

Intelligence in Future Broadband Networks: Challenges and Opportunities in High-Speed Active Networking

James P.G. Sterbenz

Abstract– This invited paper considers the challenges and opportunities in the provision of active processing in broadband networks. Active networking aims to provide a systematic mechanism to use processing capabilities within network nodes (switches and routers) to allow the dynamic provisioning and composition of advanced services. Many of these services are traditionally offered at the application layer or require changes to the network layer standards. Active networking has the potential for improving performance by providing functionality at the right place and layer, without the need to go up to the application layer. The processing gains achieved by Moore's Law are frequently touted as the motivation for active networking. It is the ratio of processing and memory to bandwidth, however, that governs how much active processing is achievable. Active node architectures should support active processing with a flexible mix of software and programmable hardware based on the granularity of the processing. Significant practical challenges remain before active networking technology will be deployed and adopted. In particular, active networking is a technology awaiting the killer application.

Index Terms– active high-speed broadband optical networking

I. INTRODUCTION TO ACTIVE NETWORKING

Active networking is a technology that uses processing in network nodes (switches and routers) to allow the dynamic provisioning and composition of enhanced services [1,18]. Many of these services are traditionally provided at the application layer (such as content caching), or require changes to the network layer standards (such as for multicast).

Active networking has recently been a topic of intense research. In its current form, most of this research is centered on IP-based networking, motivated by the premise that processing and memory have become cheap enough to allow significant processing by network nodes (IP routers and switches). There is no fundamental reason, however, why active networking cannot be based on other network layer technologies, such as ATM. The implications of resource cost will be discussed in more detail in Section IV.

This work was sponsored in part by the Defense Advanced Research Projects Agency under contract F3060299C-0131 issued by the AFRL. James P.G. Sterbenz is with BBN Technologies in Cambridge MA, 02138-1191, USA, jpgs@ieee.org.

First, this paper begins with a short summary of active network architecture. Various factors that influence performance include how active processing is invoked, the granularity of processing, the location of active processing in the network, and the architecture of the active node.

A. Active Network Flavors

Active networking can be divided into two major types: strong and moderate¹. *Strong AN* is the type originally proposed in the context of active networking [17], in which users inject *capsules* into the network that contain code to be executed in active nodes. Strong AN raises a number of significant policy, security, and resource concerns, to prevent active applications from eavesdropping or interfering with one another, as well as the normal passive packet forwarding. These concerns must have solid solutions before strong AN is viable outside the research community.

Moderate AN is the case in which a service provider provisions code into network nodes to dynamically deploy new protocols and services, without the traditional software upgrade process. Note, however, that one of the services that can be provisioned is an interpreter, providing the equivalent of strong AN functionality when desired. The main difference is that moderate AN is intended primarily as a way for *network providers* to monitor, control, manage, deploy, and evolve network protocols and services, whereas strong AN is a way for *end users* to deploy and execute active services. Moderate AN is somewhat less risky in terms of security and denial of service, and gives the network provider more control over what active services are deployed and how they are executed.

B. Active Processing Granularity

A significant determinant of the amount of processing required for active networking services is the *granularity* of processing. This affects how often active processing must occur, ranging from per bit, through per byte, word, cell, frame, burst of frames, to per flow or connection. Four ranges of granularity are important.

¹ The term *strong AN* and *moderate AN* were coined by Bhattacharjee and Sterbenz, respectively in 1997.

At the coarsest granularity are *global control plane* functions, which consist of processing that affects the node as a whole, and perhaps the entire network. An example of global control plane active processing is monitoring traffic patterns to exert changes in traffic management and forwarding functions to optimise resource utilisation in the node or perform load balancing across the network as a whole.

Per flow control plane functions operate on the flow or connection state machine or control packets at the flow level. Examples include altering the RSVP flow or ATM connection state machines in each node, or issuing ICMP messages based on long-term flow state.

Significantly finer granularity takes place at the packet level with *per packet control plane* active processing, in which every packet is examined in a flow or connection, which potentially result in alterations of control mechanisms. An example is active congestion control in which the contents of packets are examined to intelligently discard packets, e.g. B and P frames in an MPEG stream should be discarded rather than I frames.

