PSIRP
PUBLISH-SUBSCRIBE
INTERNET ROUTING
PARADIGM

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

DELIVERABLE D2.1

State-of-the-Art Report and Technical Requirements

Title of Contract	Publish-Subscribe Internet Routing Paradigm
Acronym	PSIRP
Contract Number	FP7-INFSO-IST 216173
Start date of the project	1.1.2008
Duration	30 months, until 30.6.2010
Document Title:	State-of-the-Art Report and Technical Requirements
Date of preparation	30.06.2008
Authors	Arto Karila, Mark Ain, Dmitrij Lagutin, Sasu Tarkoma, Xie Xiaolei (TKK-HIIT), Dirk Trossen, Trevor Burbridge (BT), Mikko Särelä, Pekka Nikander, Teemu Rinta-aho (LMF), András Zahemszky (ETH), Jarno Rajahalme (NSNF), Janne Riihijärvi, Borislava Gajic (RWTH), George Xylomenos, Giannis Marias, Nikos Fotiou (AUEB)
Responsible of the deliverable	Arto Karila (TKK-HIIT)
	Phone: +358 50 384 1549
	Fax: +358 9 694 9768
	Email: arto.karila@hiit.fi
Reviewed by	Dirk Trossen (BT)
Target Dissemination Level:	Public
Status of the Document:	Completed
Version	1.0
Document location	http://www.psirp.org/deliverables/
Project web site	http://www.psirp.org/

SEVENTH FRAMEWORK PROGRAMME

European Commission
Information Society and Media

Table of Contents

This document has been produced in the context of the PSIRP Project. The PSIRP Project is part of the European Community's Seventh Framework Program for research and is as such funded by the European Commission.

All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

1 Introduction

The PSIRP project aims to solve some major issues of the current Internet by applying a (information-centric) publish/subscribe paradigm throughout all layers of the network, in particular throughout the internetworking layer, which to our knowledge has not been done before. Our research hypothesis is that this will provide a better basis for the Future Internet than the current topology-based model.

Some of the key background for this work is that, e.g., many widely used Internet applications already are essentially publish/subscribe in nature. For example, distribution of software updates is currently performed in a poll/unicast fashion that is clearly non-optimal. Instead, subscribing to the updates that are needed and distributing them via multicast, caching etc. would be much easier and more efficient from the point of view of using network resources. The same applies to multimedia, such as IPTV, and many other applications, such as those for the dissemination and sharing of massive amounts of information (e.g. BitTorrent) or even those distributing very low amounts of information, like RSS feeds.

In order to push this project beyond the state-of-the-art (SoA), we have to first study the current state-of-the-art and the solutions proposed to solve various problems of the current Internet. To be successful, the PSIRP architecture must not only employ but also extend the state-of-the-art from many specific sub-areas of communications. This SoA report expands the brief SoA study presented in the original Project Proposal and the subsequent Description of Work (Annex I of the Grant Agreement), going deeper into specific areas and their key publications, summarizing prior work done, and outlining the results that were gained. With this extension of the state-of-the-art, we intend to build a common basis and terminology to work on within the project, including the production of a quick guide for finding appropriate references at the later stages of the work.

It is important to note however that the deliverable D2.1 only presents a snap-shot of the SoA as of late June 2008. The work does not end here since our understanding of other SoA continues to evolve. Hence, the SoA will be a live document in the project wiki that goes beyond this deliverable, being expanded throughout the project and updated as new developments unfold.

2 Guiding Principles, Scope, and Methodology

This section provides an overview of the rationale by which this document was designed and compiled. This rationale follows our clear formulation of architectural design principles to guide the project's progress, the scope of the state-of-the-art which is largely influenced by PSIRP's guiding principles, and the methodology by which SoA research is collected, contrasted to various aspects of PSIRP, and maintained.

2.1 Guiding Principles

The goals of the PSIRP project are very ambitious with respect to its scope of work (embodying areas such as routing, forwarding, rendezvous, identifiers, and many more) and intended scale (inter-domain as a replacement of the current IP layer). In order to support achieving the intended results in the project, we need a clearly guided investigation of the state-of-the-art, and this guidance is provided by the overall vision and goals of the PSIRP project.

As also outlined in our architectural design process (D2.2), we can observe that our vision revolves around the following major concepts:

- Everything is information, building up from simple forms of information to very complex knowledge on application level.
- Different forms of information reachability exist throughout all levels of the design and they can be changed and adapted in real-time.
- Control is handed back to the receiver by virtue of a communication model that allows for a choice of reception without needing to receive everything that is sent.

Starting from these goals, we intend to study the state-of-the-art that will allow us to reflect on these goals and extend the SoA towards the novel solutions that this project intends to devise.

2.2 Scope

The scope of this SoA study is defined by the overall goals of the project, as outlined above. Subsequently, we map these goals onto areas of investigation that are seen as relevant in providing "food for thought" for implementing the final project goals. It is important to note, however, that the scope of this deliverable is purely centred on the design aspects of our work. Hence, issues surrounding validation, including the *red-team* approach for security validation of our solutions, is not included in this document and will be covered in a separate technical report later in the project.

Given our project goals, we've identified the following two key areas to be investigated:

2.2.1 Future Internet Architecture

General work on other attempts to build a future Internet architecture is important to PSIRP. Within this area, the following main architectural aspects are chiefly involved:

- Protocols
  - Naming, Addressing, and Routing
  - Multicast (necessary for efficient dissemination of large quantities of information, which will be the norm (not the exception) in the PSIRP architecture)
- Mechanisms

  - Compensation, Caching, and Security
  - Network Coding (has wide applicability in the reliable, timely and efficient delivery of information over heterogeneous networks)
- Publish/Subscribe (related system solutions are important due to the relevance of receiver-driven communications in our approach)

2.2.2 Design Considerations

This section will cover general design considerations for systems of the envisioned scale.

- *Economics* is an area whose relevance can be derived directly from the focus on the required compensation mechanisms and the economic impact that a PSIRP system would embody.
- *Socio-economic aspects* are important due to the increasing understanding that any change to the Internet at the scale PSIRP envisions will have tremendous impact on society as a whole. The PSIRP vision revolves around the ability to flexibly adapt to and reflect the social structures of society.
- *Security* must be designed in the architecture and built into its implementation from the very beginning. The security problems of the current Internet clearly demonstrate that security cannot be efficiently added to an architecture as an afterthought. A good understanding of previous work in this space is therefore crucial.
- *Trust* is an important aspect of networking in the era of so called information and communication technology (ICT) diffusion, where information networks have become an inherent part of all human activities.
- *Privacy* has also become an increasingly important consideration in the deployment of ICT. Modern technologies allow for constant monitoring of individuals' movements and behaviours, and there must be a fine balance between privacy and accountability in the Future Internet to protect the privacy of people while enabling, e.g., legal interception, when it is applicable. An understanding of current approaches and viewpoints is important to steer our own thinking.

2.3 Methodology

The methodology used to compile this SoA report is dictated by the envisioned scope, as outlined above. The relevant state-of-the-art within this scope is gathered by experts in the identified sub-areas and reviewed against the bearing of the work towards our own project goals. The findings of these reviews can be found in this report.

It is important to note that this report is only one step towards understanding the relevant prior art in our space of work. Given the iterative methodology of the PSIRP project (design, implement, and validate/break), we expect this report to grow over time, reflected in our project-internal tools and also through the availability of revised versions of this report later in the project. Furthermore, the state-of-the-art is not solely limited to design considerations and will be extended towards validation and other areas beyond this current deliverable.

3 Future Internet Architecture

Leading-edge research involving advanced internetworking technologies and their applications to the future Internet are of great importance to the PSIRP effort. This section discusses the SoA aspects which are most relevant to PSIRP's outlook, including fundamental network components (e.g. naming, addressing, routing etc.), advanced operational features, tactics to enhance efficiency and reduce resource usage, and overall information delivery design philosophies.

3.1 Introduction

"A system as complex as the Internet can only be designed effectively if it is based on a core set of design principles, or tenets, that identify points in the architecture where there must be common understanding and agreement" [Cla2003]. The original Internet was built by people who shared a common goal of connecting their computing equipment and the group was small enough for social enforcement of behaviour in the net. The main guiding principle for the design of the Internet was the end-to-end principle [Sal1984], which has enabled a wide variety of un-foreseen applications to be deployed.

As the Internet grew, a number of problems in its architecture became apparent. Blumenthal et al. [Blu2001] identify a number of challenges for the end-to-end principle: operation in an untrustworthy Internet, more demanding applications, the rise of third party involvement, ISP service differentiation, and less sophisticated users.

The most remarkable feature of the Internet is its socio-economic complexity [Pap2001]. This means that solving *tussles* (i.e. conflicts of interest) [Cla2002] [Cla2003] [Cla2005] is one of the key problems for the future Internet. In this sense, the Internet has more resemblance to a society than to a traditional piece of technical engineering. As societies and the needs (and capabilities) of people within change, so does and should the Internet. This leads to design for change [Cla2003] and to the requirement of evolvability [Rat2005].

The ability to trust people (i.e. the ability to rely on the benevolence and good intentions of a typical person) is generally considered as a requisite for democracy and working markets [Put1993] [Fuk1995] [OEC2001]. With its resemblance to society at large, trust is also important in the Internet for many contexts [Cla2003] [Cla2007].

3.2 Protocols

Network protocols and specifications are a key factor in determining the functionality of a network infrastructure. In particular, aspects such as naming conventions, device addressing, routing strategies, and information dissemination through multicasting are chiefly dominant concepts which influence the operability of the future Internet. The following sub-sections cover the relevant leading work in these areas and their applicability within PSIRP.

3.2.1 Naming

In the current Internet architecture, “naming” usually means service-level naming (e.g. Domain Name Systems (DNS) names and namespaces rooted on DNS such as e-mail addresses, uniform resource locators (URLs), etc.).

The Domain Name System is a static distribution tree with a hierarchically organized namespace. The top-level domains are managed by name registrar companies contracted by the Internet Corporation for Assigned Names and Numbers (ICANN). Other domains are established under the top-level domains and maintain their own names. The name-domain hierarchy is independent of network-level administrative domains (i.e. autonomous systems (AS)), enabling multiple names, usually from different domains, to be mapped to the same IP address. This is widely used, for example, for hosting multiple web sites at a single web server.

Research in the Internet naming area includes designs for alternative name resolution systems [Ram2004a] and replacing the DNS namespace with self-certifying hash-based names [Kop2007], among other topics.

In [Ram2004b], a prefix-matching *distributed hash table* (DHT) (e.g. Pastry [Row2001], Tapestry [Zha2001], etc.) based alternative for the legacy Domain Name System is presented. The Cooperative Domain Name System (CoDoNS) enables fast dynamic updates and faster resolution of DNS queries. The updates are managed with a proactive replication scheme which does not require explicit recording of the replica locations, as the locations are algorithmically determined based on name-popularity rank and the desired average lookup latency of the system [Ram2004a]. The cooperative system is secured by relying on name certification through the DNS Security Extensions (DNSSec) [Are2005a] [Are2005b] [Are2005c].

For some uses the random distribution inherent in DHT-systems may be problematic, as some level of trust needs to be placed on each DHT participant. For this reason, it seems that the use of global DHTs *may* not be an optimal approach for building Internet-scale systems.

[Gan2004] defines a general method by which hierarchical DHTs can be formed using existing non-hierarchical DHTs. The created hierarchy has the isolation property by which the scope of the DHT at a certain level of the hierarchy is bounded to the domains under that level. This is a nice property for hierarchies where all domains in a sub-hierarchy are co-operative, but it seems that this design cannot be directly mapped to the Internet AS-domain hierarchy. This has been attempted by Routing over Flat Labels (ROFL) [Cae2006], but in that case, it is possible that traffic between certain domains traverses through a set of unwanted domains, which is contrary to current inter-domain policies.

In [Kop2007], a Data Oriented Network Architecture (DONA) is proposed that replaces the hierarchical DNS namespace with a cryptographic, self-certifying namespace. This enables totally distributed namespace control but offers no alternative for the current usage of human-readable DNS names in, e.g., print advertising or text messages. It also seems that protection against phishing, for example, would still require trusted third party certification. How would a user otherwise know to whom the cryptographic identifier belongs to?

The DONA namespace is intended for naming data, and not hosts or their network interfaces. It is also remarkable that the namespace proposed in DONA is not totally flat, as the names are composed of two parts: the principal's identifier and a label. This makes it possible to name data items that are not explicitly announced in the system. It would also be possible to develop the proposed DONA system so that the domains in the network could aggregate names at the principal level (at the potential cost of not always locating the nearest copy of a given data item).

[Cal2007] presents an approach where only *channels* (with unique identifiers) are named in the network. No central authority is required for naming and the architecture has no naming or addressing hierarchy. A channel is a logical means of transmitting packets from one location to another. The proposed architecture provides support for abstraction by aggregating channels into higher hierarchies. As identifiers are selected randomly, mobility is more easily supported.

In [Cro2003], the authors use the notion of *contexts* that are by definition collections of network elements that lie in a homogenous environment regarding naming services, addresses, packet format, routing solutions etc. Interstitial Functions (IFs) connect different contexts and perform the mappings between them. The authors argue for mappings between a variety of naming systems as a replacement to a single global namespace.

3.2.2 Addressing

The Internet addressing model [Hin2006] divides the total address space into variable size prefixes. The original addressing model defined three prefix classes of 8, 16, and 24 bits (classes A, B, and C, respectively). This made it necessary for each prefix to appear individually for all included host addresses within global routing tables. In 1993, the current classless inter-domain routing (CIDR) scheme was introduced to enable provider aggregated addressing and thus more compact inter-domain routing tables.

Interestingly, in the recent paper [And2007] a new addressing structure is proposed, which in a way returns to the original class-based model. Here the subnet prefix is replaced with a self-certifying *autonomous domain identifier* (AD), and the suffix (called the interface identifier in the IPv6 addressing architecture [Hin2006]) is replaced with a self-certifying *host identifier* (EID). IP addresses would then take the form *AD:EID*. The private keys bound to the AS and the EID are held by the domain and the host, respectively. This design is based on the finding that the inter-domain routing scales better if routing is done at the domain level, as there are less (autonomous) domains than there are advertised prefixes. Additionally, the self-certifying property would protect against the nowadays common Border Gateway Protocol (BGP) spoofing attacks (e.g. hijacking more specific prefixes). The EID portion is globally unique by itself, which would enable hosts to move or multi-home between domains and assure correspondents that they are communicating with the same host. This tactic is similar to that which is applied by the Host Identity Protocol (HIP) [Mos2006].

ROFL [Cae2006] proposes a radical alternative to the current IP addressing model: the dichotomy between topologically significant addresses and end-point identifiers is solved by routing on totally topology-independent, i.e. flat, labels. ROFL applies hierarchical DHT overlay solutions [Gan2004] on the network layer without the underlying IP. The paper concludes that while it really does not scale, routing on flat labels cannot be dismissed as impossible. This presents the question of whether or not flat label-based routing might scale for service names, where real-time requirements are not as strict as those of routing protocols used for packet forwarding.

In the architecture presented in [Cal2007], nodes are anonymous and are addressed through their incoming channels. Furthermore, there exists a service in the network that maps end-channels to a border channel of the targeted endpoint's domain (i.e. its realm). Routing is then performed with a forwarding directive containing a loose-source path of channel identifiers that is filled recursively any time it is needed (when a packet travels through a domain whose topology is hidden from the source).

In [Cro2003], specific addresses are bound to different addresses in the different contexts. With this mechanism, devices (e.g. sensors) can reach the Internet without implementing an IP stack by binding an address in their local context to a gateway node. The main message of this work is that our architecture has to be prepared for largely heterogeneous networks.

The current Internet communication model enables all Internet connected hosts to send packets to all public IP addresses. This reachability is a significant enabling mechanism for attacks against Internet hosts. In [Han2004], Handley and Greenhalgh present seven steps towards an Internet architecture that is resistant against denial-of-service (DoS) attacks. The first two steps call for separation of client and server addresses and removal of globally reachable client addresses altogether. They also suggest domain-level routing for clients, but with paths encoded into the packets themselves as the client request for the server traverses the inter-domain links. We'll return to this topic in the next section.

3.2.3 Routing

The current inter-domain routing protocol of the Internet (i.e. BGP [Lou1989]) is facing increasing scaling problems. The main reason for this is the robustness-minded design, where each inter-domain router in the global network needs to know how to forward IP packets to all valid destinations. This requires internet-wide route updates whenever something changes for a globally visible prefix. The number of globally visible prefixes is increasing for reasons such as provider independent addressing, site multi-homing, protection against prefix hijacking etc.

3.2.3.1 Domain-level Routing

One way to address the BGP scaling issues is to route at the domain level as discussed earlier in the context of [And2007]. Many recent research proposals push this concept further by also proposing the removal of path selection from the packet forwarding-level routing function [Gri2001] [Yan2007]. Explicit domain-level path construction also fits well with name-based routing, as outlined by the *Translating Relaying Internet Architecture integrating Active Directories* (TRIAD) project [Gri2001] and DONA [Kop2007]. Here, the server IP addresses also become redundant if the domain-level path is extended with intra-domain addresses for the server (as well as for the client). A part of the TRIAD project known as the Wide Area Relay Protocol (WRAP) [Gri2001] provides an encapsulation protocol by which explicit path-based forwarding can be performed on top of the current Internet. The WRAP header contains the (loose) source route. Domain-level source routing is achieved when the WRAP gateways are placed in domain border gateways. The New Inter-domain Routing Architecture (NIRA) [Yan2007] encodes the domain-level path to source and destination IPv6 addresses.

[Lak2006] proposes providing the path selection function as a separate routing service. Other proposals [Key2006] allow the path selection to be optimized by the sending host based upon congestion information. This can allow the spread of traffic load over the network and improves resilience. NIRA [Yan2007] proposes running a separate path discovery protocol for the up-graph, using a Name-to-Route Lookup Service (NRLS) for the downhill (destination) route, and allowing the endpoints to further negotiate end-to-end path selection. Some aspects of these functionalities are already needed by multi-path capable transport protocols, such as the Stream Control Transmission Protocol (SCTP) [Ste2000]. Furthermore, [Fea2004] proposes removing the routing function from routers altogether to allow for better domain-level control of routing policies and allow a more direct domain-level mechanism for inter-domain routing.

Another interesting feature of ROFL [Cae2006] is that it uses domain-level source routes as the means to route packets between endpoints. The first packet of a communication session takes the penalty of hierarchical Distributed Hash Table (DHT) routing, but after that the endpoints have the option to perform NIRA-like [Yan2007] end-to-end domain-level path control that enables to reduce the stretch for the remaining packets to 1.

3.2.3.2 Compact Routing

Current BGP routing has weak scaling properties when it has to follow the current growth of the global network. Both the routing table sizes and the communication cost are increasing exponentially [Kri2007]. Theoretically, routing on AS numbers instead of prefixes doesn't seem to solve the problem, as it offers only a constant reduction and cannot modify the scaling behaviour. Compact routing aims to decrease the size of the routing tables, while it allows non-shortest paths to be used. It has been proved that traditional link-state and distance-vector algorithms (that find the shortest path) have routing tables with the size of O[n*log(n)] [Gav1996]. Besides routing table sizes, an important factor describing routing algorithms is the *stretch* they produce (i.e. the worst-case ratio of the path they compute vs. the corresponding shortest path). This means that traditional distance vector and link state algorithms are stretch-1 algorithms.

By definition, a routing scheme is compact if it produces logarithmic address and header sizes, sub-linear routing table sizes and a (multiplicative) stretch bounded by a constant. If the scheme works correctly only on some specific graph classes, it is a *specialized* scheme. If the scheme works correctly and satisfies the scaling bounds on all graphs, it is called a *universal* scheme.

It is easy to achieve stretch-1 routing in grids, where each node is named with its (x,y) coordinate and the message is always forwarded to the neighbour closest to the destination. Here the routing table size scales logarithmically as it depends only on the number of bits needed to write down the name of the nodes. Also, there is a simple compact routing algorithm for binary trees that produces stretch-1. In this case the nodes are named according to their positions in the depth-first-search. Moreover, in [Tho2001] a stretch-1 compact routing scheme is presented for arbitrary trees.

Another important classification of the compact routing schemes is distinguishing *name-dependent* and *name-independent* schemes. Name-dependent schemes exclude the use of arbitrary addresses, as the name (or label) of the nodes contain some topological information. The simple algorithms introduced above for trees and grids are both name-dependent schemes. On the contrary, name-independent schemes can operate on flat labels, which seems desirable for the future Internet.

Two compact routing schemes that have minimal stretches are the *Cowen scheme* [Cow1999] and the *Thorup-Zwick (TZ) scheme* [Tho2001]. They are both non-hierarchical stretch-3 algorithms and the TZ scheme is the improvement of the Cowen scheme. In these name-dependent compact routing schemes, a *landmark* set is defined (the choice of the landmark set is different for the different schemes). In the Cowen scheme the set is based on dominating set construction, while the Thorup-Zwick scheme use a randomized technique, and the size of the landmark set is also different. The other parts of the algorithms are basically the same operations. Besides the landmark set, there is another important set of nodes in the schemes. By definition, for every node v, the *cluster* is the set of the nodes that are closer to v than their closest landmark node. Then each node in the network gets a new label (name/identifier) that consists of three parts for each node v. The first part is the original identifier v; the second part is identifier of the landmark node that is the closest from v: L(v); while the third part of the label is the identifier of the interface at L(v) that lies on the shortest path from L(v) to v. The routing table at node n will contain entries for the shortest paths to all landmark nodes and the nodes in its cluster. The forwarding of the messages is based on the information the label tells in the header and on the routing tables in the nodes. The TZ name-dependent scheme thus has a routing table size of:

$$O[n^{1/2}(\log n)^{1/2}]$$

while the Cowen-scheme has a routing table size of:

$$O[n^{2/3}(\log n)^{4/3}]$$

which is eventually determined by the size of the landmark set.

Name-independent compact routing schemes assume that the nodes are named arbitrarily. Arias et. al. [Ari2003] present a name-independent compact routing scheme with:

$$O[n^{1/2}(\log n)^{1/2}]$$

routing table size and a stretch of 5. However, earlier it was proven that the minimal stretch for compact routing schemes is 3 [Gav2001]. The authors in [Abr2005] improved the results of Arias et. al and presented the first optimal name-independent compact routing scheme. The algorithm assigns a colour to each node by a special hash function. A special colour is designated to be a landmark colour. A node has an entry in its routing table for all neighbours, for all landmark nodes and for all the nodes having the same colour. If a packet arrives which is not in the routing table then it will be forwarded to the closest neighbour with the same colour (it is ensured that every node's neighbourhood contains one node with each colour).

