Towards a new generation of information-oriented internetworking architectures

Christian Esteve, Fábio L. Verdi and Maurício F. Magalhães
University of Campinas (UNICAMP)
School of Electrical and Computer Engineering
{chesteve, verdi, mauricio}@dca.fee.unicamp.br

ABSTRACT

In response to the limitations of the Internet architecture when used for applications for which it was not originally designed, a series of *clean slate* efforts have emerged to shape the so-called *future Internet*. Recently, visionary voices have advised a shift in the networking problem under research, moving from seamless host-reachability to *internetworking of information*. We contribute to the healthy debate on future Internet design and discuss ongoing information oriented efforts. Inspired by recent works in Bloom-filter-like data structures, we propose the *SPSwitch* as a novel switching engine to make wire speed forwarding decisions on flat information labels. We address part of the scalability issues in a data-oriented forwarding layer by trading overdeliveries for state reduction and line speed operations.

Categories and Subject Descriptors

C.2.1 [**Packet-switching networks**]: Network communication

1. INTRODUCTION

For a few years, funding agencies around the world have been promoting the research towards the so-called future Internet. *Clean-slate design* has been a buzz term for networking project proposals. However, the lack of palpable results and clear business models raises doubts whether network revolution makes sense at all. Today's use of the Internet reveals well known limitations in terms of mobility, security, address space exhaustion, routing and content delivery efficiency. Nevertheless, it works reasonably well [14]. For the long term, continuously patching the Internet with ad-hoc protocol extensions and overlay solutions seems to be a complex and costly solution.

ReArch'08, December 9, 2008, Madrid, SPAIN

The Internet has shifted from being a simple host connectivity infrastructure to a platform enabling massive content production and content delivery, transforming the way information is generated and consumed. From its original design, the Internet carries datagrams inserted by sending hosts in a best effort manner, agnostic to the semantics and purpose of the data transport. There is a sense that the network could do more and better given that today's use of the network is about retrieval of named pieces of data (e.g., URL, service, user identity) rather than specific destination host connections [15]. Hence, the enhancements at the internetworking layer should not be limited to QoS or routing efficiency: data *persistence*, *availability* and *authentication* [17] of the data itself leveraged with timeliness are beneficial in-network data-centric capabilities to be embraced from design.

Last decade's efforts towards rearchitecturing the Internet have mainly focused on end-host reachability with novel concepts (e.g., id/loc split) addressing the 'classic' end-to-end security, mobility and routing issues. All of these proposals are more-or-less host centric. However, this trend is changing, and senior researchers that have participated in the Internet development since its beginning, have advised tackling the future Internet problem from an *information interconnection* perspective. Van Jacobsen [15] provides a vision to understand the motivation for a networking revolution; while the first networking generation was about wiring (telephony) and the second generation was about interconnecting wires (TCP/IP), the next generation should be about interconnecting information at large. This architectural shift implies rethinking many fundamentals by handling information (data, content) as a first-class object.

Recent concerning events (and more to come) may potentially promote and accelerate the adoption of new internetworking paradigms. Today's economy is Internet-sensitive, service outages due to DDoS attacks[1] or due to limitations of BGP insecure routing[2] carry important worries and ex-

[1] Internet reports claim potential costs of $31.000 per minute for Amazon's two hour outage in June 2008.

[2] Pakistan Telecom routing mis-configuration for YouTube's address block propagated internationally, breaking the reachability of the popular video service in February 2008.

penses. Additionally, new forms of SPAM and evolving phishing methods are threatening today's so successful IP-based communication's experience.

This paper is certainly not the first to turn into data-oriented networking or to leverage the publish / subscribe communication paradigm. First, we discuss on the significance of future internetworking research (§ 2) and gather a set of newly emerged concepts from ongoing information-oriented Internetworking activities (§ 3) that set the context and motivation of our work. We move a step towards the feasibility of new data-oriented architectures by proposing the SPSwitch, a novel switching application based on recent Bloom-filter-inspired data structures (§ 4), validated through preliminary experimental results (§ 5). The goals of our future work (§ 6) are hardware-friendly forwarding schemes for a global scale information-oriented architecture.