Finally, at the finest granularity, *per packet data plane* processing not only examines each packet in a data flow, but alters the contents and may destroy or generate new packets. Thus the granularity of operation is per word, byte, or bit. An example is active transcoding that determines the optimum place in the network to transcode, and recodes the data within the active node. Clearly, this is the most demanding granularity, requiring sufficient processing to transform the data at line rate.

C. Active Network Architecture

From the perspective of overall network architecture, the significant aspect is *where* the active processing resides. Active processing can be added to an existing network as an *overlay*, with minimal changes to the original network. Communication between active nodes **A** participating in an active application of service occurs on the overlay, which is connected to conventional nodes where appropriate, as shown in Fig. 1 [7]

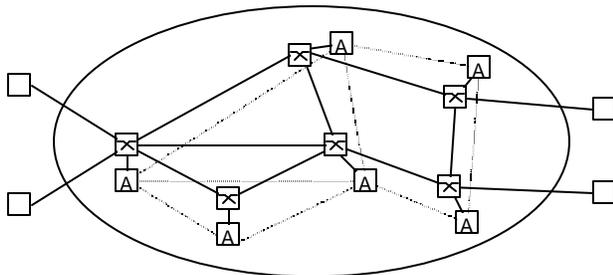


Fig. 1. Overlay Active Network.

The overlay may be physically distinct, as with the PSTN (public switched telephone network) SS7 network supporting IN (intelligent network) functionality. This is required when the basic network lacks the capability to

perform active processing or to transport and process signalling messages. Alternatively, a *virtual overlay* may be employed, which provides logical separation from normal data and control while using the same physical links and network nodes.

The other extreme is for all nodes in the network to natively support active processing; this is a *fully embedded* active network. A more flexible approach is a *partially embedded* active network, in which only some of the network nodes need support active processing, as shown by the nodes labelled **A** in Fig. 2.

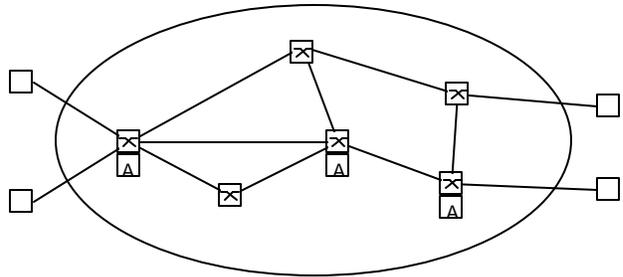


Fig. 2. Embedded Active Network.

Selective embedding has the significant advantage of allowing gradual migration from a passive to active network. Furthermore, for some active network services, not all nodes need to be active with the corresponding increased cost. In some cases tunnels are established between active nodes, and in other cases it is sufficient for passive nodes to simply forward packets until an active node happens to be reached.

D. Active Node Architecture

The other significant architectural perspective is from the active node (switch or router). A reference model for active network nodes has been created by the DARPA Active Networks research community [3] to capture the essential requirements of active nodes, as well as to provide interface definitions, as shown in Fig. 3.

The lower portion of the figure consists of the normal switch (or router) functionality: a switch fabric which interconnects a set of inputs to a set of outputs. Packet filters (or classifiers) are responsible for detecting active packets, and sending them up to the active processing portion of the node. A nodeOS provides the typical operating system functionality, such as resource management, to the *execution environments* (EEs). EEs are interpreters that provide both an execution API and language environment to the *active applications* (AAs). The *management execution environment* (MEE) serves as the privileged EE providing monitoring and control the active node and other EEs.

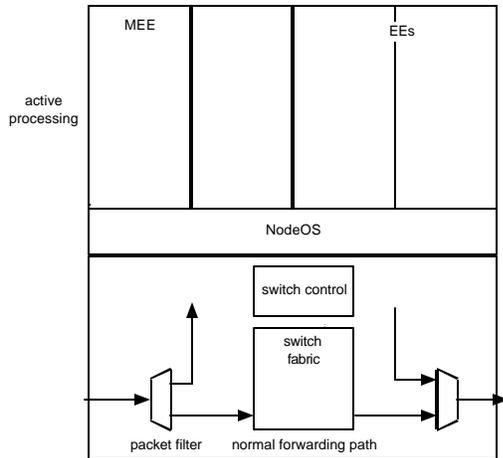


Fig. 3 Active Node Reference Model.