A relevant work [Kri2004] focuses on the characteristics of the TZ-scheme in Internet-like graphs. They provide both analytical and simulation results and find that the average (multiplicative) stretch is around 1.1, while the memory needed is much less than the theoretical upper bound (~50 routing table entries in a cc. 10000 node AS-topology, while the worst-case is around 2200). The *"BC" scheme* [Bra2006] is an algorithm utilizing some shortest path trees in the network. The authors experimentally demonstrated that their algorithm has only a small additive stretch on power-law random graphs. A hybrid scheme [Bra2006] that runs the TZ and BC scheme parallel outperforms both algorithms in Internet-like graphs in terms of average routing stretch. Additionally, in [Kri2007] simulation results showed that all three algorithms produce lower average routing stretch values than the name-independent scheme in [Abr2005]. All the abovementioned algorithms require that all nodes have a complete view on the topology at any time in the network. If the topology is changing, update messages are needed to refresh the view of the nodes. Communication cost is defined as the number of update messages needed after a change in the topology. In [Kri2007] the authors showed that the routing schemes on scale-free graphs cannot scale slower than linearly regarding the communication cost, moreover, Internet-like graphs has higher lower bounds than general graphs in terms of the communication cost, although it is still better than the exponential cost.

Compact routing is an area that is worth considering and evaluating if we aim to create a scalable architecture during the project. However, it is not clear how policy relations can be considered in compact routing schemes, and how the full view on topology could be avoided.

3.2.3.3 Overlay Routing

This short review of overlay routing focuses on the solutions that can be exploited in PSIRP.

DHT techniques can be utilized in several parts of the PSIRP architecture. For example, rendezvous can be implemented either in a hierarchical or non-hierarchical manner. If the latter option is chosen, DHTs, with some modifications, are good candidates to distribute the rendezvous functionality among rendezvous nodes. *Event routing*, which is done on a content-based overlay network, forms the basis of the content-based publish/subscribe mechanisms. Their relevance to PSIRP is clear as they offer the ability to the receivers to signal what they want to receive; however they rely on the underlying IP layer.

In overlay routing, a logical topology is formed over the underlying network. A link between two overlay nodes may take several physical hops in the underlying network. Overlay networks usually offer more functionality than just routing (e.g. lookup service, application-level multicast etc.).

DHTs are good examples of overlay routing. They provide a lookup and resource location service and are used e.g. in P2P systems. Some relevant DHT-based solutions are the Content Addressable Network (CAN) [Rat2001a], Chord [Sto2001], Pastry [Row2001] and Tapestry [Zha2001]. They are all structured DHTs, meaning that the nodes form a strict logical topology.

CAN is based on a d-dimensional Cartesian-space and each node has a coordinate zone that it is responsible for. Each node knows its neighbours in the logical topology (by storing their IP addresses and their coordinates). Packets (lookup messages) sent to a coordinate are delivered to the node that is responsible for the coordinate's zone. Each node forwards the packet to the neighbour that is the closest to the destination (it can determine the closest by checking the coordinates). One example is shown on Figure 3.1 when a message is routed from Node A to Node B. Application-level multicast can also be implemented utilizing CAN

[Rat2001b], which has the attractive feature that only the participating nodes store group-specific states, which is achieved by forming mini-CANs for each multicast group. This feature can be exploited also in PSIRP: if state-explosion is foreseen in some network elements (core routers) it is worth considering distributing states among the participating entities (subscribers in our architecture).

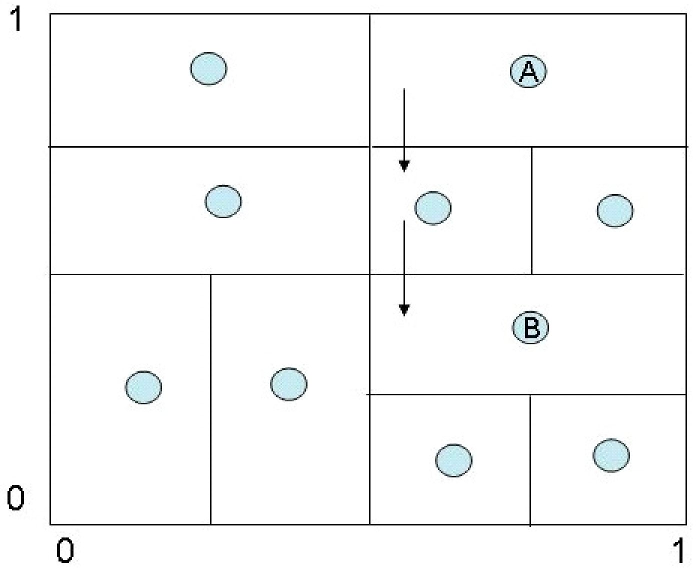

Figure 3.1 – The nodes and their zones; a route from A to B is shown

In Chord, the participating nodes have unique identifiers and form a one-dimensional ring. In the basic solution, each node maintains a pointer to its successor and predecessor node (determined by their ids), but as an optimization they maintain additional pointers (*fingers*) to other nodes in the network. A Chord ring and pointers of one node are presented in Figure 3.2. A message is forwarded in a greedy fashion: the node holding the message forwards it to the node in its finger table with the highest id value not greater than the id of the destination node.

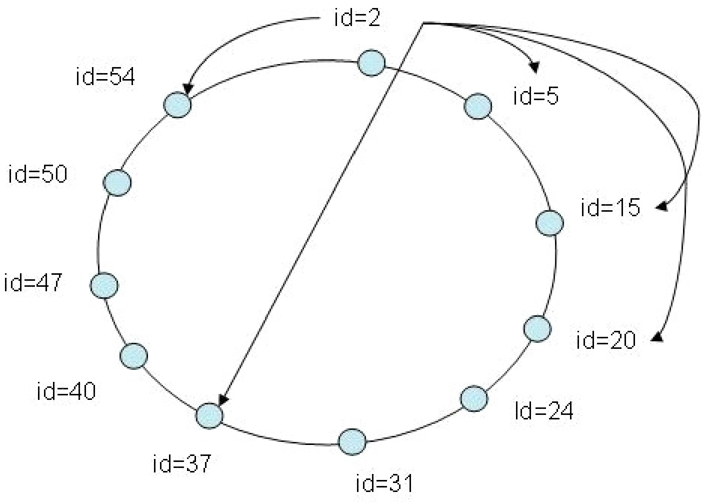

Figure 3.2 – A Chord ring and the pointers of the node with ID = 2

Pastry and Tapestry are tree-based solutions. Again, the nodes have unique identifiers, and at each step the message is routed to a node whose identifier corresponds to the destination identifier in one more digit (Pastry starts from the prefix, while Tapestry from the suffix). One example overlay route in a Pastry system from node 3254 to 7326 (assuming hexadecimal digits) is shown in Figure 3.3.

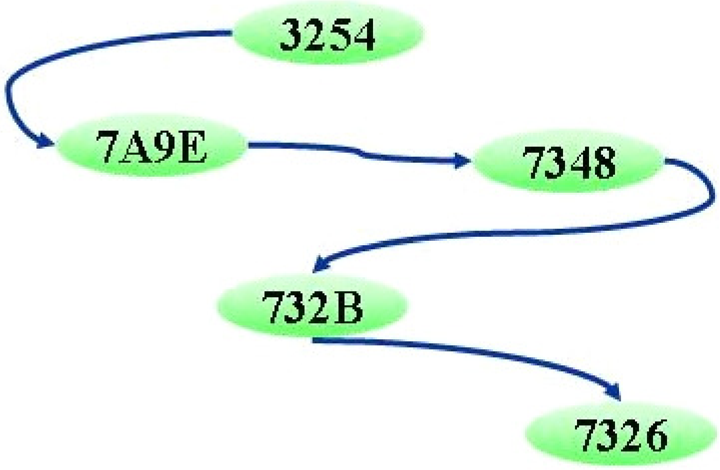

Figure 3.3 – A route from 3254 to 7326 in the Pastry DHT

3.2.3.4 Content-based Publish/Subscribe Routing

Content-based publish/subscribe routing is an important piece of prior work for PSIRP and a major contribution to the publish/subscribe research.

In *content-based publish/subscribe* hosts subscribe to content by specifying filters on the events. In content-based and data-centric routing, the data of messages defines their ultimate destination. Information subscribers use an interest registration facility provided by the network to set up and tear down data delivery paths. In this area, pub/sub has been proposed as a replacement for TCP/IP [Bri2004], but the idea has been only presented in a rudimentary level and, to our knowledge, has never been realised. Such a paradigm shift changes the economic model of the network considerably, from sender oriented communications to receiver oriented.

As an example, an event can be at a stocking system the announcement of a price of a stock: Price = 800. A subscription can be a filter on the price, e.g. 200 < Price < 1000, i.e. the user is interested in the stocks that are cheaper than $1000 but not cheaper than $200. The two different approaches to content-based event routing are filter-based solutions and multicast-based solutions.

In *filter-based event routing*, the pub/sub servers are organized into an acyclic tree topology. When a client subscribes, its subscription is flooded along the tree topology, so each server can build a routing table that contains the directions for each subscription received. When an event is received, a matching action is performed in the routing table, then it is forwarded hop-by-hop to the clients interested. For example this approach is used in Siena [Car1998].

In *multicast-based event routing* the event space is partitioned into a number of multicast groups, and a multicast tree is built for every group that spans all the servers interested in (having subscriptions to) any event in the group. When an event is published, it is mapped to a group, and forwarded along the specific multicast tree. This approach can be utilized in PSIRP as well without underlying IP, when we assume that the operator maintains some forwarding trees in the network domain and the rendezvous functionality match the publications into the best tree covering the subscribers and has minimal number of uninterested routers.

Kyra [Cao2004] is an approach that combines the latter two approaches. A two-level hierarchy is built: the pub/sub servers are partitioned into cliques based on network-proximity. Here all nodes know each other. On the higher level, routing trees (minimum spanning trees) for different partition of events are built to connect the different cliques. In a clique, each server is assigned to a partition of a content space and is designated to be the proxy server for that set of subscriptions. When a server gets a subscription, it forwards it to the appropriate server in its clique. The servers will be part of those trees whose zones are overlapped with the server's zone. When an event is received, it is forwarded along the appropriate routing tree and finally it will be the proxy server's responsibility to forward it to the appropriate servers in the local clique.

The Siena system can be considered to be a classic example of a distributed content-based routing system that was implemented in the application layer [Car2001]. Siena is envisaged to integrate at the network service level, coexisting, for example, with TCP/IP instead of working above the network level. This would eliminate an extra protocol layer, and provide greater efficiency in routing and forwarding.

A number of overlay-based routing algorithms and router configurations have been proposed for distributed pub/sub. An application-layer overlay network is implemented on top of the network layer, and typically overlays provide useful features such as fast deployment time, resilience, and fault tolerance. An overlay routing algorithm leverages underlying packet-routing facilities and provides additional services on the higher level, such as searching, storage, and synchronization services.

Overlay networks allow the introduction of more complex networking functionality on top of the basic IP routing functionality. For example, filter-based routing [Cao2004], onion routing [Din2004], DHTs, and trigger-based forwarding [Sto2002] are examples of new kinds of communication paradigms.

Good overlay routing configuration follows the network level placement of routers. Many DHTs work by hashing data to routers, or brokers, and using a variant of prefix routing to find the proper data broker for a given data item. The i3 [Sto2002] is a DHT-based overlay network that aims to provide a more flexible communication model than the current IP addressing The Delegation-Oriented Architecture (DOA) was proposed to circumvent the harmful side-effects of middle-boxes [Wal2004]. Recent proposals, such as DONA [Kop2007], aim to introduce data-centric operations to the networking architecture. DONA inserts a data-handling shim layer right above the network layer and resolves names by directly routing to data.

Math Early with DYnamic Multicast (MEDYM) partitions the event space into non-overlapping partitions with balanced load. In this architecture, each server acts as a matcher for one or more partitions [Cao2005]. A channelization technique is presented in [Ria2002] that partitions the event space into a number of multicast groups. A multicast tree is built for each group that spans servers with subscriptions for any event in that group.

Hermes and Scribe are examples of pub/sub systems implemented on top of an overlay network and are based on the rendezvous point routing model [Tar2006a]. The Hermes routing model is based on advertisement semantics and an overlay topology with rendezvous points. Typical fixed-network pub/sub routing algorithms are deterministic in nature. Basic routing algorithms do not cope with topology changes, and dynamic connections have been investigated only recently.

In the currently deployed systems, under the pub/sub and/or overlay routing substrates there lies always a "classic", IP-based routing and forwarding mechanism, which has been relatively stable since the introduction of Classless Intern-Domain Routing (CIDR) [Ful1993] in 1993. More recently, this system has started to show indications of severe stress [Wit2001] [Cha2002] [Mey2007], leading to a resurge of research on the area. For example, the

Cooperative Associated for Internet Data Analysis (CAIDA)'s NeTS-NR project [CAI2008] aims towards construction of practically acceptable next generation routing protocols based on mathematically rigorous routing algorithms, based on recent results in the area of compact routing [Kri2004]. For PSIRP, however, the areas of multicast routing [Dee1991] [Pau2002] and Byzantine routing [Per1998] may turn out to be even more important. There remains the challenge of combining the recent results [Avr2004] [Awe2005] [Gui2005] into a whole that serves the PSIRP purposes of providing a secure and scalable pub/sub substrate.

3.2.4 Multicast

Multicasting is a data delivery method whereby data is delivered to a group of receivers. While the same effect can be achieved by multiple unicasts, multicast differs from unicast in two ways: first, the receivers are referred to via a common identifier, thus decoupling the sender from the receivers and, second, receivers along the same route from the sender can be served by a single data transmission, thus conserving network resources.

In general, multicast is implemented by combining local and global mechanisms. The local mechanisms operate within a single local area network which may provide native support for multicast (e.g. shared Ethernet) while the global mechanisms operate in the wide area, between local area networks. Since wide area networks do not provide native multicast support, multicast is implemented by constructing distribution trees from a root node towards all receivers.

The construction of an optimal multicast tree with respect to any single link metric is equivalent to the *Steiner tree problem*, which is known to be NP complete [Hak1971]. The Steiner tree problem is similar to the *Minimum Spanning Tree problem*, but instead of reaching all nodes, the Steiner tree only reaches a specific group of nodes. Practical multicast trees are therefore normally constructed by combining the unicast shortest paths, provided by an underlying unicast routing protocol, between a root node and all receivers. It should be noted that such trees can only minimize pair-wise metrics such as the delay to each receiver, not global metrics such as the cost to reach all receivers.

Practical multicast trees can be classified into two categories, *source-based trees* and *shared trees*. A source-based tree is rooted at the router serving the source. Therefore, when source-based trees are used, the source injects multicast traffic directly into the tree. With this approach a separate tree needs to be constructed for each source. Each one of these trees will be optimal (with respect to whatever metric we are interested in), but it will require its own forwarding state in each router.

A shared multicast tree on the other hand is rooted at a prearranged node, called the *Core* or *Rendezvous Point* (RP) of the tree. Therefore, when shared trees are used, each source first sends its data to the RP, essentially in unicast mode, which then forwards it in multicast mode to all receivers. Shared trees are only optimal from the viewpoint of the RP and may be quite suboptimal with respect to the sources, but they only require a single piece of forwarding state in each router.

3.2.4.1 IP Multicast

The classical IP multicast model identifies each multicast group by a class D IP address. Receivers join this address in order to receive data sent by anyone to the group [Che1985]. This model is sometimes referred to as *Any Source Multicast* (ASM), in contrast to the *Source Specific Multicast* (SSM) where groups are identified both by a source and a group address, thus allowing only the indicated source to transmit to the group [Hol2006]. It should be noted that the *Multimedia Broadcast / Multicast Service* (MBMS) standardized for third generation cellular networks also implicitly adopts the SSM model, since only a designated network node can transmit data to each group [Xyl2008].

The original multicast routing protocol of the Internet was the *Distance Vector Multicast Routing Protocol* (DVMRP). In the original version of DVMRP, each router receiving a multicast datagram would forward it to all routers except the one it arrived through, but only if it arrived via the shortest path to the sender, essentially using a source specific tree composed of the shortest paths from all routers to the sender (i.e. the reverse shortest paths) calculated via a distance vector protocol. The latest version of DVMRP allows routers to be pruned of these trees if they are not serving any group members [Wai1998]. Another proposal based on source specific trees is the *Multicast Open Shortest Path First* (MOSPF) protocol [Moy1994], where a link state protocol is used to calculate the shortest path tree from each source router towards all routers serving group members. The original proposal for using shared trees was the *Core Based Trees* (CBT) protocol [Bal1993], where a Core is arbitrarily chosen as the root of the distribution tree and routers serving group members construct the tree by sending join messages towards the core. Essentially this is again a reverse shortest path tree as in DVMRP, but rooted at the Core rather than each individual sender.

All of these approaches are unified under the umbrella of *Protocol Independent Multicast* (PIM), which supports two modes, the Sparse Mode (PIM-SM) [Fen2006] and the Dense Mode (PIM-DM) [Ada2005]. There also exists a less widely used PIM Bi-directional multicast protocol. PIM-SM assumes that the receivers of multicast traffic are sparsely distributed throughout the network, hence most local area networks will not want to receive multicast packets. By default, PIM-SM uses shared multicast distribution trees which are rooted at RPs. The actual traffic is encapsulated into PIM control messages and sent by unicast to the nearest RP by a Designated Router (DR), located in the source's local network. The RP then forwards the traffic to the receivers via the shared tree. PIM-SM allows receivers to switch their trees to shortest path trees upon request. PIM-DM, on the other hand, assumes that most local networks will want to receive multicast traffic, therefore it uses only source based trees, obviating the need for RPs; essentially PIM-DM is the same as DVMRP but without any reliance to the underlying unicast routing protocol.

Within a local area network, the *Internet Group Management* (IGMP) protocol is used by the receivers to indicate to their router their interest in a specific group; the latest version of IGMP allows receivers to indicate their interest in specific (source, group) pairs so as to also support source specific multicast [Cai2002]. Based on this information, the local router will receive multicast packets via an appropriate multicast protocol and forward them to the local network in a network specific manner, either as native link layer multicast (Ethernet, WLAN, and other broadcast based networks) or as individual unicast packets (non-broadcast networks). It should be noted that in switched (wired) Ethernet, multicast is essentially equivalent to broadcast, since switches are generally unaware of group membership.

3.2.4.2 Multicast Challenges and Issues

While multicasting is considered valuable, it is not universally supported over the Internet. Many explanations for this are proposed by [Dio2000] and [Qui2001], including security and scalability issues. Although the traditional security issues raised by unicast, such as data confidentiality and integrity, are also valid for multicast, they are harder to address in the multicast context. For example, secure group communication can be provided by using independent end-to-end secure unicast channels between all pairs of participants, albeit by negating the link sharing advantages of multicast. In the host-group model adopted by the Internet, group membership is unknown to the sender; therefore, it is impossible to setup security associations between the sender and the receivers without tracking additional information.

Another security issue is raised by the fact the Any Source Multicast model is problematic for many media distribution applications. The Source Specific Multicast model prevents unauthorized senders from sending data to the group, and it may also be used to allow the sender to control group membership, albeit by partially sacrificing the decoupling between the sender and group members.

Regarding scalability, one problem with many multicast protocols is the use of broadcast (flooding) during the construction of the multicast tree to determine which routers should be part of the tree; DVMRP and PIM-DM belong in this category, as they initially flood the network, requiring uninterested routers to explicitly prune themselves off the multicast distribution tree. CBT and PIM-SM only construct trees towards routers that have explicitly stated their interest, therefore they are more sensible approaches for the wide area.

Another scalability issue relates to the amount of forwarding state required at each multicast router. Even if a shared tree is used for delivery, separate entries are needed at each router on the distribution tree for every multicast group. If shortest path trees are used, the number of forwarding entries must be multiplied by the number of senders to the group. In unicast routing, this problem is solved by aggregating the forwarding state based on the fact that networks with similar unicast IP addresses are usually geographically close. Unfortunately, multicast groups may have members everywhere on the Internet, therefore aggregation is not generally possible. Various other aggregation methods have been proposed, as described in [Zha2003].

Finally, an important scalability problem with multicast in general is providing feedback to the sender from a potentially huge number of receivers. Such feedback may be required in order to provide error, flow and congestion control. The problem is that each receiver may provide completely different feedback to the sender, thus making feedback aggregation hard or even impossible. For example, some receivers may be experiencing severe congestion, while others may experience perfect conditions. As there is no evident solution to this problem, several approaches exist emphasizing different goals (see [Pas1998] for a summary). Therefore, they are often left for the transport layer so as to allow each application to select an appropriate set of tools.

3.2.4.3 Recent Trends in Multicast

Since support for IP multicast on the Internet remains sketchy, many researches have proposed alternative methods of supporting multicast that are either more scalable or easier to deploy than the approaches mentioned above. We can split these approaches in three rough categories: a) those relying on router support for multicast, b) those relying exclusively on end-host support for multicast, and c) those relying on overlay (i.e. DHT) support.

Router-based Approaches

A service-centric multicast architecture is discussed in [Yan2008], its main idea is to construct and maintain the multicast tree in a more centralized manner compared to most other methods. Such an approach significantly reduces bandwidth consumption, since it avoids using broadcast traffic during the construction of a multicast tree. This approach uses separate master routers (m-routers), which have the information about the global network topology and are responsible for managing the whole domain (i.e., an ISP could use a m-router to manage its whole network). The m-router builds a shared multicast tree for the domain which is rooted at the m-router itself. It is assumed that all other routers in the domain know the IP address of the local m-router. Simulation results show that such an approach achieves the better performance and the lower overhead compared to existing multicast solutions.

The *Data-In-Network loop* (DINloop) is a multicast scheme utilizing Multiprotocol Label Switching (MPLS) proposed in [Guo2005]. It aims to improve the inter-domain scalability of multicast by forming a DINloop using special DIN Nodes which reside in the core network. This loop utilizes MPLS and inter-domain multicast traffic is forwarded along it. The advantage of this approach is that there is no need to construct a separate multicast tree for each multicast session since multiple sessions can share a single DINloop. This approach uses two labels to route packets within DINloop. The top label is the same for all multicast traffic while the bottom label corresponds to the destination address and it differentiates multicast messages. Each DIN Node knows all the receivers of the multicast traffic in its domain and when the DIN Node receives a multicast packet, it examines the lower label to determine if its domain contains receivers of that multicast packet. If it does, then DIN Node copies the packet and sends it in its domain. In all cases the DIN Node also forwards the packet along the DINloop. The scalability of DINloop is significantly improved compared to traditional approaches, since the size of the routing tables in core routers does not grow linearly as the number of multicast groups increases. The downside is the additional delay introduced by DINloop. The average latency of multicast traffic is about twice as high compared to traditional tree-based multicast approaches. A different take on the idea of using the same tree for multiple groups appears in the *Bi-directional Aggregate* Multicast (BEAM) protocol [Cui2003] where one of many pre-existing trees is selected for use by each multicast group depending, so as to economize on multicast routing state.

