Protocols for High-Speed Networks: A Brief Retrospective Survey of High-Speed Networking Research

James P.G. Sterbenz

BBN Technologies, 10 Moulton St., Cambridge, MA 02138-1191 USA
jpgs@ieee.org
http://www.ir.bbn.com/~jpgs

Abstract. This paper considers high-speed networking research from a historical perspective, and in the context of the development of networks. A set of axioms guiding high-speed network research and design are first presented: Ø KNOW THE PAST; I APPLICATION PRIMACY; II HIGH PERFORMANCE PATHS; III LIMITING CONSTRAINTS; IV SYSTEMIC OPTIMISATION. A framework of network generations is used as the basis for the historical development of high-speed networking: 1st – Emergence; 2nd – Internet; 3rd – Convergence and the Web; 4th – Scale, Ubiquity, and Mobility. Each generation is described in terms of its application drivers, and important infrastructure and architectural characteristics. Woven into this historical thread are the important research thrusts and sub-disciplines of high-speed networking, and their impact on deployment of the Global Information Infrastructure. Based on this historical perspective, a set of SYSTEMIC OPTIMISATION PRINCIPLES are identified: 1 SELECTIVE OPTIMISATION; 2 RESOURCE TRADEOFFS; 3 END-TO-END ARGUMENTS; 4 PROTOCOL LAYERING; 5 STATE MANAGEMENT; 6 CONTROL MECHANISM LATENCY; 7 DISTRIBUTED DATA; 8 PROTOCOL DATA UNITS. We are now in the state where everything has some aspect of high speed networking, and nothing is only about high-speed networking. This is a double-edged sword — while it reflects the maturity of the discipline, it also means that very few people are looking after the performance of the entire Internet as a *system of systems*. Rather, performance analysis tends to be isolated to individual network components, protocols, or applications. Furthermore, the high-speed networking community is not pushing back at the multitude of deployment hacks by network and application service providers (such as middleboxes) without regard to global network performance effects. Thus, this paper argues that the high-speed networking community should have the future role of caring about high-speed network deployment on a *global* scale, and throughout the entire protocol stack from layers 1 through 7.

1 Introduction

Over the last twenty years or so, the discipline of high-speed networking has seen an emergence, significant activity, and melding into the mainstream of network research as a mature field. This paper aims to survey some of the most significant thrusts of high-speed networking in a historical context.

G. Carle and M. Zitterbart (Eds.): PfHSN 2002, LNCS 2334, pp. 243–265, 2002.

High-speed networking is difficult to define, because what constitutes "high-speed" changes with time, as technology progresses and applications develop. Over time, the switching rates of electronic and photonic components increase, and the density of VLSI chips and optical components increase. This results in higher available bandwidths (data rates), increased processing capabilities, and larger memories available in the network and in end systems. Furthermore, the effective rate at which network components operate decreases as we move up the protocol stack and from the network core out to the end system. There are two reasons for this. The need for the network to aggregate vast numbers of high-performance interapplication flows dictates that the core of the network must be higher capacity than the end-systems. Furthermore, it is easier to design components for high-speed communication whose sole purpose is networking, than it is to optimise end systems with multiple roles and applications.

In the late 1990s, deployed link layers and multiplexors (such SONET) operated on the order of tens of Gb/s, switches and routers at several Gb/s *per* link, end system host–network interfaces in the range of 10–100 Mb/s, and applications typically on the order of several Mb/s. By the early 2000s, link bandwidth and switch bandwidth had increased an order of magnitude or two, but local access and end system bandwidth continued to lag.

High-speed networking consists not only of the quest for high bandwidth, but also for low latency (or the perception thereof) and in the ability to cope with high bandwidth-×-delay product paths; these will be motivated in the next section.

This paper draws heavily and quotes from two earlier works by the same author. The historical framework as a series of networking generations was introduced in 1994 at *Protocols for High Speed Networks* in [25], and later updated by [26] with the addition of a fourth generation. The high-speed networking axioms and principles were developed for [26], from which the figures and much of the text in Section 3 is derived.

The rest of this paper is organised as follows: Section 2 introduces a set of axioms to guide and motivate high-speed networking research. Section 3 presents a historical view of networking as a sequence of generations. Into this generational perspective are woven some of the most important research pursuits within the high-speed networking discipline, as are a set of high-speed networking principles. While a few references to the literature are provided, [26] should be consulted for a significantly more comprehensive and complete bibliography. Section 4 considers the future role of high-speed networking research.

2 Axioms for High-Speed Networking Research

High-speed networking is a mature discipline, but there has been little attempt to structure and document the axioms and principles that have guided high-speed networking research and system design. In this section a guiding set of axioms are presented (quoted from [26]):

Ø. KNOW THE PAST, PRESENT, AND FUTURE: *Genuinely new ideas are extremely rare. Almost every "new" idea has a past full of lessons that can*

either be learned or ignored. "Old" ideas look different in the present because the context in which they have reappeared is different. Understanding the difference tells us which lessons to learn from the past and which to ignore. The future hasn't happened yet, and is guaranteed to contain at least one completely unexpected discovery that changes everything.

I. APPLICATION PRIMACY: *The sole and entire point of building a high-performance network infrastructure is to support the distributed applications that need it. Interapplication delay drives the need for high-bandwidth low-latency networks.*

This principle is the motivation for why we need high-speed networking; if inter-application delay is low enough, there is no difference to the user between a centralised application and one that is distributed across the globe. End-to-end latency must clearly be low enough, and bandwidth must be high enough that the transmission delay of data (first bit to last bit) is small enough.

II. HIGH-PERFORMANCE PATHS GOAL: *The network and end systems must provide a low-latency high-bandwidth path between applications to support low interapplication delay.*

III. LIMITING CONSTRAINTS: *Real-world constraints make it difficult to provide high-performance paths to applications.*

These constraints include the speed of light, limits on channel capacity and switching rate, heterogeneity, policy and administration, cost and feasibility, backward compatibility, and standards. It is important to attempt to distinguish reasonable constraints from those that do not have sound basis; this will be reconsidered in Section 4.

IV. SYSTEMIC OPTIMISATION PRINCIPLE: *Networks are* systems of systems *with complex compositions and interactions at multiple levels of hardware and software. These pieces must be analysed and optimised in concert with one another.*

It is this axiom that can be refined into a number of high-speed networking principles. In the next section, these refinements will be introduced in the historical context of high-speed networking research.

3 A Brief History of High-Speed Network Research

This section traces the intertwined history of network development with high-speed networking research. The history of networking can be divided into generations [25, 26] that capture significantly different characteristics in their development and deployment. Coincidentally, these generations correspond roughly to the four decades since the 1970s.

3.1 First Generation – Emergence

The first generation lasted through roughly the 1970s and is characterised by three distinct categories: voice communication, broadcast entertainment, and data networking, each of which was carried by a different infrastructure. Voice

communication was either analog circuit switched over copper wire (and microwave relays) in the public switched telephone network (PSTN), or free space analog radio transmission between transceivers. Entertainment broadcasts to radio receivers and televisions were carried by free space broadcast of RF (radio frequency) transmissions. There was little need for high-speed networking since all of these applications had well-defined, and relatively low bandwidth requirements.

Data communications was the latest entrant, and provided only a means to connect terminals to a host. This was accomplished either by serial link local communications (*e.g.* RS-232 or Binary Synchronous Communications used on mainframes), or by modem connections over telephone lines for remote access; in both cases copper wire was the physical medium. Packet networking began to emerge in the wired network in ARPANET and packet radio, primarily for military applications [3]. Early LAN research led to the emergence of Ethernet [15] and token ring, which used a shared medium for communication among multiple end systems.

