed by other caches, therefore OARs cannot rely on forwarded traffic in order to deduce cache availability below them. OARs further propagate received CACHE UPDATE messages towards the root if they have not already done so for another downstream cache, thus avoiding feedback implosion.

When an end-host requests data, the Scribe JOIN message sent by its proxy OAR is suppressed at the first OAR that has already joined the respective tree, henceforth termed as a meta-cache OAR. What happens next depends on the state of the meta-cache OAR with respect to the indicated object. If the requested object has been cached by the meta-cache OAR itself, the cached data will be directly delivered to the requesting node (direct cache hit). If the data are not cached but the meta-cache OAR has previously completed forwarding the data object to its descendants, it will anycast a CACHE REQUEST message to the sub-tree below it in a depth first search (DFS) fashion, carrying all extra information inserted in the JOIN message. In particular, at each level of the traversed sub-tree this message is forwarded to one of the children that have previously issued a CACHE UPDATE message. At each step, preference is given to children belonging to the same AS with the requesting proxy OAR. Among equivalent candidates, a randomized selection ensures the uniform distribution of load to the available caches. Eventually, a CACHE REQUEST reaches an OAR that has cached the data, and a TCP connection is established between the caching OAR and the proxy OAR for the delivery of the cached data.

Finally, if the meta-cache OAR is currently forwarding the requested data, it forwards the arriving multicast data to the joining node, also keeping track of the part of the data object that was not delivered due to the late arrival of the JOIN message. Upon the arrival of the first CACHE UPDATE notification a data receiver, a CACHE REQUEST is issued for the missing data for each partially served child of a meta-cache OAR, thereby reverting to the previous case.

Cache eviction

In MultiCache, cache availability is correlated with the overlay multicast forwarding state. This allows requests for content to lead to either the multicast-based delivery of data or to a cache hit, while preserving the locality properties of the established tree structure and avoiding extra control overhead for the discovery of cached objects. In practice, this means that cached items are not evicted from a cache unless the corresponding multicast forwarding state is torn down.

To synchronize caching and forwarding state, we have slightly altered Scribe's leave procedure. When a caching OAR issues a Scribe LEAVE message for the tree serving the cached object marked for eviction, this message propagates up the tree until either the first node with additional children or the RV point is encountered. Then a LEAVE RESPONSE message is issued towards the leaving child, thus tearing down the forwarding state and removing the cached data. In contrast, in regular Scribe the forwarding state is removed immediately upon the reception of a LEAVE message. The new procedure ensures that eventually, all requests for data reach either a caching OAR (in the form of CACHE REQUEST messages) or the root if no other cache location is available (in the form of Scribe JOIN messages). In the latter case, the content provider is solicited to provide the desired object again via the established multicast delivery path. The above procedure is illustrated in the simple example of Figure 19 below. Node 2 will not remove node 1 from its forwarding table until a LEAVE RESPONSE message is received from node 3. Meanwhile, node 4's issued CACHE REQUEST will be served normally.

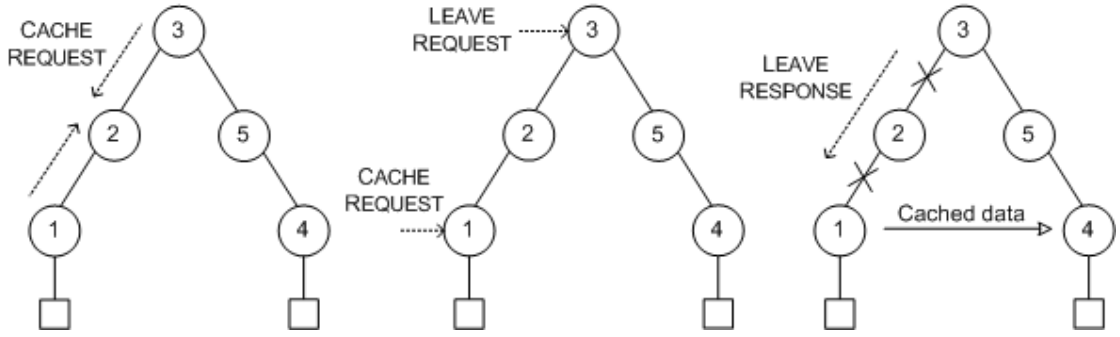

Figure 19: MultiCache Leave procedure.

Cache replacement

As mentioned above, OARs always cache the content delivered due to an end host request. When a request arrives at an OAR with an exhausted cache space, a cache replacement policy is employed to select an item for eviction. Common replacement policies (e.g., Least Recently Used (LRU)) aim at adjusting cache contents to request patterns so that less popular items leave space for more popular ones, thus increasing the cache hit ratio. In MultiCache, cache replacement aims at taking advantage of the multiplicity of cache locations inside an administrative domain. To this end, caching OARs keep track of the popularity of each cached item with respect to the frequency and/or recency of hits from other OARs inside the domain. Since all content delivered to an OAR is locally cached, such hits imply that additional copies of the same object are probably cached nearby. Hence, in MultiCache we examine the suitability of the Most Recently Used (MRU) and Most Frequently Used (MFU) policies; these policies favor the selection of items that are most likely to be available at other cache locations. The rationale behind the MRU policy is that the most recently served object is more likely to be still available at the served OAR, i.e., not to have been evicted yet, therefore the existence of an alternative cache location allows replacing that item. In the case of MFU, the probability of eviction increases with the anticipated number of alternative cache locations.

It must be stressed that common replacement policies such as LRU and LFU refer to the recency/frequency of requests referring to the entire data item (file). In MultiCache, caching,

and therefore cache replacement, takes place at a fragment level, allowing the partial caching of files. The examined MFU and MRU policies aim at reflecting the existence of specific cache locations and therefore are enforced on fragments, regardless of the data item that each fragment belongs to. In this manner there is no need for control signaling and state overhead for the association of single pieces with the corresponding data items.

4.1.1.4 Locality Properties

MultiCache favors localized cache hits by building upon Pastry's locality properties and the multiplicity of cache locations. According to Pastry's route convergence property, since caching OARs are essentially leaves of the data item's Scribe multicast tree, a Scribe JOIN message from a proxy OAR is expected to reach a meta-cache OAR at a distance approximately equal to the distance between the proxy OAR and a caching OAR in the proximity space. At the same time, following Pastry's prefix based routing, Scribe JOIN messages are initially expected to travel short distances at each overlay routing step. Hence, as demonstrated in [Cas2003b], in cases of multiple cache locations, Scribe JOIN messages from proxy OARs are expected to first reach nearby meta-cache OARs, thus leading to closely located caches. In effect, cache search messages and cached data are expected to traverse short network distances, with respect to Pastry's proximity metric, leading to the localization of traffic. This is further enhanced by the simple AS number-based cache selection mechanism.

4.1.1.5 Content Fragmentation

MultiCache allows the fragmentation of large files into pieces, in a BitTorrent fashion, leading to the creation of a forest of Scribe trees, resembling SplitStream [Cas2003a], but without the explicit goal of creating disjoint trees. This fragmentation serves several important goals. First, it facilitates the establishment of parallel data flows towards a recipient node, possibly exploiting the available downlink bandwidth and avoiding the sequential delivery of large files. Furthermore, it allows the partial caching of large data volumes, i.e., certain pieces can be cached independently of others, enabling the fine grained management of caching space [Hef2008]. The establishment of partial caches in different network locations favors the establishment of disjoint delivery paths, facilitating the distribution of forwarding load and the localization of traffic. However, these benefits come at the cost of forwarding state which increases with the size of the resulting forest. Pieces are further partitioned into blocks, again as in BitTorrent fashion. This second level of fragmentation facilitates the provision of data from multiple sources. For example, as explained in the previous section, an OAR may join a multicast tree while data are in transit, in which case the first part of the piece shall be later provided by a cache.

4.1.2 Performance Evaluation

In order to provide a realistic application model for the evaluation of the proposed architecture, we have designed a MultiCache-based content distribution application that can be directly compared to regular BitTorrent. In this application a content provider employs content fragmentation to create multiple trees for the delivery of a single file. All identifiers are retrieved by end-hosts via out-of-band means, e.g., a MultiCache-torrent file. In order to reduce forwarding dependencies [Dio2000] piece identifiers are assumed to have been appropriately selected so that the RV functionality is provided by OARs residing at the content provider's domain. Upon arrival at the network, end-hosts connect to their proxy OAR and submit requests for pieces of the file. The number of pending requests is capped, in an analogy to regular BitTorrent. Each node submits its requests independently of other end-hosts, since we cannot assume any form of collaboration between end-hosts. Once a piece has been entirely downloaded, the next piece is requested from the proxy OAR until the file download has completed.

4.1.2.1 Simulation Environment

The evaluation of MultiCache is based on a detailed full stack simulation environment based on the OMNeT++ Simulator [Var2008] and the OverSim Framework [Bau2007]. The MultiCache content distribution application is compared against our own BitTorrent implementation for OMNeT++ [Kat2009]. In our simulations we used Internet-like topologies generated by the Georgia Tech Internet Topology Model (GT-ITM). For our measurements, we created topologies comprised of 1225 routers hierarchically organized in 25 stub and 5 transit domains. In all topologies, the default link establishment probabilities were used.

In order to study the properties of our caching scheme, we generated synthetic traces of request arrivals for several files across the network. To generate this workload we used features of the ProWGen trace generation tool [Bus2002]. To better reflect the characteristics of a P2P application we replaced the Zipf distribution of file popularities with the Mandelbrot-Zipf distribution proposed in [Hef2008]. A certain number of requests were generated for each file in the workload, according to the file's popularity. All file requests followed the exponentially decreasing arrival rate process described in [Guo2007], parameterized according to the popularity of the corresponding file. We interleaved these single file traces by placing the first request of each file at a constant time interval after the first request for the previous file. This reflects the constant torrent arrival rate observed with BitTorrent in [Guo2007]. File sizes were sampled from the traces in [Bel2004]. The content providers, one per file, are uniformly distributed across the entire network. Each of the generated requests is then assigned to one of the 100 end hosts we attach at a randomly chosen access router of the topology.

Finally, we used the default parameters for both the Peer-Wire and Tracker protocols of BitTorrent. In the case of MultiCache, we use 16KB blocks and set the default size of a piece to 16MB.

4.1.2.2 Evaluation Framework

In order to study the properties of MultiCache's caching scheme, our first metric is the achieved cache hit ratio (CHR). To further study whether traffic is localized within AS boundaries, we also measure the intra-domain cache hit ratio (CHR-Intra) which reflects only cache hits on OARs residing in the same AS as the end-host receiving the cached data. Finally, we measure the distance to block source, i.e., the average number of physical hops traversed by blocks arriving at end hosts, in order to assess the overall locality properties of data transfers.