Free Riding Multicast (FRM) [Rat2006] is an inter-domain multicast approach which separates multicast membership discovery from route discovery. Its aim is to use existing unicast links for inter-domain multicast traffic. FRM uses an extended version of BGP for multicast group membership advertisement while forwarding of multicast traffic is done as follows. As the packet arrives to the border router of the source domain, the router constructs an AS-level multicast tree based on group membership information. The packet is sent to neighbouring domains together with the constructed multicast tree. Therefore, the border routers in other domains can forward the packet simply based on the attached tree information; they do not need to perform multicast tree construction again. Since FRM is designed only for inter-domain multicast, a multicast protocol must be used together with FRM to handle intra-domain multicast. This does not affect scalability since traditional multicast protocols scale well within a single domain. As an advantage, FRM offers good inter-domain multicast scalability since it does not require distributed construction of multicast trees over the whole network. FRM is also a relatively simple protocol compared to other approaches because it does not use rendezvous points. However, FRM requires that border routers have more computational resources such as memory and processing power.

Host-based Approaches

There are numerous approaches to multicast relying exclusively on functionality at the end hosts participating in a multicast group. In all approaches the end hosts essentially form a multicast routing overlay based on underlying unicast routing functionality and then use this overlay in order to distribute multicast traffic. A simple example is *Narada* [Chu2002], where each newcomer to a multicast group initially gets a list of other group members via an out-of-band mechanism. The new member then randomly creates mesh (overlay) links to some of these members. Periodically the mesh is adjusted by adding and dropping links so as to improve the overlay paths and heal any partitions. Group members run a distance vector algorithm over the mesh in order to calculate a shortest path multicast tree to all other group members. While Narada requires no support from the network, the multicast trees that it constructs can be quite suboptimal and each group member must be aware of all other members.

A more complex example is the *NICE system* [Ban2002] where the group members are organized in a hierarchy, consisting solely of end hosts. Unlike Narada, in NICE each end host only needs to maintain full state about its neighbours in the hierarchy, with limited state about other group members, thus allowing the scheme to operate with much larger multicast groups. A different approach to reducing state management requirements is the *Application Level Multicast Infrastructure* (ALMI) [Pen2001] where one of the end hosts (or a separate server) is responsible for the entire group, handling group management and recalculating the multicast distribution tree, thus allowing the other end hosts to only maintain state regarding their neighbours in the topology. Obviously, in ALMI the responsible host is a centralized component limiting the scheme's scalability. With both NICE and ALMI, the trees can be as suboptimal as in Narada.

A solution residing somewhere between the router and end host based approaches is *Small Group Multicast* (SGM) [Boi2000], where each end host participating in a group is aware of all other group members. In SGM each multicast message incorporates the addresses of all group members, thus allowing intermediate SGM enabled routers to replicate each message on the way to the receivers: each router determines the next hop on the shortest path to each receiver and then forwards over each outgoing link a copy of the packet that only lists the receivers for whom the shortest path begins with that outgoing link. The advantage of SGM is that the distribution tree is optimal with respect to the unicast routing metric; however, each group member needs to continuously track all other group members, thus limiting the schemes scalability.

Overlay (DHT)-based Approaches

One of the main problem with other host based multicast solutions is that each group member needs to maintain state about many, if not all, of the other group members, in order to construct and maintain the multicast distribution trees. Distributed hash tables can be used to distribute this state among the participants. The multicast group identifier is mapped to a specific DHT-node, which is then used as the rendezvous node for the group.

There are two approaches in which a DHT can be used for multicast distribution. The first, exemplified by Scribe [Cas2002b] and Hermes [Pie2002], is to share the same DHT for all groups, but create a separate multicast distribution tree within the single DHT for each group. In this approach, group members send a join message towards the rendezvous node using the DHT; as these messages are propagated towards the rendezvous node, reverse path forwarding state is created in the intermediate DHT nodes, essentially forming a shared tree over the DHT rooted at the RP. It is notable that the propagation of the join messages can be stopped when a node having state for the group is reached, i.e. the rendezvous node need not see all the join messages. Senders can then forward their multicast traffic towards the rendezvous node, again using the DHT, so that it may then be forwarded to all group members. The second approach, exemplified by CAN-multicast [Rat2001b], is to create a separate overlay per group. In this approach, group members first identify the RP via a global DHT and then create a separate mini DHT consisting only of group members. Messages to the group are then flooded over the mini DHT.

While both these approaches require additional state per group, in the overlay per group approach only group members participate in the routing process; this however means that the tree per group approach may provide better routing performance by also exploiting non group members. It should be noted that while the tree per group approach was originally proposed for the Pastry DHT and the overlay per group approach was originally proposed for the CAN DHT, they are actually independent of the DHT in use. A comparison of the two approaches in terms of performance is provided in [Cas2003]; from this study it seems that the tree per group approach is preferable and that it is better to construct these trees via Pastry rather than via CAN.

The advantage of the DHT-based multicast is that the routing overlay is created and maintained by a separate mechanism that may also serve other needs; however, the routing paths used may be quite suboptimal since DHTs operate in a virtual network topology that may be quite different than the underlying network topology. These schemes are however of potentially great relevance to PSIRP due to their reliance on identifier based routing, which is one of the main premises of PSIRP.

3.3 Mechanisms

Here we delve into certain “operational tactics” whose importance is often underrated in the state of the current Internet. These features will play a key role within PSIRP, and their interplay is fundamental for the data-centric design of the PSIRP system.

3.3.1 Compensation

Fundamentally, the purpose of compensation is to facilitate efficient resource use through providing the resource "owner" some assurance that they will eventually benefit from consumption of "their" resources. That is, if Alice has invested in some resources (such as network capacity or caching space) and Bob would benefit from being able to use those resources, without any kind of compensation Alice would not have any incentive for allowing Bob to use those resources. With some sort of compensation, Alice is led to believe that she will gain, eventually and somehow, from Bob's consumption of those resources, thereby creating an incentive for her to allow the consumption. Bob being able to use those resources, in turn, leads potentially both to increased efficiency and increased wealth.

Given this definition, there are a number of very different forms of compensation, including the following:

- Authorisation: For example, a company having invested in resources may find it sufficient to believe that its employees will consume those resources in order to benefit the company. This belief may be sufficient ground to take a simple authorisation decision, based on whether the user is an employee or not, to function as a means of compensation.
- Community membership: Based on strong human reciprocity [Gin2000] [Feh2002], it may be sufficient that the resource provider believes the resource user to be a member of a certain (loose) community.
- Resource exchange or barter: In some cases, the resource user may have at their disposal some other resource that the resource provider can use immediately, thereby leading to compensation by barter exchange.
- Sacrifice or evidence of deliberate waste of users' resources: In the so called puzzle-based mechanisms, the resource owner gains assurance that the forthcoming resource user has sacrificed or deliberately wasted some of their own resources, in order to show their "honesty.” While these systems do not compensate in the strict sense of the word, for the consumption of the resources, they partially serve a similar kind of function in certain settings; see below.
- Payment or promise of future reimbursement: These systems include the traditional formal currency-based systems (i.e. traditional money) as well as systems based on community currencies, e.g. [Tur2004].

Structurally and architecturally, the available compensation systems are limited by the relative values or costs of the resource in question and the components of the transaction. When the relative value of the resource is high compared to the transaction costs, we can more-or-less rely on our informal understanding of exchange and base the compensation, primarily, on authorisation or payment. However, whenever the value of the resource and the transaction costs are of the same level, and especially in the case where the transaction costs may be higher than the value of the resource itself, more careful analysis is needed, especially if the system allows one to gain by harvesting lots of low-value resources.

For the purpose of such an analysis, we have to make a distinction between at least the following types of transaction-related costs:

- The immediate technical costs related to creating the desired level of assurance, including storage, computation, and communication.
- The informational search costs related to the process of gaining assurance, including searching for credentials, etc.
- The collateral costs associated with the consumption of the resource (such as communication costs related to utilising remote storage).

This is in stark contrast to the traditional Transaction Cost Economics (TCE), where the transaction costs are considered to consist of the following components:

- *Researching potential suppliers*
- *Collecting information on prices*
- *Negotiating contracts*
- *Monitoring the supplier's output*
- *Legal costs incurred should the supplier breach contractual negotiations*

From another point of view, the list can be contrasted with the micropayment transaction cost analysis by Papaefstathiou and Manifavas [Pap2004], where they make a distinction between fixed technical costs, storage costs, computational costs, communication costs, administrative costs, cost of non-availability, and publishing costs. From our point of view, this list is particularly problematic since usually the resources that we deal with include just these, e.g. storage, computation etc,, thereby creating circular dependencies.

Following Weber [Web1978], Biggard and Delbridge [Big2004] define the term exchange to refer to a "voluntary agreement involving the offer of any sort of present, continuing, or future utility in exchange for utilities of any sort offered in return" and that may involve money, goods, or services.

Based on the structures of social relationships and the type of rationality (partially founded on values), they divide the systems of exchange into four categories: price-based, associative, moral, and communal systems.

- *Price System:* In the categorisation, the classical neo-classist price-based exchange systems depict settings where strangers compete primarily on price and quality (i.e. "free" markets). In principle, actions are motivated by self-interest and unaffected by social or moral considerations beyond the self-interested morality of "greed is good."
- *Associative System:* The foundation of associative systems lies in alliances between economic actors; such associations are defined as "voluntary arrangements involving durable exchange, sharing, or co-development of new products and technologies". These systems are based on the assumption that mutual support and reciprocity will result in the best economic outcome for the parties. Like the price-based systems, they are oriented toward instrumental rationality and profit maximisation.
- *Moral System:* Moral systems are based on some belief in a substantive good or value. Actors are rational but only insofar as their actions are oriented toward putting in place a value or as their substantively rational actions are bound by a moral code. Perhaps the most familiar example is the so called fair trade goods, where the producer of the good is assured to get a "fair" share of the price instead of the lowest possible one.
- *Communal System:* In communal systems the exchange occurs between parties characterised by social relations. The relationship influences the terms of exchange, including whether or not the exchange takes place and the price set. Members of the group are treated preferentially, while outsiders are less well treated or are rejected entirely as exchange partners. The bases on which exchange takes place are often dictated by the customary rules of participation and distribution established by the group.

Finally, we note that this preliminary analysis does not cover the problems related to public or common goods at all. The sole focus here has been on compensation related to using privately held goods.

3.3.2 Caching

[Pit 2008] studies the performance of caching by the nodes in a Delay Tolerant Network (DTN) network. DTN can provide ad-hoc communication services within (sparse) mobile user communities when end-to-end IP communication is not available. By definition, the nodes cache the data for some time, as DTN operates as a store-carry-and-forward network. The data that is being "carried" can also be used to serve requests from other nodes before its lifetime has expired. This implicitly provides a caching functionality into nodes whose primary goal is to perform forwarding

The next step from caching is to use the network as a distributed storage system. A user may send data to the network and later fetch it with the same or a different device. In contrast to the traditional Internet model, the data does not need to be sent to and fetched from a specific server. The paper presents simulation results from different DTN routing algorithms and two different mobility models. The performance gain from caching can be seen quite clearly.

These results encourage the study and implementation of caching functionality in PSIRP.

3.3.3 Security Mechanisms

There are a number of security mechanisms which are often used to enable certain properties for protocols and architectures. Examples of such mechanisms include access control, hash chains, hash trees, and computational puzzles.

3.3.3.1 Scope Security

Scopes in pub/sub architectures control the dissemination of messages within a certain range of nodes with similar interests. Information scoping allows scopes to be hierarchically organized to form larger more broad scopes. The basic idea is that application components are arranged into groups that share a common scope without typically being aware of their membership within these groups; notifications are never propagated outside the groups..

[Fie2004] proposes an extension to a large scale pub/sub system, known as *Rebeca*, to support scopes. This extension uses routing tables of nodes that have been split into multiple tables, one for each local scope. When a scope is created, a broker is responsible for its announcements and creates an empty routing table for its scope. Whenever a node wishes to join a scope, it issues a join message which is subsequently routed through routing brokers until it reaches the first node that is member of the scope. This node sends a reply using the same path, followed by its own join message. If the reply is affirmative, it contains management information as well as information needed by all the involved brokers to setup their routing tables. In this fashion, all of the involved brokers become part of the scope's overlay. When a message is required to transit two scopes, these scopes must have at least one common broker.

Access control is realised through *attribute certificates* (ACs) [Far2002]. They are used to identify nodes as well as their privileges. Whenever a node is about to invoke an action, it sends the appropriate message accompanied with its AC. The routing brokers examine the message and the attached AC; if the source node has the privilege to issue that message, the message will be forwarded. Moreover, a hierarchy of trust is employed to ensure infrastructure security. When a node requests to join a scope, it has to send its trust certificates along with the join request. If that node is directly connected with a node that is part of the scope, and both share a common ancestor in the trust hierarchy, the join message is accepted. However, there is always the case when messages of a given scope have to traverse the network via untrusted nodes. For this purpose, secure tunnelling is used, provided that the intermediate nodes are willing to participate and route the encrypted messages.

3.3.3.2 Packet Layer Authentication

In broadcast communications it is always challenging to authenticate the source of a transmission. Symmetric encryption cannot solve the problem, as each node owning the shared key is able to inject bogus or malicious packets. Asymmetric solutions, such as digital certificates and signing, provide a more efficient solution. However, digitally signing every sent packet poses significant computational overhead since it mandates per-packet signature verification. Moreover, malicious users might send bogus packets that contain fake signatures, and cause a clogging DoS attack, as nodes commit extreme amounts of resources to verify every signature that they receive.

[Per2002] presents the *Timed Efficient Stream Loss-Tolerant Authentication* (TESLA) protocol, which can authenticate broadcast sources with low communication and computational overhead. TESLA bases its operation on loose time synchronization between sender and receiver, as well as through the use of *one-way chains*. A one-way chain is a cryptographic primitive in which every element S_n in the chain can verify all the elements S_k where $k > n$. The first element of the chain, for example, can verify all other elements, and is thus the last element revealed to end-users.

At the initiation phase, the TESLA client and sender loosely synchronize their clocks. Then, the sender splits the time into equal size intervals and assigns an element of the one-way chain to each interval. For each time interval, the sender computes a message authentication code (MAC) using the corresponding element of the one-way chain as a cryptographic key. Finally, the sender broadcasts the packet and reveals the value of the one-way chain after a known delay. The receivers must buffer packets until they receive the requisite values in the one-way chain that can be used to validate them. Even if the receiver looses a disclosed key, it can recover by using the keys that it will received afterwards. Moreover, receivers discard any packet which contains a MAC that was computed with a key that was already revealed since these packets may be forged. In this manner, a repeat attack is avoided.

3.3.3.3 Transparency and Information Accountability

It is generally recognised that social rules are supposed to more easily invoke compliance than abuse. This is because the rules are generally known and social institutions tend to make the results associated with compliance easier than the consequences of violation. If we adopt this societal paradigm in the internetworking environment, such as with pub/sub networks, then large-scale information systems might be regulated to be reliable, robust, secure, trusted, misuse-free, and efficient.

In a step towards this radical proposal, Weitzner et al. [Wei2007] introduce transparency and accountability as the attributes of information systems that might force compliance and collaboration, rather that violation and misuse. These attributes can be supported by *policy awareness*, defined as a property of information systems that provides all “participants with accessible and understandable views of the policies associated with information resources, provides machine-readable representations of policies in order to facilitate compliance with stated rules, and enables accountability when rules are intentionally or accidentally broken” [Wei2007]. A critical implementation question when making new internetworking paradigms policy-aware is whether it would require significant re-engineering of existing protocols. The work in [Wei2007] suggests the use of accountability appliances that are distributed throughout the Internet and communicate using well-defined protocols. For the realization of this innovative approach, policy languages, reasoning tools, and transaction logs are certainly required.

3.3.4 Network Coding

In general, the problem of data transmission from source to destination can be seen as the problem of communicating over an erasure channel between the sender and the receiver with unknown erasure probability. In order to improve channel capacity different coding techniques are applied but none of them meet all requirements.

Traditional approaches suffer from possibly large numbers of transmissions in situations where one of the receivers did not receive one of the packets. Especially in the case of broadcast communications, most of the receivers will have already received majority of the retransmitted packets. According to this, the aim of source and network coding is to reduce unnecessary retransmissions as much as possible, improving overall reliability and capacity of the network.

In the following we give first a short account on SoA in source coding, going from traditional (rateful) forward error correction (FEC) codes to modern (rateless) codes, namely digital fountain codes. We then move on to discuss network coding schemes in which relay nodes participate into the coding process.

3.3.4.1 Reed-Solomon Codes [Ree1960] [Che1993] [Skl2001] [Wic1999]

One of the most significant traditional block codes with erasure correction is Reed-Solomon code. The main property of Reed-Solomon code (N,K) with q^m symbols in alphabet is that after receiving any K symbols of N sent symbols, original message of K symbols can be decoded. Generally, encoder takes K information symbols of m bits each and adds N-K parity symbols to make N symbols codeword (see Figure 3.4). Thanks to this added redundancy receiver is able to decode complete messages even if some of the symbols are lost.

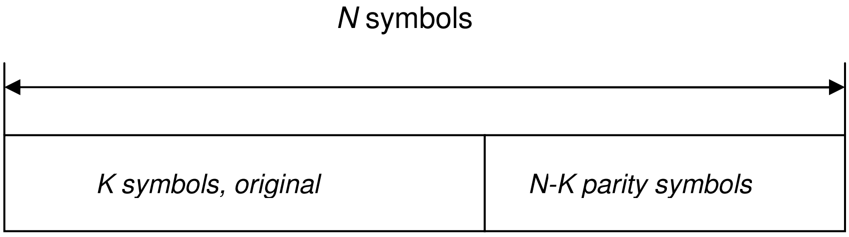

Figure 3.4 - Packet of N symbols, encoded using Reed-Solomon code

Reed-Solomon code has the largest possible code minimum distance among all linear codes with the same encoding input and output block length. The code distance for Reed-Solomon code is:

$$d = N - K + 1$$

Generally, the code with distance *d* is capable to decode *t* or less error where *t* is given by:

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

For Reed-Solomon code we have:

$$t = \frac{N-K}{2}$$

This implies that decoder can correct up to $t = \frac{N-K}{2}$ error symbols in received codeword. In the most general case:

$$RS(N,K) = (\ 2^m\text{-}1,\ 2^m\text{-}1\text{-}2t\ )$$

where *m* is a number of bits representing each symbol and *t* is symbol-error correcting capability.

Considering a binary code *RS(N,K),* only 2^K of possible 2^N are code symbols. For non-binary codes, redundancy is even larger, for instance if we examine the case where each symbol is represented with 3 bits *p = 3* then the number of code symbols will be 2^{p*K} of possible 2^{p*N}. The number of symbols used for code words will be dramatically reduced in comparison with number of symbols at disposal which implies that redundancy increases as well as code distance.

For example, the most used Reed-Solomon code is *RS(255,223),* which means that it consists of 255 code words of 8 bits, the number of parity symbols is 32, which implies that it can correct up to 16 error symbols. Errors can occur at the single bit in the symbol, or at all 8 bits of it. Both of mentioned situations will be considered as one error symbol. This means that algorithm can correct up to *m*(N-K)/2* error bits, or in this example 8*16 bits. It can decode a symbol either the error was caused by one bit being corrupted or all bits in the symbol being corrupted with the same success. This gives a Reed-Solomon code great burst-noise decoding capability, making it especially appropriate for transmissions over wireless channels.

When position of an erroneously transmitted symbol is known it's called an erase. Reed-Solomon algorithm can decode up to *(N-K)* erases (twice as much as errors).

The decoder fails to correct error message when the number of errors exceeds *(N-K)/2*. When this situation comes to pass decoder either recognizes the problem using built in filters or decodes message wrong.

Main disadvantage of this approach is that it's applicable just for small number of *N, K* and *q*. As code redundancy increases with the length of symbols, its implementation complexity grows, as well as bandwidth for real time applications based on RS codes. Standard encoding and decoding have a cost of order *K* (*N-K*) log *N* packet operations. In addition to this, estimation of erasure probability as well as code rate *K/N* has to be done before transmission, which causes problems if the erasure probability is higher then expected and the receiver got less than *K* symbols. Modifying code on the fly by reducing code rate would be the best solution in this case, but it is not applicable in Reed-Solomon code. This was the starting point in developing new codes which could support “on the fly” approach.

3.3.4.2 Fountain Codes [Mit2004]

In contrast to traditional transmission and coding techniques which chop the message to be transmitted into parts, send each part of it separately, and wait for acknowledgement from destination, fountain techniques send randomly all parts of the message which is slightly extended beforehand by adding redundant data. Fountain codes are rateless since the limitation in number of encoded packets generated from the source message does not exist, and can be changed on the fly. Source can send as many encoded packets as is needed for receiver to successfully recover data. This number just has to be slightly larger than *K*. We now give an overview of some of the major subtypes of fountain codes.

Random Linear Fountain Code [Mac2005]

Redundant packets for transmission (t_k) are obtained as a XOR of original set of N source packets ($p_1,p_2 \ldots p_N$) and randomly generated set of N bits at each clock cycle k, G_{kn}:

$$t_k = \sum_{n=1}^{N} G_{kn} p_n$$

If G_{kn} is considered as column of a matrix, at the receiver side, after receiving N transmitted packets (t), a matrix G of dimension N*N will be obtained. Main assumption is that the receiver knows matrix G. It is possible, for instance, for randomly generated set of bits to be used, obtained using pseudorandom number generator with the seed which is stored in the header of packets. Receiver which has the same generator, knowing the seed can produce the same set of bits. Decoding is processed by simply inverting matrix G:

$$p_k = \sum_{n=1}^{N} G_{kn}^{-1} t_n$$

Decoding is not possible if the number of received packets is less than N or if the matrix G is not invertible. Probability that G is invertible is very small for number of received packets equalling N. It increases dramatically with transmitting excess packets to probability equal to ($1-2^{-E}$) where E is number of excess packets.

Unfortunately, adding more packets in transmission is not perfect approach due to increasing of computational complexity by quadratic and cubic fashion with number of encoded packets.