From a performance perspective, the key requirements are for the active processing not to interfere with the normal fast (or critical) path of packet forwarding, and for active processing to happen sufficiently fast that input queues not build and latency bounds be met. This can be stated as the Active Network Processing Principle [16]:

Active network processing should not impede the nonactive fast path; packet filters in the critical path must operate at line rate to pass nonactive packets. The ability to perform active packet processing requires sufficient processing power to sustain the required active packet throughput.

II. PSTN AND INTERNET INTELLIGENCE

It is useful to consider the location of intelligence in the Internet (evolved from the original ARPANET) and the PSTN (evolved from an analog circuit switched voice network). With the broadest brush, the PSTN is a smart network connecting dumb end systems and the Internet is a dumb network connecting smart end systems. The notion of a “dumb” and “smart” network are relative, however, and the traditional location of intelligence in both cases has blurred in the recent past.

A. PSTN: Intelligence in the Network

The PSTN has evolved from, and is still primarily, a network of circuit switches connecting telephones. Telephones are dumb terminals; in the extreme, POTS (plain ordinary telephone service) telephones are only capable of analog voice and extremely simple signalling (pulse or DTMF dialing, hook flash, and DTMF in-band signalling). Similarly, the network nodes (telephone switches) were originally capable only of establishing circuits for analog telephones. Even before the advent of digital switches, there was a desire to offer more sophisticated services than a point-to-point telephone call, for example multipoint conference calls and toll-free number redirection. Initially,

these services had to be added to each switch. With the advent of common channel signalling, which became SS7 (signalling system no. 7), the signalling network was separated into a physically distinct overlay data network. This provided the opportunity to provide enhanced network services and intelligence using the SS7 network as a basis. The intelligent network (IN) became the basis for systematically adding intelligence to the PSTN, and this predates AN in the context of the Internet by many years. By nature, IN is a form of moderate AN using an overlay model (SS7 network) with per flow granularity; in this case a flow corresponds to the circuit established for a telephone call. Active processing takes place in SCP (service control point) processors attached to the telephone switches. The connection state machine is represented by a basic call model (BCM), for which triggers can be defined to perform intelligent processing. New services can be deployed by provisioning new service code into the SCPs, which is a significant improvement over upgrading signalling code in the switch itself.

Unfortunately, the architecture of the PSTN is deeply rooted in its voice legacy, and the complete separation of the data and control planes. This prevents an intelligent end system from establishing peer-to-peer signalling associations with network nodes. While there have been efforts to evolve IN to a broadband environment (e.g. [8]), the PSTN is fundamentally the wrong architecture to support intelligent processing for integrated services packet networks.

B. ARPANET to Internet: Intelligence at the Edges

The Internet is based on the opposite model of intelligence: a simple core network connecting intelligent end systems. One of the key aspects of the ARPANET design philosophy was to keep the core network simple and stateless [4,5]. This requires that significant functionality be located in the end systems, and is only possible when end systems are intelligent processors, such as mainframes, minicomputers, workstations or personal computers. This design philosophy allowed the network as a whole to be robust in the presence of component (link or switch) failures or sabotage. Note that this is *not* the same as the end-to-end arguments, which will be discussed in Section III. In summary, the ARPANET model was for a (relatively) dumb network interconnecting smart end systems. The ARPANET model held through the explosive growth years of the NSFNET infrastructure in the early 1990s; its preservation of transparent end-to-end semantics allowed open development of new applications such as the Web and real-time media streaming.

C. Evolving Internet: Hacked Intelligence

The Internet of the late 1990s and early 2000s is significantly diverging from the stateless core with transparent end-to-end semantics. A number of services and applications are being hacked into the Internet in a plethora of proprietary closed solutions for services such as firewalls, VPNs, caching, and address reuse (using NATs – network address translators). Many of these obscure the end-to-end

transparency, and prevent open application and service development using IP based protocols. Dynamically changing IP addresses and NATs, in particular, prevent applications from using simple addressing mechanisms to communicate with one another.

Active networking, however, provides us the opportunity to embed functionality in the network in a *systematic* way; and in a manner that does not necessarily break traditional addressing transparency in an uncontrolled manner². So the debate should not be over the desirability of intelligence in the Internet; this is already given. Rather, the debate should be over *how* is the best way to introduce intelligence into the network in a systematic way, and in a way that does not break the end-to-end semantics of the Internet that have been critical to its growth and openness.