Early packet routers used a bus-based general-purpose computer system with multiple network interfaces (NIs), which stored each packet in main memory to be forwarded out the appropriate interface after network layer protocol processing was performed. This architecture is shown in Figure 1.

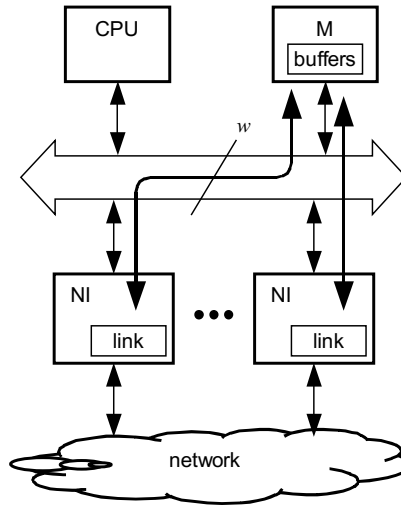

Fig. 1. First Generation Store-and-Forward IP Router

During these formative years of data networking, the discipline of high-speed networking didn't exist as such, but rather performance was one of many considerations that was brought to bear as necessary for network research and development. Store-and-forward message and packet switching and shared medium LANs were generally not the limiting performance factor given the computing capabilities of end systems and the link technologies of the 1970s.

3.2 Second Generation – the Internet

In roughly the 1980s a dramatic jump in the types and scope of networking occurred, but the three categories of communication (voice, entertainment, and data) remained relatively distinct. This period took us from the experimental ARPANET to the ubiquitous Internet.

While the end user of the voice network generally continued to use analog telephone sets, the internal network switches and trunks became largely digital, but transmission remained mostly over copper wire and microwave relay links. Additionally, there was widespread deployment of digital PBXs (private branch exchange telephone switches) on large customer premises. Mobile communications began to emerge in form of cellular telephony. The significant addition to the entertainment category of networking was the wide scale deployment of cable television (CATV) networks for entertainment video over copper coaxial cable. So while the PSTN evolved internally and CATV infrastructure was built, the application requirements saw little change. The impetus for high-speed networking consisted primarily of increasing aggregate link and switching capacity deployed deep within the PSTN.

In data networking, we first saw the emergence of consumer access, but in the primitive form of bulletin board systems (BBSs) and consumer online services (such as America Online, CompuServe, and Prodigy). These were essentially first generation networks made available to consumers, with modems connecting to a central server farm, and there was little impact on the emerging Internet.

Connection oriented corporate enterprise networks using protocols such as BNA, DECNET, and SNA were widely deployed, along with the deployment of public X.25 networks (used primarily as corporate virtual private networks). Most of these networks used copper wire as the predominant physical medium. These networks used incompatible architectures that were poorly interconnected with one another, if at all. While there were significant research advances in the context of enterprise networks in the second generation, there was little direct impact on the emerging Internet.

The collection of research and education networks such as BITNET, CSNET, and UUNET were collectively referred to as the Matrix [21] before they began to join the Internet, unified by IP addresses, with DNS symbolic addressing replacing bang paths. The growth in data networking for universities and the research community was significant during this period, for purposes of file transfer, remote login, electronic mail, and Usenet news. The technology employed was the packet switched Internet, utilising the TCP/IP protocol suite. In the wide area, the backbone network consisted of store-and-forward routers connected by leased 56kb/s telephone lines. The NSFNET upgraded the infrastructure to 1.5 Mb/s T1 lines (and ultimately 45Mb/s T3 lines at the transition into the third generation). In the local area, shared media Ethernet and token ring networks became ubiquitous and allowed clusters of workstations and PCs to network with file and compute servers.

The second generation is the time in which high-speed networking came into existence as a distinct discipline. As the Internet and enterprise networks came into wide use by the research and business communities, respectively, there was a corresponding desire to support more sophisticated applications with better

performance. Remote terminal access was significantly slower than local connections, while file and document transfers took longer than users desired. It was clear that higher speed networks would enhance productivity. Similarly, it was recognised that significantly higher aggregate bandwidth was needed to satisfy the increasing demand for network services (as in the case of the PSTN). This drove the Internet link bandwidth mentioned above, enabling more users to put greater demand on the network. The cycle of demand and capacity increase had begun, and high-speed networking research rose to meet the challenge, initially at the lower layers of the protocol stack.

Network. By the mid 1980s, the Internet was seeing significant growth, and shared medium LANs were reaching capacity, requiring the deployment of bridges to increase spatial reuse, and demanding faster link rates (*e.g.* from 4Mb/s to 16Mb/s token ring, and driving research into technologies such as FDDI). The goal of the high-speed networking community was to increase network link bandwidth by a couple orders of magnitude beyond that supported by the store-and-forward IP routers and deployed shared medium LANs, to gigabit per second data rates. The initial steps were the SONET OC-3 (155Mb/s) and OC-12 (622 Mb/s) rates. Furthermore, there was the desire to support integrated networks, in which data, voice, and video could be carried on the same network.

Since conventional *per* packet datagram forwarding was too complicated to consider at these data rates in the technology of the time, fast packet switching was proposed (e.g [34]). By substantially reducing the complexity of packet processing, hardware implementation of the switching function was possible. There are four key motivations that drove fast packet switching:

1. Dramatically simplifying packet processing and forwarding lookups by establishing connection state.
2. Eliminating the store and forward latency.
3. Eliminating the contention of the general-purpose computer bus as the switching medium.
4. Adding the ability to provide QOS guarantees to emerging multimedia applications, facilitated by resources reservations for connections.

The architecture of a fast packet switch is shown in Figure 2. The goal is to blast packets through the switch without the delays of blocking due to contention in the switch fabric or need for store-and-forward buffering.

PROTOCOL DATA UNIT PRINCIPLE: *The size and structure of protocol data units are critical to high-bandwidth low-latency communication.*

Fast packet switches are based on maintaining connection state to simplify the *per* packet processing as much as possible, so that the data path can be efficiently implemented in hardware. This requires the latency of connection setup before data can be transferred; for long connections this cost is amortised over many packets, but the user and application requirements for fast connection setup were not very often considered, and there was very little emphasis on the high-speed implications of signaling and control.