2. RE-ARCHITECTING THE INTERNET

Research to circumvent current Internet limitations can be divided into those advocating a completely new architecture (*clean-slate*), and those defending an *evolutionary* approach due to incremental deployability concerns. From a research perspective, *clean-slate design* does not presume *clean-slate deployment* and aims at innovation through questioning fundamentals.

A key question is to what extent a new paradigm thinking 'out-of-the-TCP/IP-box' for the future network is really necessary, e.g., as packet switching was to circuit switching in the 70's. The reasoning is based on the large scale use of the Internet for dissemination of data [15]. Tons of connected devices are generating and consuming content, without caring about the actual data source as long as integrity and authenticity are assured [17].

We can also observe this shift toward information-centric networking in the momentum of service oriented architectures (SOA) and infrastructures (SOI), XML routers, deep packet inspection (DPI), content delivery networks (CDN) and P2P overlay technologies. A common issue is the necessity to manage a huge quantity of data items, which is a quite different task than reaching a particular host. In today's Internet, forwarding decisions are made not only by IP routers, but also by middleboxes, VLAN switches, MPLS routers, DPIs, load balancers, mesh routing nodes and other cross-layer approaches. Moving down data-centric functions to the lower networking layers could be in tune with the trend in access and backbone technologies represented by the coupling of the dominant Ethernet access protocol and label switched all optical transport networks.

More than an endless discussion around clean-slate design and actual network (r)evolution deployment, what we really need for the future Internetworking is 1) 'clean-slate thinking' beyond the TCP/IP heritage to foster innovation through questioning paradigms; and 2) feasibility work on an information-oriented infrastructure capable of supporting the actual and future demands over the network of networks.

3. PARADIGMS OF INFORMATION ORIENTED INTER-NETWORKING

Until recently, research in a new generation Internet has prompted architectural proposals (e.g., TRIAD, FARA, Plutarch, UIP, IPNL, HIP, ROFL) that mainly aimed at solving the *host reachability problem* by providing more flexible, expressive, and comprehensive naming and addressing frameworks than the Internet hierarchical IP address space. A move towards *information interconnection* can be observed in recent projects addressing the future Internet such as PSIRP [8], 4Ward [12], Trilogy, ICT's FIRE and other activities within EU FP7 and NSF FIND. Data-centric architectural proposals have started to emerge (e.g., DOA, i3, DONA, Haggle) and are similar in spirit to 'peer-to-peer', 'content-delivery', 'sensor' and 'delay-tolerant' networks.

3.1 Information-centric concepts

In *information/content/data - oriented/centric* networks, the flow of messages is driven by the nodes that have expressed their interest and the information identifiers of the messages, rather than by explicit destination host interface names (IP addresses) assigned by senders. Reachability of destinations is not anymore delimited by topological information but by the notion of *information scope* [21]. Having the data location hidden makes the semantics of what defines a sender or receiver of data less relevant than the data itself, intuitively providing enhanced security (e.g., DDoS mitigation) and bridging connectivity challenged underlying networks (e.g., DTN).

The *publish/subscribe* paradigm [11] is a promising trend to instantiate the so sought modern communication API [10] for information-centric systems. Pub/sub systems have been widely studied and employed for specific event-dissemination applications and have appealing characteristics like spatial and temporal decoupling [11]. In Internet-scale topic-based pub/sub internetworking [8, 20, 21], topics are unique information identifiers at different layers in the architecture accommodating different granularities and semantics (e.g., messages, channels, documents) to support every type of communications (e.g., transactional, interactive, etc.).

The suitability and benefits of moving the pub/sub layer downwards into the networking stack is one of the challenging objectives of interest-driven architectures where naming, routing, forwarding and addressing get fresh semantics (see Table 1).

In the envisioned *internetworking service bus*, information objects are first-class citizens introducing a new global unmanaged*namespace*. A form of publication *metadata* information is required to enable the self-authentication of the data, fragmentation, scope delimitation, inter-domain policies, in-network management, caching, and so on [8].

A global namespace for data items enables *caching* capabilities for every type of communications. In comparison, caching over TCP/IP is costly and application-specific. In case of non-mutable information objects caching becomes

Table 1: Concepts of information-oriented networking versus the original Internet design. Rethinking fundamentals.