One of the improvement possibilities is dividing packets into sub-packets of constant size. On each sub-packet the same procedure as in previous case is performed. Obtained packets are then used as starting point and XOR is performed on them once again. Main arguments are in decision when to send acknowledgements of successful decoding, after each sub-packet or after overall message has been successfully received and decoded. This approach increases efficiency but encoding/decoding complexity still remains.

Tornado Codes [Bye1998]

These codes are one of erasure block codes, primarily constructed to speed up coding/decoding in traditional erasure codes. Given the erasure channel with erasure probability p, tornado codes can decode up to $p(1-\varepsilon)$ symbols, with speed $n \log(1/ \varepsilon)$.

Tornado codes can be described in terms of graphs. The construction of a tornado graph is based on layers containing nodes. First layer contains nodes representing symbols of original message. Each node represents one of the symbols where symbols can be considered as packets of bits. Second layer consists of nodes with redundant content-exclusive-or of neighbours of a corresponding node. Iteratively all layer nodes can be obtained.

All nodes of subsequent layers are considered as restrictions of first layer nodes. By choosing original symbols and their restrictions in random but appropriate way, decoding of original message is possible as soon as receiver gets enough information. In tornado codes each restriction on subsequent layers depends only on few symbols from alphabet and not on all symbols like it was in the case of Reed-Solomon codes. This makes tornado codes less computationally expensive and speed up its coding and decoding.

It is possible to construct a tornado code and its restrictions in the way that for a given overhead rate ε and number of message packets k, and number of encoded packets n it is possible to decode k(1-ε) packets with speed n log(1/ ε).

Despite the fact that tornado code ensures increase in coding/decoding speed it is not widely deployed due to serious drawback. Before coding with use of tornado code, exact number of encoded packets which can be generated has to be known, as encoded packets are determined by graph representing a tornado code.

LT Fountain Code [Lub2002] [Rob2002]

This code was the first practical realization of rateless codes. Its main encoding idea is quite simple:

1. Chop message to be encoded into *n* blocks of roughly equal size.
2. Choose *d*, degree of an encoded symbol, according to predetermined distribution.
3. For every block of message randomly select d packets from the original message and XOR them.

Decoding is performed mainly based on existence of two areas: message queue area and buffer area. All packets of degree *d* = 1 are stored in message queue area and considered as decoded, as all packets of degree d > 1 are stored in buffer area and matched (XOR-ed) with decoded packets from message queue. After XOR-ing packet of degree 1 from message queue (for instance P_k) area with packet of degree more than 1 from buffer area which contains P_k, degree of a packet in buffer is decreased. Iterative repeating of this process leads to decreasing degrees of all packets to 1, at which point the receiver is able to decode the message and potentially sends acknowledgement to transmitter.

To perform these kinds of operations, the receiver has to be aware of the content of each packet, its degree, and the indices of the packets that it consists of. In order to avoid overhead in transmission this additional required information the same pseudo-random generator is used on both sides, source and destination. The transmission of "code key" which represents a seed for pseudo-random generator is sufficient to resolve both degree d and packet indices.

The most critical part of LT Fountain Code design is the probability distribution of degree d. Generally, majority of packets have to have low degrees in order to represent starting point for algorithm and provide prerequisites for its continuity. On the other hand, some of the packets need to have high degree to make sure that there will not be the packets which are not included and not in relation with anyone else.

In the ideal case at every iteration just one packet would have degree *d* = 1 and performing XOR operation with other packets would result in appearing again just one degree-one packet. The probability distribution would be ρ(1) $=\frac{1}{N}$ for a given number of N encoded packets, and ρ(d) $=\frac{1}{d(d-1)}$ for d = 2, 3, …N. Unfortunately this probability does not give good performance in practice due to unexpected changes.

3.3.4.3 XOR Coding [Kat2006] [Fra2007]

The main idea of the exclusive "OR" (XOR) approach, the first real "network coding" technique we consider, is to use intelligent mixing of packets in order to increase network throughput instead of automatically sending packets from transmitter to receiver based on their addresses.

Considering the scenario where two sides (Alice and Bob) want to exchange pair of packets via a router four transmissions are required (see Figure 3.5). First, Alice sends packet to router, which forwards it to Bob, and then Bob sends packet to router which forwards it to Alice. Instead of this, intelligent combination of packets is possible at routers side: Both, Alice and Bob send their packets, router XORs them and broadcasts the XOR-ed version. After receiving the XOR-ed packet $A \oplus B$, both Bob and Alice are able to decode the packet sent from other side by simple XOR-ing received packet on their own because $A \oplus A=0$ (see Figure 3.6). Moreover, encryption is achieved by the fact that it's impossible to reverse the operation (decode message) without knowing the content of one of two initial messages.

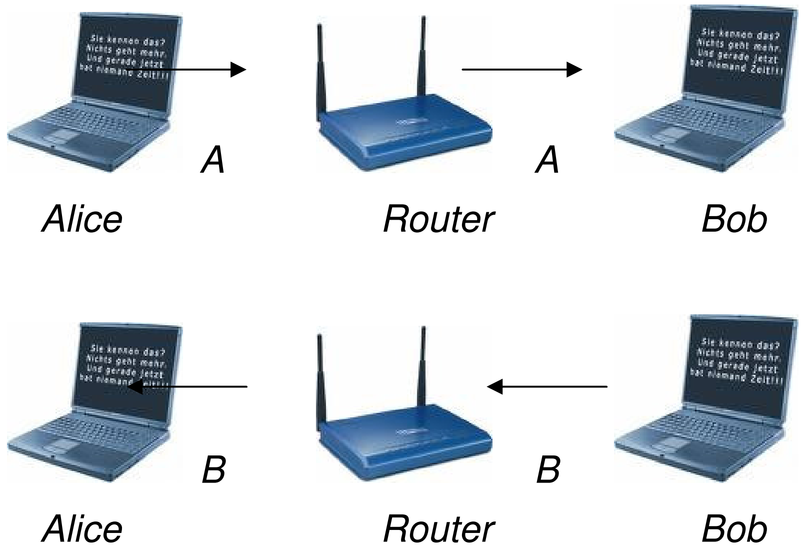

Figure 3.5 – Exchange of two packets without network coding requires four transmissions

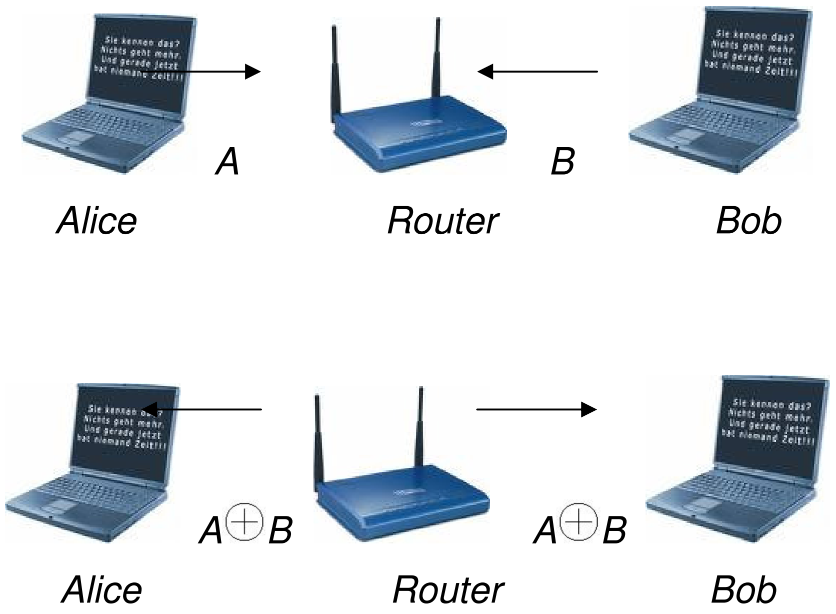

Figure 3.6 – Exchange of two packets using XOR network coding requires three transmissions

One of the main issues in the implementation of this simple coding approach is gathering the information about packets stored in neighbour nodes. This information is essential for the node in order to be able to intelligently combine packets for broadcast in the way that the next hop has enough data to decode the packet intended to it (to transmit n packets $p_1,p_2,\ldots p_n$ to n next hops $r_1,r_2\ldots r_n$ a node can XOR all n packets together only if each next-hop r_i has all n-1 packets p_j for $j\neq i$). A node may have multiple options how to encode, but it always follows the aim to maximize the number of packets which could be successfully decoded at the receiver side after receiving XOR-ed combination. In order to obtain enough information about packets stored in the network at particular nodes several mechanisms have been proposed. The network can let nodes snoop all communications over the network and to store overhead data for a limited time interval. This approach works well in a wireless environment, but is not directly applicable in modern switched / routed networks. Moreover, each node can send reports about packets it contains together with packets the node transmits in piggybacking manner. In the case that there are no packets to transmit, the node can just periodically send information about packets it contains in special control packets.

Exchanging report packets opens another question: is it enough for network coding to rely just on information obtained by those reports? The main issue is the possibility of report packets being lost in case of congestion and their too late arrival in the case of light traffic when nodes have already made decisions without waiting for reports.

To decrease the probability of inadequate encoding due to the lack of information needed from the neighbours, the XOR mechanism relies also on the *Expected Transmission Count* (ETX) metric [Dec2005] used in selecting routing paths. The approach in ETX is to compute the delivery probability on each link and assign each link a weight equal to 1/(delivery probability). These weights are not just used in a link state routing protocol to compute the shortest path; they are also used as auxiliary information for intelligent packet combination in absence of report packets. The probability that a particular neighbour has a packet is estimated as delivery probability between packets previous hop and the neighbour.

For reliable transmission of XOR-ed packets, 802.11 broadcast is not an acceptable approach. It lacks receiver acknowledgement (left out due to the broadcast storm problem) and backoff which is ensured by the unicast mode. One of possible compromises would be pseudo-broadcast which is modified 802.11 unicast in the sense that it unicasts packets intended for broadcast, so it provides reliability and backoff. Pseudo-broadcast is designed to have the media access control (MAC) address of one of the recipients written in the link layer destination field. All next hops of the packet are listed after the link layer header. When a node receives a packet with a destination address different from its own, it checks if it is a next hop and if so, it processes the packet further, else it stores a packet in a buffer as an opportunistically received packet. In the case that the destination address of the packet matches with the address of the node which received it, an acknowledgement message is generated and sent to the source.

In order to improve reliability a scheduler for retransmission events can be introduced. When a node sends a packet it schedules a retransmission event for each of the packets. If any of them is not acknowledged in a certain period of T seconds, the packet is retransmitted. At the receiver side, when it decodes successfully it automatically generates an acknowledgment (ACK) and schedules an ACK event for it. Those ACKs are sent in piggy backing mode during the transmission of information packets. When the node sends packets, it first checks for its pending ACK events and incorporates them in packet header.

In experiments this approach, generally, shows very good results, but analysis mostly assumes certain prerequisites: identical nodes, omni-directional radios, perfect hearing within some radius with the addition that the signal is not heard outside this radius, a pair of nodes can hear each other the routing will pick the direct link, infinite flows and steady state. Regarding memory, nodes need to store recently overheard packets for future decoding.

Consequently, the storage requirement should be slightly higher than the delay bandwidth product. Also, lack of power is not taken into consideration, and it is assumed that nodes have an unlimited power supply. Also, the XOR approach requires high node coordination, which is more difficult to achieve in larger networks.

3.3.4.4 Linear Network Coding [Ho2005] [Kat2007] [Fra2006]

This approach is, in general, similar to XOR coding with the difference that the XOR operation is replaced with linear combination of data (in essence, matrix multiplication) where coefficients of linear combination are taken from certain finite field. This provides more flexibility in how the packets can be combined. Similar to erasure coding, successful reception of information does not depend on receiving particular data packet but on receiving sufficient number of independent packets.

Let M^1, M^2 …M^n denote the original packets generated by several sources, then encoded the packet would be a linear combination of M^1, M^2 …M^n with associated set of coefficients g^1, g^2, …g^n from a certain finite field F which implies that it has a form of:

$$X = \sum_{i=1}^{n} g_i M^i$$

In other words two vectors exist; first, g=(g_1, g_2, …g_n)-encoded vector, which is used at the receiver side to decode the message, and, second:

$$X = \sum_{i=1}^{n} g_i M^i$$ - information vector

Encoding can be performed recursively, with already encoded packets.

Considering the node that has already received a set of encoded packets (g^1,X^1), (g^2,X^2)…(g^m,X^m), new encoded packet can be generated from them by choosing coefficients h_1, h_2, …h_m and computing the linear combination:

$$X^{'} = \sum_{i=1}^{m} h_i X^i$$ (see Figure 3.7 for illustration)

The corresponding encoding vector of new encoded packet is not simply equal to h, since also the encoding vector of a starting packet has to be calculated. According to this, the new encoding vector can be easily calculated as:

$$g^{'}{}_j = \sum_{i=1}^{m} h_i g_j^{\,i}$$

A node stores the encoded vectors it receives as well as the original packets, in a so called decoding matrix, row by row. Initially it contains just non-encoded packets issued by this node with the corresponding encoding vectors. A received packet is innovative if it increases the rank of the matrix. If a packet is not innovative it is converted to row of zeros by Gaussian elimination and ignored.

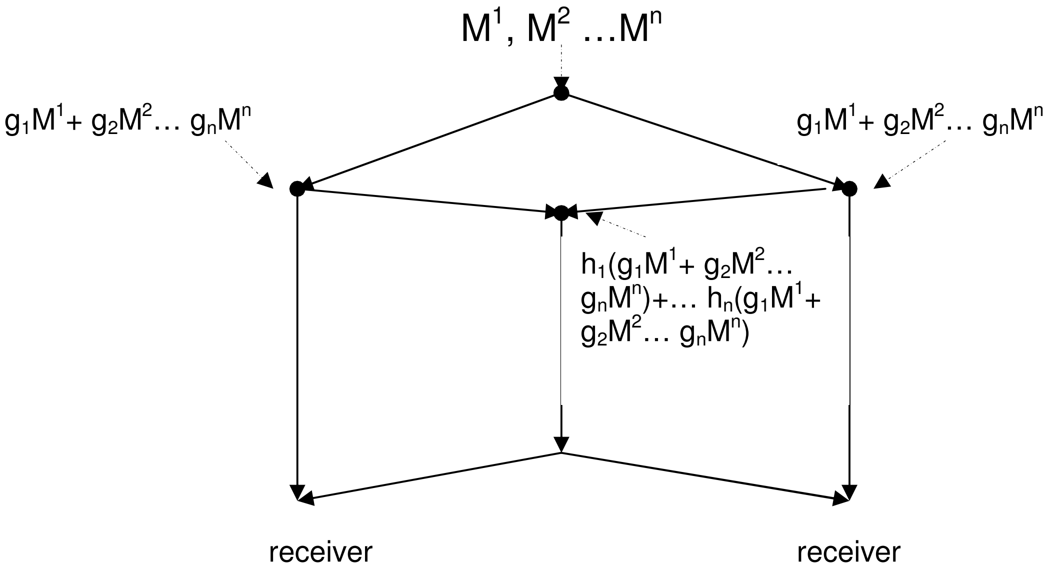
M^1, M^2 …M^n
g_1M^1+ g_2M^2… g_nM^n
g_1M^1+ g_2M^2… g_nM^n
h_1(g_1M^1+ g_2M^2… g_nM^n)+… h_n(g_1M^1+ g_2M^2… g_nM^n)
receiver
receiver

Figure 3.7 - An example of linear network coding, where M^1, M^2 …M^n are source packets multicast to the receivers, and coefficients g_i and h_i are randomly chosen elements of a finite field.

In order to retrieve the original message the decoder has to solve the system:

$$X^j = \sum_{i=1}^{n} g_i{}^j M^i$$

using Gaussian elimination algorithm, where unknowns are M^i. This system with m equations has n unknowns, and having m ≥ n is prerequisite for decoding. Fulfilling this requirement is not a guarantee that the message will be decoded since some of the linear combinations might be linearly dependent.

With random network coding (randomly choosing coefficients) there is a certain probability of selecting linearly dependent combinations which is related to field size. Simulation results shows that even for relatively small fields this probability becomes negligible.

Moreover, linear network coding shows good results in synergy with multicast, making it highly relevant to publish/subscribe architectures which usually rely on multicast-like forwarding patterns. In Figure 3.8 a scenario is illustrated where the source multicasts four packets to three destinations. In the case that some of the packets are lost during the transmission, without network coding sender has to retransmit the union of all four packets.

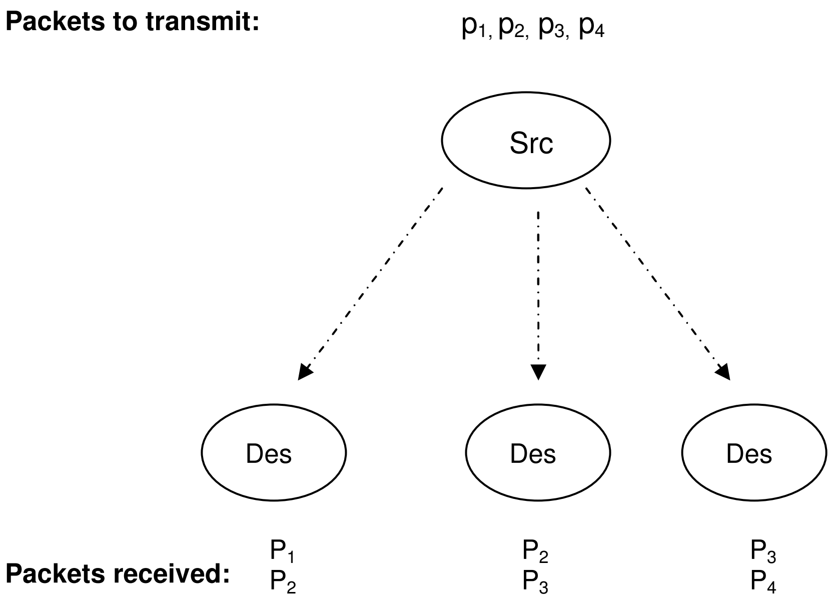

Figure 3.8 - Source multicasts four packets to three destination, and two of them are lost for every receiver.

In contrast to network coding it is sufficient to retransmit only 2 randomly coded packets, for example p_1'=p_1+p_2+p_3+p_4 and p_2'= p_1+2p_2+3p_3+4p_4. Despite the fact that they lost different packets all three destinations will be able to retrieve all four original packets by inverting the matrix of coefficients and multiplying it with the packets it received. For example, if the receiver receives p_1, p_2, p_1' and p_2' reconstruction of original packets is effected by the following matrix equation:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} p_1^{'} \\ p_2^{'} \\ p_1 \\ p_2 \end{pmatrix}$$

Despite the fact that linear network coding provides more flexibility and performs in many ways better in comparison to XOR mechanism, linear combining requires enhanced computational capabilities at the nodes. Since processing costs are becoming less expensive the bottleneck is shifted especially in wireless networks to network bandwidth due to growing demands in applications and quality-of-service (QoS) guarantees. In that sense network coding utilizes cheap computational power to increase network efficiency.

3.3.4.5 Conclusions

Many hardware and software implementations of Reed-Solomon code exist. Some of them are “off the shelf” integrated circuits which can encode and decode Reed-Solomon codes, and others are more related to *Very High Speed Integrated Circuit (VHSIC) Hardware Description Language* (VHDL) and *Verilog* design. Software implementation is more difficult because general purpose processors do not natively Galois field arithmetic operations. But, the main disadvantage is the suitability for just small number of N, K and q. As code redundancy increases with the length of symbols, the implementation complexity grows, as well as the required bandwidth. Also, estimation of the erasure probability as well as code rate K/N has to be done before transmission. Modifying code on the fly by reducing code rate is not applicable in Reed-Solomon code.

Fountain code introduces solution in the sense that the limitation in number of encoded packets generated from the source message does not exist, and can be changed on the fly. Source can send as many encoded packets as needed for receiver to successfully recover data. The main issue is in finding a compromise solution for number of packets encoded. This number has to be large in order to produce an invertible encoding matrix, but from the other side encoding a bigger number of packets leads to great complexity increases. Also, for tornado code, exact number of encoded packets which can be generated has to be known before encoding, as encoded packets are determined by graph representing a tornado code. The main issue of LT codes is finding optimal degree distribution which is from the essential importance.

XOR coding is a simple network technique which substantially improves network throughput by intelligently combining packets. But, in order to be able to combine packets in an optimal manner, each node in the network must have information about the overall network (which node contains which packet). Gathering this kind of information can cause significant overhead in terms of required memory and processing data. Also, XOR encoding ties the MAC address to routing, imposing a strict schedule on routers' access to medium. In the case that node closer to destination overheard the packet it can't send it due to the fixed schedule. This prevents spatial reuse and thus underutilizes the medium.

Linear network coding gives better results in the case of increasing number of nodes, because central scheduling is not necessary. Nodes make decisions on how to propagate packets based on local information only (each user knows about the blocks he downloaded and the blocks that exist in the neighbours). Replacing XOR operation on packets with their linear combination interpreted as coefficients over some finite field allows much larger flexibility in the way packets can be combined. This approach has two main benefits: potential throughput improvements and high degree of robustness. The receivers need only to know the overall linear combination of source processes in each of their incoming transmissions. This information can be sent at each transmission block or packet as a vector of coefficients corresponding to each of the source processes, and updated at the each node by applying the same linear combination to the coefficient vector as to the information vector. The relative overhead of transmitting these coefficients is low. Requirement for successful decoding is receiving sufficient number of independent packets. Decoding is done by using Gaussian elimination technique which needs certain computational power and increases complexity especially for larger finite fields. In addition to problem of linear coding complexity and speed of encoding/decoding, one of the main concerns is presence of malicious nodes which can input false packets and thus make decoding problem even harder.

3.4 Publish/Subscribe Paradigm

Event-based computing and the pub/sub paradigm are crucial for future services and applications. The event paradigm allows asynchronous and decoupled many-to-many communication. Typically, event systems consist of publishers, subscribers, and the event service. The service ensures that information is routed properly from publishers to subscribers. Typical application areas of pub/sub have been workflow systems, stock market systems, air traffic control and other industry applications.

Pub/sub is a frequently used paradigm in current Internet and telecommunications services. RSS feeds can be seen as a primitive pub/sub system. The Session Initiation Protocol (SIP)

[Ros2002], used in the Internet Multimedia Subsystem (IMS) and Beyond 3G Systems (B3G), has extensive support for events [Roa2002].

One recent observation regarding pub/sub is that one size does not fit all [Rai2006], and it is challenging to meet all application requirements within one system. This observation motivates our focus in PSIRP to develop a simple, efficient, and scalable pub/sub substrate, which can then be used to build more elaborate routing systems.