These metrics are studied for the described cache replacement policies, as well as for various cache sizes. Following the methodology in [Fan2000], we consider relative cache sizes (S_r), i.e., the cache size is expressed as a fraction of the infinite cache size, which is the minimum cache size required to avoid replacements. We also examine the effect on these metrics of MultiCache deployment density d, defined as the fraction of the access routers that are enhanced with MultiCache functionality. In the example deployment of Figure 17, the density for domain A is d_A = 2/8 =0.25 while for domain B it is d_B = 1. In this paper, we assume uniform density values across all ASs.

Finally, we investigate the ability of MultiCache to localize traffic inside domain boundaries depending on the popularity of data objects inside the domain. For this reason we define the localizability parameter l taking values in [0,1] as an indicator of the concentration of end-hosts (and the corresponding requests) inside domain boundaries. If we denote as D the total number of administrative domains in the topology, then end hosts are uniformly distributed across max[(1 - l)D, 1] administrative domains. At l = 1, all end-hosts reside in the same AS, while at l = 0, they are uniformly distributed across all ASs.

4.1.2.3 Results

Cache size and replacement policies

Figure 20 and Figure 21 show the CHR and CHR-Intra achieved with the LRU, MFU and MRU cache replacement policies, for relative cache sizes (S_r from 0.5% to 20%). Interestingly, all these policies exhibit approximately the same behavior for all cache sizes considered. Note that the CHR reaches values up to 98.5% for higher S_r values, thus reducing the amount of data delivered via overlay multicast and taking advantage of available caches throughout the entire network. As a result, MultiCache reduces the impact of overlay multicast stretch and takes advantage of Pastry's proximity properties in order to locate nearby copies of the desired data. This is clearly demonstrated in Figure 22 which depicts the distance to block source. As the cache size increases, average distance decreases, denoting the delivery of content from nearby caches. Again, no cache replacement policy exhibits superior performance, leading us to the adoption of the MFU policy due to its implementation simplicity.

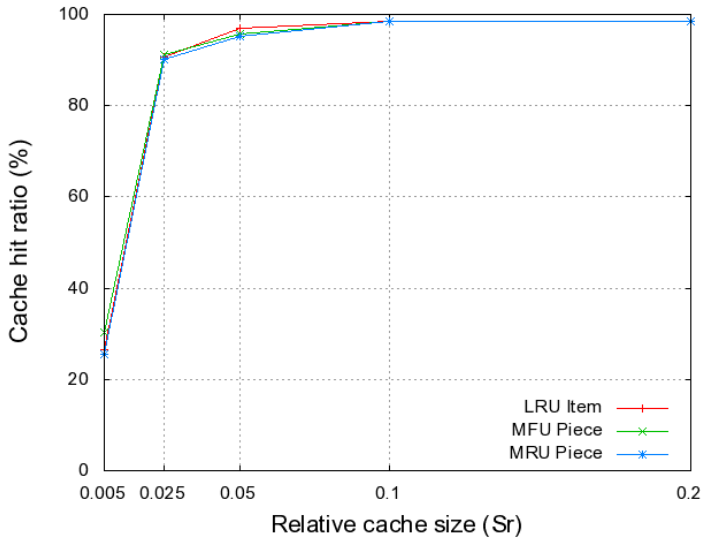

Figure 20: Effect of cache replacement policies: cache hit ratio.

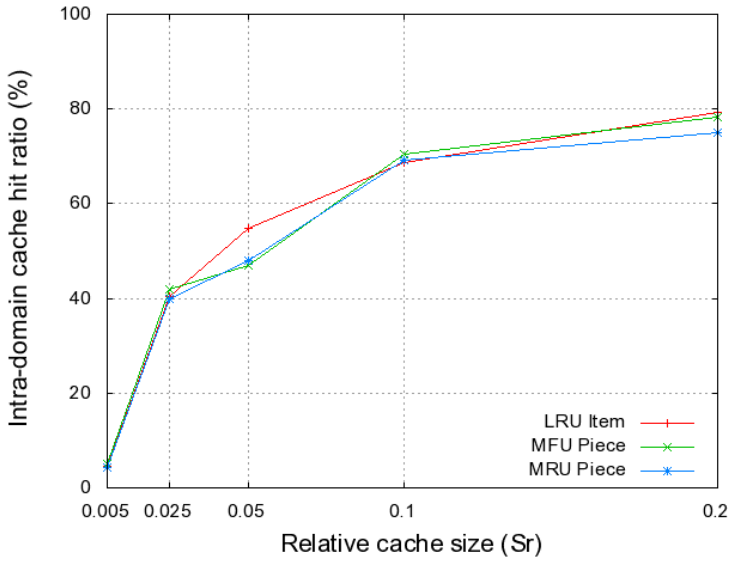

Figure 21: Effect of cache replacement policies: intradomain cache hit ratio.

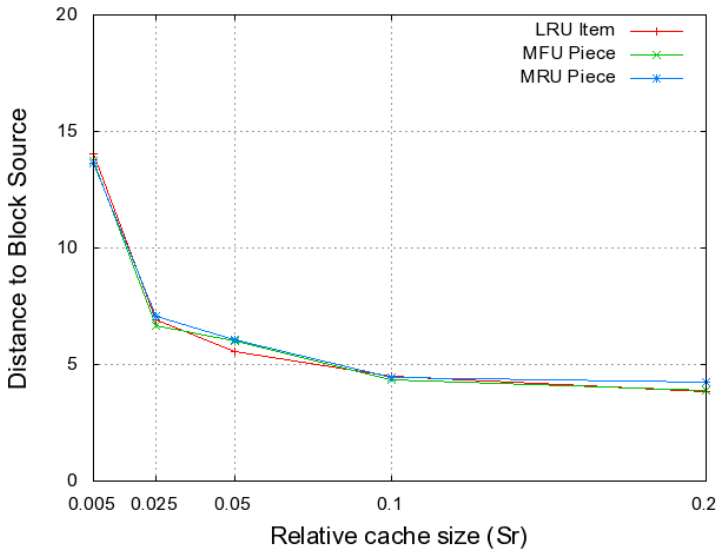

Figure 22: Effect of cache replacement policies: localization of traffic.

Deployment Density

Since MultiCache necessitates the deployment of additional infrastructure (OARs) by network operators, a crucial issue for the viability of the proposed architecture is the magnitude of the investment required. Figure 23Figure 25 show the cache hit ratio for various deployment densities and relative cache sizes. Figure 23 shows that even though CHR increases with deployment density, for relative cache sizes ranging from 2.5% to 20% of the infinite cache size the perceived CHR is always greater than 80% at a deployment density of 25%. Figure 24 shows that CHR-Intra ranges from 46% to 88% for higher relative cache sizes, meaning that this portion of traffic is held inside domain boundaries. As the density increases however, local cache hits decrease, due to the reduced degree of request aggregation at caching OARs and the corresponding reduction of direct cache hits at proxy OARs. This is also depicted by the modest reduction of the average network distance travelled by data blocks, shown in Figure 25. While denser deployments result in more cache locations, and therefore shorter distances between proxy and caching OARs, they also mean that similar requests and the resulting cached content are distributed across a correspondingly larger number of locations. Therefore, the modest investment required to achieve a deployment density of 25% to 50% is sufficient to reap all the benefits of MultiCache.

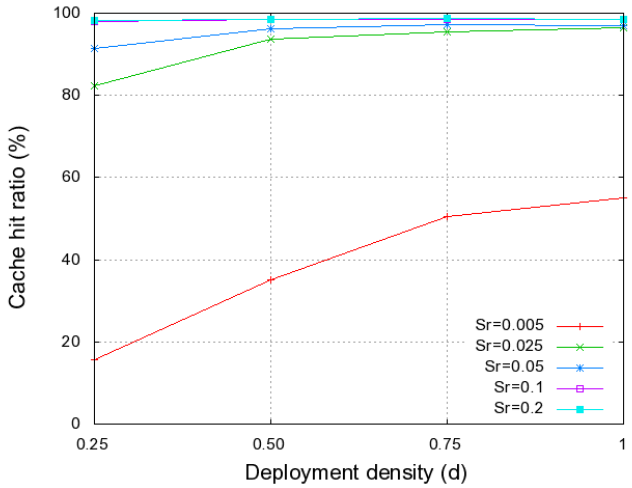

Figure 23: Effect of deployment density: cache hit ratio.

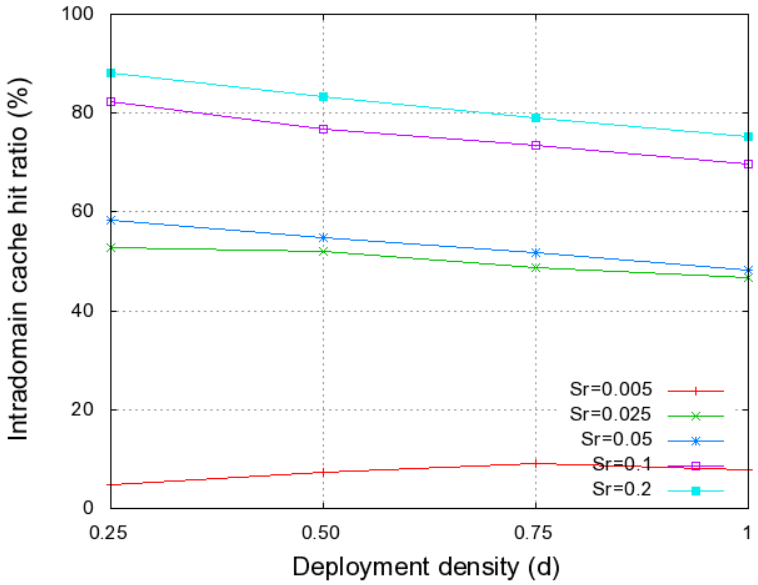

Figure 24: Effect of deployment density: intradomain cache hit ratio.

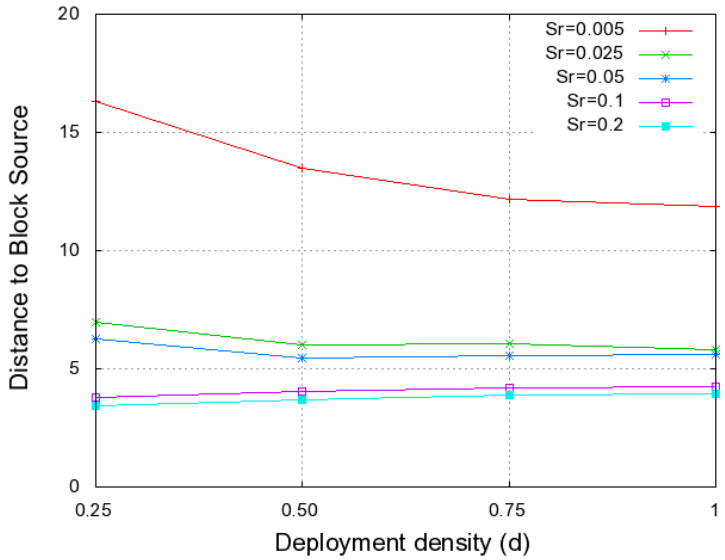

Figure 25: Effect of deployment density: localization of traffic.