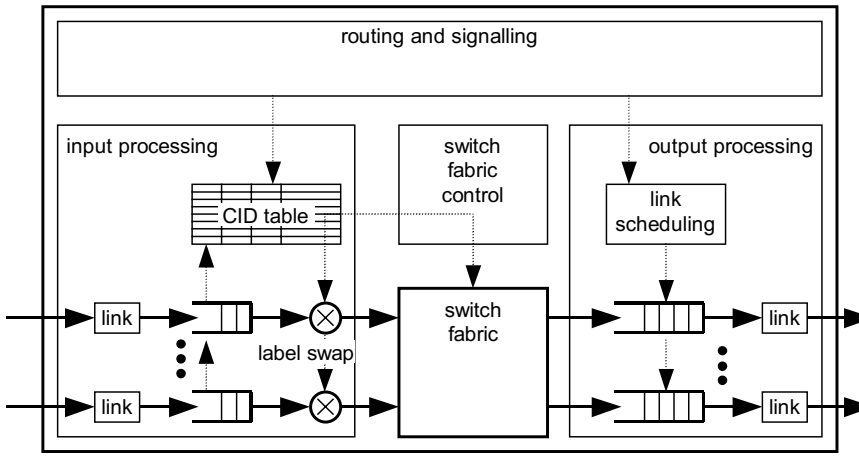

Fig. 2. Connection-Oriented Fast Packet Switch

CONTROL MECHANISM LATENCY PRINCIPLE: *Effective network control depends on the availability of accurate and current information. Control mechanisms must operate within convergence bounds that are matched to the rate of change in the network, and latency bounds to provide low interapplication delay.*

Further refinements of this principle to high-speed networking include minimising round trips for control messages, exploiting local knowledge, anticipation of future state, the proper balancing of open-and closed-loop control mechanisms, and the separation of distinct control mechanisms (such as error, flow, and congestion control). The existence of any state in the network requires its management, and difficult tradeoffs must be made:

STATE MANAGEMENT PRINCIPLE: *The mechanisms for installation and management of state should be carefully chosen to balance fast, approximate, and coarse-grained against slow, accurate, and fine-grained.*

In addition to enabling higher link rates, switched networks overcame many of the scalability limitation of shared medium networks, such as (the original) Ethernet and token ring. By using link protocols scalable in data rate (such as SONET), in conjunction with scalable switch architectures, networks could be easily grown in capacity, by increasing link rate and adding links, respectively.

The fast packet switching research took on a life of its own however, with the codification of ATM standards. Unfortunately, rather than migrating fast packet switching research technology into the Internet, an entire layer 3 routing, addressing, and signaling architecture was built for ATM, intended by its proponents to replace IP. And ATM was fraught with significant design flaws, from the small cell size that pushed the technology curve so far out that point-to-point 100Mb/s Ethernet chips became cheaper and more ubiquitous than ATM UNI chips, to complex traffic

management and inefficient signaling messages. For better or worse, by the late 1980s, the IP based Internet had become the global information infrastructure, and any attempt to replace it was a futile exercise.

Transport Layer and End Systems. The late 1980s (into the early 1990s) saw intense research at the transport layer and in end system and host–network interface architecture (*e.g.* [7,9,23,33,36]. It also produced one of the most important principles to guide where functionality could, and should be placed, the *end-to-end arguments* [22], paraphrased here:

END-TO-END ARGUMENT: Functions required by communicating applications can be correctly and completely implemented only with the knowledge and help of the applications themselves. Providing these functions as features within the network is self is not possible.

This principle tells us that certain functions, such as end-to-end error control and encryption *must* be provided at the endpoints. Providing this functionality in the network does not preclude the need for end system implementation, and thus may be a waste of resources in the network.

END-TO-END PERFORMANCE ENHANCEMENT COROLLARY: *It may be useful to duplicate an end-to-end function hop-by-hop, if doing so results in an overall (end-to-end) improvement in performance.*

There are indeed justifications for a simple network that does not heavily rely on embedded stateful functions, including better resilience to link or node failures; this is one of the key ARPANET design decisions [14]. But the end-to-end arguments do not argue for a simple network *per se*. Rather the argument is that end-to-end functions should not be *redundantly* located in the network, but rather replicated where necessary to only to improve performance. This is a key principle in high-speed networking that indicates, for example, that hop-by-hop error control can shorten control loops such that end-to-end error control can be exerted less frequently with an overall reduction in latency to applications.

As deployed network bandwidth increased, and fast packet switch prototypes were built, it was recognised that the bottleneck in end-to-end communication was moving to the edges of the network. There was a period when the grand challenge of communications was to design networks capable of transferring data at rates in excess of 1 Gb/s. While fast packet switching research suggested that this was feasible in the network, delivering this bandwidth end-to-end was (and still is) more challenging. Protocol processing was constraining distributed processing, and it was commonly thought that the key bottleneck lay in the transport protocols. This resulted in significant debate between the advocates of new transport protocols, those who thought that protocols should be implemented in hardware in the network adapter, and those who thought that TCP would perform quite well if implemented properly. The following conjectures sumarise these positions:

EC1: Designing a new transport protocol enables high-speed communication
EC2: Implementing protocols on the host–network interface will enable high-speed networking
EC3: Implementing protocol functionality in hardware speeds it up

While there is some basis for each of these statements, the mere replacement of an existing transport protocol (such as TCP) by a new transport protocol and implementing it in hardware on the host–network interface does not in itself solve the problem.

In the end, the transport protocol debate was irrelevant, due to the explosion of the Internet and pervasiveness of TCP. TCP is now the legacy data transport protocol of the global Internet, and for better or worse will be with us indefinitely. Thus the main thrust of research became how to optimise TCP for high performance given the evolution of high-speed network infrastructure, and what changes can be made in the protocol without breaking previous implementations [11].

It is critical to analyse existing end system architectures to determine where overhead and bottlenecks lie; this is the SYSTEMIC OPTIMISATION PRINCIPLE. It does little good to highly optimise operations that are not part of the bottleneck, or to create other bottlenecks as a side effect of an optimization, which leads to a particularly important refinement:

SELECTIVE OPTIMISATION PRINCIPLE: *It is neither practical nor feasible to optimise everything. Spend time and system cost on the most important constituents to performance.*

Considerations of the tradeoffs between hardware and software protocol functionality [4] and wide dissemination of the analysis of an existing protocol (TCP over IP) [5] provided needed perspective on where the bottlenecks really are, and what needed fixing. It was observed that the significant overheads were in the operating system and in *per*-byte operations such as checksumming and copying, as well as timer management. The approach shifted to systemic analysis and elimination of bottlenecks in the *critical path* with emphasis on related operating system and protocol implementation efficiencies, as well as providing sufficient memory bandwidth to the network interface [6,18,23]. These systematic analyses showed that areas to consider for reform included eliminating or reducing copies [23] and revisiting the I/O abstraction for communications [8,24]; that is, communication should be a first-class citizen, like memory or native graphics interfaces.

Protocol bypass [37] is a technique to optimise critical path processing, as shown in Figure 3.

The entire protocol stack is analysed to identify frequent operations, which are put in the bypass path, consisting of a single process without internal concurrency. A template is used to store packet header fields to quickly match or build headers (as in TCP header prediction). Data in the bypass path is shared with the conventional protocol stack. The templates are state that can be created by connection setup, or created dynamically in a data driven manner.

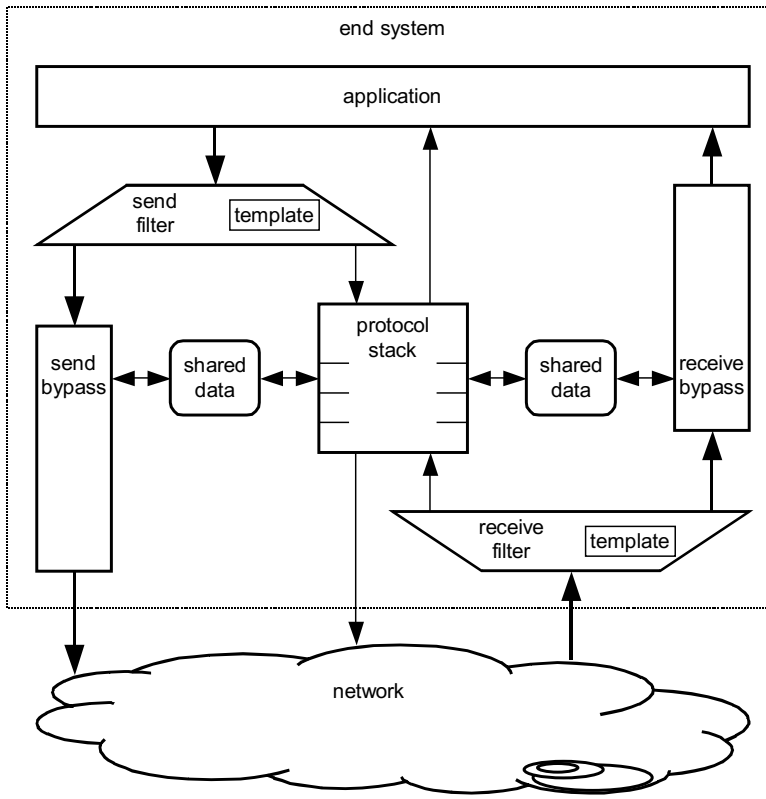

Fig. 3. Protocol Bypass

Many of the control overheads were a result of process *per* layer implementation of the protocol stack in conjunction with I/O mechanisms that were never designed for extremely high data rates, leading to the important principle that layering as an abstraction need not lead to layering as an implementation technique:

PROTOCOL LAYERING PRINCIPLE: *Layering is a useful abstraction for thinking about networking system architecture and for organizing protocols based on network structure. Layered protocol architecture should not be confused with inefficient layer implementation techniques.*

Integrated layer processing [6] is a way to overcome the overhead of layered system implementations, and can be viewed as the way to efficiently implement the bypass path described above. In a conventional layered protocol implementation, transport and network layer processing of data would consist of multiple distinct loops, as shown in Figure 4a.

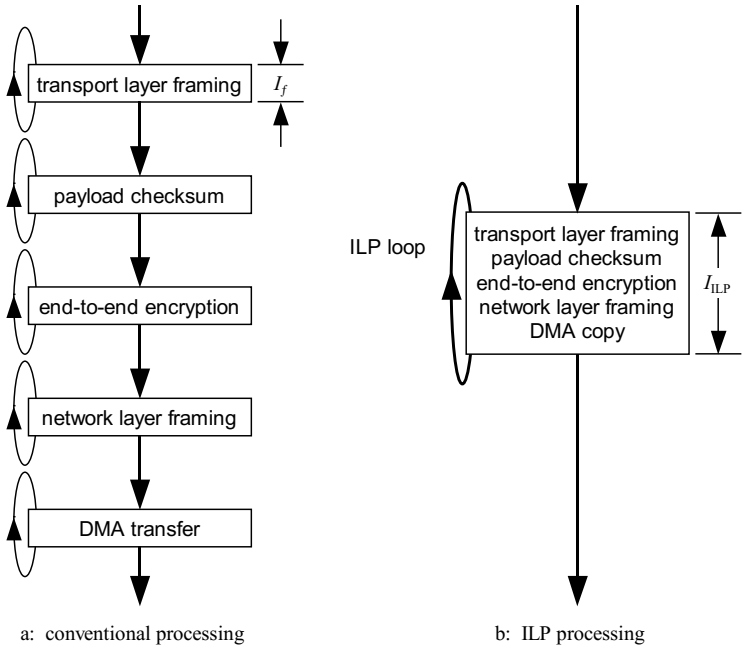

Fig. 4. Conventional and ILP Processing for TCP/IP

By employing ILP, all of the functions are processed in a single ILP code loop, as shown in Figure 4b. There are substantial savings in datapath processing by doing this. By leaving the data in place, copies between layers have been eliminated. Furthermore, joint code optimisations within a layer for *per* byte operations such as checksum and encryption may be possible. Additionally, by merging the processing loops for the various functions and putting them together, the overhead involved with transfer of control between the layers and functions is reduced. Hardware versions of ILP are also possible [23,10].

3.3 Third Generation – Convergence and the Web

The 1990s saw the emergence of integrated services: the merging of data, voice, and entertainment video on a single network infrastructure. With the advent of IP-telephony gateways, the PSTN started to become a subnet of the Internet, and with the advent of streaming multimedia, the same became imaginable for entertainment audio and video. Network service providers scrambled to keep capacity ahead of demand in over-provisioned networks, since the QOS mechanisms to support real-time and interactive applications were just beginning to emerge.

The second generation was characterised by the packet switched Internet, X.25, and enterprise networks. The third generation was characterised by an IP based global information infrastructure (GII) increasingly based on fast packet switching

technology interconnected by fiber optic cable, and IP as the single network layer unifying previously disjoint networks.

The second significant characteristic of the third generation is in the scope of access, with consumers universally accessing the Internet with personal computers *via* Internet service providers (ISPs). Disconnected BBSs are a thing of the past, and online consumer services have become merely value-added versions of ISPs. The Internet went from being a kilobit kilonode network, through megabit meganode, approaching gigabit giganodes.

The final distinguishing characteristic of the third generation is the World Wide Web, which provided a common protocol infrastructure (HTTP), display language (HTML), and interface (Web browsers) to enable users to easily provide and access content. Web browsers became the way to access not only data in web pages, but images and streaming multimedia content. The Web became the killer app that drove bandwidth demand, and the rate of adoption of Internet connections vastly exceeded the rate of new telephone connections. In spite of the fact that the Internet and Web became the primary reason for users to have PCs, these devices were still not designed with networking as a significant architectural consideration.

In the third generation, high-speed networking research moved up the protocol stack to be more concerned with applications. Additionally, the failure of ATM and decreasing cost in hardware finally led to practical application of fast packet switching technology to IP routers in the late 1990s, which became IP switches. Optical networking saw some significant advances, which lead all but the most skeptical to consider that optical switching finally held some promise for future deployment.

This divergence of high-speed networking research into the upper and lower layers, respectively application layer and switch design, had the effect of fragmenting the discipline, and in mainstreaming high-speed networking into other sub-disciplines of communications, such as router/switch design and applications.

Internet. Demand for the Internet was steadily increasing by the end of the second generation, and short term solutions were necessary. This lead to a significant optimisation of IP router architecture to eliminate the shared CPU and memory as source of contention among the various network links. Distributing and offloading the network layer protocol processing to the NIs, as shown in Figure 5, accomplished this goal. This architecture was used in the NSFNET routers of the mid-1990s.

Packets are moved between NIs across the bus using *third party* transfers, without going through main memory. Each network interface contains a network interface processor (NIP), which performs the network layer processing, along with buffer memory for packet processing. While this significantly reduces the contention for a single memory and distributed the processing to each NI, this architecture still requires a store-and-forward hop on the NI. Furthermore, a single bus as the interconnect between all NIs significantly limits scalability.

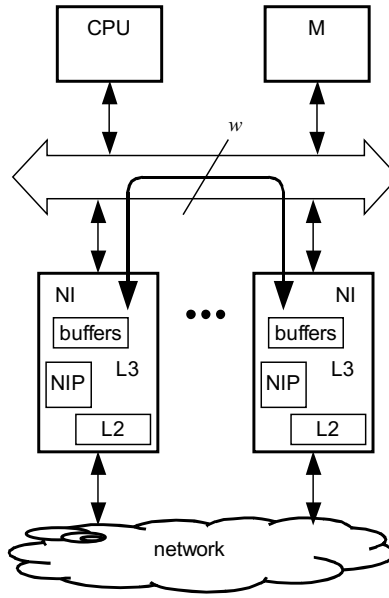

Fig. 5. Third Generation IP Router with Third-Party Bus Forwarding

Network. While research in fast packet switching had flourished in the second generation, this translated to only limited deployment in the third generation. ATM switches were deployed, but generally as islands of PVC meshes under IP. Performance was not particularly good, initially due to the inexcusable assumption that traffic was Poisson, and even after reasonably deep switch buffers were added, to incompatible forwarding and signaling mechanisms. While there were some hopes [25] and attempts (such as I-PNNI) to unify the IP and ATM frameworks, very little progress was made.