Original Internet	Information-Oriented / Content-Centric Internetworking
Sender	Content producer (publisher)
Receiver	Content consumer (subscriber)
Sender-based control	Receiver-based control
Client/Server communications	Publish/Subscribe Sender and Receiver uncoupled
Host-to-host	Service access / Information retrieval
Topology / Domain	Information scope
Unicast	Unified uni-, multi- and anycast
Explicit destination	Implicit destination
End-to-End (E2E)	End-to-Data (E2D)
Host name (look-up oriented)	Data/Content name ("search" activity)
Secure channels, host authentication	Integrity and trust derived from the data

trivial, whereas for streaming applications, caching can be seen as long in-network buffers. Hence, the architecture natively plays the role of current CDNs and avoids redundant traffic over network links [1]. Furthermore, a new namespace for information objects could easily accommodate multi-, any-, con- and unicast types of communication in addition to novel forms of network coding to increase the network's efficiency and resilience.

3.2 A few reference architectures

In this section, we briefly introduce the basics of two recent design choices from the EU FP7 Publish/Subscribe Internetworking Routing Paradigm (PSIRP) project [8] we selected as reference architectures.

The RTFM architecture [20] gets its name from the functional building blocks that are recursively applied. The *rendezvous* (R) is in charge of matching subscriptions to publications and information scoping. The *topology* (T) management creates and maintains (sub-optimal) delivery trees used for traffic forwarding, acting both proactively (optimization) and re-actively (on-demand). The *forwarding* (F) functions perform the actual datagram delivery based on label switching techniques. Finally, *mediation* (M) refers to the node-to-node physical data transmission.

A high-level operational overview of the RTFM could be as follows. After a node subscribes to a publication, a distributed rendezvous system (e.g. a type of DHT or semi-hierarchical solution as in DONA [17]) must first find a copy of the publication's metadata. Using the distributed rendezvous structure to route to a copy of the wanted data, the topology management systems are expected to gather enough information to identify the delivery trees needed to forward the actual data to the subscriber(s). Note that the RTF functions are not necessary co-located in nodes and are distributed and recursive in nature.

In the *black box* rendezvous based networking approach [21], the key idea is to regard the network as a collection of black boxes based on a set of recursive rendezvous functions. The boxes operate in trusted domains hiding their internal topology and exposing outwards only labels and interest definitions. Recursivity [9] and *scoped information layers* are pivotal architectural patterns with a major goal: *scalability*. With the same goal but at a lower layer, efficient data structures enabling the data-centric networking functions (e.g., switching, label processing, caching) are called for to achieve the challenging scalability requirements of information-oriented networks heavily based on virtually 'unlimited' set of flat identifiers.

4. FAST FORWARDING ON FLAT LABELS

The overall picture of an information-oriented network architecture is complex and deserves very detailed discussions spanning multiple disciplines (internetworking, network management, semantic layer, etc.). However, there is a common challenge in any data-oriented paradigm: the need to take switching decisions at wire speed (Gbps) based on a large universe of flat (non-topological, non-aggregatable) identifiers (e.g., 256-bit cryptographic hash values).

Related work relying on flat labels includes ROFL [7], a proposal for Internet-scale routing on flat host identifiers based on neat DHT constructs. In our work we focus on flat identifiers with fundamentally different architectural principles (see Table 1). DONA [17] employs flat self-certifying labels for data objects operated by *find/register* primitives over IP networks, whereas our work is more ambitious and could run on top of L2 and L3.

For the sake of generality and the objectives of this paper, we use the term *flat label* for data identifiers or any topology-independent packet header forwarding identifiers.

4.1 Publish/Subscribe Switch

The Publish/Subscribe Switch (SPSwitch) is an abstract switching element that relays messages through strict port-forwarding operations. In a more elaborated design, the SPSwitch performs more complex actions like label switching or querying the cache system.

For the purposes of this work, it is enough to consider the generic problem of having to take switching decisions based on large flat identifiers (labels). Note that output destinations (ports) are not just limited to physical port-in/out interfaces but should be regarded as generic outputs, including also local processes, virtual ports, recursive operations, and cache systems.