Localizability

Figure 26 shows the effect of the localizability factor on the cache hit ratios for S_r = 2.5%. Since the delivery of content to an OAR results in the availability of that content at the corresponding cache, localizability essentially expresses the availability of content inside domain boundaries. However, as localizability increases, so does the competition for caching space: the more data being delivered to a domain, the more content has to be cached locally within a fixed cache size. Hence, the content arrives at a domain but gets evicted due to the increased load on the caches. At the same time, content concentrates at intra-domain cache locations, due to co-located requests, therefore the portion of intra-domain cache hits rises, up to the point where all cached content is provisioned by a local cache. This means that MultiCache does take advantage of localized request patterns, but only up to the point where cache size limitations do not allow further improvements.

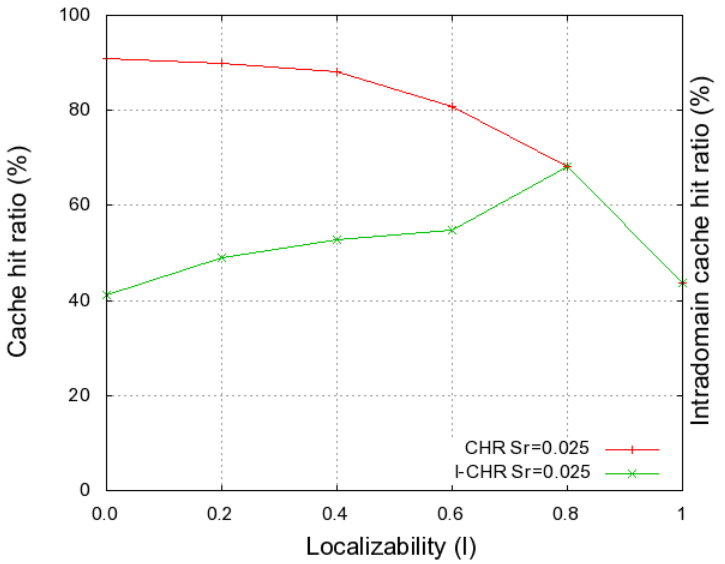

Figure 26: Effect of localizability on cache hit ratio (*d*=0.25, S_r = 2.5%, MFU).

4.1.3 Conclusions

Our findings show that MultiCache takes advantage of the multiplicity of cache locations, avoiding as far as possible the employment of overlay multicast for already transmitted content. Moreover, our results show that sparse MultiCache deployments can yield high intradomain cache hit ratios, thus localizing traffic inside domain boundaries.

4.2 Intra-Domain Topology Management

One of the important aspects in our evaluation work is testing of intra-domain topology management functionality. Its main role is in discovering of topology information within one administrative domain. This data is further used as an input for computing necessary forwarding information for involved nodes, and appropriate updating of such information as soon as the topology changes. It is intended to work cooperatively with inter-domain topology functionality in order to build overall topology and forwarding states. In other words it is responsible for configuring and maintaining intra-domain topology information and forwarding paths as a part of global topology creation. Our current Python based implementation of topology management is compliant with the recent PSIRP (alpha3) prototype and performs intra-domain topology formation taking into account relevant network conditions and specific application requirements besides information about present network entities.

Main functionality is carried out within two modules: topology manager - responsible for building topology paths, and connectivity helper - mainly serving for discovery of local connectivity. Additionally, each forwarding node can be equipped with a link-state helper module which maintains the table of “known” links along with link related available information, e.g., throughput, and delay. This information about network conditions provides valuable input for optimization of forwarding paths computation considering not only the shortest paths criteria, but the present network state. Moreover, application requirements for specific network conditions determine additional parameters affecting the topology generation algorithm. This input is provided by the application helper module. The basic operational mode of topology management is similar to common link state advertisement algorithms. Each node is publishing its own existence in the network, while being simultaneously subscribed to the same data coming from its neighbors. After collecting relevant information about its surroundings, each node publishes this data in the form of Link State Advertisement to be processed by the topology manager module.

In order to test the current topology management implementation we apply two modes of operation single-domain and networking mode, with respect to Blackhawk prototype functionality. In the first setup we utilize one-domain scenario without deploying separate Rendezvous point as dispatching entity between publishers and subscribers. Thus, all nodes in the network are communicating directly within predefined ScopeID/RendezvousID pairs. For each type of information to be transmitted over the network we define separate SID/RID pairs, e.g., Hello SID/RID pair for exchanging the data about node’s existence.

Our scenario consists of up to 10 connectivity helpers in one domain, having a single link-state helper module, application helper module and topology manager within the domain. We monitor different metrics of topology management functionality e.g., introduced delay of Dijkstra algorithm and igraph [igraph] setup. We perform multiple runs of the same scenarios in order to obtain clearer insight of system performance. The current implementation of forwarding states computation mechanism relies heavily on a weighted Dijkstra algorithm from igraph. Appropriate link weights are added based on input from the link-state helper function. Due to its important role in topology management functionality, we estimate the impact of Dijkstra algorithm to overall performance, measuring the delay it introduces. Initial tests are carried out over the networks of variable sizes, ranging from 3 to 10 connectivity helpers, running the single-domain scenario explained above. Results show that the average delay introduced by execution of the Dijkstra algorithm varies based on the network size, but does not exceed 30ms. Average processing time of the Dijkstra algorithm increases as the network

size grows, due to the larger number of input data which needs to be treated. In order to more accurately estimate the influence of applying the Dijkstra algorithm in our topology management implementation we measure the time needed for execution of sole Dijkstra algorithm over an arbitrary network size and different link existence probability between nodes. Obtained results, illustrated in Figure 27 and Figure 28, are generic and demonstrate average delay introduced by execution of Dijkstra algorithm in igraph, regardless of the implementation in which the Dijkstra algorithm is utilized.

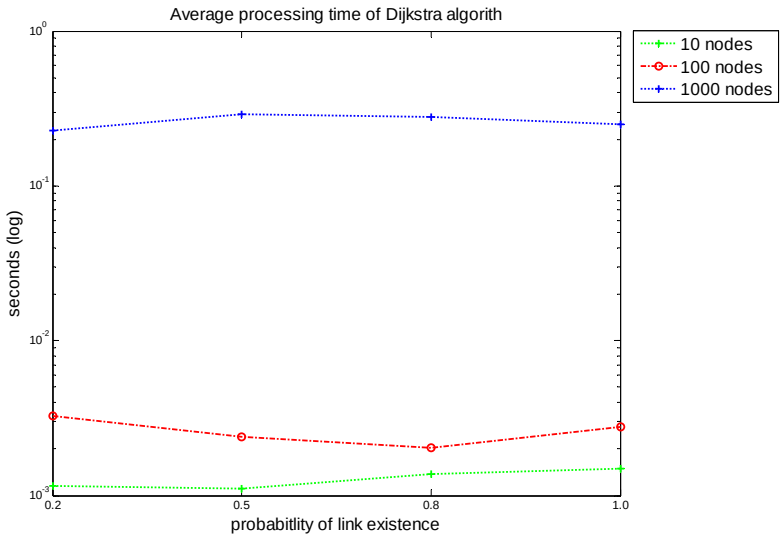

Figure 27: Average processing time of Dijkstra algorithm applied over the arbitrary graph of 10, 100 and 1000 nodes with different probability of link existence.

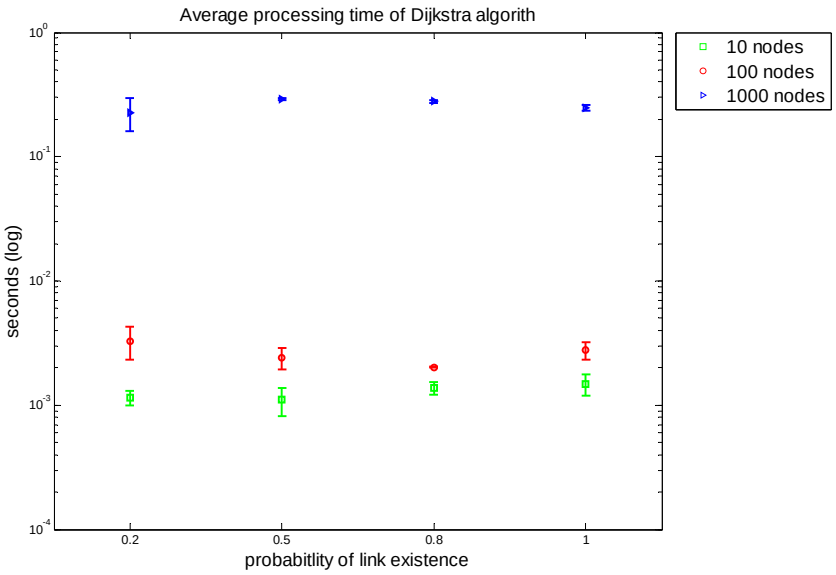

Figure 28: Average processing time and standard error of Dijkstra algorithm applied over the arbitrary graph of 10, 100 and 1000 nodes with different probability of link existence.

Prior to actually applying the Dijkstra path creation algorithm, appropriate igraph functionality is utilized. Based on received Link State Advertisement messages obtained from nodes in the network, topology manager generates the graph of nodes. The graph is dynamically modified

by adding and removing vertices and connecting links between them. After updates due to the reception of new Link State Advertisements and assignment of related link weights, the Dijkstra algorithm is applied. The setting up phase of the graph, its maintenance and updating of relevant link weights consumes considerable time. Figure 29 shows the time needed for igraph initialization and related tasks, together with Dijkstra algorithm execution.

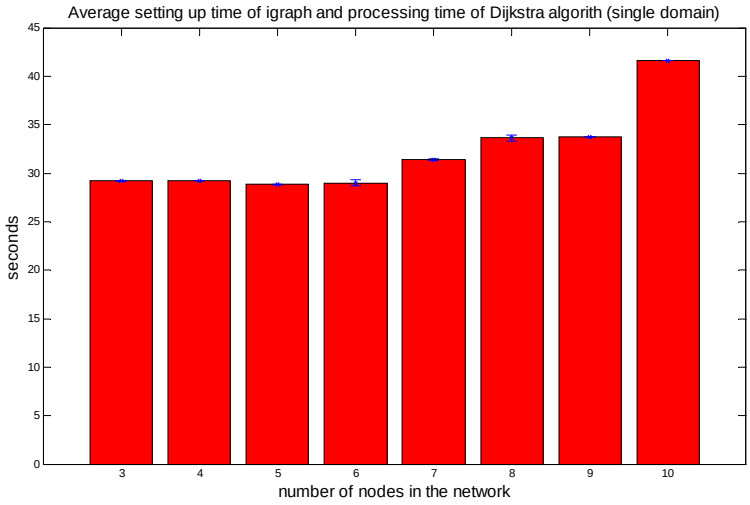

Figure 29: Average initialization time of igraph and relevant tasks together with Dijkstra algorithm execution.