Two forces resisted the global deployment of a connection-oriented network layer, such as ATM. First, the explosion of the IP based Internet and Web in the mid 1990s entrenched TCP as the end-to-end protocol, and IP as the single global network layer; the *hourglass principle* indicates that there should only be one network layer. In the cases where connection oriented network protocols were deployed in backbone networks (such as ATM or X.25), IP traffic was run over these other network layers in a kludge of inefficient layering and incompatible control mechanisms that resulted in the native network layer being used as if it were a link layer. In the end, there was little motivation to create native ATM applications and transport protocols, or to use the ISO application protocols such as FTAM (file transfer, access, and management) or VT (virtual terminal).

Second, the limitations of shared medium link protocols such as Ethernet and token ring were overcome by the evolution of Ethernet to a switched point-to-point link protocol, with order-of-magnitude increases in data rate. This further reduced the motivation for adoption of ATM using scalable SONET links to increase the bandwidth on network links.

Finally, the dramatically increasing capabilities of VLSI processing in the 1990s finally made it feasible to consider *per* packet datagram forwarding and *per* flow queueing in switches.

RESOURCE TRADEOFF PRINCIPLE: *Networks are collections of resources. The relative composition of these resources must be balanced to optimise cost and performance. This relative cost changes over time, due to nonuniform advances in different aspects of technology.*

Therefore, much of the research community shifted their attention to speeding up connectionless datagram forwarding (*e.g.* [29,19,17]. Decreasing cost in processing resulted in shifts in resource tradeoffs that made it feasible to consider datagram processing at line rate by the mid 1990s. A full ATM layer 3 infrastructure became unnecessary, and deployments of IP over SONET (POS – packets over SONET) began, with research into IP directly over WDM (POW – packets over wavelengths). At the same time, the important characteristics of fast packet switching technologies began to be incorporated into the Internet, for example IP switches based on the fast switch fabrics, and protocol optimisations such as MPLS.

At a high level, the architecture of a fast connectionless datagram switch depicted in Figure 6 has the same functional blocks as the fast connection oriented packet switch that was shown in Figure 2.

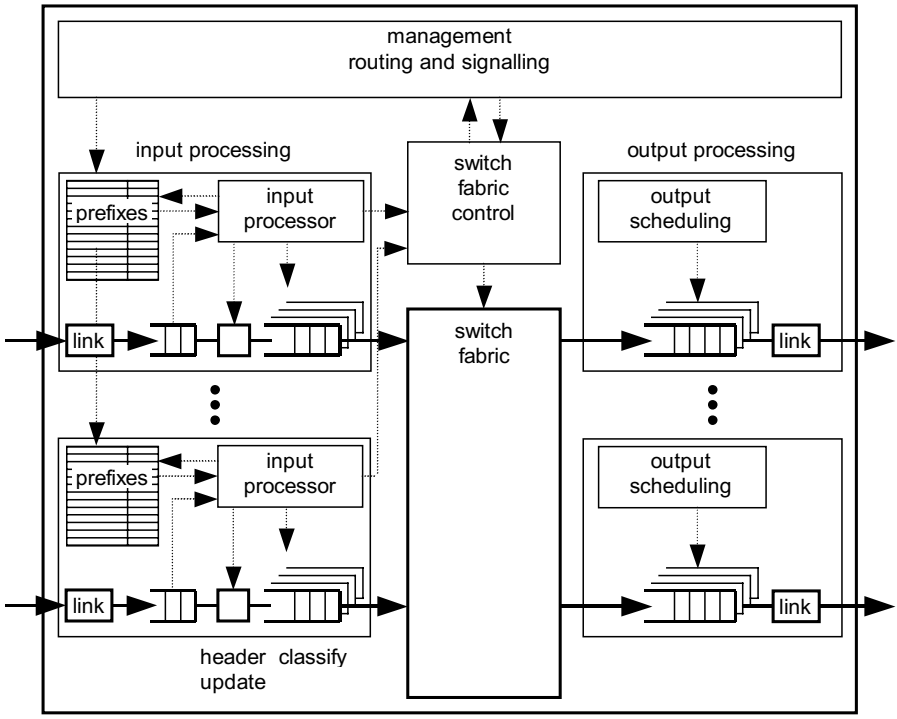

Fig. 6. Fast Datagram Switch

There is set of control software, including routing, signalling, and management, and a switch fabric core, as in a typical fast packet switch. Input and output processing are also present, and this is where the primary difference lies:

Input processing is considerably more complex, with an input processor (either a small fast RISC embedded controller or a specialised network processor) performing address lookup using a prefix table, as well as packet classification.

Output processing is also more complex, with a significant packet scheduling (including traffic shaping) to meet QOS requirement for flows, and to insure fairness among best-effort flows.

The input and output processing can either consist of custom hardware engines, or be implemented in emerging network processors. The use of network processors for this functionality opens the door to active and programmable networks, in spite or resistance by switch vendors [27].

Optical Networks. While there had long been research on optical networking, the late third generation saw significant advances in the development of optical switching components, including MEMS switch elements and the resulting optical switch fabrics; 1024×1024 research prototypes have been constructed (*e.g.* [1]). Optical switching technology provides a fast datapath, but optical logic and control circuits are beyond the ability of early fourth generation (2000s) networking. This means that all-optical *packet switching* is impractical in the near future, since the packet header cannot be decoded and processed in the optical domain. Furthermore, the switching rate of optical switch elements is relatively slow, on the order of a microsecond. Therefore, data flows must be assigned to lightpaths that are switched only infrequently. Optical burst switching [35,20] aggregates packets into bursts that can utilise the network more efficiently than circuits.

Applications. While the early third generation saw a steady increase in traffic on the Internet, primarily from educational institutions, it also saw the birth of the Web. By the mid 1990s, the exponential increase in traffic was driven by the Web, which had become ubiquitously available in universities, particularly to undergraduate students.

Applications can be classified in several ways related to their performance demands: by bandwidth aggregation, bandwidth scalability, latency requirements, and communication characteristics.

Aggregate bandwidth. An important measure of the impact of an application on the network infrastructure is the demand it places in aggregate. This is measured by the product of the *per* instance bandwidth × the number of simultaneous instantiations of the application [12]. Thus, an aggregate gigabit application might consist either of 100 simultaneous instances of a 10 Mb/s application, or 10 simultaneous instances of a 100 Mb/s application. The aggregate bandwidth of the PSTN (public switched telephone network) is generally estimated at O(1 Tb/s) as was the bandwidth of data networks in the mid 1990s (particularly the Internet, SNA, and X.25 packet networks). While it is expected that PSTN bandwidth will remain relatively flat, the aggregate bandwidth of the Internet continues to grow dramatically with no end in

sight. In the early 2000s, bundles of fibers are being laid and switches deployed that exceed 1 Tb/s.

Individual bandwidth. A single instance of an application that requires a significant fraction of the bandwidth available on a high-speed network link or high-performance end system interface can be considered a *high-speed application*. Supporting this sort of application requires high-bandwidth network infrastructure, high-speed transport protocols, and high-performance end systems. These applications clearly need to be the focus of high-speed networking research.