3.4.1 Formal Methods in Publish/Subscribe System Design

Formal modelling of publish/subscribe systems and the correctness of content-based routing protocols were examined in [Müh2002b]. A routing protocol is correct if it maintains required safety and liveness properties. Since it may be difficult to maintain these properties in dynamic pub/sub systems they may be relaxed. A self-stabilizing pub/sub system ensures correctness of the routing algorithm against the specification and convergence [Müh2002b]. The safety property may be modified to take self-stabilization into account by requiring eventual safety.

The safety and liveness properties were extended in [Tan2004] with the notion of message-completeness and using propositional temporal logic. A message-complete pub/sub system eventually acknowledges subscriptions and guarantees the delivery of notifications matching acknowledged subscriptions.

A formal framework for modelling pub/sub systems is presented in [Bal2005]. The framework is based on two delays, namely the subscription/unsubscription delay and the diffusion delay. The motivation for this abstraction is to model concurrent execution of the system without waiting for the stability of the system state. This work differs from the previous liveness and safety properties, because they focus on analytically to characterize the quality of the system.

Subscriber and publisher mobility requires that the routing topology is updated to ensure that data is sent to the proper location. To solve this synchronization problem, the Siena event system was extended with generic mobility support, which uses existing pub/sub primitives: publish and subscribe [Cap2003]. The mobility-safety of the protocol was formally verified. The benefits of a generic protocol are that it may work on top of various pub/sub systems and requires no changes to the system application programming interface (API). On the other hand, the performance of the mobility support decreases, because mobility-specific optimizations are difficult to realize when the underlying topology is hidden by the API. In addition to Siena, several other event systems have been

A formal discrete model for both publisher and subscriber mobility was presented in [Tar2007]. In this work, two new properties are defined for the pub/sub topology, namely mobility-safety and completeness. A handover protocol is mobility-safe if it prevents false negatives. A topology or a part of a topology is complete if subscriptions and advertisements are fully established (propagated) throughout it. Mobility-safety of a generic stateful handover was shown for acyclic pub/sub networks. The completeness of the topology is used to characterize pub/sub handover protocols and optimize them. One of the results of this work is that rendezvous points are good for pub/sub mobility, because they can be used to limit signalling and flooding of updates.

4 Design Considerations

In contrast to Section 3, this section outlines design considerations whose involvement is necessary to govern the application of the previously discussed architectural and conceptual properties of the future Internet. This includes both fiscal and social economic factors, past and current network security concerns, and characteristics of trust models and information privacy.

4.1 Economics

As economics is a vast field in itself, we do not aim to give a state-of-the-art survey on economics per se. Instead, we focus on commenting key earlier works in which techniques or principles from economics have been applied either on architecture or mechanism designs in networking. As the present document is meant to discuss SoA related to architecture, we shall also consider economics-motivated analysis techniques as being outside the scope of this deliverable and instead comment on those issues in deliverable D4.1.

The key application areas of economics to design of networking architectures and mechanism are mainly related to cost of communications. Some of the key questions related to these issues are:

- Which aspects of network usage are being charged for?
- Related to above, which are the entities involved?
- How is charging accomplished?
- What happens at domain boundaries?
- What are the objectives of charging?
- Which economical "fundamentals" limit architectural choices?

We shall focus mainly on existing work related to mechanism design, especially focusing on QoS and congestion pricing. We also briefly comment on SoA related to inter-domain routing with cost-related metrics. Mechanisms for compensation are discussed elsewhere in this document (see Section 3.3.1).

Much of the work related to internetworking has targeted matching user demands with resources in proportion to their willingness to pay for those. Seminal work has been done in this space by Kelley together with his collaborators (see, for example, [Kel1998] [Gib1999a] [Gib1999b]. The key insight arising from this work is the application of cost fairness instead of often used ad hoc fairness metrics, such as flow rate fairness. For a forceful argument with ample references to supporting literature, see [Bri2007]. A simple mechanism for enforcing cost fairness in traditional end-point centric networking has been proposed in [Bri2005]. For a related discussion on different charging schemes, see [Cou2000]. Auction-based and market-driven mechanisms for pricing best-effort traffic have also been proposed (see, for example, [Mac1995]). Pricing and revenue sharing especially from the point of view of ISPs with relations to net neutrality have been recently studied by Walrand with his group (see [He2006] and [Mus2008]).

Another interesting body of work has arisen from QoS-related considerations. Vast body of work has arisen from problems related to soft or guaranteed resource reservation, key concept usually being assignment of packets or flows into one of fixed QoS classes. An interesting alternative without per-flow assignments has been proposed by Odlyzko in [Odl1999]. One of the key problems to consider is the differences in application requirements related to the QoS provided by the network. For a discussion on the roles of different application utilities in network architecture design see the seminal paper by Shenker [She1995]. Micropayments are an often-mentioned technique for enabling per-packet QoS compensation. However, despite a vast amount of work on mechanism design for enabling micropayments, they have not become widely adopted in deployed networks. An interesting discussion on reasons for this is given in [Odl2003].

Regarding applications of the above-discussed mechanism to the PSIRP architecture, the key problem is how to share "cost" of multicast-like transmission patterns. In the context of traditional multicasting an early analysis of the central problems such as the receiver-oriented nature of communications was given by Herzog et al. in [Her1997]. Algorithm design has been considered for implementing the actual cost sharing in [Fei2001]. Also relevant is the existing work on incentive-compatible inter-domain routing. Unfortunately very little exists in this space in terms of multicast routing, as the state-of-the-art work has been almost solely targeting BGP. For a discussion on the key problems and related mechanisms, see [Fei2006] and [Fei2007].

4.2 Socio-economic Aspects

In contrast to efforts that use methodologies and approaches from the fields of economics (see Section 4.1), socio-economic aspects are more concerned with the overall design of systems and value chains under the economic angle. It is important to keep this difference in mind when looking at state-of-the-art in this space.

4.2.1 Value-chain Dynamics

[Fin1998] investigates the speed of evolving value chains, based on observations in well-developed industries such as the automotive industry. He then maps these observations within these rather long-lived industrial structures onto faster moving industries such as the computer industry (with Microsoft being a well-covered case at the time of writing the book due to the monopoly case against the giant through federal agencies).

Fine asserts that life cycles in complex value chains are following a curve often described as the *double helix* [Fin1998] [CFP2005]. This life cycle follows through phases of integration, market differentiation, verticalisation, and disintegration. Mappings onto different industries exist, such as onto the IP service industry [Tro2007]. Although simplifying in its depiction, the double helix visualizes a complex trigger dynamics analysis that leads to the observed integration/disintegration effects.

The work within the Communications Futures Program [CFP2008] is most relevant to the PSIRP project although its results are not widely available since the consortium is closed to its members. [CFP2005] has been since evolved to a set of value chain analysis methodologies that allow for segmenting particular solutions into value chains or value networks, leading to a model of control point constellations (or business models) that is investigated under a multi-dimensional set of triggers, ranging from regulatory over technology to different market and corporate triggers.

4.2.2 Bullwhip Effect

The *bullwhip* or *Forester effect* [Fin1998] observes the dynamics of demand fluctuations (leading to inventory build-ups) throughout a complex value chain and the impact that changes in demand rather down the value chain might have on supply rather up the value chain, the impact being an exaggeration of demand up the supply chain. The resulting bullwhip effect describes this demand/supply exaggeration similar to the effect of a bullwhip in terms of immediate impact at the beginning and increasing amplitudes towards the beginning of the value chain (in other words, the farther one sits from the end consumer the harder one is hit by demand changes by stock piling up in inventories).

While this seems to more traditionally relate to inventory-based value chains, such as the automotive industry, a similar behaviour can be observed in industries like telecommunication (equipment stock), computer industry (investment in R&D) and others.

Smoothening the bullwhip effect can be seen as being desirable (e.g., reducing fabrication times in microelectronics), which lends to the assumption that solutions allowing for this reduction are to be favoured although a full study of this aspect is still to be found.

4.2.3 Overlay Economics

[Cla2006] provide insight into the particular field of overlay and economic structure as a relevant field for PSIRP (due to the aspect of potentially deploying first PSIRP solutions as an overlay to IP). The authors outline a taxonomy of thinking about overlays "that reflects the rationale for their existence/emergence and provides further elaboration of the sorts of technical, business/economic, and policy questions that overlays raise". The taxonomy is presented based on three examples (content, routing, and security), all of which are relevant to PSIRP. The argumentation used in the taxonomy as to why overlays emerge in the first place connect in spirit very well to the notion of the double helix although without asserting that there is some form of integrative/disintegrative repetitive pattern (as asserted in [Tro2007].

[Far2007] extends the work in [Cla2006] towards an investigation on the underlying economics for infrastructure-based content delivery networking (CDN) vs. peer-to-peer CDN solutions. His findings on the economic superiority of infrastructure-based solutions, such as Akamia offerings, are based on an industrial organizations model given the current structure of inter-domain peering and transit policies.

4.2.4 Design for Tussle

The seminal paper on *Design for Tussle* [Cla2007] can be seen as the first attempt to introduce socio-economic views on the architectural design front in a systematic manner. While economic arguments had never been absent from design debates in the Internet (and in other systems of large scale), the discussions in [Cla2007] embedded the economic angle of system design into architectural foundations of many projects since.

Apart from PSIRP itself, the Trilogy project [Tri2008a] particularly embeds the notion of Design for Tussle in its work programme and intends to shed some more light on designing under tussle principles in its set of deliverables. The public deliverable D2 [Tri2008b] lists case studies under the aspect of design for tussle which shed some light on the tussle and its resolution in currently available designs and solutions.

[Cla2007] introduces an evolution or extension of the original end-to-end (E2E) design principle of the Internet by specifically taking economic and trust aspects into account. The resulting *trust-to-trust principle* is a direct input in the PSIRP design process due to its economic importance in creating (trust-based) markets.

The presentation in [Sol2007] extends the Design for Tussle concepts towards a vision for a flexible execution environment that incorporates tussles (and their underlying concerns) directly into the formation of the (dynamic) execution environments. The presented *tussle networking vision* serves as a foundation for our work, as outlined in the project vision.

4.2.5 Reductionism vs. Evolution

[Hol2008] discusses the transformation of socio-economics from the reductionist Newtonian-Descartian view towards a Darwinian evolution approach that emphasizes evolutionary changes of complex systems through mechanisms of self-organization. The article outlines the multi-level approaches required to understand and observe complex systems where reductionist approaches seemingly do not help. Analogies from nature and other disciplines are seen at the centre of explaining the behaviour of large and complex systems and societies.

This evolutionary type of angle can be seen, e.g., in the work by Fine [Fin1998] where analogies of fruit flies are used to explain complex industrial lifecycles. Integration and des-integration cycles are described in the abovementioned double helix, underpinned by simulation techniques stemming from game theory or system dynamics. The described trigger analysis in work performed, e.g., in [CFP2008], is multi-dimensional and often captured in a rather '*fuzzy' way.*

This work and discussion on socio-economic approach is seen as very relevant to our work due to the intended system scale.

4.3 Security

Building safe and secure network systems is paramount to the success of the PSIRP effort. To achieve an efficient end result, it is necessary to consider both historical and present accounts of network security properties and directly involve this information when designing the network infrastructure from the ground up. From this methodology, we hope to arrive at a system which is not only resistant to attack through added functionality, but also naturally secure by virtue of its underlying design and construction.

SoA concerns in this section address network attacks, threat analysis, solution methodologies, and formal methods of modelling security protocols, requirements, and operating tactics.

4.3.1 Distributed Denial-of-Service (DoS) Attacks

Bandwidth consumption attacks are the most difficult distributed denial-of-service (DDoS) attacks to defend against, as the target of the attack is the network and solutions cannot be locally deployed. The capabilities of such attacks arise from the architecture of the Internet, which allows anyone to send packets to anyone, with or without the consent of the receiver, needing only the knowledge of the IP address of the target. The main research towards mitigating or preventing DDoS attacks is based on filtering, diffusion, replication, and hiding.

Filtering can be proactive or reactive and be based either on the data packets or a separate control information from the recipient of the data.

[And2003] [Par2007] utilize a two-pronged approach. The network bandwidth is divided into control channel and data channel, with control channel having only a small portion (a few percent) of the total bandwidth. Over the control channel anybody can send packets to a destination asking for a permission, a capability, to send data traffic. The capability is added to every packet sent over the data channel and only packets with valid capabilities will be allowed to pass through the network. Thus, the filtering of the data channel is proactive and based (mostly) on the data in the packets.

The control channel of the capability approach faces the same problem any reactive approach does. How to distinguish between attacking hosts and prevent attackers from flooding the system? In the case of capabilities the problem, specifically, is how to prevent attack against the control channel, constituting a denial of capability attack in itself [Arg2005]. A number of techniques have been developed that utilize bandwidth [Wal2005], computational, and memory puzzles [Par2007] to even the playing field between attacking and legitimate hosts. The idea is to increase the chances that a legitimate client is served by helping the probabilities of actual clients against attacking hosts. This can be done, because basic first-in first-out (FIFO) queues favour attackers who are sending at full rate as opposed to actual clients who send requests at relatively low rates.

Filtering can also be done in the network. Typically, the victim requests the network to stop packets with certain properties, such as they are coming from a certain host. In order to do this, however, the victim or the network needs to be able to tell which packets to filter and where they come from. SAVA [xxx] and ingress filtering [xxx] are considered important to solve the information problems of filtering the right traffic as close to the source as possible.

Other techniques, such as packet marking [xxx] or the encapsulation architecture [Hui2007], also exist.

Off by default [Bal2005] proposes a proactive filtering scheme based on bloom-filters and source routes. In it, each host indicates the sources in the Internet allowed to contact it, which is aggregated with other indications (into a bloom filter) and forwarded through the Internet. Only packets matching the filters, or with explicit source routes to destination are allowed to pass. The latter case, so that reverse routes to clients automatically work.

A much less studied approach to DDoS protection is utilizing diffusion, replication, and hiding. These techniques try focus on making it harder for an attacker to concentrate its attack on a single vulnerable point in the service it targets by either dispersing the service or hiding it from the attacker. i3 [Sto2002], SoS [Ker2002], and Hi3 [Gur2005] spread the attack over a large overlay, and also enable the victim to hide its IP addresses, at least to a point. Pushing DNS [Han2005] and DONA [Kop2007] replicate data in many parts, and thus, make it harder for an attacker to deny access to a given piece of data.

Pub/sub architectures are working differently from the underlying IP network, but still denial-of-service attacks are possible although they are not thoroughly studied for this type of systems. In [Wun2007], a first attempt is presented to classify DoS attacks for this type of system.

According to [Wun2007], pub/sub systems DoS attacks might have unpredictable effects. For instance, if a broker is being flooded with publications, then this attack has no significant impact to the internal brokers that are responsible for routing. However, other edge brokers responsible for notifying subscribers about new publications have significant more impact than the attacked broker. This effect is called *localization effect* and shows that pub/sub systems can be vulnerable to remote attacks.

Content-based pub/sub systems base their routing decisions on flexible messages, and, thus, routing nodes need to have sustainable computational power. Flexible messages allow the system to perform complex operations; however routing-scope flooding DoS attacks containing complex messages might drive the system to recover slowly after the attack. This happens because the CPU and the memory of routing nodes becomes overloaded and does not process these complex messages in high-speed. This is called *workload complexity effect* and it shows that there should be a upper threshold of routing message complexity in order to allow the system to recover quickly after the DoS attack. Another characteristic of pub/sub systems is that the routing nodes should maintain state for performing filtering, as well as event matching. However DoS attacks can take advantage of this fact to introduce severe effects to the system. For instance it is measured that a DoS attack that includes subscription messages has more severe effects than a DoS attack that uses the same amount of publish messages. This happens because for each new subscription, the routing nodes need to keep a state. This is called *message state effect* and it shows that there is a need for mechanism that will manage malicious states.

[Wun2007] introduces a very useful taxonomy of the DoS attacks in pub/sub systems. DoS attacks are being classified according to the exploitation type, the attack source and target, the attack propagation, the content dependence, and the statefulness of the effects. Each class of attack has a different impact on the system performance and different countermeasures should be taken to protect pub/sub system against each class of DoS attack.

4.3.2 Threat Analysis and Research

To survey existing attacks that exploit potential vulnerabilities in PSIRP framework, we classify three different domains of functionality:

- *The end-user domain,* consisting of publishers and subscribers. Publishers and subscribers may not trust each other, and may not trust the pub/sub network service, or the infrastructure

- *The pub/sub service provision domain,* consisting of the pub/sub network service providers (brokers) and the end-users (publishers and subscribers). The provider may not trust publishers and subscribers, and vice-versa
- *The infrastructure domain;* its components (cache elements, label switches routers, forwarded nodes, multicast points, network coders) may not necessarily trust each other

In the pub/sub service provision domain, providers and end-users should have a symbiotic relationship. In that sense, strong authentication might be used, although spoofing attacks such as *replay* and *sybil* attacks might be present:

- *Replay Attack:* The attacker eavesdrops the communication channels (sniffing) and stores packets. It resends them at a later time, trying to copy and replay packets that contain authentication credentials. When successful, the attacker gains access credentials and pretends that they are a legitimate authorised user.
- *Sybil Attack*: Usually when a system aims to self-protect against faulty or malicious actions, it replicates tasks among several remote entities. Each entity is then identified by an identity. However, when a local host has no direct evidence of the remote entities, it is difficult to ensure that specific identities refer to distinct entities. In the Sybil attack, a malicious entity is self-presented as multiply identities and undermines the redundancy employed by the system [Dou2002].

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

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

- *Cache Poisoning:* Exploits the absence of an authentication layer and forces the network elements to believe that they have received authentic caching elements, whilst this is incorrect. Bogus caches could contain malicious content, such as a worms or viruses.

- *Routing Service Attacks*: Malicious routing attacks target the routing discovery or maintenance phases. Examples include the routing message flooding, such as hello, route-request, and acknowledgement flooding, routing table overflow, and routing cache poisoning or fabrication [Hu2004]. Proactive routing discovers routes before they are actually needed, while reactive algorithms create routes on-demand, i.e., only when they are needed. Thus, proactive routing is more vulnerable to routing table overflow attacks. More sophisticated attacks include the Wormhole and Byzantine attacks. In the former case, an attacker records packets at one location in the network and tunnels them to another location [Hu2002]. In the latter case, a set of compromised intermediate nodes collude to create routing loops, forward packets through non-optimal paths, or selectively drop packets, resulting in disrupted or degraded routing services [Awe2002]
- *Forwarding Phase Attacks:* Once the route is established, on the fast data path, selfish or malicious entities drop data packets selectively, fabricate data content, or produce packet replay attacks for hijacking. They can also delay forwarding time-sensitive packets, or inject junk packets [Wu2006].
- *Eclipse Attack*: A sufficient number of malicious nodes collude, trying to deceive legitimate nodes into accepting malicious ones as trusted, with the goal of dominating a neighbour of the legitimate nodes. This way the attackers mediate most overlay traffic and effectively “eclipse” correct nodes from each others' view [Cas2002a].
- *Amplification:* Is a type of flooding and DoS attack where an adversary induces delivery of multiple messages to a single entity by injecting a single malicious message [Wun2007]. For instance, a new advertisement may attract many dormant subscriptions or an un-subscription may trigger multiple re-subscriptions to other publications.
- *Resource Consumption Attack:* Also known as the sleep deprivation attacks. They aim to consume a victim’s resources. A clogging attack is a common type of this category. The target node is requested to verify signatures or key exchanges during Diffie-Hellman handshaking, tasks that require significant processing cycles. The threat appears when multiple demands arrive simultaneously on a target node from several compromised peers. Thrashing is a special case of this type of attack. Unlike typical flooding attacks, in a thrashing attack, an attacker induces load by abusing repeated state changes that are process intensive. This can be accomplished using a set of messages that will likely include e.g., unsubscriptions [Wun2007].
- *Message State Effect:* Another characteristic of pub/sub systems is that the routing nodes are stateful for performing filtering as well as event matching. However DoS attacks can take advantage of this fact. For instance, it is measured that DoS attacks that include subscription messages have more severe effects than a DoS attacks that use the same amount of publish messages [Wun2007]. This happens because for each new subscription the routing nodes need to keep a state. This shows that there is a need for mechanism that will manage malicious states.

Service layer confidentiality is associated with end-users’ choice to remain anonymous and use the service provider’s facilities without the risk of revealing their identities. Additionally, the content itself should be sufficiently encrypted when delivered to service providers. In this direction, Man-in-the-Middle (MitM) attacks should be avoided.

- *MITM Attack:* An attacker intercepts and replaces the public keys of two communication parties with its own selected public keys. This allows the attacker to decrypt communications using the related private key.

Availability is already discussed within the scope of service integrity. *Service availability* means that the publication, notification, announcement, subscription, registration and rendezvous facilities are available when requested. As previously mentioned, several service integrity threats affect availability. Vulnerabilities in this scope are mainly exploited by DoS attacks. Sophisticated DoS attacks are camouflaged as routine flooding circumstances, but their aggregation is the actual threat.

Infrastructure availability means that the elements should always be available and robust enough to provide routing, caching, coding, multicasting, and other lower layer functions. To achieve service and infrastructure availability, (D)DoS mitigation is essential, and this is a twofold objective. The (D)DoS attack, when identified, should be spread in a minimum network span, or otherwise it should be populated with a minimum harsh risk. It is widely recognized that high availability protocols, redundant network architectures, and system design without single points of failure ensure availability and robustness.

Lastly, in a broad sense, spamming might be considered as an end-user domain availability threat. As it is shown in [Tar2006b], although pub/sub architectures are less vulnerable to spam messages than email, this threat might actually exist. Spam messages can be classified into two categories; inbound and outbound. Different techniques should be applied to fight spam messages for each category. Spam may also exist in bogus brokers, which can be used as black boxes that insert spam messages while dropping all legitimate messages, or as normal brokers which monitor network traffic in order to learn users' preferences and later on insert more effective spam messages. One key issue in pub/sub architectures is event replication; an event can be replicated to neighbour routers as long as it matches their filters. In case of poor filter designs, a spammer may construct a single message that will flood the network.

When *accounting issues* arise in pub/sub network, an adversary model exists when misbehaving entities observe, store, and then re-sell the contents or chunks for personal profit [Khu2005].

4.3.3 Existing Solutions

The following three sub-sections discuss research areas designed to address the security issues discussed in Section 4.3.2 and publish/subscribe network operations in general.

4.3.3.1 Access Control

Access control is a security requirement, especially in commercial pub/sub applications. It is used to assign privileges to all parts participating in the pub/sub architecture.