Introduced delay is increasing with the number of nodes in the network similar to the observations related to sole Dijkstra algorithm. Another interesting metric relevant to our topology implementation model is the measure of publish/subscribe latency. We analyze the delay introduced by monitoring the time difference between actual generation of publications and their reception after corresponding subscriptions. Figure 30 illustrates obtained results for the single domain case.

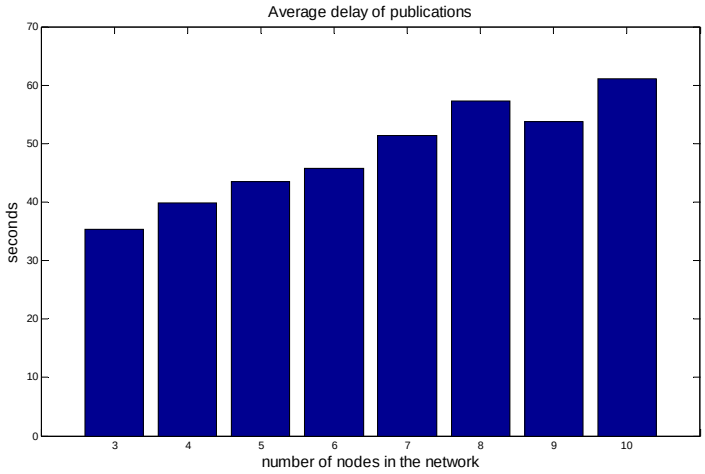

Figure 30: Average delay of publications in single domain.

Publication delay evaluation results confirm previous conclusions: the performance deteriorates with the network growth. Publications latency becomes significantly larger for bigger networks. A similar test has been performed for topology management functioning over the network of virtual machines. For testing purposes we use VMWare Workstation Tool for creating virtual machine FreeBSD instances running Blackhawk prototype. We deploy the

network of four nodes, where three of them are native publishers/subscribers, acting as simple network entities. Each of them is running single instances of topology manager, connectivity helper, link-state helper function and application helper function. The remaining node is responsible for dispatching publications and subscriptions accordingly, thus it has a function of Rendezvous node. Repeating the same tests running networking daemon, averaging results and obtaining their standard deviation, we get better understanding of topology management functionality over the network. Results related to Dijkstra algorithm performance alone show a decrease in introduced delay compared to the one-domain scenario, since the average processing time of Dijkstra algorithm does not exceed 10ms. Following the same idea as for one-domain case we estimate the time needed for initialization of igraph in networking scenario, building and maintaining the graph of vertices, assignment of appropriate weights and finally applying Dijkstra algorithm. Figure 31 illustrates results obtained over multiple runs measuring the delay introduced by igraph relevant operations.

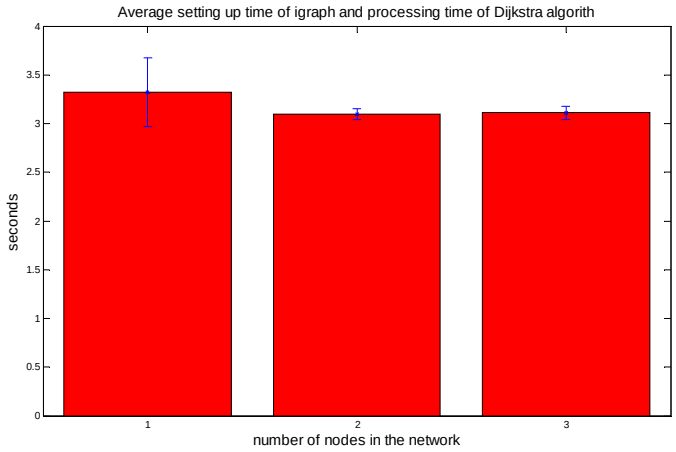

Figure 31: Average initialization time of igraph and relevant tasks together with Dijkstra algorithm execution for networking functionality.

Results acquired from single-domain tests differ compared to the networking case. This is mainly caused by different settings applied due to the nature of experiments. In the single-domain case all connectivity helper instances are residing physically on one machine and one domain. In this scenario actual topology relevant information gathering and data processing, e.g., subscribing to hello messages, link-state messages and application requests, followed by parsing the data and computing necessary forwarding state is done by a single topology manager. Furthermore, it is responsible for managing all issued publications in the domain of operation, thus experiencing a certain delay in topology management functionality is expected. On the other hand, having separated topology managers residing on each virtual machine in the network with separated helper functions for handling additional network information causes fewer overloads. Additional tests carried out within topology management networking experiments relate to simulation of different network conditions using the Dummynet tool [Dummynet]. Dummynet is designed mainly for testing purposes serving as a valuable tool for evaluation of new protocols. We utilize Dummynet in order to enforce specific conditions on particular links and to monitor changes in topology management output. More precisely, we focus on changing the delay of links, which directly affects optimal path decisions and thus changes the final forwarding IDs generated by topology management.

Future work

Due to the role of the topology management module in network path creation, the most natural approach in extending its current functionality and further tests is to rely on network daemon. Thus, future evaluation efforts will be heavily focused on testing of topology management functionality over the network. The first step in extending the current experiments will be

enlarging the network size. Bigger network experiments will show topology management limitations in the number of network entities it can handle, as well as the impact on processing delay. Further tests would include performance evaluation over real networks, e.g., real LAN or OpenVPN. Having more realistic experimental conditions we can include additional parameters coming from helper functions in order to optimize path creation, e.g. actual throughput of the link.

4.3 Feasibility evaluation of Multiprotocol Stateless Switching

4.3.1 MPSS: Multiprotocol Stateless Switching

The forwarding layer in the PSIRP architecture is achieved by using in-packet Bloom filters. Briefly, an in-packet Bloom Filter (iBF) is a source routing-style forwarding identifier. The links the packet needs to traverse are encoded into a small Bloom filter (aka zFilter), and put into the packet header. When making the forwarding decision, nodes simply check which of its outgoing links are present in the Bloom filter and copy and forward the packet accordingly. Description of the forwarding mechanism can be found in [Jok2009], while a performance analysis can be found in [D42].

MPSS (Multiprotocol Stateless Switching) is a continuation of the work on the forwarding layer by more focusing on current networks and deployability issues. In short, MPSS targets those networks where currently MPLS is installed for reasons such as flexible management of traffic, rerouting around failures, or providing Layer 2 or Layer 3 VPN services, both unicast and multicast. The primary difference in the concept of the two systems that in MPSS, MPLS labels are replaced by small Bloom filters, encoding the path or the tree the packet needs to follow. Thus, in the default case, the state related to the Label Switched Paths (LSPs) are drastically reduced, as in MPSS, the iBF already holds sufficient forwarding information about the whole path or tree in the MPSS-enabled network.

To provide the sender with an iBF, the network needs to perform three or four steps: a) compute the path, b) determine an iBF corresponding to the path, c) optionally, reserve the resources on the forwarding nodes, and d) provide the sender with the iBF. These steps can each be performed separately, and each can be accomplished either in an off-path or in an on-path, hop-by-hop manner. The off-path solutions utilize a link state routing protocol such as OSPF-TE, IS-IS-TE, or other similar protocol, for distributing link state information. The on-path solutions rely on extending existing hop-by-hop protocols. Most often the path computation and iBF determination steps are combined, but it is possible to use, e.g., off-path path computation and on-path iBF collection. Another possibility is to combine resource reservation and iBF provisioning into a single step.

While we refer to [Zah2010] for further details on the architecture, we briefly sketch two scenarios to emphasize the flexibility of MPSS. In the first scenario, consider that the tree is requested by the source node with requirements such as bandwidth constraints from a remote Path Computation Element (PCE). The PCE computes the tree satisfying the constraints, and replies with it to the source. With the strict source routing information, the source initiates an RSVP-TE process, where the resources are reserved and the iBF is calculated hop-by-hop according to the forwarding decision (and based on some flow information, optionally, cf. as in zFormation [Est2009]). In another scenario, the source node can compute the iBF, as OSPF-TE could be extended to advertise the link identifiers. Now, each node can compute the tree and even the iBF, and if resource reservation is not needed, the iBF can be immediately used for communication, without any additional signalling delay (cf. RSVP-TE explicit routes with zero bandwidth reservation, where the hop-by-hop path setup is still needed to configure the forwarding tables).

One promising application of MPSS is provisioning Multicast VPN services to organizations. In the current L3VPN architecture [Ros2006], remote sites of different organizations are connected via the service provider's backbone. The service provider needs to ensure that

VPNs can have overlapping IP address spaces, and no traffic is leaking between different VPNs. When only unicast communication is allowed in the VPN sites, the most wide-spread solution utilizes MPLS forwarding inside the service provider network: an inner MPLS label is a so-called "VPN label" for identifying the VPN, while the outer MPLS label is used for forwarding inside the provider's network. As core routers only check the outer label, and the VPN label is checked only at the edges, forwarding scalability is ensured, as the amount of state in the provider routers does not depend on the number of VPNs the network serves. Unfortunately, when multicast communication is allowed between the VPN sites, scalability is harder to ensure. Obviously, a multicast solution should be deployed in the service provider's network; the forwarding can be either simple broadcast, multiple IP or MPLS unicast, IP multicast or the usage of point-to-multipoint MPLS trees built by either LDP or RSVP-TE. However, network operators face a difficult problem when choosing the most suitable forwarding solution to use, and fine-tuning it. Conflicting viewpoints, such as bandwidth efficiency, amount of forwarding state, amount of signalling traffic should be considered. As it is impossible to find a point of operation that is optimal in all viewpoints, operators try to set up rules of thumb and use complex optimization processes to find a point of operation, where the network operates efficiently enough.

As a forwarding solution in the service provider's network provisioning Multicast VPNs, MPSS has the promise of easing the trade-off and the difficult process of fine-tuning, as it offers stateless multicast, though with the penalty of false positives, i.e. a controllable amount of unnecessary packet forwardings due to probabilistic reasons. By exploiting the features of MPSS, the signalling inside the operator's network could be reduced, and because of zero-signalling, changes in the network (e.g. new members joining a multicast group in one VPN) could be handled faster. While we are still in the phase of exploiting the potential of MPSS for Multicast VPNs and other use cases, we present a short analysis evaluating the feasibility of the MPSS architecture. This analysis is a direct continuation of the work describing the forwarding performance in [D42]. The efficiency analysis compares MPSS with MPLS, in terms of state requirements and bandwidth overhead. It is followed by discussing the flexibility of the solution, and finally, the architecture is evaluated from the security perspective.