The bandwidth that an individual application requires is generally not a fixed quantity; most applications operate over a range of bandwidths. Thus, it is important to understand how application utility scales with available bandwidth [31]. Some applications remain structurally unchanged, becoming only faster or perceptually better; other applications have difficulty keeping pace as bandwidths scale. The bandwidth scalability of applications can be described using the following taxonomy [16]:

Bandwidth Enhanced. The application operates at various bandwidths. Although the application is functional at low bandwidths, it increases in utility given high-speed networking, and does not require fundamental restructuring. Streaming multimedia is the canonical example, because high bandwidth increases the achievable resolution and frame rate, with an increased perceptual quality to users.

Bandwidth Challenged. The application is useful at various bandwidths, but either requires substantial revision, or operates in a different way at high-speed. An example of a bandwidth challenged application is distributed computing. Some computations, such as *monte carlo* simulations, work with infrequent state exchange. As bandwidth increases, more sophisticated distributions of computation are possible, requiring greater control interaction and data exchange.

Bandwidth Enabled. The application is usable *only when* a high-bandwidth path is available. This may be dictated by particular bandwidth requirements of the application, for example the high data rates of uncompressed video for networked studio production of movies. It may also be the case that without a base bandwidth certain applications just don't make sense. Distributed scientific visualisation and collaborative CAD/CAE (computer aided design / engineering) fall into this type.

While all of these applications drive aggregate network bandwidth, is the bandwidth challenged and bandwidth enabled applications that present the most serious high-speed networking demands end-to-end and application-to-application.

Latency Characteristics. Latency is the other important characteristic of application demand, and can be characterised as best-effort, interactive, real-time, and deadline. Application utility curves, depicted in Figure 7, indicate how tolerant applications are to latency, and thus indicate the latency bound and its tolerance that must be provided by the network and end systems.

Clearly there is a range of tolerance ranging from best-effort (tolerant), through interactive (moderate) to deadline and real-time (intolerant).

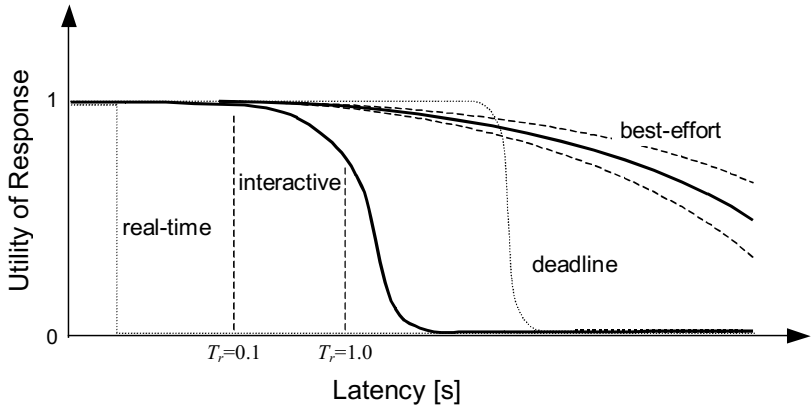

Fig. 7. Application Utility Functions

Characteristics. Additionally, applications can be categorised by characteristics class [16]: information access, telepresence, and distributed computing. Information access applications, such as the Web, are client/server with highly asymmetric bandwidth requirements and a fairly large granularity of transfers. Telepresence applications involve the exchange of information that allows users to maintain a distributed virtual presence; frequently in the form of multimedia. Telepresnece thus tends to be more symmetric in its bandwidth requirements, and individual transfers are either at small granularity or a continuous stream. Finally, distributed computing involves the distribution of computations beyond a room, and involves an arbitrary exchange of data. While requirements are highly variable on the particular computation (which is in turn designed on network capabilities), in the general case the bandwidth, latency, and synchronisation requirements can be very challenging. Other more complex application scenarios are compositions of the three core classes. For example, distance learning is a composition of telepresence (virtual classroom participation) and information access (student access to course materials).

The Web became the killer app in the mid third generation. Web browsing is an interactive information access application. It is a challenging and ubiquitous high-speed networking application, not only in aggregate, but also for each individual user browsing. Web browsing has traditionally been considered a best-effort application, but this is point, click, and wait mode. For Web browsing to meet the requirements for interactive applications (point-click-point-click), a response time in the approximate range of 100 ms to 1 second is required [25]. This latency bound drives the bandwidth requirement, especially for large web objects, such as those including embedded images. Medical and photographic-resolution images are particularly demanding.

Figure 8 [32] shows the bandwidth requirements for different types of web pages. The horizontal bands represent different types of web pages. The vertical bars indicate now much data can be transmitted over various link technologies in the 100 ms interactive response time budget, assuming the given link is the bandwidth bottleneck. Note how even modest web pages can stress analog modem and ISDN

rates. As web page sizes increase with higher resolution and 3-dimensional images (allowing local rotation), bandwidth requirements increase into the Gb/s realm.

This bandwidth demand is fueled in the consumer arena by high-resolution digital cameras and printers, coupled with the desire to deliver of digital photographs on the Web. Note that this analysis only considers the propagation delay, which assumes the entire 100 ms of response time budget can be used for data transmission. Client, server, and network node delays also contribute to the end-to-end interapplication delay, and may consume a significant portion of the latency budget, requiring even higher link bandwidths. Servers outside a 100 ms propagation radius (around 5000 km) cannot be accessed within this latency bound at all with direct request-response techniques; this motivates techniques that masking latency.

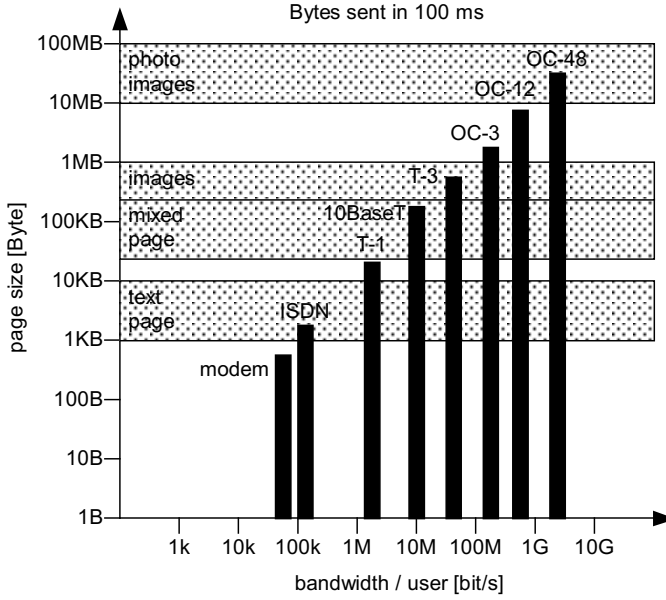

Fig. 8. Web Browsing Bandwidth Requirements

Application matching the network. Significant performance gains are possible if applications are aware of the underlying network data structures and control mechanisms; matching them can dramatically reduce the control overhead and data transformation. The PROTOCOL LAYERING PRINCIPLE introduced previously indicates that layering as an abstraction need not lead to poor implementation. ILP is one embodiment of this principle. Another is *application layer framing* (ALF), which is a technique that allows the application to more directly adapt network protocol formats and data units [6]. This reduces the overhead in protocol encapsulation, decapsulation, fragmentation, and reassembly. There are benefits in matching in the control plane as well. Unfortunately, HTTP is an application layer transaction protocol hacked onto a connection-oriented transport protocol designed for long-lived data transfers. While there were attempts to modify TCP for transactions [2] they

were not deployed, due to security flaws in TCP connection management. Furthermore, there is no attempt to match the structure of Web pages to the underlying protocol data units. Thus the Web benefits neither from a native transaction protocol, nor from ALF.