In the SPSwitch representation of Figure 1, each possible message output is represented by a Bloom filter [3], forming our first p-bank switching approach (§ 4.3) and a reference switching model for an enhanced data structure (§ 4.4).

4.2 The role of Bloom filters and space efficient probabilistic data structures

Given the huge space and flatness of the information identifiers, our intuition is that Bloom filters and other compact

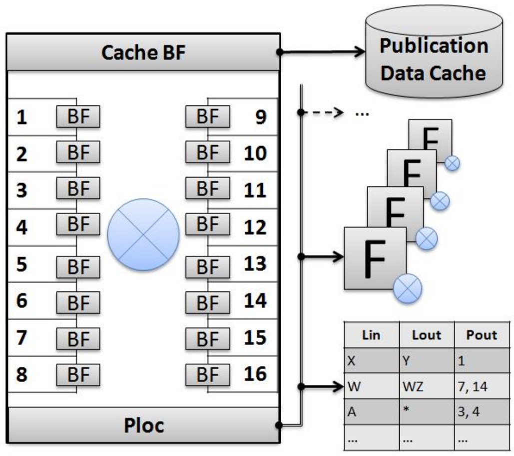

Figure 1: Switching model for incoming datagrams carrying flat labels. Generic destinations (ports) are aggregated in Bloom filters.

hash-based data structures will play a fundamental role as efficient data aggregators in any information-centric architecture. Basically, a Bloom filter (BF) [3] is a space-efficient hashing-based data structure that answers set membership-queries (e.g., *is label*$_x$ *in output* P_y*?*) with some probability of being wrong (false positive rate). BFs are useful whenever you have a set of elements and space is an issue. Then, an approximate representation like a BF may be a powerful alternative if the effects of false positives can be managed. The performance of a BF does not depend at all on the size of the items but on the ratio $memory/elements$. Therefore, hashing-based data structures are an ideal room to handle the large set of flat identifiers. We refer to the large literature on BFs [3, 4, 6, 16] for details and mathematical background.

Bloom filters are commonly used in IP forwarding and other widely studied networking applications (e.g., caches, P2P, measurement, packet classification) [6]. We expect increasingly more useful applications of BFs and its derivatives in new data-intense networking proposals (e.g., Internet accountability [2], flow management [4], credential-based network security [22], IP multicast revisited [19]) with strict performance and memory requirements. The authors of [13] briefly sketched the idea of aggregating active IP multicast addresses per output interface to achieve scalability.

Due to space limitation we do not compare the SPSwitch design with existing hardware designs for fast networking. We are aware that compact hash tables and hashing functions are a daily aid in IP networking. However, there are notable operational differences and challenges (e.g., longest IP prefix vs. long flat identifier matching).

4.2.1 False positives

It is important to place emphasis on the *bounded effect of false positives* in data-centric interest-driven architectures. First, the pub/sub paradigm inherently tolerates false positives, since datagrams corresponding to non subscribed items do not progress in the network and do not create forwarding states. Moreover, end-nodes will only process explicitly subscribed pieces of information. Second, with support for *opportunistic caching*, copies of data can be used to fulfill possible future requests of close by subscribers. Finally, packets forwarded due to false positives are not propagated over many hops due to the large label space and the decreasing probability of consecutive false positives.

4.2.2 Hardware requirements

Since hashing is performed on a per packet basis, the hash tables are implemented directly into the hardware to achieve low processing times ('constant' time to hash and easily parallelized). The position of the element in the memory array can be directly given by the hash value. Built-in dedicated hashing modules are already available in networking hardware. Moreover, we can even skip the hashing operations by taking advantage of the randomness of the hash-based labels under consideration.

In order to achieve to achieve high data rates, memory accesses and computations must be kept to a minimum during packet processing. Routers are limited in expensive high-speed memory. Projections for routers capacity for the next decade [2, 18] let us assume the availability of high speed memory in routers in the order of tens of Mbits[3].

4.3 Naive p-bank Bloom filter approach

Our first natural approach was to define a SPSwitch formed by a bank of BFs, maintaining a BF for each possible output (2^p). The 'control plane' inserts the label(s) in the required output BFs and upon message arrival all possible outputs are queried in parallel to make the forwarding decisions. Recall that a BF does not return false negatives.