III. THE ROLE OF THE END-TO-ARGUMENTS

The end-to-end arguments have been used as justification against active networking, but in fact the end-to-end arguments *support* the proper application of active networking [2]. While the ARPANET design philosophy pushed functions to the edges of the network for the sake of robustness, the end-to-end arguments are about functional composition of hop-by-hop services, and formalised what services *must* be provided end-to-end. In particular, the end-to-end argument is [14]:

The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible.

The canonical example lies in the security domain: end-to-end assurance is not guaranteed by hop-by-hop link encryption; while data is encrypted on the links themselves, is in plaintext within the network nodes. Users who require end-to-end assurance *must* encrypt end-to-end, making any hop-by-hop encryption irrelevant. Similarly, the only way to guarantee against bit errors end-to-end is to provide end-to-end error control.

On the other hand, there is nothing wrong with putting functionality in the network if the motivation is performance enhancement; this is clearly stated as part of the end-to-end arguments [14]:

Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.

For an example of this case, again consider error control. While end-to-end assurance is only guaranteed by an end-to-end integrity check (checksum, CRC, or FEC), performance may be significantly enhanced in some cases with hop-by-hop checks. In the case of a lossy wireless link that is part of a long latency multihop path, every errored packet requires a round-trip time to correct. Using link layer error control

significantly shortens the control loop, significantly enhancing performance to the end user.

Thus, we can imagine two reasons to put intelligence in the network using active networking: to enhance the basic network layer services, or to perform functionality that is traditionally end-to-end to enhance performance to end users, or the network as a whole. An example includes active congestion control mentioned earlier: by intelligently discarding the packets *least* important to an application (rather than randomly), end-to-end performance is maximised while minimising overall network congestion. Similarly, active caching snoops on the application layer headers in packets (e.g. HTTP) to intelligently redirect requests to a near cache; this minimises response time to the application while reducing overall network bandwidth.

IV. MOORE'S LAW: OPPORTUNITIES AND CHALLENGES

The factor that is generally recognised to enable active networking is the dramatic increase in processing and memory capabilities due to Moore's Law, which states that for a given cost point, processing power doubles roughly every 18 months. Similar trends also exist for memory capacity (but not access time). While this indeed has enabled significant processing in network nodes the situation is a bit more complicated.

A. Resource Tradeoffs

It is useful to consider a network as a set of resources that must be traded against one-another [10], subject to some constraints [16]. In particular, a network consists of processing, memory, and bandwidth forming the triple $\langle B, P, M \rangle$. Furthermore, various constraints also come into play. In the context of high-speed networking, the most important constraint is latency L , and in the context of mobile nodes energy consumption W . Thus, the tuple becomes $\langle B, P, M | L, W \rangle$ representing a tradeoff of bandwidth, processing, and memory, constrained by the latency requirements of applications and energy capacity of self-powered mobile nodes. Note that the constraints may also be traded against the three fundamental resources. For example, FEC increases the bandwidth and processing requirements to lower the latency on lossy links.

B. Processing/Bandwidth Ratio

In this light, what really matters is not the absolute cost of processing and memory, but rather the *ratio* of processing and memory to bandwidth, that is $(P+M)/B$ [11]. Thus, there is ample opportunity for significant active network processing on a 10Mb/s link, and significantly less on a 10Gb/s link, by three orders of magnitude. As the $(P+M)/B$ ratio decreases, the amount of per packet active processing that can be done at line rate decreases. Similarly, as the flow or connection length decreases, the amount of per flow active processing that can be done decreases.

Note that the fastest wireless networks tend to have a significantly higher $(P+M)/B$ ratio than high-speed optical

² This is not to say that active networking, in itself, solves the addressing limits of IPv4.

networks, and thus provide significant opportunities for active processing.

C. Location of Active Processing

The reference model shown in Fig. 3 has served the active networking research community well, but it is not suitable for high-speed active networking. In particular, locating all active processing in a centralized processor of the node will create bottlenecks for active processing at the packet granularity.

Thus, it is more reasonable to place all per packet control and data plane processing in the appropriate input and output processors for each interface of the node [6], as shown in Fig. 4 [16].