4.3.2 Efficiency evaluation

To understand the state requirements and the bandwidth overhead of MPSS and MPLS, we implemented iBF-forwarding in ns-3 [Hen2008]. The service provider network consists of Provider (P) routers, and Provider Edge (PE) routers. Provider Edge routers are connected to Customer Edge (CE) devices, residing in the customer networks. In the L3VPN case, the PE routers are those that add MPLS labels or an in-packet Bloom filter to the packets; also, PE routers do the demultiplexing, i.e. deciding which VPN to forward the packet on. In the core, P routers simply forward the packet based on the MPLS label or on the iBF.

We study the behavior of the system in the Rocketfuel topologies [Spri2004] and on a so-called snowflake network with 1000 Provider Edges, and 55 Provider nodes (The related parameters are: S(1)=5, M(1)=10, M(2)=20 with the notation of [Yas2009]).

4.3.2.1 State

It is useful to differentiate between control plane and forwarding state, and also between state in the edges and in the core of the network.

PE state. First, there is forwarding state required in the edges for flow classification. This state depends largely on the context where MPSS or MPLS is used, and what classification policy the operator implements. Without losing generality, we can say that the number of entries required is the same as for MPLS; instead of storing MPLS labels as the output of the mapping process, the PE stores iBFs. As we will see later, the typical size of an iBF for a given flow is between 64 and 512 bits. Considering an IPv4 5-tuple as the classification index, an MPLS entry (with its 20 bits label) would take minimally 124 bits, and an MPSS entry 168-606 bits, a worst case increase of 35-380%.

In the context of L3VPNs, the PE first needs to decide the VPN label and the egress PE for the unicast packet, by looking up the Virtual Routing and Forwarding table (VRF). Then, it needs to determine the iBF to reach the particular egress PE. For example, with 4 different paths between each egress PEs in a network with 1000 PEs need a forwarding table of around 384 kbits, which is still feasible. Considering multicast, the ingress PE has to store one iBF for each customer group. In its VRF, there is one entry for each group, mapping the group to an iBF. Hence, as the number of entries in the VRF does not change, while state requirements increase, the lookup performance is similar to current deployments. Furthermore, by pushing the flow classification state to the CE, the iBFs can be stored in slower memory at the PEs. With this, the costly upgrades of PE routers can be delayed; the trigger for the upgrade can be merely the throughput of the device.

P state. In standardized MPLS procedures, with point-to-multipoint LSPs, one forwarding entry is needed in each node for each tree. In contrast, in MPSS, sparse mode multicast trees do not require state at all; see below. The total forwarding table requirement in P nodes in MPSS for sparse mode multicast groups is constant; e.g., a node with 10 neighbors, 8 LITs/interface (see e.g. [D42]), and 256 bit long iBF needs approximately 20 kbits of forwarding plane state. With MPLS, accounting 8 bytes per each rule (label swapping and incoming, and outgoing interface), 320 entries can fit into the same amount of memory. With dynamically computed link identifiers, it is sufficient to store only the key. If the size of the key is 200 bits, the state requirement is equivalent to around 3 MPLS entries.

In the control plane of the nodes, the resource consumption depends on the path setup method applied in the network. For example, if there is no bandwidth reservation on-path, the node does not need to keep state for the path in the control plane. However, if TE is needed and the network does not have a bandwidth broker, the node has to keep the same amount of control plane state as it would with MPLS.

4.3.2.2 Bandwidth

There are two factors that constitute to bandwidth overhead in MPSS. First, the larger packet header consumes some extra bandwidth. Second, false positives cause some unnecessary packet forwarding.

In MPSS, there is a tension between the Bloom filter size and the false positive rate. If the iBF is longer, the header size gets increased, and even correctly routed packets cause more bandwidth overhead, but at the same time there is less excess traffic due to false positives.

Table 1 shows the header overhead with different encapsulation methods for different packet sizes, typical to multicast traffic, according to measurements [Bev2003]. Note that besides the Bloom filter, an MPSS header might contain additional control information - we assume 2 bytes for this purpose.

Packet size	GRE	MPLS	64-bit iBF	256-bit iBF	512-bit iBF
88 B	27.3%	4.5%	11.4%	38.6%	75%
566 B	4.2%	0.7%	1.8%	6%	11.7%
775 B	3.1%	0.5%	1.3%	4.4%	8.5%
1480 B	1.6%	0.3%	0.7%	2.3%	4.5%

Table 1: Per-packet overhead of different encapsulation methods.

To quantify the overall bandwidth overhead caused by the header overhead and falsely routed packets, we start with the mean bandwidth overhead with small multicast groups and large

packets in Figure 32. The number of receivers is counted in terms of PEs and Bloom filters are quite small with respect to the used Rocketfuel AS6461 topology. Figures Figure 33, Figure 34 and Figure 35 show similar results for two different topologies (Snowflake and Rocketfuel AS3257) with larger iBFs.

Overall, selecting a suitably sized Bloom filter to match the number of egress PEs and the packet size, the bandwidth overhead compared to MPLS can be easily in the 5-15% range for all but the smallest packets, with approximately 10% overhead for 35 receivers and 1480 bytes long packets.

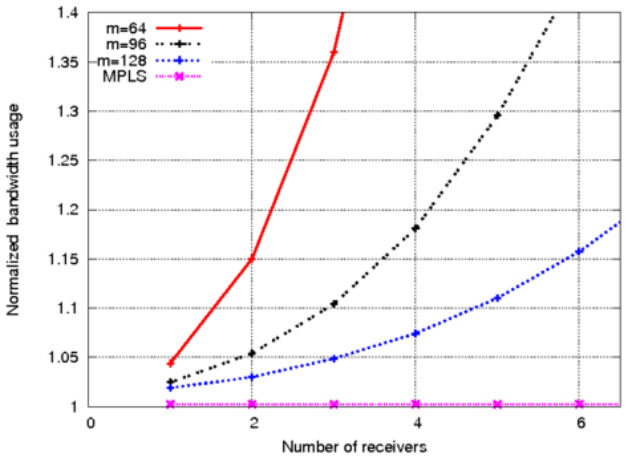

Figure 32: The bandwidth overhead of MPSS with different size iBFs and MPLS in AS6461 (138 nodes, 372 links), for packet size=1480 bytes.

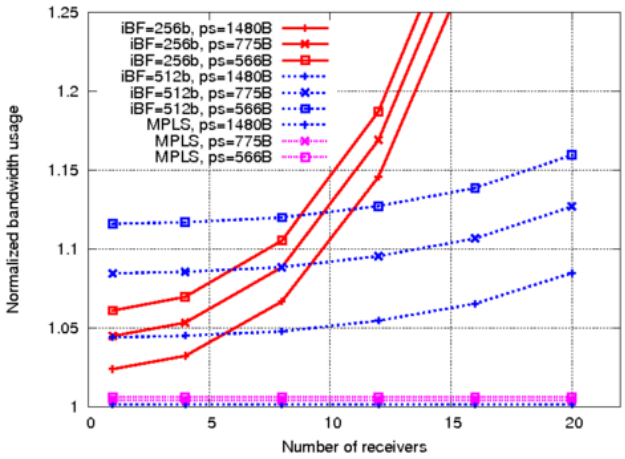

Figure 33: The bandwidth overhead of MPSS with different size iBFs and MPLS in the Snowflake topology with 1000 PEs, with different packet sizes.

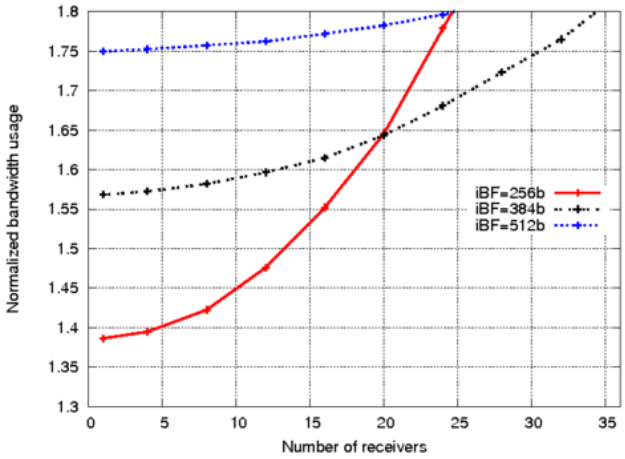

Figure 34: The bandwidth overhead of MPSS with different size iBFs in AS3257 (161 nodes, 328 links), for packet size=88 bytes.

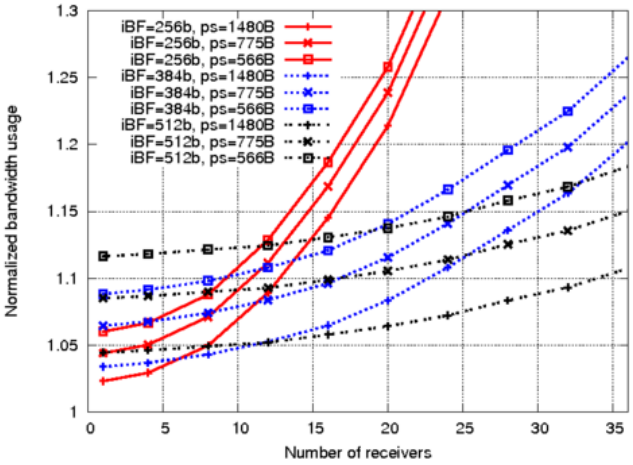

Figure 35: The bandwidth overhead of MPSS with different size iBFs in AS3257 (161 nodes, 328 links), for different packet sizes.

4.3.3 Discussion

To put the simulation results into context, we need to understand the typical multicast traffic patterns in networks. While there has been plenty of effort to characterize multicast traffic in the Internet [Cha2001], [Alm1997], to our knowledge, only Karpilovsky et al [Kar2009] have analyzed multicast traffic in a large operator network that provides VPN services to enterprises. They clustered the multicast flows to 4 types, summarized in Table 2. Furthermore, according to them, most of the flows had low bandwidth requirements, with peak rates < 10 kb/s.

The large majority of all flows had a limited number of egress PEs, typically 3 at maximum. These flows contributed to a large amount of multicast routing state in current deployments. The second biggest cluster contained moderate bandwidth flows with a relatively large number of receivers. The high bandwidth flows were almost unicast like. In the smallest cluster of long lived flows, the average of the maximum number receivers was below 10. Unfortunately, the size of the networks where the measurement were taken is not reported.