After gaining some practical experiences, we identified some limitations of our naive p-bank BF approach inherent to basic BF constructs [4]: a) lack of associated values: just binary probabilistic set-membership responses; b) expensive deletion[4]: counting BFs are costly in memory sizes; c) no notion of time: costly association of filter elements or cells with timing information; d) unbalanced usage of memory per output: unpredictable destination demands difficult the overall system design and memory allocation.

We realized that a more flexible and expressive data structure was required enabling dynamic port-value assignment and per element handling capabilities. Nevertheless, standard BFs are expected to play a role as filters for large caching systems and in other elements of the architecture due to its simplicity, ease of use, and excellent performance.

[3]Typically SRAM, DRAM, (T)CAM and newly RLDRAM. CAMs are expensive fully associative memories - highest available single-chip CAM is 18Mbit [18].

[4]Deletions potentially introduce false negatives. This is inherent to any probabilistic data structure and needs to be carefully handled.

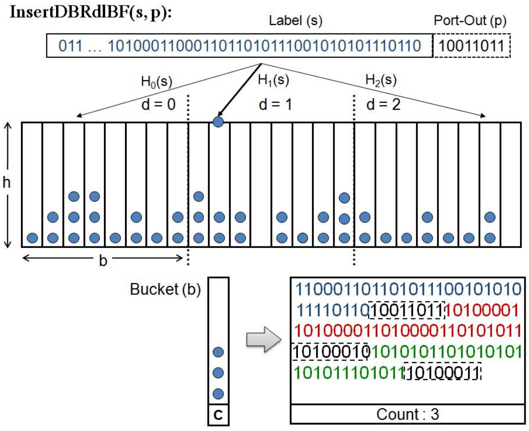

Figure 2: In the stateful d-left Bloom filter, a label fingerprint and its output value are inserted into the least loaded bucket among d candidates.

4.4 Stateful d-left Bloom filter with Dynamic Bit Reassignment

The goal is to build upon a more flexible data structure that could return the output port-value(s) for each programmed label, while still allowing false positives in the form that labels *not in* the set return a value and labels *in* the set return more than one value (aka 'multicast') when only one entry was programmed.

Looking for a candidate structure that could store values (stateful), we took on recent results in BF-inspired data structures by Bonomi et al [4, 5], where the authors define a d-left fingerprint compressed filter (FCF) to track the state of network traffic flows in a dynamic environment. Due to space limitation we refer to [4, 5, 6] for deeper details and rigorous mathematical analysis. In this work, we present our application of the d-left FCF data structure to the specifics of our switching problem and validate it with experimental results.

The d-left scheme is based on the power of two choices (or multiple-choice hashing) and its key feature is yielding near-perfect hash functions, something impractical for dynamic sets. The procedures for *insertion* (label:port) and *look-up* (label) of the stateful d-left FCF are as follows (see Fig.2). A hash table stored in memory is divided into d equal subtables. On *insertion*, d hash functions ($H_i(s)$) uniformly provides a candidate bucket b in each subtable. The least loaded bucket is chosen (breaking ties to the left) to place the item's f-bit fingerprint (e.g., last f bits of the label hash) and the p-bit value. On *look-up*, d buckets need to be checked for fingerprint match and the companion port-value(s) is retrieved.

False positives happen after a fingerprint match when inspecting h elements in d buckets ($Pr = d \cdot h \cdot 2^{-f}$). The neat idea of dynamic bit re-assignment (DBR) [5] is to adjust the size of the fingerprint in function of the bucket's load, yielding better false positive rates due to larger average fingerprints ($f`_{DBR} > f$). The counterpart is having to maintain a counter per bucket and the increased complexity on element insertion due to fingerprint down-resizing (easily implementable with bit shift operations). Further bucket space optimization via semi-sorting of items can be achieved, however, the gain/complexity trade-off is not appealing in our setting where we require to store additional p bits per entry.