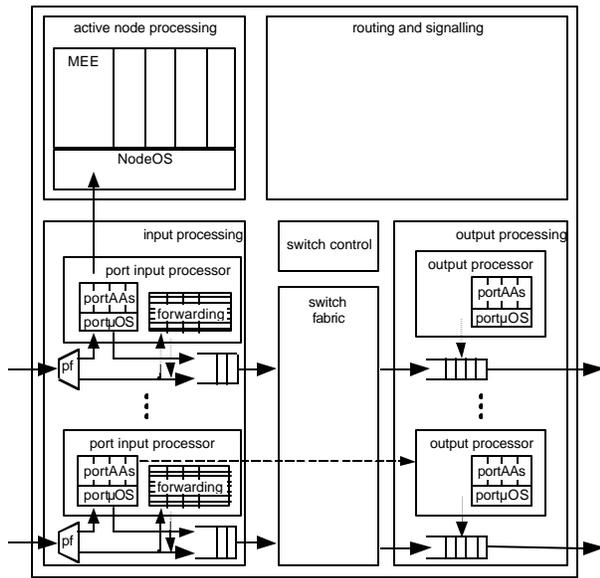


Fig. 4. High Performance Active Router Architecture.

Note that as in Fig. 3, packet filters pull packets from the normal fast path for active processing. In this architecture, however, per packet active processing occurs either in a per port microcontroller or in programmable hardware [9]. Only global control plane (and perhaps per flow) active packets go up into the per node active processing. Similarly, active processing may occur per output port, particularly to dynamically modify the output scheduling algorithm. This processing may be coordinated with the input port processing, global node active processing, or both.

An additional motivation for moving active processing to the interface processors occurs in mobile wireless networks [11]. Wireless networks can benefit from adaptive physical, link, and MAC layer processing. For example, power control can be used to adaptively control the degree of connectivity in an ad hoc network [13]. Nodes may wish to dynamically select physical layer coding and MAC parameters based on time varying channel conditions. In the extreme, mobile nodes may negotiate protocol selection, e.g. TDMA vs. CDMA. Software radios enable active processing all the way down to

the physical layer. High performance mobile nodes with multiple interfaces (e.g. using directional antennae) have the dual motivation for per interface active processing from the physical layer, through the link and MAC layers, to the network layer. Ad hoc nodes serve as both end system and intermediate system, and the distinction between active processing for transit traffic and application processing blurs significantly.

D. Higher Layer Processing

A number of desirable active network functions involve processing above the network layer, particularly at the transport layer 4, session layer 5 and application layer 7. Examples of such processing include active multicast, multiparty session control, firewalls, and active caching.

A number of these functions are being hacked into the Internet now, either in the form of *middle boxes* [15] that terminate transport layer associations, or by using application specific layer 4-7 switches that forward based on transport and application layer headers.

Active networking provides the opportunity to perform these functions not only in a systematic manner, but at considerably higher performance without the need to terminate transport or application layer associations. The proper partitioning of functionality among per interface programmable hardware, microcontroller software, and node control software can be made to optimise performance, both in terms of bandwidth and latency.

While high bandwidth, low $(P+M)/B$ ratio networks challenge active processing, this effect is less prominent than by use of application layer overlays and middle boxes.

E. Optical Networks

All-optical networks (AONs) have the characteristic that no processing can occur in the data path, since optical logic and memory are beyond the capability of current technology. Thus, control is generally at circuit granularity, or, in the case of optical burst switching (OBS) [12,19] over long bursts of data. In this case, active processing is limited to per burst or circuit, essentially the same as per flow granularity, with the ability to alter circuit and burst signalling and control.

Optical packet processing has been proposed by converting the header to the electronic domain for decoding, and in this case limited per packet control plane active processing is possible (for example active congestion control). Any active processing that relies on higher layer protocol headers (such as active caching), as well as data plane active processing, is not currently possible since the network layer payloads must remain in the form of photons.

The major opportunity for active optical networking is at the edge of the optical network, as will be discussed next.

F. Activity at Subnetwork Boundaries

A number of network technologies have characteristics that are fundamentally different from one another. Even in the

case where applications run over IP as the waist of the hourglass, significant adaptation can occur between IP and the subnetwork protocols, or at the boundaries between subnetworks. Examples of subnetwork technologies significantly different from the conventional wired Internet include transparent all optical networks, mobile wireless ad hoc networks, and sensor networks.