Assuming a reasonably sided network, MPSS can easily handle type 1 flows with a reasonable per-packet overhead and rare (if any) false positives. For type 3, the flows can be handled with small iBFs and virtually no false positives. Type 4 can be handled with 256-bits long iBFs, resulting in 3-40% traffic overhead mostly caused by the headers.

While the penalty of the iBFs is highest in cluster 2, it can still be feasible to route these flows in a purely stateless manner with iBFs. On the other hand, if the bandwidth overhead would be too high for these flows, the operator can add some state to reduce bandwidth overhead. It remains an open question how to optimize this; indeed, Karpilovsky et al. note that "optimizing [these flows] may not be of primary importance".

Flow type	Average number of receivers	Fraction of flows
Few receivers	2.1	86.7%
Well fitted	19.6	13.3%
High bandwidth	1.2	0.1%
Long lived	3.1	<0.1%

Table 2: VPN multicast flow clustering by Karpilovsky et al [Kar2009].

4.3.4 Flexibility

The MPSS architecture provides a few important benefits. MPSS enables PEs to create any tree in the network without signalling. This is possible, because there is no need to add state into the P routers, unless resource reservations are needed and done at the routers. In contrast, MPLS each new path either requires one round trip signalling (e.g. with RSVP) or is constrained to shortest path (with LDP). This also makes multicast tree management more flexible, as changes in the tree can be done locally without signalling to the Ps in the multicast tree.

Controlling the iBF size affects the excess bandwidth by changing the amount of header space required and the number of false positives. This gives MPSS flexibility in optimizing efficiency, as both in-network state and iBF size can be used to control bandwidth waste.

4.3.5 Security

The so-called cross connection threat, where packets are delivered to wrong outputs, is one of the main MPLS security concerns [Fan2009]. A false positive leading to a flow being duplicated across an incorrect link may not by itself be a large security concern. However, forwarding traffic to a wrong customer, e.g., due to a false positive, would create a security problem. To prevent this, we mark the packets with VPN or egress PE labels.

Pushing the state to CE (i.e. when the CE adds the iBF it received for the PE to the packet) creates new potential attack vectors, as an attacker having physical access to a CE can try to manipulate the packet headers, either for self interest (e.g. upgrading quality of service) or in an attempt to harm the network. Encrypting the iBF could help with iBF tampering; for securely associating the iBF with the flow, secure iBFs can be used.

One often cited problem in inter-domain label switching is domain privacy. Service providers are not willing to reveal the identity or connectivity of their forwarding nodes to their neighbors. With iBFs, it is possible to reveal paths, in the iBF form, from a domain border to any given router, and even bind it to a specific flow (e.g. certain kind of control traffic). Secure iBFs and secure bit permutations can be employed to provide useful iBFs to the neighbors without revealing network details.

Other than domain privacy, the use of contiguous iBFs in a multi-domain environment results mostly in similar security problems to the single domain case. One potential additional problem is created by a case where two domains use the same inner MPLS label for different VPNs, in the case a false positive would cause a the tree to include an egress PE which uses the label for the wrong VPN. Hence, for inter-domain MPSS VPNs, it may be necessary to use

secure VPN or egress PE names, or otherwise provide coordination over the inner MPLS label space.

4.4 Rendezvous Design

The evaluation of the rendezvous framework developed in PSIRP was originally described in detail in deliverable D4.2. In this section we shall give an update on the progress after that document was finalized in terms of the evaluation work carried out in that environment. We shall not repeat in detail the rendezvous network design itself, nor give a complete description of the simulation model. For these, the reader is referred to either D4.2, or the self-contained technical report [Raj2009]. The implementation of the simulation environment is described in detail in D4.4.

4.4.1 Updated Results

Since D4.2 was issued, we have rerun the simulations for the performance evaluation of the rendezvous architecture in the new Octave-based simulation framework, as well as extended the simulations to also yield results on the sensitivity of the results on individual parameter choices. We shall first give the basic performance results in terms of the stretch, latency and load achieved by the design, and then comment the sensitivity analysis in some detail.

In Figure 36 we report the cumulative distribution function (CDF) of the measured stretch, the multiplier to AS-level hop count required to route a rendezvous message from a randomly selected source domain to the domain where the sought after object is located, via the overlay structure vs. routing directly using a routing policy compliant valley-free path (bypassing the overlay). The small number of requests with stretch below one are due to the overlay node functioning as a detour [Sav1999], offering a shortcut, typically utilizing peering links otherwise not usable for policy-compliant end-to-end paths.

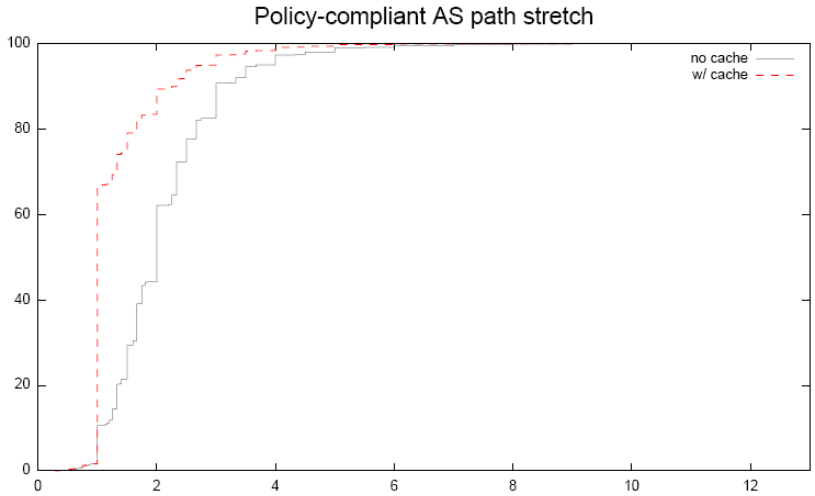

Figure 36: AS-level stretch CDF: 2.11 (mean), 3.75 (95%) (without caching); 1.35 (mean), 3.00 (95%) (with caching).

In Figure 37 we show the histogram for the number of hops in the overlay needed to reach the scope pointer.

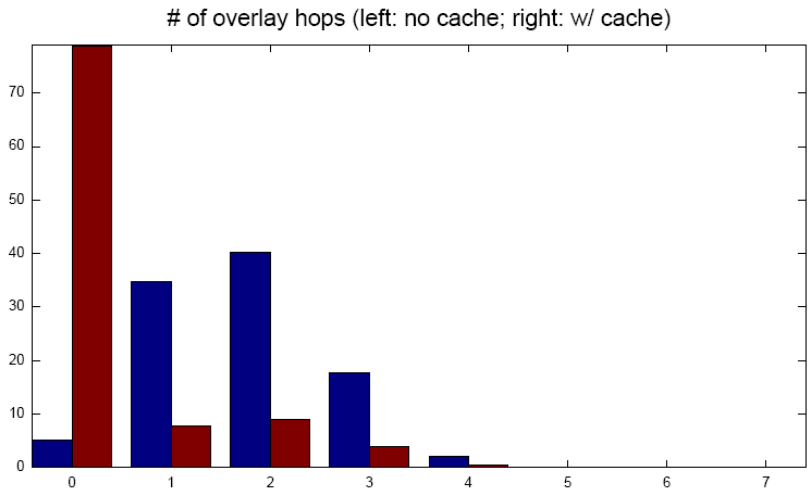

Figure 37: Rendezvous overlay hops needed to reach the scope pointer (% of requests): 1.77 (mean), 3.00 (95%) (without caching); 0.39 (mean), 2.00 (95%) (with caching).

Figure 38 shows the cumulative distribution of overlay routing latency in milliseconds, depicting the time to reach an overlay node holding the scope pointer, starting from the user's home AS. The latency model used is as was in D4.2.

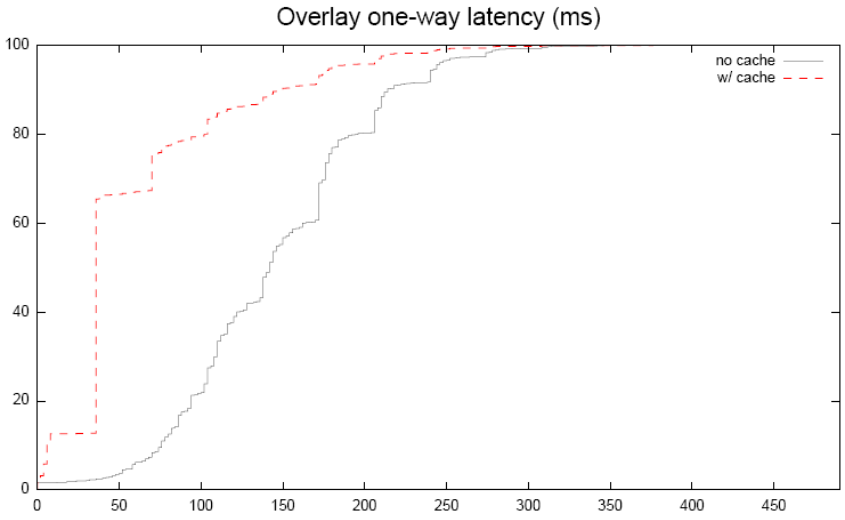

Figure 38: Query latency CDF (ms): 144 (mean), 244 (95%) (without caching); 62 (mean), 180 (95%) (with caching).

Figure 39 shows the distribution of node load, measured by the number of times each node forwards or handles a rendezvous request message during a simulation run of 100000 rendezvous requests. It can be seen that most of the nodes are lightly utilized, and the heaviest load concentrates on a rather small set of nodes. However, the ratio between the most heavily loaded nodes and the average is not too big, and caching reduces that ratio considerably.

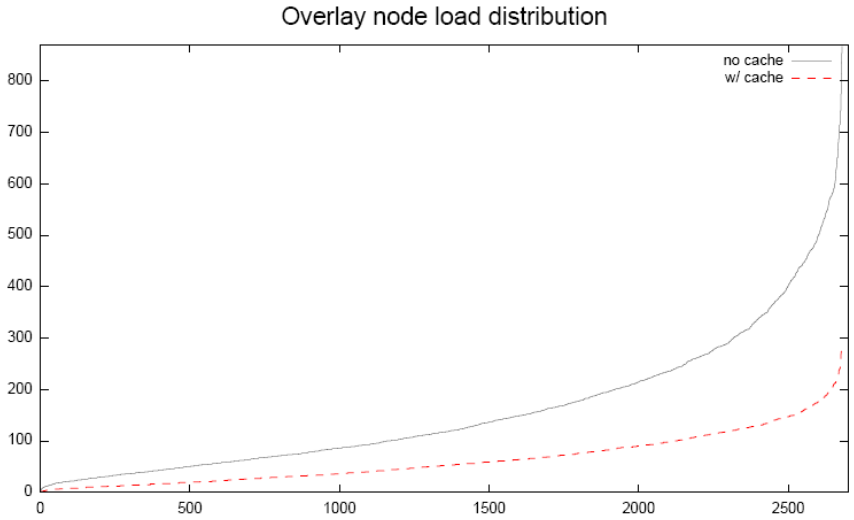

Figure 39: Figure 5.4: Node load distribution of 100000 rendezvous requests on the overlay nodes: 159 (mean), 441 (95%) (without caching); 63 (mean), 157 (95%) (with caching).