5. EXPERIMENTAL RESULTS

The relation between the number of targeted labels *n*, the amount of buckets b and subtables d determines the average load per bucket that can be determined asymptotically as in [5]. By choosing b equal to $n/12$ and $d = 3$, a bucket high $h = 6$ provides a safe margin for overflow ($\approx 10^{-31}$) and a reasonable table utilization $u = n/(dbh) = 2/3$ [4]. Finally, we used the following construction for our experiments: $1M$ 256-bit flat labels (n), 20-bit fingerprints (f) and 10-bit outputs (p). We inserted the n elements into the filters and used a disjoint test set of $10 \cdot n$ labels for counting actual false positives.

Table 2 compares the different forwarding table schemes analytically and includes the predicted calculations and the actual values averaged over 50 experiments. As expected, a simple forwarding table indexed per flat labels has prohibitive costs in both memory and computation. When fixing the available memory, the d-left FCF based approach outperforms the alternative probabilistic structures in actual false positive rates. The gains in performance from the d-choice technique [5] comes from the reduction of the maximum bucket load to about $loglogn/logd$ in the well studied *balls into bins* problem. The DBR optimization results in larger fingerprints (equiv. $f' \approx 22.27$) and thereby substantially lower false positive rates.

We should stress that the beauty of the d-left FCF data structure is not only the gains in terms of memory but in the low and constant memory accesses ($O(1)$). Moreover, a key differentiator of the fingerprint-based approach is to have a powerful probabilistic key (the fingerprint) for managing inserted elements (e.g., querying, updating, deleting).

The actual low performance of the *p* BFs can be explained due to 1) the fact of aiming very low error rates ($0.6185^{M/n} \approx 10^{-9}$) per BF compared to the theoretical convergence rate $\theta(1/n_{BF})$; and 2) practical issues of the *double hashing* technique [16] to efficiently generate the optimal *k* hash functions ($ln(2) \cdot M/n \approx 31$).

The actual results are very promising; we may conclude that our envisioned forwarding layer (SPSwitch) based on d-left FCF data structures could evolve to a real system with e.g., 18 Mbits high speed single-chip memories per subtable and handle labels in the order of millions with very low false positive rates (10^{-6}) and bounded worst case performance. The specific system parameters are not so relevant (yet) and mainly serve as a first proof of concept. The most important results are the orders of magnitude and the performance that can be achieved with an optimized d-left FCF inspired data structure.

Table 2: Analytical and experimental comparison of different data structures for the switching procedures.

	Mem. access	Mem. size M	(Mbits)**	(bpe)	False positive	(predicted)**	(actual)**
Standard Table	$O(n) — O(1)$*	$n * (s + p)$	253.68	266.0	0	0	-
Fingerpr. Table	$O(n) — O(1)$*	$n * (f + p)$	28.61	30.00	2^{-f}	$9.54 * 10^{-7}$	-
p-bank BF	$O(1)$	$2^p * m$ ***	43.63	45.75	$\approx 2^p * 0.62^{M/n}$	$2.91 * 10^{-7}$	$4.33 * 10^{-3}$
d-left FCF	$O(1)$	$d * b * h * (f + p)$	42.92	45.00	$< d * h * 2^{-f}$	$1.72 * 10^{-5}$	$1.51 * 10^{-5}$
d-left FCF DBR	$O(1)$	$d * b * (h * (f + p) + c)$	43.63	45.75	$< d * h * 2^{-f'}$	$3.57 * 10^{-6}$	$3.46 * 10^{-6}$
* Assumes a perfect hash function. ** Parameters: $n = 1.000.008; d = 3; b = 83.334; f = 20; p = 10; h = 6; c = 3; s = 256$.							
*** Total memory of the p-bank Bloom filters equal to the value M of the d-left FCF DBR. $m = M/2^p$; $k_{opt} = 31$.							

6. FUTURE WORK

Our very next step is to leverage the proposed data structure to work in steady states alternating deletions and insertions, with deletions being handled by timing mechanisms or explicitly. We require further studies on how to apply cache management algorithms (e.g., LRU, LFU) and BF extensions to handle deletions. Upcoming efforts include design optimizations considering memory technology specifics and efficient in/off-chip memory element reallocation. Last but not least, our roadmap includes experimental validation with regard to hardware implementation (NetFPGA) and feasibility on a large scale testbed infrastructure (e.g., Onelab2).