Active networks can play a significant role in allowing this adaptation without the need for fixed interworking units, and allowing IP to adapt as necessary to the appropriate underlying network technology without requiring changes to the IP standard.

In the context of high-speed networks, this active processing can be targeted to reducing the length of control loops, minimising the effects of large bandwidth- \times -delay products.

V. PRACTICAL CHALLENGES

The opportunities and challenges previously discussed have little impact if there are significant practical barriers to the deployment and adoption of active networking technology. There are many recent examples of promising Internet technology that have faced such barriers, including multicast, QOS, and IPv6. Not surprisingly, there are a number of such practical challenges facing active networking.

A. Open Architectures and Signalling

A prerequisite for active networking is open architectures and signalling. Open flexible interfaces allow the provisioning of new protocols and services without long protocol standardisation and software design cycles, and by the network provider without the need for switch/router vendor participation and agreement. This was the primary motivation for IN in the PSTN. Active networking uses these interfaces to allow dynamic provisioning of protocols and services, and the motivation in the Internet is at least as strong as in the PSTN.

Unfortunately, switch and router vendors are not particularly eager to open their interfaces or expose their signalling and routing code. Only significant demand from consumers and network service providers will force the open architectures and signalling that enable active networking.

One glimmer of hope is in the trend to put programmable network processors in switches to support the increasingly sophisticated packet classification and scheduling desired by network service providers. This results in a switch architecture similar to Fig. 4, and may allow service providers to sneak active code into commodity switches.

B. Conservative Service Providers

Unfortunately, network service providers are also extremely conservative, with an exceptionally short planning horizon that focuses mostly on projecting the bandwidth capacity needed to meet near-term predicted user demand. It is not clear what forces must come into play to convince network service providers that active network monitoring, control, management, and service deployment can ultimately save

them time and bandwidth. The cost of deploying active networking is greater than any single typical service deployment (although arguably no worse than a major protocol upgrade, such as from IPv4 to IPv6).

It is interesting to note that network service providers have long been averse to only being a “bit pipe”. The ultimate antithesis to being a bit pipe is to provide arbitrary services and charge for compute cycles. Therefore, if network service providers were somehow able to get past their conservative nature, and if security and resource research problems are solved, active networking could be viewed as a way for network service providers to compete by selling compute cycles for user deployed and composed services.

C. Finding the Killer Application

Finding the killer application for active networking ought to be sufficient to force active network deployment. Unfortunately, this is a chicken-and-egg problem that can be expressed as the Field of Dreams vs. Killer App Dilemma [16]:

The emergence of the next “killer application” is difficult without sufficient network infrastructure. The incentive to build network infrastructure is viewed as a “field of dreams” without concrete projections of application and user demand.

While dynamic service and protocol deployment is a potential killer app to the network service providers, they are too conservative to demand the infrastructure (and perhaps too conservative to use it even if deployed). The Web became the killer app driving network bandwidth capacity, but could be deployed by *anyone* on the Internet without new infrastructure. The Web also ought to be driving low latency (sub-second) networks to provide truly interactive service, but this hasn't yet happened.

Perhaps the advent of network processors, as mentioned above, in conjunction with increasingly complex demands for packet classification for service level agreements and VPNs (virtual private networks) will drive the desire to enable active networking infrastructure.

Research programs that promote the construction of infrastructure, such as the US DARPA sponsored ABone (active networks backbone overlay) [20] and the EU FAIN testbed [21], can help provide the field of dreams on which to find the killer app. These testbeds need to be large scale and widely accessible from the Internet, however, to provide such an opportunity driven by a large community of researchers and users.

VI. ACKNOWLEDGEMENTS

Rajesh Krishnan and Alden Jackson provided helpful comments during the preparation of this paper. The ideas reflected in this paper are the result of conversations with many people, both in the preparation of [16], as well as in the DARPA active networks community.