After conducting the simulations reported above, a factorial analysis was performed to get an understanding of the sensitivity of the model to some of the simulation parameters. The analysis was conducted without caching to better show the effect of individual parameters. The non-caching case can be considered the \worst case scenario", as even modest caching dramatically improves the average performance figures. Analysis results based on 8 x 64 replications are summarized in Table 3. As can be seen from the confidence interval figures in the Table 3, the average stretch is highly stable with modest number of replications (8), while analyzing the tail (95 percentile) of the latency would in the case of some parameters need more replications in order to yield statistically significant results. The difference in the parameter values for the base level and the alternate level were set on the intuition that alternate values should produce better performance. This was indeed the case for most of the variables:

Node Memory Amount of memory nodes dedicate to store object pointers: Less memory, more nodes needed to serve the target 10^{10} scopes.

SmallNetLimit limits the size of networks accepted as Rendezvous network customers. Bigger rendezvous networks lead to lower average latency, as their internal routing is policy-compliant, but amount of state that needs to be managed by a single rendezvous service provider grows.

Util. Fraction allows networks with small impact on load to become rendezvous customers, higher value leads to lower total number of rendezvous networks created, but more load on individual rendezvous service providers.

Min. Nodes. Minimum number of overlay nodes for each rendezvous service provider. More than one may be necessary for fault tolerance (compare to DNS operative guidelines).

Repl. Count. Number of replicas of object pointers to keep on the global level. One is the functional minimum, more than one may be needed for fault tolerance. Wider replication should help lowering the routing latency, but requires more storage space in the system.

Target Links. 70 is close to functional minimum with the simulation parameters used. More overlay links implies denser connectivity; more opportunities to select shorter routes and higher the maintenance cost for managing the links.

The only (initially) counter-intuitive response was with the minimum number of nodes in a rendezvous service provider domain: While increasing overlay node count via reducing the

per-node memory allocation resulted in consistently (somewhat) better performance, increasing the minimum number of overlay nodes in all rendezvous networks from 1 to 2 resulted in worse performance. This may be explained by the fact that the minimum applies also with rendezvous service providers with lightly loaded overlay nodes, hence the main effect is somewhat increased stretch in the system.

Parameters:	Node Mem.	SmallNetLim.	Util.Frac.	Min.Nodes	Repl.Count	Links
Base level:	4GB	2	0.1	1	1	70
Alt. level:	2GB	8	0.3	2	4	140
Effect:						
Stretch (mean):	-0.1691	-0.1142	-0.0781	0.1775	-0.5747	-0.1090
CI (95%):	0.0001	0.0019	0.0002	0.0002	0.0009	0.00004
Stretch (95%):	-0.2991	-0.2373	-0.1149	0.2978	-1.0550	-0.2002
CI (95%):	0.0030	0.0202	0.0053	0.0077	0.0122	0.0024
Latency (mean):	-10.4120	-8.4217	-6.8646	13.1244	-37.3722	-9.7002
CI (95%):	0.1047	5.8838	1.4313	0.3427	0.5786	0.3221
Latency (95%):	-13.9063	-18.5000	-9.9375	12.3594	-51.7656	-11.0469
CI (95%):	1.6636	18.5044	7.7257	3.5080	6.8988	0.7126

Table 3: Sensitivity analysis results (8x64 replications, no caching). Numbers indicate the average effect of a given variable change from base to alternative level. CI lines report the half length of the 95% confidence interval for the parameter effect above.

To assess the effect of caching on the parameter sensitivity, we compared the performance of the worst performing combination and the best performing combination of the parameters in the Table 3, with and without caching. The performance figures are summarized in Table 4. Contrary to our anticipation, caching helps also the 95 percentile latency performance (243 ms vs. 186 ms for the best performing parameter set). Also contrary to our anticipation, caching did not mask the performance difference between the best and worst performing parameter sets: Both with and without caching the increase in the 95 percentile latency is approximately 50% more with the worst performing parameter set, compared to the best performing parameter set.

	Stretch (mean)	Stretch (95%)	Latency (ms,mean)	Latency (ms,95%)
Without caching:				
Best:	2.075	3.79	142	243
Worst:	3.217	5.94	222	347
With caching (~75% hit ratio):				
Best:	1.3410	2.74	62	186
Worst:	1.5950	4.058	79	281

Table 4: Best vs. worst parameter combination performance (with and without caching).

Finally, we compared the performance of our provider based model against the variant where there are only singular rendezvous networks, i.e. a model where each AS operates as their own rendezvous service provider. Similar hierarchical clustering was conducted in each case. On the average over multiple simulation rounds the alternative resulted in about 25% increase in both stretch and latency. This shows that our incentive-based separation between the roles of the ASes may indeed perform better than the variant without such separation.

4.4.2 Conclusions

Our sensitivity analysis shows that our model gives rather robust performance indicators for our chosen metrics, other than for the 95 percentile latency, which exhibits significant statistical fluctuation between simulation runs for some parameters. Even so, we conclude that

the additional (95 percentile) latency contribution of our rendezvous design compares favorably to DNS, where the comparable worst case performance is measured in seconds [Jun2002].

The analysis above shows that our system performance is better than the variant without the incentive motivated separation of the edge-based rendezvous networks from the rendezvous service providers. Also, it can be argued that our stretch and latency figures show that the system performance is close enough to stretch-1 systems, when considering the huge decrease in the required number of servers and the related incentive challenges.

The presence of transit loops and the lack of policy-compliant connectivity between some domains in the CAIDA dataset forced us to add a degree of realism we did not originally plan for in the simulation. To manage the observed non-connectivity, we probed the connectivity in the dataset for each candidate overlay link and only used links that had policy-compliant connectivity. This also enabled us to mimic the real-world latency measurement capabilities used in overlay structures, resulting in realistic analysis of the effect of choosing topologically short links.

The major deficiency in the presented analysis is the lack of modeling of defection, or untrustworthy operation by some of the overlay participants. However, there are couple of approaches that could be utilized in practice to detect and route around such behavior. Firstly, overlay nodes could ask a random set of other nodes to test the reachability for the object(s) registered by the first node. This would help against lying nodes that would respond favorably the original registrar, but still claim non-availability to the other requesting nodes. Secondly, having detected untrustworthy behavior, the first node could initiate wider replication of its object registrations, thus quickly minimizing the effect of the misbehaving node. Furthermore the first node could initiate measures for excluding the lying node from the overlay.

4.5 Deploying a Multi-Site PSIRP Test Network

During its 3 months extension, PSIRP will establish a test bed facility that will showcase integration as well as isolated technologies that have been developed within PSIRP. It will further serve as a playground for evaluation of various kinds. The following section outlines the setup, applications and envisioned evaluations within this test bed.

4.5.1 Test Bed Set-up

The test bed utilizes the current node implementation, i.e., the Blackhawk implementation. Several nodes, depending on availability of hardware, are installed at various partner sites (see below). These local nodes are directly connected through Ethernet. Each local site is interconnected with other sites through an openVNP configuration, i.e., Ethernet frames are tunneled via the public Internet. This will also allow for public demonstration through laptop setups, i.e., a local demo (with one or more laptops) can be connected to the overall setup, assuming that proper openVPN configuration and authorization is in place. The openVPN server is currently setup at the University of Essex.

Initially, the different sites are administered as a single PSIRP domain with a single topology manager [D2.4]. Eventually, however, each site will be configured as a single PSIRP domain to fully enable inter-domain operations. Also flexible rendezvous configuration will be implemented, based on the current rendezvous point implementation. All forwarding elements implement packet forwarding with in-packet Bloom Filters. The Link Identities will be based on either fixed LIDs or Z-formation based LID calculations. (Using Z-formation requires support from the Topology manager). Furthermore, several partners have expressed interest in setting up forwarding nodes based on the existing NetFPGA implementation for the Bloom filter forwarding approach, based on fixed LITs and/or Z-formation based LITs.

The currently envisioned setup for the testbed until the end of September is:

- Essex University (non-PSIRP partner): 5 PSIRP nodes, acting as publishers and/or subscribers as well as forwarding nodes
- Cambridge University: 2 PSIRP nodes
- Aachen University: up to tens of PSIRP nodes through cluster machine setup
- Athens University: several dedicated machines
- Helsinki University: several dedicated machines
- Institute for Parallel Processing - BAS: One powerful server with possibility to hold many virtual machines + another two dedicated machines

With this initial setup, in the order of ten or more distributed machines are setup while having additional cluster capabilities. As an additional site, discussions are ongoing with MIT to setup at least one or two machines as a PSIRP domain in the US.

4.5.2 Applications

The following applications are envisioned (without any guarantee that all of them will be realized):

- Simple file transfer: one publisher, one or more subscribers
- Non-realtime video streaming: one frame per publications with re-publication of new frames, one or more subscribers
- Collaborative working (e.g., shared applications): real-time audio/video conferencing, requiring bidirectional traffic. This is likely to be implemented at beta-stage only!
- Legacy applications through a socket emulator: the socket emulator, implemented at AUEB, will be utilized for, e.g., audio streaming applications over standard IP
- Web applications with modification to the HTTP model: utilizes the Firefox plugin for the PSIRP protocol, i.e., metadata document for the rendering is sent with embedded RId for active objects such as weather or sensor information

4.5.3 Envisioned Tests

The setup, as described above, allows for evaluating various parts of our implementation architecture. The following functions will be particularly tested:

- Node implementation: given the larger number of Blackhawk nodes in the system, test on performance as well as stability of the implementation will be the focus of our tests.
- Topology manager: the first version of the topology manager will be tested, first within a single domain and then in several instances where each site will implement a single domain.
- Rendezvous function: the first integrated rendezvous code will be tested regarding its stability and conformance to the expected functionality. For this, simple publication scenarios with various scopes will be run.
- Forwarding function: the current forwarding is realized using the LIPSIN [LIPSIN] mechanisms, implemented in the Blackhawk node as well as in NetFPGA. Both implementations will be tested regarding their performance and stability. For the NetFPGA implementation, the setup will need to be extended with some of the existing NetFPGA boxes available at various partner sites.