Distributed Data. The way in which data is structured and distributed across the network can have a profound influence on interapplication delay, and thus on application-to-application performance.

DISTRIBUTED DATA PRINCIPLE: *Distributed applications should select and organise the data they exchange to minimise the amount and latency of data transferred and to allow incremental processing of data.*

Unfortunately, there has been little tangible evidence of this principle in widely deployed applications. Many applications perform poorly and *seem* to need high-speed networks simply because they are poorly designed and partitioned. The Web is again an example of this problem; web content is generally not structured with performance in mind. Rather than organising data into easily transferable, displayable, and cacheable units, web page designers use authoring tools that have no cognisance of this; the overuse and misuse of dynamic content is a reflection of this problem. Furthermore, administrative and policy decisions are frequently are at odds with performance (recall the LIMITING CONSTRAINTS axiom from Section 2). While some of these limiting constraints have practical justification, they are frequently not balanced against the needs of high-speed applications. The way in which banner advertisements are implemented is an example of this problem.

Application Adaptation and Latency Masking. In an attempt to adapt to constraints on latency (primarily due to the speed-of-light over long distances) and limited bandwidth, applications can adapt to mask these effects. This depends on network feedback, and may benefit from user control [28]. Mirroring and caching are the canonical techniques to mask latency and reduce aggregate bandwidth, and this became an intense area of research in the late 1990s. There are limits to the benefits of these techniques, however, and the next step are anticipatory techniques that prefetch and preload, in an attempt to reduce response time for pages that are not yet cached. Simple examples are to prefetch pages hyperlinked in the page just requested [30] and to push preload based on user profile information located on a server. Intelligent rate adaptation and layered coding help applications to gracefully degrade as bandwidth becomes constrained.

3.4 Fourth Generation

The first decade of the new millennium inaugurates a new network generation, which will be characterised by orders of magnitude increase in network scale, and by the ubiquity of mobile wireless computing devices. The third generation was largely a wired network; the fourth generation will be largely wireless at the edges, with access to a high-speed optical backbone infrastructure, including optical switches. In the

extreme, one can envisage a network that consists *only* of wireless access to an optical backbone, although in practice copper wire will be in use for a very long time.

Advances in the interface between biological and microelectronic interfaces, and the benefits of applying micro- and macro-biotic behaviour and organisation techniques to networks may begin to emerge, or may be a characteristic of a later fifth generation.

Ubiquitous computing, smart spaces, and sensor network research suggest that individual users will carry tens to hundreds of networked devices. These will perhaps include a personal node for computing and communications to other individuals, and a wireless network of wearable I/O devices and sensors (environmental and biological). Rooms and vehicles will consist of perhaps thousands of embedded sensors and networked computing platforms. Thus we need to consider teranode and petanode networks. There are profound implications to high-speed networking. The aggregate wireless bandwidth demanded will vastly exceed the capacity of the shared medium using third generation protocols and techniques, and the highly dynamic nature will stress routing and control protocols. The ability to manage power among the massive number of autonomous wireless devices, and to do high-speed networking where power is a constraint will be a major challenge.

We will see not only end systems, the sole purpose of which is networking, but also a blurring of functionality between end systems and network nodes. In mobile networking, many devices will serve both as application platforms and as switches or routers.

While the capacity in processing, memory, and bandwidth will dramatically increase, resource tradeoffs will continue to shift. If the shifts are significant (for example several orders of magnitude increase in only one of processing, memory, or bandwidth), the future of high-speed networking will be drastically different.

The relative decrease in the cost of processing enabled the field of active networking, which may play a significant role in the fourth generation. We note that speed-of-light latency will continue to be a significant challenge, and increasingly so as we begin to build the Interplanetary Internet, initially for the Mars missions, but with an eye toward the Jupiter system.

4 The Future of High-Speed Networking as a Discipline

High-speed networking has become a mature discipline, to the point that everything has *some* aspect of high-speed networking, and nothing is *only* high-speed networking. In the late 1990s, this seemed like a reasonable state of affairs.

Unfortunately, the decline in high-speed networking as a distinct discipline seems to have lead to the situation where nobody is looking after the performance of the *entire* network as a *system of systems*. At best, component manufacturers are building high-performance subsystems (such as fast IP switches), but typically service providers deploy them in a haphazard manner to barely stay ahead of the demand curve. At worst, network providers are working at odds with one another deploying bad topologies with complex and irrational peering points that obscure performance. ASPs are deploying hacks and middleboxes without regard to the overall performance of the Internet.

Active and programmable networks may provide the mechanisms to evolve the network in a systematic manner; switches that contain network processors may allow this to happen in a rational manner in spite of switch vendors that do not wish to open their boxes, and network service providers that can't see beyond the next bandwidth capacity planning cycle.

At best, high-speed networking is in a rut [13]; at worst it has been fragmented and absorbed into the mainstream. As long as there is a community of people deeply interested in high-speed networking, there is hope. Whether this translates into an effort to restore order and performance to a chaotic network remains to be seen.

References

1. David J. Bishop, C. Randy Giles, and Gary P. Austin, "The Lucent LambdaRouter: MEMS Technology of the Future Here Today", *IEEE Communications*, vol.40 #3, IEEE, New York NY US, Mar. 2002, pp. 75–79
2. Robert Braden, *Extending TCP for Transactions – Concepts*, RFC 1379, Nov. 1992.
3. Vinton G. Cerf and Edward Cain, "The DoD Internet Architecture Model", *Computer Networks*, vol.7 #5, Elsevier Science / North-Holland, Amsterdam NL, Oct. 1983, pp. 307–318.
4. Greg Chesson, "XTP/PE Design Considerations", in *Protocols for High-Speed Networks*, IFIP PfHSN'89 (Zürich CH), May 1989, Harry Rudin and Robin Williamson editors, Elsevier / North-Holland, Amsterdam NL, 1989, pp. 27–33.
5. David D. Clark, Van Jacobson, John Romkey, and Howard Salwen, "An Analysis of TCP Processing Overhead", *IEEE Communications*, vol.27 #.6, IEEE, New York NY US, June 1989, pp. 23–29.
6. David D. Clark and David L. Tennenhouse, "Architectural Considerations for a New Generation of Protocols", *Proceedings ACM SIGCOMM'90* (Philadelpha PA US), *Computer Communication Review*, vol.20 #4, ACM, New York NY US, Sep. 1990, pp. 200–208.
7. Bruce S. Davie, "A Host–Network Interface Architecture for ATM", *Proceedings of ACM SIGCOMM'91* (Zürich CH), *Computer Communication Review*, vol.21 #4, ACM, New York NY US, Sep. 1991, pp. 307–315.
8. Gary S. Delp, Adarshpal S. Sethi, and David J. Farber, "An Analysis of Memnet: An Experiment in High-Speed Shared-Memory Local Networking", *Proceedings of ACM SIGCOMM'88* (Stanford CA US), *Computer Communication Review*, vol.18 #4, ACM, New York NY US, Aug. 1988, pp. 165–174.
9. David C. Feldmeier, "A Framework of Architectural Concepts for High-Speed Communications Systems", *IEEE Journal on Selected Areas in Communications*, vol.11 #4, IEEE, New York NY US, May 1993, pp. 480–488.
10. Zygmunt Haas, "A Communication Architecture for High Speed Networking", *Proceedings of IEEE INFOCOM'90* (San Francisco CA US), IEEE, New York NY US, June 1990, pp. 433–441.
11. Van Jacobson, Robert Braden, and David A. Borman, *TCP Extensions for High Performance*, RFC 1323 (standards track), May 1992.
12. J. Bryan Lyles, Ira Richer, and James P.G. Sterbenz, "Applications Enabling the Wide Scale Deployment of Gigabit Networks" (editorial), *IEEE Journal on Selected Areas in Communications*, vol.13 #5, IEEE, New York NY US, June 1995, pp.765–767.
13. J. Bryan Lyles, keynote address, Protocols for High-Speed Networks, Berlin DE, Apr. 2002.