VII. REFERENCES

- [1] S. Bhattacharjee, K.L. Calvert, E.W. Zegura, and J.P.G. Sterbenz, "Directions in Active Networks", *IEEE Communications*, vol.36 #10, Oct. 1998, pp. 72–78.
- [2] S. Bhattacharjee, K.L. Calvert, E.W. Zegura; C. Partridge, T. Strayer, B. Schwartz, A.W. Jackson; D.P. Reed, J.H. Slatzer, D.D. Clark "Commentaries on 'Active Networking and End-to-End Arguments'", *IEEE Network* vol.12 #3, May/Jun. 1998, pp. 66–71.
- [3] K.L. Calvert editor, *Architectural Framework for Active Networks* version 1.0, DARPA Active Network architecture document, Jul. 1999, available from <http://www.dcs.uky.edu/~calvert/arch-docs.html>.
- [4] V.G. Cerf and R.E. Lyons, "Military Requirements for Packet-Switched Networks and Their Implications for Protocol Standardization", *Computer Networks*, vol.7 #5, Elsevier Science / North-Holland, Amsterdam NL, Oct. 1983, pp. 293–306.
- [5] D.D. Clark, "Design Philosophy of the DARPA Internet Protocols", *Proceedings of ACM SIGCOMM'88* (Stanford CA US), *Computer Communication Review*, vol.18 #4, Aug. 1988, pp. 186–114.
- [6] D.S. Decasper, B. Plattner, G.M. Parulkar, S. Choi, J.D. DeHart, and T. Wolf, "A Scalable, High-Performance Active Network Node", *IEEE Network* vol.13 #1, Jan./Feb/ 1999, pp. 8–19.
- [7] A.W. Jackson and J.P.G. Sterbenz, *Introduction to Active Networks*, tutorial presentation, available from <http://www.ir.bbn.com/projects/sencomm/doc/an-tutorial.pdf>.
- [8] G. Lauer, J.P.G. Sterbenz, and I. Zibman, "Broadband Intelligent Network Architecture", *Proceedings of the IEEE/ICCC Intelligent Network Workshop (IN'95)*, Ottawa, ON, May 1995.
- [9] J.W. Lockwood, J.S. Turner, and D.E. Taylor, "Field Programmable Port Extender (FPX) for Distributed Routing and Queuing", *ACM International Symposium on Field Programmable Gate Arrays FPGA'2000* (Monterey, CA), February 2000, pp. 137–144.
- [10] J-P. Nussbaumer, B.V. Patel, F. Schaffa, and J.P.G. Sterbenz, "Networking Requirements for Interactive Video on Demand", *IEEE Journal on Selected Areas in Communications*, vol.13, #5, June 1995, pp.779–787.
- [11] B. Plattner and J.P.G. Sterbenz, *Mobile Wireless Active Networking: Issues and Research Agenda*, presented at the DARPA Active Nets Next Generation Workshop, Orlando FL, Dec. 2001, available from <http://www.ir.bbn.com/projects/sencomm/papers/mwan-research.html>.
- [12] C. Qiao and M. Yoo, "Optical Burst Switching – A New Paradigm for an Optical Internet", *Journal of High Speed Networks*, vol.8 #1, 1999, pp. 69–84.
- [13] R. Ramanathan and R.R. Hain, "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment", *Proceedings of IEEE INFOCOM 2000* (Tel Aviv IL), vol.2, Mar. 2000, pp.404–413.
- [14] J.H. Saltzer, D.P. Reed, and D.D. Clark, "End-to-End Arguments in System Design," *ACM Transactions on Computer Systems*, vol.2 #4, ACM, Nov. 1984, 227–288.
- [15] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, *Middlebox Communication Architecture and Framework*, Internet Draft `draft-ietf-midcom-framework-06.txt`, work in progress, Dec. 2001.
- [16] J.P.G. Sterbenz and J.D. Touch, *High Speed Networking: A Systematic Approach to High-Bandwidth Low-Latency Communication*, John Wiley, New York, 2001.
- [17] D.L. Tennenhouse and D.J. Wetherall, "Toward an Active Network Architecture", *ACM Computer Communication Review*, vol.26, no.2, April 1996, pp. 5–18.
- [18] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, and G.J. Minden, "A Survey of Active Network Research", *IEEE Communications*, vol.35 #1, Jan. 1997, pp. 80–86.
- [19] J.S. Turner, "Terabit Burst Switching", *Journal of High Speed Networks*, vol.8 #1, IOS Press, Amsterdam NL, 1999, pp. 3–16.
- [20] Active Network Backbone, <http://www.isi.edu/abone>
- [21] Future Active IP Networks, <http://www.ist-fain.org>