[Bel2003] suggests that access control can be based on roles. This architecture is referenced to as Hermes [Pie2002] pub/sub system, originally modified to support OASIS role-based access control system [Bac2002]. The goal of the suggested architecture is to provide a system in which security is managed within the pub/sub middleware, and access control is transparent to publishers and subscribers. In this architecture, each event has an owner who is identified with the use of a X.509 certificate. These owners set the access policies for their events. Users are assigned roles and privileges are assigned to each role. The users are never assigned privileges directly.

This approach has two obvious advantages: administration of privileges becomes easier and policy control becomes decoupled from the software being protected. Publishers and subscribers need to be authenticated. Every request they make to brokers is sent along with their credentials, based on these credentials brokers can accept, partially accept or reject the request. Policies are expressed with the usage of a policy language provided by OASIS. Access control decisions are based on predicates. Generic predicates are used, and they are handled as black-boxes, for instance, a predicate could make decisions based on the size of the message. They can be publish/subscribe restriction predicates, as well. In that case, predicates are understood by the pub/sub system and they make use of the event type hierarchy. For example, if a subscriber attempts to subscribe to an event which it is not authorized to access, the system will check if the subscriber is authorized to subscribe to any event's sub-types and thus the original subscription is transformed to a different subscription scope.

In order for this approach to be effective, brokers must be trusted for using access control policies. The proposed architecture suggests the usage of certificate chains that will form a *web-of-trust*. In this web-of-trust an event owner signs the certificates of the brokers it trusts, and these brokers sign the certificates of their immediate brokers and so forth. Providing that publishers and subscribers have a trusted root certificate for the event owners, they can verify whether their local brokers are eligible to process a certain event.

4.3.3.2 EventGuard [Sri2005]

EventGuard, is a mechanism that aims at providing security for content-based pub/sub systems. Its goal is to provide authentication for publications, confidentiality and integrity for publications and subscriptions as well as to ensure availability while keeping in mind performance, scalability and ease of use. Eventguard is a modular system operating above a content-based pub/sub core. It uses six “guards”, that secure six critical pub/sub operations (subscribe, advertise, publish unsubscribe, unadvertised, and routing) as well as a meta-service that generates tokens and keys. Tokens are used as an identification of the publication, such as a hash function over publication topic, and keys are used for encrypting messages' contents. All pub/sub operations involve communication with the meta-service before sending any message. Eventguard uses El-Gamal for encryption, signatures and the creation of tokens.

4.3.3.3 QUIP [Cor2007]

QUIP is a protocol for securing content distribution in pub/sub networks. Its aim is to provide encryption and authentication mechanisms to existing pub/sub systems. QUIP's security goals are to protect content from unauthorized users, to protect payment methods, to authenticate publishers and to protect the integrity of the exchanged messages. QUIP does not consider privacy in subscriptions. QUIP assumes a single trusted authority responsible for keys and payments handling, named *key server*. Each participant in the pub/sub network willing to use QUIP has to download in advance a QUIP client that will provide him with a unique random ID as well as with the key server's public key. At the initiation phase the key server provides to QUIP participants a certificate that links their public key to their id. Publisher wishing to publish a protected publication, contact the key server, receiving in return a content key, which is used for encryption. Subscribers that want to read the encrypted publication have to contact the key server, and if necessary to pay, in order to obtain the content key. QUIP purposes the usage of a public key traitor tracing scheme designed by Tzeng and Tzeng which has two main advantages, namely the ability to revoke the keys of some subscribers without affecting the keys of the others and each subscriber has a unique key which makes it easier to tell who has leaked a key.

QUIP considers two problems, ensuring that subscribers can authenticate the messages they receive from publishers, and ensuring that publishers can control who receives their content. [Cor2007] The idea is to combine an efficient traitor-tracing scheme with a secure key management protocol. There is a single trusted authority which will handle key management and payment called the key server. The focus in the paper is on DRM-like content control.

4.3.4 Formal Modelling and Analysis of Security Protocols

Based on our initial work, analysing publish/subscribe-based cryptographic protocols is essentially similar to analysing those based on send/receive, as the protocol nor the semantics have changed. The largest obvious change is that publish/subscribe versions of existing protocols need explicit “channels”, or pre-agreed message names, instead of relying on the network to “magically” deliver the messages to the intended receiver. Some of the intuitions may have changed, too, due to the recipients being replaced with (unique) message names. That is, the basic elements of traditional cryptographic protocol analysis appear to be essentially the same in publish/subscribe and the more conventional send/receive worlds. The only difference is that the sender need not know the network-level topological identity of the intended recipient. However, as most “standard” cryptographic protocols do expect that the sender simply must know some (cryptographic) identifier for the recipient (cf. e.g. [Syv2001]), such an “insight” does not lead us far.

Hence, we have to look at other intended purposes (beyond simple authentication) that a cryptographic protocol may have. For example, instead of knowing the identity of the communication peer, it may be enough to know that there is only one peer (e.g. a group of fully synchronised nodes) that remains the same throughout some session. More generally, it may be necessary to look at the intention more from the application point of view, and try to understand the economic mechanism, contract, or other purpose which the protocol has been build for. Some of the properties from more traditional protocols may still apply though, such making sure that the holder of a particular key is currently reachable (freshness), etc. (cf. also e.g. [Syv2001]).

Another aspect that we haven’t yet considered adequately is labels. If the labels are cryptographically meaningful, they per se create a set of implicit protocols, needing explicit design and analysis. For example, in a publish/subscribe network it may be meaningful to establish a cryptographically strong relationship between a certain (application-level) principal and a set of message labels.

4.3.4.1 Towards a Problem Statement

The way protocols are designed may need more fundamental changes. Hence, given the pub/sub communication model and its constraints, we tentatively can make the following observations.

- While the traditional Alice & Bob like protocols with the Dolev-Yao intruder model still pertain, they form only a small subset of the interesting problems. Furthermore, the existing models may need to be extended and enriched by the facts that all communication in the pub/sub network is naturally multicast and that two-way communication requires explicit establishment of a return channel (message name).
- Moving focus from authenticating principals to various security properties related to the data itself may require completely new methods.
- The group communication aspects of publish/subscribe seem to change the nature of many problems, and lead focus from typical Alice & Bob two-party protocols to protocols traditionally used for group communication.
- Another set of open problems can be found from within the infrastructure. Apparently, a number of new publish/subscribe based protocols are needed. A large open problem in designing such protocols is that of resource control, including issues related to fairness, compensation, and authorization.

Given this all, it becomes necessary to reconsider what we mean with authentication goals and assumptions. As the network provides no names for the active entities (nodes), the next generation applications are likely to be more interested in the ability to receive correct and properly protected information rather than communicating with predetermined nodes.

The threats and security goals can be divided, in a perhaps more standard fashion, as follows:

- Secrecy of security-related entity identities and identity protection.

- Secrecy of keys and other related information, typically needed for confidentiality and data integrity of the transmitted information.
- Denial of service, including unsolicited bulk traffic (spam).
- Threats to fairness, including mechanisms such as compensation and authorization.
- Authenticity and accountability of the information, including its integrity and trustworthiness, reputation of the origin, and evidence of past behavior, if available.
- Privacy and integrity of subscriptions to information.
- Privacy and integrity of the forwarding state (as a result of subscriptions).

At the mechanism level, there must be in place mechanisms to enable communication through potentially malicious networks and nodes, as well as to establish mutual trust between different administrative domains. This may require new kinds of cryptographic protocols that draw insight from micro-economics, e.g. algorithmic mechanism design [Nis1999], and have explicit structures for handling compensation, authorization, and reputation instead of relying solely on more traditional identity authentication and key distribution.

4.3.4.2 Design and Modelling of Cryptographic Protocols

The majority of work in the area of cryptographic protocol design and modelling has been based on the two-party communication model, with a Dolev-Yao [Dol1983] intruder. As discussed above, such a model appears insufficient for pure publish/subscribe networks, where the network provides no identity (other than the implicit identity provided by the location-related forwarding information) for the active parties. Furthermore, the set of interesting security problems goes beyond the standard end-to-end examples, such as authentication, key distribution, and secure file transfer; in addition to those, we need to consider group communication, denial of service, security goals related directly to data or database transactions, and the overall security of the network infrastructure itself. In this section, we briefly look at existing work, trying to figure out possible ways to enhance them to cover some of the new challenges.

Adversary Model

The standard attacker model in cryptographic protocol design and analysis is that of Dolev and Yao [Dol1983], often enriched with the correspondence assertions by Woo and Lam [Woo1993]. The Dolev-Yao model assumes two honest parties that are able to exchange messages through a powerful adversary that is able to intercept, eavesdrop, and inject arbitrary messages. Given that in our model primary communication is expected to be one way data transfer rather than two way transactions, requires two distinct channels for two way communication, and that in a more realistic model the attackers are typically able to compromise only part of the infrastructure (a byzantine model) instead of having complete control over it, a richer attacker model is needed.

Given the primarily multicast nature of the publish/subscribe paradigm, some insights may be attainable from the work on group protocols. It may even turn out that discrete attacker models are not sufficient, but that instead one has to turn attention to probabilistic or micro-economic models, such as Meadows' model for analysing resource-exhausting denial of service [Mea2001] or Buttyán and Hubaux micro-economics flavoured models [But2002].

Modelling Logic and Beliefs

To our knowledge, the vast majority if not all the work on logic-based modelling and verification of cryptographic protocols is inspired by the Alice & Bob two-party setting (see e.g. [Cal2006] and [Syv2001]), sometimes enriched with a Server. Considering the publish/subscribe paradigm, this does not appear very useful. In the case of a single publication (channel), the publisher basically knows nothing, or, rather, does not gain any new knowledge when publishing. The subscribers, on the other hand, may learn new knowledge from the message contents. However, some properties, like freshness, appear impossible to implement without either two-way communication or additional, external data (such as roughly-synchronised clocks).

Digging slightly deeper, it becomes evident that also in the publish/subscribe world there will necessarily be two-party or multi-party protocols. Using our basic model, the initial messages will contain information that allows the receivers to subscribe to some messages expected to be published in the future, or publish messages in a way where they can expect there to be a subscriber. Hence, already here we have some basic beliefs:

Alice believes that there is a party ("Bob") that is subscribed to a message named M and will do some well-specified action X once it receives a valid M.

As this belief expresses expectations about the allowed future states of the system, an open question is whether adding temporal modalities some of the existing modal-logic based approaches would be sufficient.

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

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

Information-Theoretic Models: At the time of this writing, it is a completely open problem how the more information theoretic models, such as the one underlying Huima's tools [Hui1999] or developments thereof (e.g. [Mil2001]), could be applied to publish/subscribe.

4.3.5 Formal Methods in Security

The earliest attempt to formally analyze security protocols is arguably the *BAN-Logic* put forth by *B*urrows, *A*badi and *N*eedham (hence "BAN") in 1990 []. BAN-Logic can be described as a set of mathematical notations combined with a few commonly held beliefs upon which security properties such as authentication and secrecy can be formally or at least semi-formally discussed and proven. However, Ban-Logic only conveys a protocol from a static vantage point, i.e. it implicitly assumes that attackers are only capable of passive eavesdropping, which, in many cases, is untrue - attackers are not only able to eavesdrop but to insert or block messages as well as to perform encryption or decryption on messages using its own or acquired keys.

The combination of Casper/FDR, in contrast, does provide a dynamic perspective. Developed by Gavin Lowe [Casper], Casper translates a high-level description of a security protocol into communication sequential processes' (CSP) terminologies that can be fed into the FDR model checker for verification against defined specifications, such as agreement and secrecy. Casper/FDR has been successfully applied to a number a of security protocols [Low2001]

[Low1996]. However, Casper, as represented by its latest version provided by Gavin Lowe [], provides no means to model the Diffie-Hellman exchange.

The "Strand Spaces" advocated by Thayer et al. [] is a mathematics theory that allows such security properties as authentication and secrecy to be expressed and proven in terms of origins and action sequences. [] does not include a model for the Diffie-Hellman exchange, but one of its authors, Jonathan C. Herzog, developed one in 2003 [Her2003]. The model treated the Diffie-Hellman exchange as a function that maps the two parties' public parameters into the resultant session key. For protocols that incorporate the Diffie-Hellman exchange in the conventional way, this model could be of interest.

Other formal methods are also seen in the literature, such as Brackin's Automated Authentication Protocol Analyzer [], the Common Authentication Protocol Language [], and the chi-space [] model. The author, however, has not had an opportunity to study them in more details.

4.4 Trust

The term “trust” is used in many different ways, both in the literature and in the everyday parlance, to the extent that sometimes its use seems to cause more confusion than clarity.

There is a large body of work on trust from a computer science point of view. Starting from the seminal work by Burrows, Abadi, and Needham on the so called BAN logic [Bur1990], there has been a large body of papers analysing the underlying assumptions about the parties' intentions and knowledge in the protocol context, often formulated in terms of trust assumptions; for example, see the summary papers by Meadows [Mea2003] and by Caleiro, Vigan, and Basin [Cal2006]. Another body of work, partly building on protocol analysis and trying to formally model trust in more social context, was created by Yahalom et al. in the early 1990s; for example, see [Gon1990] [Yah1993] [Yah1994]. The later work by Audun Josang, creating a multidimensional concept that models both trust(worthiness) and knowledge [Jøs1996] [Jøs1999] [Jøs2001], was inspired partly by the aforementioned work and partly by the Dempster-Schafer theory of evidence [Dem1968] [Sha1976]. A relatively independent body of work considers how to represent trust relationships in distributed systems; for example, see the work by Blaze et al. [Bla1996] [Bla1999], and, for example, by Ellison [Ell1999], Nikander et al. [Leh1998] [Nik????], and Aura [Aur1999].

From society and social point of view, ability to trust people; i.e., the ability to rely on the benevolence and good intentions of a typical person, is generally considered as a requisite for democracy and working markets [Put1993] [Fuk1995] [OEC2001]. Furthermore, elaborate checks and balances have been developed over generations to institutionalise the trust to some extent. Consequently, the present erosion of trust and growing distrust within the Internet is believed to seriously hamper the development of new communities and marketplaces.

4.5 Privacy

For every new service that is launched and massively adopted, privacy issues will inevitably arise. As such, privacy and anonymity in the context of communications over publish/subscribe networks gains substantial consideration in the technical, procedural, and legal domains. There are various reasons why an end-user would wish to remain anonymous when communicating over a pub/sub network. Firstly, a subscriber might wish to conceal his/her identity when selecting published artifacts, files, and other material, therefore remaining hidden from, e.g., marketing campaigns and unwanted advertisements, directed through inference over the personal preferences expressed in client subscriptions. On the other hand, a publisher might also desire to remain anonymous when publishing articles that might link his/her identity with personal information such as age, market profiles, political ideologies, or even sexual preference.

The aforementioned examples belong to an information privacy scope that is related to the unsanctioned invasion of privacy by, e.g., the government, corporations, and/or individuals, in order to identify or even manipulate sensitive personal information. Alan Westin identifies privacy as “the desire of people to choose freely under what circumstances and to what extent they will expose themselves, their attitude and their behavior to others.” Nowadays, we can define privacy in different horizontally-overlapping domains:

- Physical Privacy – e.g. DNA searching
- Information Privacy – as previously mentioned
- Contextual Privacy – an individual’s fundamental right not to be linked with places, people, locations, and preferences encountered as a result of their daily life; threats include surveillance devices, sensor networks, radio frequency identification (RFID) - tagging systems etc.

Consider a model in which an attacker wishes to reveal the identity of end-users (subscribers or publishers). Defining four legitimate parties in a pub/sub session (i.e. the subscriber, the publisher, the service provider of the subscriber, and the service provider of the publisher), we can define the following privacy protection classes:

- End-user absolute anonymity, where the subscriber/publisher does not expose the user’s identity to, or otherwise his/her identity cannot be exposed by, any other entity
- Subscriber/publisher eponymity only towards peers, where the identity of the subscriber/publisher should only be revealed to the peering publisher/subscriber, respectively
- Publisher/subscriber eponymity only towards the provider, where the identity of the publisher/subscriber should only be revealed to a client’s personal provider
- Publisher/subscriber eponymity only towards a peer subscriber’s/publisher’s provider; same as the above case, except identity information is only revealed to the peer’s provider

To support these privacy classes, an anonymity architecture should make an attacker unable to distinguish between the occasions when a publisher publishes an article or a subscriber selects a publication, and the occasions when (s)he does not. Moreover, any anonymity architecture should protect the physical location of the end-user. No user within the system, nor the system itself, should know from which point an end-user is connected. Even if the relation of the publications and subscriptions with a particular user is not possible, the anonymity system should prevent attackers from linking messages with physical locations. This avoids the provable exposed conditions [Rei1998] in which an attacker can prove the identity of publishers/subscribers to others.

4.5.1 Anonymity Architectures

A theoretical model for ensuring anonymity is the k-Anonymity concept [Sam1998], originally introduced in the context of relational data privacy. It addresses the question of “how a data holder can release its private data with guarantees that the individual subjects of the data cannot be identified whereas the data remain practically useful” [Swe2002].

To provide or improve baseline privacy in the realm of Internet services, several privacy enhancement technologies (PET) have been proposed. Chaum’s Mixes [Cha1981], Stop-and-Go Mixes and MixNets [Kes1998], Crowds [Rei1998], Hordes [Lev2002], Onion Routing [Ree1998], and Mist [Muh2002a] are examples of such anonymity preservation techniques.

Mixes [Cha1981] arguably introduced the notion of anonymous digital communication. The Mix system provides “unlinkability” between sender and receiver. This ensures that while an attacker is able to determine that the sender and receiver are actually sending and/or receiving messages, (s)he cannot determine with whom they are communicating. The system consists of a special mix of nodes which store, mix, and then forward the messages in transit. The sender predetermines the route of the message through one or more mix nodes using a well-defined protocol. A public key cryptography protocol is also used to ensure that any message cannot be tracked by an attacker as it passes through the mix network. In its simplest form (called a *threshold mix*), a mix node waits until it collects a number of messages as input. It then uses its private key to reveal the address of the next mix node (or final destination) and reorders the received and buffered messages by some metric before forwarding them. In that sense, an omnipresent attacker cannot trace a message from its source to its destination without the collusion of the mix nodes.

To provide a mix-network routing protocol, Kesdokan et al. introduced the Free Route and Mix Cascade concepts [Kes1998]. The former gives autonomy to the sender for dynamically choosing the trust path of the mix-nodes, whilst in the latter the routing paths are pre-defined. Mix networks introduce delays due to buffering and mixing and different padding patterns for mixing real and dummy traffic. Continuous mixes attempt to avoid delay issues by introducing fixed delay distributions for buffering and mixing. Mixes became subject to several attacks, such as timing attacks, statistical analyses of message distributions, and statistical analysis of the properties of randomly constructed routes.

Crowds [Rei1998] is a network that consists of voluntarily collaborating nodes. It is based on the idea that the anonymity of a single being can be protected better when that being is moving within a crowd. According to [Rei1998], Crowds' web servers are unable to learn the true source of a request because it is equally likely to have originated from any member of the crowd of potential requestors. Even collaborating crowd members cannot distinguish the originator of a request from a member who is merely forwarding the request on behalf of another. In Crowds, each user (browser) is represented in the system by a "jondo" process. A message that requires user anonymity enters into the Crowd node, its presence is announced via the local jondo, and it is sent to another randomly chosen jondo with probability p or to the actual server with probability 1-p. When the server (or recipient jondo) receives the message, it responds using the same forward path. Crowds can effectively deter traceback attacks and also mitigate collusion attacks if the users randomly select the set of forwarding jondos.

Onion Routing [Ree1998] is an overlay infrastructure for providing anonymous communications over a public network. It supports anonymous connections through three phases: connection setup, data exchange, and connection termination.

In the setup phase, the initiator creates a layered data structure called "onion" which implicitly defines the route path through the network. An onion is recursively encrypted using public key cryptography. The number of encryptions is equal to the number of onion routes that the structure should deliver and process towards the destination. The outer cryptographic control information refers to the first onion router in the path, whilst the inner cryptographic control information refers to the last onion router in the path (i.e. the predecessor to the destination), etc. Each onion router along the route uses its public key to decrypt the entire onion that it receives. This operation exposes the embedded onion, and as a result, the identity of the next onion router. Each onion router pads the embedded onion after decrypting a "cortex" to maintain a fixed size, and sends it to the next onion router. Once the onion reaches the destination, all of the inner control data appears as plaintext. This establishes the anonymous end-to-end connection, and then data can be sent in both directions.

As data are routed through the anonymous end-to-end connection, each onion-router removes one layer of encryption, so the data arrives in plain form at the next recipient. This layering occurs in the reverse order (using different algorithms and keys) for data moving backwards through the connection.

Connection tear-down can be initiated by either end, or in the middle of the path if needed. All of the messages (onions and real data) transferred through the Onion Routing network are identically sized. The messages arrive at an onion router at fixed time intervals. They are mixed to avoid correlation by potential attackers. Additionally, cover traffic in the semipermanent connections between onion-routers deludes external eavesdroppers. As such, Onion Routing can effectively resist traffic analysis.

Hordes [Lev2002] is an anonymity infrastructure that combines elements from Onion Routing and Crowds. It is the first protocol that uses multicast transmission when the destination answers the sender. It includes two phases, the initialization and the transmission phase. In the first phase, Hordes borrows the jondos concept from Crowds, and a public key scheme is used to add authentication services. The sender sends a join-request message to a proxy server, and the proxy authenticates the sender and returns a signed message that contains the multicast address of jondos, and informs the multicast group of the new entry. In the second stage, for the data transmission phase of a message, the sender selects a subset of jondos for the forwarding path and a multicast group address for the reverse path. When a data message is scheduled for transmission, the sender chooses a jondo member of the forwarding subset and sends this message to this peer as an encrypted onion data structure. The chosen jondo then sends this message to another randomly chosen jondo with probability p, or to the receiver with probability 1-p, using encryption layers as well. The receiver replies on the backward path, and for that reason, it sends an acknowledgment as a plaintext message to the multicast group.

A promising system that overcomes some of the previously discussed privacy drawbacks is “the Mist” [Muh2002a]. The Mist handles the problem of routing a message though a network while keeping the sender’s location hidden from intermediate devices (routers, caching elements etc), the receiver, and any potential eavesdroppers. The system consists of a number of routers, known as Mist routers, which are ordered in a hierarchical structure. According to Mist, special routers, called “portals”, are aware of the user’s location, without knowing the corresponding identity, whilst “lighthouse” routers, hereafter referenced as “LIGs”, are aware of the user’s identity without knowing his/her exact location. The key point of the Mist architecture is the distribution of knowledge. Due to its decentralized structure, a possible collusion between the aforementioned Mist routers is extremely difficult since the routers are unaware of each other’s identity. The leaf nodes in the hierarchy (i.e. portals) act as connection points where users can connect to the Mist system.