14. John M. McQuillan and David Walden, “The ARPA Network Design Decisions”, *Computer Networks*, vol.1 #5, North-Holland, Amsterdam NL, Aug. 1977, pp. 243–289.
15. Robert M. Metcalfe and David R. Boggs, “Ethernet: Distributed Packet Switching for Local Computer Networks”, *Communications of the ACM*, vol.19 #5, ACM, New York NY, Jul. 1976, pp. 395–404.
16. Craig Partridge editor, *Report of the ARPA/NSF Workshop on Research in Gigabit Networking*, Washington DC, Jul. 1994, available from `http://www.cise.nsf.gov/anir/giga/craig.txt`.
17. Craig Partridge, Philip P. Carvey, Ed Burgess, Isidro Castineyra, Tom Clarke, Lise Graham, Michael Hathaway, Phil Herman, Allen King, Steve Kolhami, Tracy Ma, John Mcallen, Trevor Mendez, Walter C. Milliken, Ronald Pettyjohn, John Rokosz, Joshua Seeger, Michael Sollins, Steve Storch, Benjamin Tober, Gregory D. Troxel, David Waitzman, and Scott Winterble, “A 50-Gb/s IP Router”, IEEE/ACM Transactions on Networking, vol.6 #3, IEEE / ACM, New York NY US, Jun. 1998, pp. 237–248.
18. Gurudatta M. Parulkar and Jonathan S. Turner, “Towards a Framework for High-Speed Communication in a Heterogeneous Networking Environment”, *IEEE Network*, vol.4 #2, IEEE, New York NY US, Mar. 1990, pp. 19–27.
19. Guru Parulkar, Douglas C. Schmidt, and Jonathan S. Turner, “$a^{I}t^{P}m$: A Strategy for Integrating IP with ATM”, *Proceedings of ACM SIGCOMM’95*, (Cambridge MA US), *Computer Communication Review*, vol.25 #4, ACM, New York NY US, Aug. 1995, pp. 49–57.
20. Chunming Qiao and Myungsik Yoo, “Optical Burst Switching – A New Paradigm for an Optical Internet”, *Journal of High Speed Networks*, vol.8 #1, 1999, pp. 69–84.
21. John S. Quarterman, *The Matrix: Computer Networks and Conferencing Systems Worldwide*, Digital Press, Maynard MA US, 1989.
22. J.H. Saltzer, D.P. Reed, and D.D. Clark, “End-to-end Arguments in System Design,” *Proceedings of the Second International Conference on Distributed Computing Systems (ICDCS)*, IEEE, New York NY US, 1981, pp. 509–512, also *ACM Transactions on Computer Systems*, vol.2 #4, ACM, New York NY US, Nov. 1984, 227–288.
23. James P.G. Sterbenz and Gurudatta M. Parulkar, “Axon: A Distributed Communication Architecture for High-Speed Networking”, *Proceedings of IEEE INFOCOM’90* (San Francisco CA US), June 1990, pp 415–425.
24. James P.G. Sterbenz and Gurudatta M. Parulkar, “Axon Network Virtual Storage for High Performance Distributed Applications”, *Proceedings of 10th International Conference on Distributed Computing Systems ICDCS* (Paris FR), IEEE, New York NY US, June 1990, pp 484–492.
25. James P.G. Sterbenz, “Protocols for High Speed Networks: Life After ATM?”, *Protocols for High Speed Networks IV*, IFIP/IEEE PfHSN’94 (Vancouver BC CA), Aug. 1994, Gerald Neufeld and Mabo Ito, editors, Chapman & Hall, London UK / Kluwer Academic Publishers, Norwell MA US, 1995, pp. 3–18.
26. James P.G. Sterbenz and Joseph D. Touch, High-Speed Networking: A Systematic Approach to High-Bandwidth Low-Latency Communication, John Wiley, New York NY US, 2001.
27. James P.G. Sterbenz, “Intelligence in Future Broadband Networks: Challenges and Opportunities in High-Speed Active Networking”, *Proceedings of IEEE International Zürich Seminar on Broadband Communications IZS 2002* (Zürich CH), IEEE, New York, Feb. 2002, pp. 2-1–2-7.
28. James P.G. Sterbenz, Rajesh Krishnan, and Tushar Saxena, *Latency Aware Information Acces with User Directed Handling of Cache Misses: Web VADE MECUM*, `http://www.ir.bbn.com/projects/wvm`.
29. Ahmed Tantawy and Martina Zitterbart, “Multiprocessing in High Performance IP Routers”, *Protocols for High Speed Networks III*, IFIP PfHSN’92 (Stockholm SE), May

1992, Per Gunningberg, Björn Perhson, and Stephen Pink editors, Elsevier / North-Holland, Amsterdam NL, 1993, pp. 235–254.
30. Joseph D. Touch, "Parallel Communication" *Proceedings of INFOCOM'93* (San Francisco CA US), IEEE, New York NY US, Mar. 1993, pp. 506–512.
31. Joseph D. Touch, "Defining 'High Speed' Protocols : Five Challenges and an Example That Survives the Challenges", *IEEE Journal on Selected Areas in Communications*, vol.13, #5, IEEE, New York NY US, June 1995, pp. 828–835.
32. Joseph D. Touch., "High Performance Web", animation session, *Protocols for High-Speed Network*, IFIP/IEEE, PfHSN'96 (Sophia-Antipolis, FR), Oct. 1996.
33. C. Brandan S. Traw and Jonathan M. Smith, "Hardware/Software Organization of a High-Performance ATM Host Interface", *IEEE Journal on Selected Areas in Communications*, vol.11 #2, IEEE, New York NY US, Feb. 1993, pp.228–239.
34. Jonathan S. Turner, "Design of an Integrated Services *Packet* Network", *IEEE Journal on Selected Areas in Communications*, vol.SAC-4 #8, IEEE, New York NY US, Nov. 1986, pp. 1373–1380.
35. Jonathan S. Turner, "Terabit Burst Switching", *Journal of High Speed Networks*, vol.8 #1, IOS Press, Amsterdam NL, 1999, pp. 3–16.
36. Richard W. Watson and Sandy A. Mamrak, "Gaining Efficiency in Transport Services by Appropriate Design and Implementation Choices", *ACM Transactions on Computer Systems*, vol.5 #2, May 1987, pp. 97–120.
37. C.M. Woodside, K. Ravinadran, and R.G. Franks, "The Protocol Bypass Concept for High Speed OSI Data Transfer." *Protocols for High-Speed Networks II*, IFIP PfHSN'1990 (Palo Alto CA US), Oct. 1990, Marjory Johnson editor, Elsevier / North-Holland, Amsterdam NL, 1991, pp. 107–122.