7. CONCLUSIONS

The information-centric usage of today's Internet has changed our daily lives with regard to content generation, consumption and communication patterns. We discussed the relevance of 'clean-slate' research on future Internetworking centered around information and move a step forwards in terms of feasibility. We presented the SPSwitch, a generic forwarding engine based on hashing data structures promising a seedbed of new lines of fast scalable forwarding on flat identifiers. Basically, the SPSwitch trades a small amount of overdeliveries for state reduction and line speed operations. We expect Bloom-filter-inspired systems to play a key role in routing and aggregation of information in data-oriented networks.

Acknowledgements

The interpretation of those ideas made by the authors may or may not agree with the vision of the PSIRP group. We want to thank specially Pekka Nikander and Dirk Trossen for their insightful discussions. We are also thankful to Adam Kirsch for his aid in the field of hash-based data structures. This work is partly funded by CNPq, Capes, FAPESP and Ericsson Research.

8. REFERENCES

[1] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker. Packet caches on routers: The implications of universal redundant traffic elimination. In *ACM SIGCOMM*, 2008.

[2] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable internet protocol (AIP). In *ACM SIGCOMM*, 2008.

[3] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, 1970.

[4] F. Bonomi, M. Mitzenmacher, R. Panigrah, S. Singh, and G. Varghese. Beyond bloom filters: from approximate membership checks to approximate state machines. In *ACM SIGCOMM*, 2006.

[5] F. Bonomi, M. Mitzenmacher, R. Panigrah, S. Singh, and G. Varghese. Bloom filters via d-left hashing and dynamic bit reassignment. In *44th Allerton Conference*, 2006.

[6] A. Z. Broder and M. Mitzenmacher. Survey: Network applications of bloom filters: A survey. *Internet Mathematics*, 1, 2003.

[7] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, and I. Stoica. Rofl: routing on flat labels. *SIGCOMM Comput. Commun. Rev.*, 36(4):363–374, 2006.

[8] D. Trossen (ed.). Conceptual Architecture of PSIRP Including Subcomponent Descriptions (D2.2). http://psirp.org/publications, June 2008.

[9] J. Day. *Patterns in Network Architecture: A Return to Fundamentals*. Prentice Hall, December 2007.

[10] M. Demmer, K. Fall, T. Koponen, and S. Shenker. Towards a modern communications api. In *Proceedings of HotNets-VI*, 2007.

[11] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[12] FP7 4WARD Project. D2.1 Technical Requirements. http://www.4ward-project.eu/, Aug 2008.

[13] B. Grönvall. Scalable multicast forwarding. *SIGCOMM Comput. Commun. Rev.*, 32(1):68–68, 2002.

[14] M. Handley. Why the internet just works. BT Technology Journal (2006) vol. 24 (3) pp. 119–129.

[15] V. Jacobson. If a clean slate is the solution what was the problem? Stanford "Clean Slate" Seminar., Feb 2006.

[16] A. Kirsch and M. Mitzenmacher. Less hashing, same performance: building a better bloom filter. In *ESA'06.*, pages 456–467, London, UK, 2006. Springer-Verlag.

[17] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. *SIGCOMM Comput. Commun. Rev.*, 37(4):181–192, 2007.

[18] K. Pagiamtzis and A. Sheikholeslami. Content-addressable memory (CAM) circuits and architectures: A tutorial and survey. *IEEE Journal of Solid-State Circuits*, 41, 2006.

[19] S. Ratnasamy, A. Ermolinskiy, and S. Shenker. Revisiting ip multicast. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15–26, New York, NY, USA, 2006. ACM.

[20] M. Särelä, T. Rinta-aho, and T. Tarkoma. RTFM: Publish/subscribe internetworking architecture. ICT Mobile Summit, Stockholm., June 2008.

[21] S. Tarkoma, D. Trossen, and M. Särelä. Black boxed rendezvous based networking. In *ACM MobiArch '08*, 2008.

[22] T. Wolf. A credential-based data path architecture for assurable global networking. In *Proc. of IEEE MILCOM*, Orlando, FL, October 2007.