Let us assume that publisher *A* requires a network service that ensures privacy and data confidentiality. Publisher *A* must first register with the Mist system. The publisher’s device interfaces directly with one of the available portals in the surrounding space. The portal, upon receiving a registration request, replies with a list of its ancestral Mist routers that exist at a higher level within the Mist hierarchy and are willing to act as a LIG (i.e. point of contact) for the user. Subscribers that intend to communicate with publisher *A* have to contact his LIG.

Following LIG selection, a virtual circuit (i.e. a Mist circuit) must be established between publisher *A* and the corresponding LIG. This process, known as “Mist circuit establishment”, aims to entitle publisher *A*’s LIG to authenticate *A* without revealing *A*’s physical location, while hiding, at the same time, from the Portal, *A*’s identity and the designated LIG. Furthermore, the Mist circuit applies a hop-to-hop handle-based routing technique for packet transmission between source and destination nodes and, in combination with data encryption, manages to conceal from intermediary nodes any information related to the identities and location of the communicating parties.

To establish a Mist Circuit, publisher *A* generates a circuit establishment packet and transmits it to the corresponding Portal, without informing the portal of the selected LIG. Upon receiving the packet, the portal assigns a special number, called a handle ID, to the communication session with publisher *A*. Thereafter, the portal encloses the assigned handle ID in the received packet and forwards it to its Mist Router ancestor. As the packet propagates through the Mist hierarchy, each LIG Router attempts to decrypt the payload using their private key. If the decryption fails, the particular router infers that it is not the recipient of this packet and forwards it to the next router in the hierarchy. This process is repeated by each intermediate Mist router until the packet reaches its final destination. In the case that the decryption of the payload is successful, this indicates that *A* has selected the current Mist Router to act as his LIG. The LIG responds to publisher *A* and confirms the registration. From this point, a secure circuit is established through which publisher *A* can communicate securely with his LIG. Note that even though the LIG of publisher *A* can infer that his/her physical location is underneath a given Mist router *Y*, it is very difficult if not impossible to determine *A*'s exact position. Following circuit establishment, the LIG undertakes the role of representing the end-user.

An issue that has to be addressed is the detection of the user's LIG. A public directory (e.g. a Lightweight Directory Access Protocol (LDAP) server) or a web server can be used for this purpose. Let us assume now that subscriber *B* intends to communicate with publisher *A* and both have previously established a Mist circuit with LIGs *B'* and *A',* respectively. Subscriber *B* transmits to his/her LIG a packet indicating that (s)he wants to set up a pub/sub service with publisher *A*. LIG *B* verifies that the originator of the message is *B*, locates the LIG of publisher *A*, and performs the initialization procedure for connection establishment. As soon as the communication path is established, users *A* and *B* are able to communicate. Note that the intermediate routers are unaware of the two ends of the communication. Moreover, it is impossible for subscriber *B* to determine the location of A and vice versa.

5 References

[Aba1998] M. Abadi and A. D. Gordon, *A Calculus for Cryptographic Protocols — The Spi Calculus*, Research report SRC 149, 1998, pp. 110.

[Abr2005] I. Abraham, C. Gavoille, D. Malkhi, N. Nisan, and M. Thorup, "Compact Name Independent Routing with Minimum Stretch," *In Proc. The sixteenth annual ACM symposium on Parallelism in algorithms and architectures*, 2004, pp. 20-24.

[Ada2005] D. Adams, J. Nicholas, and W. Siadak, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised),* IETF RFC 3973, Jan. 2005.

[And2003] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing Internet denial-of-service with capabilities," *In Proc. of the 2th ACM Workshop on Hot Topics in Networks (HotNets)*, 2003.

[And2007] D. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Holding the Internet Accountable," *In HotNets-VI*, Nov. 2007, pp. 7–12.

[Are2005a] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, *DNS Security Introduction and Requirements*, IETF RFC 4033, Mar. 2005.

[Are2005b] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, *Resource Records for the DNS Security Extensions*, IETF RFC 4034, Mar. 2005.

[Are2005c] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose, *Protocol Modifications for the DNS Security Extensions*, IETF RFC 4035, Mar. 2005.

[Arg2005] K. Argyraki and D. Cheriton, "Network capabilities: The good, the bad and the ugly," *In Proc. of the 4th ACM Workshop on Hot Topics in Networks (HotNets)*, 2005.

[Ari2003] M. Arias, L. J. Cowen, K. A. Laing, R. Rajaraman, and O. Taka, "Compact Routing with Name Independence," *In Proc. Fifteenth annual ACM symposium on Parallel algorithms and architectures*, 2003, pp. 184-192.

[Aur1999] T. Aura, "Distributed access-rights management with delegation certificates," Secure Internet Programming, 1999, pp. 211–235.

[Avr2004] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, "Highly Secure and Efficient Routing," *In Proc. of IEEE INFOCOM 2004*, Hong Kong, Mar. 2004.

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

[Awe2005] B. Awerbuch, R. G. Cole, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, *Dynamics of learning Algorithms for the On-Demand Secure Byzantine Routing Protocol*, JHU Applied Physics Laboratory Technical Report No. VIC-05-088, Nov. 2005.

[Bac2002] J. Bacon, K. Moody, and W. Yao, "A model of OASIS role-based access control and its support for active security," *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, issue 4, Nov. 2002, pp. 492-540.

[Bac2005] J. Bacon, D. M. Eyers, K. Moody, and L. I. W. Pesonen, "Securing publish/subscribe for multi-domain systems," *in Middleware, ser. Lecture Notes in Computer Science, G. Alonso, Ed.*, vol. 3790, Springer, 2005, pp. 1–20.

[Bal1993] A. Ballardie, J. Crowcroft and P. Francis, "Core Based Trees (CBR) - An Architecture for scalable inter-domain multicast routing," *Computer Communications Review*, vol. 23, no. 4, 1993, pp. 85-95.

[Bal2005] H. Ballani, Y. Chawathe, S. Ratnasamy, T. Roscoe, and S. Shenker, "Off by default!", *In Proc. of the 4th ACM Workshop on Hot Topics in Networks (HotNets)*, 2005.

[Bal2005] R. Baldoni, R. Beraldi, S. T. Piergiovanni, and A. Virgillito, "On the modelling of publish/subscribe communication systems," *Concurrency and Computation: Practice and Experience*, vol. 17, no. 12, Oct. 2005, pp. 1471-1495.

[Ban2002] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," *Proc. of the conference on Applications, technologies, architectures, and protocols for computer communications*, 2002, pp. 205-217.

[Bau2006] S. J. Bauer, P. Faratin, and Robert Beverly, "Assessing the assumptions underlying mechanism design for the Internet," *In Proc. Workshop on the Economics of Networked Systems (NetEcon06)*, 2006.

[Bel2003] A. Belokosztolszki, D. M. Eyers, P. R. Pietzuch, J. Bacon and K. Moody, “Role-based access control for publish/subscribe middleware architectures,” *Proc. of the 2nd international workshop on Distributed event-based systems (DEBS’03)*, 2003.

[Ben2006] Y. Benkler. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press New Haven, CT, USA, 2006.

[Big2004] N. W. Biggart and R. Delbridge, "Systems of Exchange," *Academy of Management Review*, vol. 29, no. 1, 2004, pp. 28–49.

[Bla1996] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," *Proc. of the 1996 IEEE Symposium on Security and Privacy*, 1996, pp. 164–173.

[Bla1999] M. Blaze, J. Feigenbaum, and A.D. Keromytis, "KeyNote: Trust Management for Public-Key Infrastructures," *Lecture Notes in Computer Science*, 1999.

[Blu2001] M. Blumenthal and D. Clark, "Rethinking the design of the Internet: The End-to-End arguments vs. The Brave New World," *ACM Transactions on Internet Technology 2001*, vol. 1, issue 1, 2001, pp. 70-109.

[Boi2000] R. Boivie, N. Feldman, and C. Metz, "Small group multicast: a new solution for multicasting on the Internet," *IEEE Internet Computing*, vol. 4, no. 3, 2000, pp. 75-79.

[Bra1998] S. Brackin, "Evaluating and improving protocol analysis by automatic proof," *In Proc. of the 11th IEEE* Computer Security Foundations Workshop*, IEEE Computer Society Press,* June 1998.

[Bra2006] A. Brady and L. Cowen, "Compact routing on power-law graphs with additive stretch," *In ALENEX*, 2006.

[Bri2004] R. Briscoe. "The implications of pervasive computing on network design," *BT Technology Journal*, vol. 22, no. 3, July 2004, pp. 170–190.

[Bri2005] B. Briscoe, A. Jacquet, C. D. Cairano-Gilfedder, A. Salvatori, A. Soppera and M. Koyabe, "Policing Congestion Response in an Internetwork using Re-feedback," *ACM SIGCOMM Computer Communications Review*, vol. 35, issue 4, Sep. 2005, pp. 277-288.

[Bri2007] B. Briscoe, "Flow Rate Fairness: Dismantling a Religion," *ACM Computer Communications Review*, vol. 37, issue 2, Apr 2007, pp. 63-74.

[Bur1990] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, issue 1, 1990, pp. 18-36.

[But2002] L. Buttyán and J. P. Hubaux, "A Formal Analysis of Syverson's Rational Exchange Protocol," *IEEE Computer Security Foundations Workshop*, 2002.

[Bye1998] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 4, Oct. 1998, pp. 56-67.

[Cae2006] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker, "ROFL: Routing on Flat Labels," *In ACM SIGCOMM*, Sep. 2006, pp. 363–374.

[Cai2002] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, *Internet Group Management Protocol, Version 3,* IETF RFC 3376, Oct. 2002.

[CAI2008] CAIDA, NeTS-NR Toward Mathematically Rigorous Next-Generation Routing Protocols for Realistic Network Topologies [online], available at: http://www.caida.org/funding/nets-nr/ [Accessed 13th June 2008].

[Cal2006] C. Caleiro, L. Viganò, and D. Basin, "On the Semantics of Alice&Bob Specifications of Security Protocols," *Theoretical Computer Science*, vol. 367, no. 1–2, 2006, pp. 88-122.

[Cal2007] K. Calvert, J. Griffioen, and L. Poutievski, "Separating Routing and Forwarding: A Clean-Slate Network Layer Design," *In Proc. of Broadnets 2007*, 2007.

[Cao2004] F. Cao, and J. P. Singh, "Efficient Event Routing in Content-based Publish-Subscribe Service Networks," *In Proc. of the 23rd Conference on Computer Communications (IEEE INFOCOM* 2004)*,* Mar. 2004.

[Cao2005] F. Cao and J. P. Singh, "MEDYM: Match-Early and Dynamic Multicast for Content-Based Publish-Subscribe Service Networks," *In Proc. of the Fourth international Workshop on Distributed Event-Based Systems (DEBS) (ICDCSW'05)*, vol. 04, Washington DC, USA, June 2005, pp. 370-376.

[Cap2003] M. Caporuscio, A. Carzaniga, and A. L. Wolf, "Design and evaluation of a support service for mobile, wireless publish/subscribe applications," *IEEE Transactions on Software Engineering*, vol. 29, no. 12, Dec. 2003, pp. 1059–1071.

[Car1998] A. Carzaniga, "Architectures for an Event Notification Service Scalable to Wide-area Networks," doctoral dissertation, Politecnico di Milano, Dec. 1998.

[Car2001] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Design and evaluation of a wide-area event notification service," *ACM Transactions on Computer Systems*, vol. 19, no. 3, Aug. 2001, pp. 332–383.

[Cas2002a] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *in Proc. of USENIX Operating System Design and Implementation(OSDI)*, Boston, MA, Dec. 2002.

[Cas2002b] M. Castro, P. Druschel, A.-M. Kermarrec, and A.I.T. Rowstron, "Scribe: a large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, 2002, pp. 1489-1499.

[Cas2003] M. Castro, M. B. Jones, A.-M. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman, "An evaluation of scalable application-level multicast built using peer-to-peer overlays," *Proc. of the IEEE INFOCOM*, vol. 2, 2003, pp. 1510-1520.

[CFP2005] D. Trossen and C. Fine (Eds), "Value Chain Dynamics in the Communication Industry," *MIT Communications Futures Program*, 2005.

[CFP2008] Communications Futures Program [online], 2008, available at: http://cfp.mit.edu [Accessed 19th June 2008].

[Cha1981] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 4, no. 2, Feb. 1981.

[Cha2002] D. Chang, R. Govindan, and J. Heidemann, "An empirical study of router response to large BGP routing table load," *SIGCOMM IMW*, 2002.

[Che1985] D. R. Cheriton and S. E. Deering, "Host groups: A multicast extension for datagram internetworks," *Proc. of the Data Communication Symposium*, vol. 9, 1985, pp. 172-179.

[Che1993] K. Cheun and W. E. Stark, "Optimal Selection of Reed-Solomon Code Rate and the Number of Frequency Slots in Asynchronous FHSS-MA Networks," *IEEE Transaction on Communications*, vol. 41, no. 2, Feb. 1993, pp. 307-311.

[Chu2002] Y.-H. Chu, S. G. Rao, S. Seshan, and H. Zhang, "A Case for End System Multicast," *Journal of Selected Areas in Communications*, vol. 20, no. 8, 2002, pp. 1456-1471.

[Cla2002] D. D. Clark, J. Wroclawski, K. R. Sollins, R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," *Proc. of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, 2002.

[Cla2003] D. D. Clark, K. Sollins, J. Wroclawski, T. Faber. "Addressing Reality: An Architectural Response to Real-World Demands on the Evolving Internet," *In Proc. of the ACM SIGCOMM workshop on Future directions in network architecture*, New York, NY, USA, 2003, pp. 247-257.

[Cla2005] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," *IEEE/ACM Transactions on Networking*, vol. 13, issue 3, 2005, pp. 462-475.

[Cla2006] D. Clark, B. Lehr, S. Bauer, P. Faratin, R. Sami, and J. Wroclawski, "Overlay Networks and the Future of the Internet," *COMMUNICATIONS & STRATEGIES*, no. 63, 2006, p. 1.

[Cla2007] D. D. Clark and M. S. Blumenthal, "End-to-end Arguments in Application Design: The Role of Trust," *In Proc. of TPRC*, 2007.

[Cor2007] A. Corman, P. Schachte, and V. Teague, "QUIP: A Protocol For Securing Content in Peer-To-Peer Publish/Subscribe Overlay Networks," *In Proc. Thirtienth Australasian Computer Science Conference (ACSC2007)*, Ballarat, Australia, 2007, pp. 35-40.

[Cou2000] C. Courcoubetis, F. P. Kelly, V. A. Siris and R. Weber, "A study of simple usage-based charging schemes for broadband networks," *Telecommunication systems*, vol. 15, issue 3-4, 2000, pp. 323-343.

[Cow1999] LJ Cowen, "Compact routing with minimum stretch," *in Proc. Tenth annual ACM-SIAM symposium on Discrete algorithms*, 1999, pp. 255-260.

[Cra2002] F. Crazzolara and G. Milicia, "Developing Security Protocols in X-Spaces," *In Proc. of the* 7th Nordic Workshop on Secure IT Systems (NordSec), Nov. 2002.

[Cro2003] J. Crowcroft, S. Hand, R. Mortier, T. Roscow, and A. Warfield, "Plutarch: An Argument for Network Pluralism," *in Proc. SIGCOMM Workshops*, Aug. 2003.

[Cui2003] J.-H. Cui, L. Lao, D. Maggiorini, and M. Gerla, "BEAM: a distributed aggregated multicast protocol using bi-directional trees," *IEEE International Conference on Communications*, vol. 1, 2003, pp. 689-695.

[DeC2005] D. S. J. DeCouto, D. Aguayo, J. Bicket, and R. Moris, “A High-Throughput Path Metric for Multi-Hop Wireless Routing,” *Wireless Networks,* Spinger, Netherlands, vol. 11, no. 4, July 2005, pp. 419-434.

[Dee1991] S. Deering, "Multicast Routing in a Datagram Network," doctoral dissertation, Dept. of Computer Science, Stanford University, 1991.

[Dem1968] A. P. Dempster, "A generalization of Bayesian inference," *Journal of the Royal Statistical Society*, vol. 30, 1968, pp. 205-247.

[Din2004] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," *In Proc. of the 13th USENIX Security Symposium*, Aug. 2004.

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

[Dol1983] D. Dolev and A. C. Yao, "On the security of public-key protocols," *IEEE Transactions on Information Theory*, vol. 2, no. 29, Mar. 1983, pp. 198-208.

[Dou2002] J. Douceur, “The Sybil attack,” *Proc. of IPTPS*, 2002, pp. 251-260.

[Ell1999] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI Certificate Theory," 1999.

[Fab1999] F. Javier Thayer Fabrega, J. C. Herzog, and J. D. Guttman, "Strand Spaces: Proving Security Protocols Correct," *Journal of Computer Security*, 1999, pp. 191-230.

[Far2002] S. Farrell and R. Housley, *An Internet Attribute Certificate Profile for Authorization*, IETF RFC 3281, April 2002.

[Far2007] P. Faratin, "Economics of Overlay Networks: An Industrial Organization Perspective on Network Economics," *Proc. of the Joint Workshop on The Economics of Networked Systems and Incentive-Based Computing (NetEcon+IBC) in conjunction with ACM Conference on Electronic Commerce (EC'07)*, 2007.

[Fea2004] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. v. d. Merwe, "The case for separating routing from routers," *In FDNA ’04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, New York, NY, USA, 2004. pp. 5-12.

[Feh2002] E. Fehr, U. FIschbacher, and S. Gächter, "Strong Reciprocity, Human Cooperation and the Enforcement of Social Norms," *Human Nature*, vol. 13, 2002, pp. 1-25.

[Fei2001] J. Feigenbaum, C. Papadimitriou, and Scott Shenker, "Sharing the Cost of Multicast Transmissions," *Journal of Computer and System Sciences*, vol. 63, 2001, pp. 21-41.

[Fei2006] J. Feigenbaum, V. Ramachandran, and M. Schapira, "Incentive-Compatible Interdomain Routing," *in Proc. of the 7th Conference on Electronic Commerce*, ACM Press, New York, 2006, pp. 130-139.

[Fei2007] J. Feigenbaum, D. Karger, V. Mirrokni, and R. Sami, "Subjective-Cost Policy Routing," *Theoretical Computer Science*, vol. 378, 2007, pp. 175-189.

[Fen2006] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised),* IETF RFC 4601, Aug. 2006.

[Fie2004] C. Fiege, A. Zeidler, A. Buchmann, R. Kilian-Kehr, and G. Mühl, “Security Aspects in Publish/Subscribe Systems,” *in Proc. Of Third International Workshop on Distributed Event-Based Systems (DEBS)*, Edinburgh, Scotland, UK, May 2004.

[Fin1998] C. Fine, *Clockspeed: Winning Industry Control in the Age of Temporary Advantage*, Sloan School of Management, 1998.

[Fra2006] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, “Network Coding: An Instant Primer,” *ACM SIGCOM Computer Communication Review*, vol. 36, no. 1, Jan. 2006, pp. 63-68.

[Fra2007] C. Fraguoli, D. Katabi, A. Markopoulou, M. Medard, and H. Rahul, “Wireless Network Coding, Opportunities and Challenges,” *Military Communications Conference, IEEE MILCOM 2007,* Oct. 2007, pp. 1-8.

[Fuk1995] F. Fukuyama, "Trust: the social virtues and the creation of prosperity," Free Press, 1995.

[Ful1993] V. Fuller, T. Li, J. Yu and K. Varadhan, *Classless Inter-Domain Routing (CIDR)*, IETF RFC 1519, Sep. 1993.

[Gan2004] P. Ganesan, K. Gummadi, H. Garcia-Molina, "Canon in G Major: Designing DHTs with Hierarchical Structure," *ICDCS*, Mar. 2004.

[Gav1996] C. Gavoille and S. P´erennes, "Memory requirement for routing in distributed networks," *In Proc. of the 15th PODC, ACM*, 1996.

[Gav2001] C. Gavoille and M. Gengler, "Space-Efficiency for Routing Schemes of Stretch Factor Three," *Journal of Parallel and Distributed Computing*, vol. 61, issue 5, 2001, pp. 679-687.

[Gib1999a] R. J. Gibbens and F. P. Kelly, "Resource pricing and the evolution of congestion control," *Automatica*, vol. 35, 1999, pp. 1969-1985.

[Gib1999b] R. J. Gibbens and F. P. Kelly, "Distributed connection acceptance control for a connectionless network," *in Proc. of Teletraffic Engineering in a Competitive World (Editors P. Key and D. Smith)*, ITC16, Elsevier, Amsterdam, 1999, pp. 941-952.

[Gil2006] S. Gillett, B. Lehr, C. A. Osorio, and M. Sirbu, *Measuring the Economic Impact of Broadband Deployment*, Final Report prepared for the U.S. Department of Commerce, Economic Development Administration, 2006.

[Gin2000] H. Gintis, "Strong Reciprocity and Human Sociality," *Journal of Theoretical Biology*, vol. 206, no. 2, 2000, pp. 169–179.

[Gon1990] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," *Proc. 1990 IEEE Symposium on Research in Security and Privacy*, 1990, pp. 234-248.

[Gri2001] M. Gritter and D. R. Cheriton, "An Architecture for Content Routing Support in the Internet," *In USITS’01: Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems*, Berkeley, CA, USA, 2001, pp. 37-48.

[Gui2005] A. Guitton and J. Moulierac, "Scalable Tree Aggregation for Multicast," *In proc. of 8th International Conference on Telecommunications (ConTEL)*, Zagreb, Croatia, June 2005.

[Guo2005] H. Q. Guo, L. H. Ngoh, and W. C. Wong, "DINloop based inter-domain multicast with MPLS," In Proc. of 24th IEEE International Conference on Performance, Computing, and Communications, IPCCC 2005, Phoenix, USA, Apr. 2005.

[Gur2005] A. Gurtov, D. Korzun, and P. Nikander, *Hi3: An efficient and secure networking architecture for mobile hosts*, Technical Report TR-2005-2, HIIT, June 2005.

[Haa2004] C. Haack and A. Jeffrey, "Pattern-matching Spi-calculus," *Proc. FAST'04*, IFIP series 173, 2004, pp. 193-205.

[Hak1971] S. L. Hakimi, "Steiner's problem in graphs and its implications," *Networks*, vol. 1, 1971, pp. 113-133.