In addition to performance and stability test of various functions, the testbed will also serve application development purposes. This effectively tests the viability of the provided APIs, as outlined in [Tro2009].

4.5.4 Future Usage

The testbed serves as a starting point for future efforts in the space of information-centric networking. One of these is the PURSUIT project which is currently being setup under call 5 of FP7. PURSUIT will directly built on top of the PSIRP architecture and investigate extensions and further technology developments. This work can be directly placed into the then existing testbed for further testing along the lines of performance, stability and usability. Other efforts will also make use of the facilities, e.g., the UK-funded PAL project and work at MIT in the area of network management.

5 Conclusions

The validation and evaluation work in the project has continued in very active manner since the release of the previous deliverable, D4.2, reporting on WP4 results a year ago. The quantitative evaluation work has continued on track, providing a continuous stream of results on diverse issues such performance of the developed prototype components, overlay solutions, and the different Inter-domain solutions developed. In addition to the specific results obtained, the quantitative evaluation work has also resulted in new methodological contributions, such as the high-level simulation environments developed to enable high-level simulation of Internet-scale systems. We strongly believe that such contributions from the project will also be useful to other researchers working on designing and evaluating large-scale networked systems. The security evaluation of the overall PSIRP architecture and the various individual components has also proceeded over the past year, and new results and open issues especially on trust establishment have been reported in this document.

In addition to the quantitative evaluation work and security validation, the project has also made significant progress on the development and application of the socio-economic validation techniques on the PSIRP architecture. The chosen system dynamics approach enables high-level modeling of the different factors affecting the adoption and successive deployment of the different technologies developed in the project, and studying the potential impact of different technical choices on this level. While still work in progress, the initial results show that the developed methodology has significant potential in validating and studying the socio-economic factors related to adoption of new Internetworking technologies. We are also working on integrating the socio-economic evaluation techniques with the high-level simulation framework developed for quantitative evaluation work. Such an integrated environment would allow study of the performance impact of, for example, different technology adoption scenarios, and to feed the results back into the systems dynamics models for the socio-economic evaluation tools. This integration work will continue over the last months of the project, and results will be made available to the community through the usual dissemination channels.

One of the unique aspects of the PSIRP validation and evaluation work has definitely been the breadth of the activities carried out. The project has not only focused on performance aspects of the architecture and related technical solutions, but has instead adopted a holistic approach covering much more than the commonly tackled performance aspects. The obtained results are very promising, demonstrating both the feasibility and potential in the information centric networking approach developed in the project.

References

[Alm1997] K. Almeroth and M. Ammar, "Multicast group behaviour in the Internet's multicast backbone (MBone)", IEEE communications Magazine, 35(6):124–129, 1997.

[Bau2007] I. Baumgart, B. Heep, and S. Krause, "OverSim: A flexible overlay network simulation framework," in Proc. of the IEEE Global Internet Symposium, Anchorage, AK, USA, Jan 2007, pp. 79–84.

[Bel2004] A. Bellissimo, B. N. Levine, and P. Shenoy, "Exploring the use of bittorrent as the basis for a large trace repository," University of Massachusetts Amherst, Tech. Rep., June 2004.

[Ber2001] P. Berends and A. Romme, "Cyclicality of capital-intensive industries: a systems dynamics simulation study of the paper industry", 2001.

[Bev2003] R. Beverly and K. Claffy. "Wide-area IP multicast traffic characterization", IEEE network, 17(1):8–15, 2003.

[Bro2009] I. Brown, "Socio-economic drivers of Internet development", 2009

[Bus2002] M. Busari and C. Williamson, "ProWGen: a synthetic workload generation tool for simulation evaluation of web proxy caches," ComputerNetworks, vol. 38, no. 6, pp. 779–794, 2002.

[Cas2002] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "SCRIBE: A large-scale and decentralized application-level multicast infrastructure," IEEE JSAC, vol. 20, no. 8, pp. 100–110, 2002.

[Cas2003a] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh, "SplitStream: High-bandwidth multicast in cooperative environments," in Proc. of the ACM SOSP, 2003, pp. 298–313.

[Cas2003b] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "Scalable application-level anycast for highly dynamic groups," in Proc. of the Networked Group Communication Workshop, 2003.

[Cha2001] R. Chalmers and K. Almeroth, "Modeling the branching characteristics and efficiency gains of global multicast trees", In IEEE INFOCOM, volume 1, pages 449–458, 2001

[D42] J. Riihijärvi, D. Trossen, G. Marias, T. Burbridge, A. Zahemszky, J. Ylitalo, D. Lagutin, K. Katsaros, G. Xylomenos, J. Rajahalme, K. Visala, M. Särelä, B. Gajic, C. Esteve, S. Arianfar, P. Nikander, T. Rinta-aho, J. Keinänen, K. Slavov, "D4.2 First report on quantitative and qualitative architecture validation", PSIRP deliverable, 2009.

[Del2.4] Mark Ain et al., "PSIRP deliverable D2.4, Update on the Architecture and Report on Security Analysis", Sept 2009

[Dio2000] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture,"Network, IEEE, vol. 14, no. 1, pp. 78–88, 2000.

[Est2009] C. Esteve, P. Jokela, P. Nikander, M. Särelä, and J. Ylitalo, "Self-routing Denial-of-Service Resistant Capabilities using In-packet Bloom Filters", Proceedings of European Conference on Computer Network Defence (EC2ND), 2009.

[Fan2000] L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary Cache: a scalable wide-area web cache sharing protocol," IEEE/ACM Transactions on Networking, vol. 8, no. 3, pp. 281–293, 2000.

[Fan2009] L. Fang, "Security Framework for MPLS and GMPLS Networks", Internet-Draft draft-ietf-mpls-mpls-and-gmpls-security-framework-07, IETF, Oct 2009. Work in progress.

[Fot10] N. Fotiou, G. C. Polyzos and G. F. Marias. "Information Ranking in Content-Centric Networks", Future Network and Mobile Summit, 2010.

[Gan04] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in G Major: Designing DHTs with Hierarchical Structure," ICDCS, March 2004.

[Guo2007] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang, “A performance study of BitTorrent-like peer-to-peer systems,” IEEE JSAC, vol. 25, no. 1, pp. 155–169, 2007.

[Hef2008] M. Hefeeda and O. Saleh, “Traffic modeling and proportional partial caching for peer-to-peer systems,” IEEE/ACM Transactions on Networking, vol. 16, no. 6, pp. 1447–1460, 2008.

[Hen2008] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. B. Kopena, "Network simulations with the ns-3 simulator", SIGCOMM’08 Demos, August 2008.

[Hui2002] J. Huigen, “OECD reviews of regulatory reform”, 2002

[IAN2009] IANA. (2009, Jun) Autonomous system (AS) numbers. [Online]. http://www.iana.org/assignments/as-numbers/as-numbers.xml

[ISO1999] ISOC, “Internet Governance: The Struggle over the Political Economy of Cyberspace”, available at http://www.isoc.org/oti/articles/0199/rao.html, 1999

[Jok2009] P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, and P. Nikander, "LIPSIN: Line speed publish/subscribe inter-networking", In SIGCOMM, 2009.

[Jun2002] J. Jung, E. Sit, H. Balakrishnan, and R. Morris, “DNS Performance and the Effectiveness of Caching,” IEEE/ACM Transactions on Networking (TON),10(5):589{603, 2002.

[Kar2009] E. Karpilovsky, L. Breslau, A. Gerber, and S. Sen, "Multicast redux: a first look at enterprise multicast traffic", In Proceedings of the 1st ACM workshop on Research on enterprise networking, pages 55–64. ACM, 2009.

[Kat2009] K. Katsaros, V. Kemerlis, C. Stais, and G. Xylomenos, “A BitTorrent Module for the OMNeT++ Simulator,” in Proc. of the IEEE MASCOTS, 2009, pp. 361–370.

[Lag08] D. Lagutin, “Redesigning Internet - The packet level authentication architecture,” licentiate's thesis, Helsinki University of Technology, Finland, June 2008.

[Mey2007] D. Meyer, “Report from the IAB workshop on routing and addressing, RFC 4984”, 2007

[Pew2009] Pew Internet and American Life Project, “The Internet and Civic Engagement”, available at http://www.pewinternet.org/~/media//Files/Reports/2009/The%20Internet%20and%20Civic%20Engagement.pdf, 2009

[Pew2010] Pew Internet and American Life Project, “The Impact of the Internet on Institutions in the Future”, March 2010

[Raj2008] J. Rajahalme, M. Särelä, P. Nikander, and S. Tarkoma, “Incentive compatible caching and peering in data-oriented networks,” in Proc. of ACM CoNEXT, Madrid, Spain, 2008, pp. 1–6.

[Raj2009] J. Rajahalme, M. Särelä, K. Visala, and J. Riihijärvi, “Inter-Domain Rendezvous Service Architecture”, PSIRP Technical Report #TR09-003, December 2009.

[Rii2009] J. Riihijärvi (ed), "Description of validation and simulation tools in PSIRP context", PSIRP deliverable D4.4, 2009

[Ros2006] A. Rosen and Y. Rekhter, "BGP/MPLS IP virtual private networks (VPNs)", RFC 4364, IETF, Aug. 2006.

[Row2001] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems,” in Proc. Of the Middleware Conference, 2001, pp. 329–350.

[Sav1999] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan, “Detour: Informed Internet Routing and Transport,” IEEE Micro, 19(1):50-59, Jan/Feb 1999.

[Spri2004] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel" IEEE/ACM Trans. Netw., 12(1):2–16, 2004.

[Ste2000] J. D. Sterman, “Business Dynamics: Systems Thinking and Modeling for a Complex World”, McGraw-Hill Higher Education, Boston, 2000

[Tew1999] R. Tewari, M. Dahlin, H. Vin, and J. Kay, “Design considerations for distributed caching on the internet,” Proc. of the ICDCS, 1999.

[Tro2009] D. Trossen (ed), “Architecture Definition, Components Descriptions and Requirements”, PSIRP, 2009

[Var2008] A. Varga and R. Hornig, “An Overview of the OMNeT++ Simulation Environment,” in Proc. of ICST SIMUTools, Brussels, Belgium, 2008, pp. 1–10. Network, IEEE, vol. 14, no. 1, pp. 78–88, 2000.

[Yas2009] S. Yasukawa, A. Farrel, and O.Komolafe, "An Analysis of Scaling Issues in MPLS-TE Core Networks", RFC 5439, IETF, Feb. 2009.

[Zah2010] A. Zahemszky, P. Jokela, M, Särelä, S. Ruponen, J. Kempf, and P. Nikander, "MPSS: Multiprotocol Stateless Switching", in proceedings of 13th IEEE Global Internet Symposium 2010, San Diego, CA, USA, March 19, 2010.