[Han2004] M. Handley and A. Greenhalgh, "Steps Towards a DoS-resistant Internet Architecture," *In FDNA '04: Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*, New York, NY, USA, 2004, pp. 49-56.

[Han2005] M. Handley and A. Greenhalgh, "The Case for Pushing DNS," *In Proc. of the 4th ACM Workshop on Hot Topics in Networks (HotNets)*, 2005.

[He2006] L. He and J. Walrand, "Pricing and Revenue Sharing Strategies for Internet Service Providers," *IEEE JSAC*, May 2006.

[Her1997] S. Herzog, S. Shenker, and D. Estrin, "Sharing the "cost" of multicast trees: an axiomatic analysis," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, Dec. 1997, pp. 847-860.

[Her2003] J. C. Herzog, "The Diffie-Hellman Key-Agreement Scheme in the Strand-Space Model," *In Proc. of the 16th IEEE Computer Security Foundations Workshop*, 2003.

[Him2007] P. Himanen, "Finnish dream: An Innovation Report," 2nd ed. (in Finnish), 2007.

[Hin2006] R. Hinden and S. Deering. *IP Version 6 Addressing Architecture*, IETF RFC 4291, Feb. 2006.

[Ho2005] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, Oct. 2005, pp. 4413-4430.

[Hol2006] H. Holbrook and B. Cain, *Source-Specific Multicast for IP,* IETF RFC 4607, Aug. 2006.

[Hol2008] R. Hollingsworth and K. Müller, "Transforming socio-economics with a new epistemology," *Socio-Economic Review*, vol. 6, 2008, pp. 395–426.

[Hu2002] Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense Against Wormhole Attacks in Wireless Ad Hoc Networks," *Proc. of IEEE INFOCOM*, 2003.

[Hu2004] Y. Hu and A. Perrig, "A Survey of Secure Wireless Ad Hoc Routing," *IEEE Security & Privacy*, 2004, pp. 28-39.

[Hui1999] A. Huima, "Efficient Infinite-State Analysis of Security Protocols," *Workshop on Formal Methods and Security Protocols*, 1999.

[Hui2007] F. Huici and M. Handley, "An edge-to-edge filtering architecture against DoS," *ACM SIGCOMM Computer Communication Review 2007*, vol. 37, issue 2, ACM Press New York, NY, USA, 2007, pp. 37-50.

[Jøs1996] A. Jøsang, "The right type of trust for distributed systems," *Proc. of the 1996 workshop on New security paradigms*, 1996, pp. 119-131.

[Jøs1999] A. Jøsang, "An algebra for assessing trust in certification chains," *Proc. of the Network and Distributed Systems Security Symposium (NDSS99)*, The Internet Society, 1999.

[Jøs2001] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, vol. 9, issue 3, 2001, pp. 279–311.

[Kat2006] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XOR in the Air," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, Oct. 2006, pages 243-254.

[Kat2007] S. Katti and D. Katabi, *MIXIT: The Network Meets the Wireless Channel*, MIT Computer Science and Artificial Intelligence Laboratory Technical Report, Sep. 2007.

[Kel1998] F. P. Kelly, A. Maulloo and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, 1998, pp. 237-252.

[Ker2002] A. D. Keromytis, V. Misra, and D. Rubenstein, "SOS: secure overlay services," *In Proc. of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM Press, New York, NY, USA, 2002, pp. 61-72.

[Kes1998] D. Kesdogan, J. Egner, and R. Buschkes, "Stop-and-go MIXes Providing Probabilistic Security in an Open System," *2nd International Workshop on Information Hiding*, 1998.

[Key2006] P. Key and L. Massoulié, "Combining Multipath Routing and Congestion Control for Robustness," *40th Conference on Information Sciences and Systems (CISS 2006)*, March 2006, pp. 345-350.

[Khu2005] H. Khurana, "Scalable Security and Accounting Services for Content-based Publish/Subscribe Systems," *ACM Symposium on Applied Computing*, 2005.

[Kop2007] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," *In SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, 2007, pp. 181-192.

[Kri2004] D. Krioukov, K. Fall, and X. Yang, "Compact Routing on Internet-like Graphs," *In Proc. INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004, pp. 208-219.

[Kri2007] D. Krioukov, kc claffy, K. Fall and A. Brady, "On Compact Routing for the Internet," *In ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, 2007, pp. 41-52.

[Lak2006] K. K. Lakshminarayanan, I. Stoica, S. Shenker, and J. Rexford, *Routing as a Service*, Technical report, UC Berkeley, 2006.

[Leh1998] I. Lehti and P. Nikander, "Certifying trust," *Proc. of the Practice and Theory in Public Key Cryptography (PKC)98*, 1998.

[Lev2002] B. N. Levine and C. Shields, "Hordes: A multicast-based protocol for anonymity", *J. of Computer Sec.*, vol. 10, issue 3, 2002, pp. 213-240.

[Lou1989] K. Lougheed and Y. Rekhter, *A Border Gateway Protocol (BGP)*, IETF RFC 1105, June 1989.

[Low1996] G. Lowe, "Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR," *in Tools and* Algorithms for the Construction and Analysis of Systems*, Springer-Verlag, 1996, pp. 147-166.*

[Low2001] G. Lowe, P. Broadfoot, and M. L. Hui, "Casper A Compiler for the Analysis of Security Protocols," [online] Dec. 2001, available at: http://web.comlab.ox.ac.uk/people/gavin.lowe/Security/Casper/ [Accessed 23th June 2008].

[Lub2002] M. Luby, "LT Codes," *in Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science,* 2002, pp. 271-280.

[Mac1995] J. MacKie-Mason and H. Varian, "Pricing congestible network resources," *IEEE Journal on Selected Areas in Communication*, vol. 13, 1995, pp. 1141-1149.

[Mac2005] D. J. C. MacKay, "Capacity Approaching Codes Design and Implementation, Fountain Codes," *IEEE Proc.-Commun.*, vol. 156, no. 6, Dec. 2005, pp. 1062-1068.

[Mea2001] C. Meadows, "A formal framework and evaluation method for network denial of service," *In Proc. of the 12th IEEE Computer Security Foundations Workshop*, IEEE Computer Society Press, vol. 9, no. 1, 2001, pp. 47–74.

[Mea2003] C. Meadows, "Formal methods for cryptographic protocol analysis: emerging issues and trends," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, 2003, pp. 44–54.

[Mey2007] D. Meyer, L. Zhang, and K. Fall, *Report from the IAB Workshop on Routing and Addressing*, IETF RFC 4984, Sep. 2007.

[Mil1997] J. K. Millen, "CAPSL: Common Authentication Protocol Specification Language," The MITRE Corporation, 1997.

[Mil2001] J. Millen and V. Shmatikov, "Constraint solving for bounded-process cryptographic protocol analysis," *in 8th ACM Conference on Computer and Communication Security*, Nov. 2001, pp. 166–175.

[Mit2004] M. Mitzenmacher, "Digital Fountains: Survey and Look Forward," *IEEE Information Theory Workshop 2004,* Oct. 2004, pp. 271-276.

[Mos2006] R. Moskowitz and P. Nikander, *Host Identity Protocol (HIP) Architecture*, IETF RFC 4423, May 2006.

[Moy1994] J. Moy, *Multicast Extensions to OSPF,* IETF RFC 1584, Mar. 1994.

[Muh2002a] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments," *in Proc. Intl. Conf. of Distributed Comp. Syst.*, 2002.

[Müh2002b] G. Mühl, "Large-Scale Content-Based Publish/Subscribe Systems," doctoral dissertation, Darmstadt University of Technology, Sep. 2002.

[Mus2008] J. Musacchio, G. Schwartz, and J. Walrand, "A Two-Sided Market Analysis of Provider Investment Incentives with an Application to the Net Neutrality Issue," *Review of Network Economics*, to be published, 2008.

[Nik1999] P. Nikander, "An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems," doctoral dissertation, Helsinki University of Technology, Mar. 1999.

[Nis1999] N. Nisan and A. Ronen, "Algorithmic Mechanism Design," *in Proc. 31st Annual ACM Symposium on Theory of Computing*, Atlanta, GA, 1999, pp. 129-140.

[Odl1999] A. Odlyzko, "Paris metro pricing for the internet," *Proc. of the 1st ACM conference on Electronic commerce*, 1999, pp. 140-147.

[Odl2003] A. Odlyzko, "The case against micropayments," *in Proc. of Financial Cryptography: 7th International Conference, FC 2003*, Guadeloupe, French West Indies, Jan. 2003.

[OEC2001] OECD "The Well-being of Nations: the Role of Human and Social Capital," 2001.

[Pap2001] C. Papadimitriou, "Algorithms, Games, and the Internet," *in Proc. of STOC'01*, July 6-8, Crete, Greece, 2001.

[Pap2004] I. Papaefstathiou and C. Manifavas, "Evaluation of Micropayment Transaction Costs," *Journal of Electronic Commerce Research*, vol. 5, no. 2, 2004, pp. 99-113.

[Par2007] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y. C. Hu, "Portcullis: protecting connection setup from denial-of-capability attacks," *In Proc. of ACM Sigcomm 2007*, 2007.

[Pas1998] J. C. Pasquale, G. C. Polyzos, and G. Xylomenos, "The multimedia multicast problem," *Multimedia Systems*, vol. 6, no. 1, 1998, pp. 43-59.

[Pau2002] P. Paul and S. V. Raghavan, "Survey of multicast routing algorithms and protocols," *Proc. of the Fifteenth International Conference on Computer Communication 2002*, Mumbai, India, Aug. 2002.

[Pen2001] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: an application level multicast infrastructure," *Proc. of the USENIX Symposium on Internet Technologies and Systems*, vol. 3, 2001.

[Per1998] R. Perlman, "Network layer protocols with Byzantine robustness," doctoral dissertation, MIT-LCS-TR-429, 1998.

[Per2002] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA Broadcast Authentication Protocol," CryptoBytes, vol. 5, no. 2, Summer/Fall 2002, pp. 2-13.

[Pes2005] L. I. W. Pesonen and J. Bacon, "Secure event types in content-based, multi-domain publish/subscribe systems," *in SEM '05: Proc. of the 5th international workshop on Software engineering and middleware*, New York, NY, USA: ACM Press, Sept. 2005, pp. 98-105.

[Pes2006] L. I. W. Pesonen, D. M. Eyers, and J. Bacon, "A capabilities-based access control architecture for multi-domain publish/subscribe systems," *in Proc. of the Symposium on Applications and the Internet (SAINT 2006). Phoenix, AZ, Jan. 2006, pp. 222-228.*

[Pes2007a] L. I. W. Pesonen, D. M. Eyers, and J. Bacon, "Access Control in Decentralised Publish/Subscribe Systems," *Journal on Networks*, vol. 2, no. 2, Apr. 2007.

[Pes2007b] L. I. W. Pesonen, D. M. Eyers, and J. Bacon, "Encryption-Enforced Access Control in Dynamic Multi-Domain Publish/Subscribe Networks," *In Proc. of the International Conference on Distributed Event-Based Systems (DEBS'07)*, ACM Press, June 2007, pp. 104-115.

[Pie2002] P. R. Pietzuch and J. M. Bacon. "Hermes: A Distributed Event-Based Middleware Architecture," *Proc. of the 1st International Workshop on Distributed Event-Based Systems (DEBS'02)*, July 2002, pp. 611-618.

[Pie2004] P. R. Pietzuch, "Hermes: A Scalable Event-Based Middleware," doctoral dissertation, Computer Laboratory, Queens' College, University of Cambridge, Feb. 2004.

[Pit2008] M. Pitkänen and J. Ott, "Enabling Opportunistic Storage for Mobile DTNs," *Elsevier,* to be published, 2008.

[Put1993] R. Putnam, R. Leonardi, and R. Nanetti, "Making democracy work: civic traditions in modern Italy," Princeton University Press, Princeton, NJ, 1993.

[Qui2001] B. Quinn and K. Almeroth, *IP Multicast Applications: Challenges and Solutions,* IETF RFC 3170, Sep. 2001.

[Rai2006] C. Raiciu, D. S. Rosenblum, and M. Handley, "Revisiting Content-Based Publish/Subscribe," *In Proc. of the ICDCS Workshop on Distributed Event Based Systems (DEBS)*, Lisbon, Portugal, July 2006.

[Ram2004a] V. Ramasubramanian and E. G. Sirer, "Beehive: Exploiting Power Law Query Distributions for O(1) Lookup Performance in Peer to Peer Overlays," *Symposium on Networked Systems Design and Implementation*, San Francisco CA, Mar. 2004.

[Ram2004b] V. Ramasubramanian, E. G. Sirer, "The Design and Implementation of a Next Generation Name Service for the Internet," *SIGCOMM*, 2004.

[Rat2001a] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *In Proceedings of ACM SIGCOMM*, Aug. 2001.

[Rat2001b] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-level multicast using content-addressable networks," *Lecture Notes in Computer Science*, 2233, 2001, pp. 14-29.

[Rat2005] S. Ratnasamy, S. Shenker, and S. McCanne, "Towards an Evolvable Internet Architecture," *In Proc. of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM Press New York, NY, USA, 2005, pp. 313-324.

[Rat2006] S. Ratnasamay, A. Ermolinskiy, and S. Shenker, "Revisiting IP Multicast," *In Proc. of the ACM SIGCOMM 2006*, Pisa, Italy, Sep. 2006.

[Ree1960] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," *SIAM Journal of Applied Math.*, vol. 8, 1960, pp. 300-304.

[Ree1998] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE JSAC*, vol. 16, issue 4, 1998, pp. 482-494.

[Rei1998] M. K. Reiter and A. D. Rubin. "Crowds: anonymity for web transactions," *ACM Trans. Information Systems Security*, vol. 1, issue 1, 1998, pp. 66-92.

[Ria2002] A. Riabov, Z. Liu, J. L. Wolf, P. S. Yu, and L. Zhang, "Clustering algorithms for content-based publication-subscription systems," *In Proc. of the 22nd IEEE International Conference on Distributed Computing Systems*, July 2002.

[Roa2002] A. B. Roach, *Session Initiation Protocol (SIP)-Specific Event Notification,* IETF RFC 3265, June 2002.

[Rob2002] S. Robinson, "Beyond Reed-Solomon: New Codes for Internet Multicasting Drive Silicon Valley Start Up," *SIAM News*, vol. 35, no. 4, May 2002.

[Ros2002] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, *SIP: Session Initiation Protocol,* IETF RFC 3261, June 2002.

[Row2001] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems," *In Proc. of Middleware 2001*, Springer, Nov. 2001, pp. 329–250.

[Rya2001] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe, *Modelling and Analysis of Security Protocols*, Addison Wesley, 2001.

[Sal1984] J. H. Saltzer, D. P. Reed and D. D. Clark, “End-to-End Arguments in System Design,” *ACM Transactions on Computer Systems*, vol. 2, issue 4, Nov. 1984, pp. 277-288.

[Sam1998] P. Samarati and L. Sweeney, “Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression,” *Proc. IEEE Symposium on Research in Security and Privacy*, 1998.

[Sco2006] J. Scott, P. Hui, J. Crowcroft, and C. Diot, "Haggle: A Networking Architecture Designed Around Mobile Users," *IFIP WONS 2006*, Les Menuires, France, 2006.

[Sha1976] G. Shafer, *A mathematical theory of evidence*, Princeton University Press, Princeton, NJ, 1976.

[She1995] S. Shenker, "Some fundamental design decisions for the future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, 1995, pp. 1176-1188.

[Sho2001] P. Sholtz, “Transaction Costs and the Social Cost of Online Privacy,” *First Monday*, vol. 6, no. 5, May 2001.

[Skl2001] B. Sklar, *Digital Communications: Fundamentals and Applications*, Second Edition, Chapter 8, Prentice Hall PTR, 2001, pp. 437-461.

[Sol2007] K. Sollins and D. Trossen, "From Visions to Understanding," presentation to the Communications Futures Program, 2007.

[Sri2005] M. Srivatsa and L. Liu, "Securing publish-subscribe overlay services with EventGuard," *Proc. of the 12th ACM conference on Computer and communications security*, Nov. 2005.

[Ste2000] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. *Stream Control Transmission Protocol*, IETF RFC 2960, Oct. 2000.

[Sto2001] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *In Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ACM Press, 2001, pp. 149-160.

[Sto2002] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," *In Proc. of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM Press, New York, NY, USA, 2002, pp 73-86.

[Sto2004] K. Lakshminarayanan, I. Stoica, S. Shenker, and J. Rexford, *Routing as a Service*, Tech. Rep. UCB-CS-04-1327, UC Berkeley, 2004.

[Sur2004] J. Surowiecki, *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*, Random House, June 2004.

[Swe2002] L. Sweeney, “k-Anonymity: A Model for Protecting Privacy,” *Int’l J. Uncertainty, Fuzziness, and Knowledge-Based Systems*, vol. 10, no. 5, 2002, pp. 557-570.

[Syv2001] P. Syverson and I. Cervesato, "The Logic of Authentication Protocols," *Lecture Notes in Computer Science,* vol. 2171, 2001, pp. 63–136.

[Tan2004] A. Tanner and G. Mühl, *A Formalisation of Message-Complete Publish/Subscribe Systems*, Technical Report, Berlin University of Technology, Nov. 2004.

[Tar2006a] S. Tarkoma, "Efficient Content-based Routing, Mobility-aware Topologies, and Temporal Subspace Matching," doctoral dissertation, Dept. of Computer Science, Univ. of Helsinki, Apr. 2006.

[Tar2006b] S. Tarkoma, “Preventing Spam in Publish/Subscribe,” *Distributed Computing Systems Workshops*, 2006.

[Tar2007] S. Tarkoma and J. Kangasharju, "On the cost and safety of handoffs in content-based routing systems," *Computer Networks*, vol. 51, no. 6. Apr. 2007.

[Tha1999] F. J. Thayer, “Strand Spaces: Proving Security Protocols Correct,” *Journal of Computer Security*, vol. 7, issue 2-3, Jan. 1999, pp.191-230.

[Tho2001] M. Thorup and U. Zwick, "Compact routing schemes," *In Proc. Thirteenth annual ACM symposium on Parallel algorithms and architectures*, 2001, pp. 1-10.

[Tri2008a] Trilogy FP7 project [online], 2008, available at: http://www.trilogy-project.org/ [Accessed 19th June 2008].

[Tri2008b] S. Schuetz (ed), "D2 - Lessons in ‘Designing for Tussle’ from Case Studies," *Trilogy FP7 project* [online], 2008, available at: http://www.trilogy-project.org/ [Accessed 19th June 2008].

[Tro2007] D. Trossen, "The Sky is Falling or at least it is creaking," presentation at PIMRC, 2007.

[Tur2004] D. A. Turner and K. W. Ross, "A Lightweight Currency Paradigm for the P2P Resource Market," *International Conference on Electronic Commerce Research*, 2004.

[Wai1998] D. Waitzman, C. Partridge, and S. Deering, *Distance Vector Multicast Routing Protocol,* IETF RFC 1075, Nov. 1998.

[Wal2000] M. Waldman, A. Rubin, and L. Cranor, “Publius, A robust, tamper-evident, censorship-resistant web publishing system,” *in Proc. of the 9th USENIX Security Symposium,* Aug. 2000.

[Wal2004] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, and S. Shenker, "Middleboxes No Longer Considered Harmful," *In Proc. 6th USENIX OSDI*, San Francisco, USA, Dec. 2004.

[Wal2005] M. Walfish, H. Balakrishnan, D. Karger, and S. Shenker, "DoS: Fighting fire with fire," *In Proc. of the 4th ACM Workshop on Hot Topics in Networks (HotNets)*, 2005.

[Wan2002] C. Wang, A. Carzaniga, D. Evans, and A. L. Wolf, “Security issues and requirements in internet-scale publish-subscribe systems,” *in Proc. of the 35th Annual Hawaii International Conference on System Sciences (HICSS’02)*, Big Island, HI, USA, 2002.

[Web1978] M. Weber, *Economy and society*, Berkeley: University of California Press, 1968/1978.

[Wei2007] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman, *Information Accountability*, Technical Report, Massachusetts Institute of Technology (MIT), June 2007.

[Wic1999] S. B. Wicker and V. K. Bhargava, *Reed-Solomon Codes and their Applications*, Chapter 1, IEEE Press, 1999, pp. 1-17.

[Wit2001] Wittbrodt, B. Woodcock, A. Ahuja, T. Li, V. Gill, and E. Chen, "Global routing system scaling issues," *NANOG Panel* [online], Feb. 2001, available at: http://www.nanog.org/mtg-0102/witt.html.

[Woo1993] T. Y. C. Woo and S. S. Lam, "A Semantic Model for Authentication Protocols," *Proc. IEEE Symposium on Security and Privacy*, 1993.

[Wu2006] B. Wu, J. Chen, J. Wu, and M. Cardei, "A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks", *Wireless/Mobile Network Security*, Y. Xiao, X. Shen, and D.-Z. Du (Eds.), Springer, 2006.

[Wun2007] A. Wun, A. Cheung, and H.A. Jacobsen, "A taxonomy for denial of service attacks in content-based publish/subscribe systems," *Proc. of the 2007 International Conference on Distributed event-based systems*, 2007.

[Xyl2008] G. Xylomenos, V. Vogkas and G. Thanos, "The Multimedia Broadcast / Multicast Service," *Wireless Communications and Mobile Computing*, vol. 8, no. 2, 2008, pp. 255-265.

[Yah1993] R. Yahalom, B. Klein, and T. Beth, "Trust Relationships in Secure Systems-A Distributed Authentication Perspective," *Proc. of the 1993 IEEE Symposium on Security and Privacy*, 1993, p. 150.

[Yah1994] R. Yahalom, B. Klein, and T. Beth, "Trust-based navigation in distributed systems," *Computing Systems*, vol. 7, issue 1, 1994, pp. 45-73.

[Yan2006] X. Yang, G. Tsudik, and X. Liu, "A Technical Approach to Net Neutrality," *In Proc. ACM SIGCOMM HotNets Workshop*, 2006.

[Yan2007] X. Yang, D. Clark, and A. Berger, "NIRA: A New Inter-Domain Routing Architecture," *IEEE/ACM Transactions on Networking*, vol. 15, no. 4, Aug. 2007, pp. 775-788.

[Yan2008] Y. Yang, J. Want, and M. Yang, "A Service-Centric Multicast Architecture and Routing Protocol," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, issue 1, Jan. 2008, pp. 35-51.

[Zha2001] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, *Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing*, Technical report, UC Berkeley, Apr. 2001.

[Zha2003] B. Zhang and H. T. Mouftah, "Forwarding state scalability for multicast provisioning in IP networks," *IEEE Communications*, vol. 41, no. 6, 2003, pp. 46-51.