TCP/IP-over-ATM Implementation for the Radiology Consultation WorkStation

Jeremy Elson

3 July 1997

Computational Bioscience and Engineering Laboratory
Division of Computer Research and Technology
National Institutes of Health
Bethesda, MD 20892, USA

Abstract

The Radiology Consultation Workstation makes extensive use of the TCP/IP protocol suite. Using TCP/IP over ATM networks is not entirely straightforward, and a number of issues must be considered carefully to avoid potential pitfalls. At the ATM layer, drivers and software on both the host and the switch must conform to compatible signaling standards. At the IP layer, there are different IP-over-ATM standards from which to choose, each with strengths and weaknesses. There are also a number of serious performance issues to be considered; some problems are inherent to running TCP/IP at high data rates, and some are inherent to ATM itself. Each of these issues must also be considered in conjunction with the realities of which technologies are immediately available to us. This paper will examine many of these issues from the practical perspective of: "What is required to build a clinical RCWS system—today?"

Contents

1 Introduction

The Radiology Consultation Workstation project [1] makes extensive use of the TCP/IP protocol suite. All application-level protocols, implemented both by our software and various system utilities, use the TCP/IP and UDP/IP services provided by the Solaris kernel. For example, EVE's message-passing system is implemented using TCP/IP, encompassing both short, bursty messages such as those used for synchronization, and the very large messages used for distribution of images before a consultation session. UDP/IP is used for system functions such as clock synchronization and domain nameserver (DNS) requests.

IP [2] is a network-layer protocol (at OSI layer 3) implemented by the Solaris kernel. It is a connectionless, datagram-oriented protocol responsible for routing individual datagrams through a heterogeneous internetwork using best-effort delivery. "Best-effort" means that correct delivery of datagrams is not guaranteed: they can be lost, duplicated, or arrive out-of-order.

TCP and UDP [2] are both transport-layer protocols (at OSI layer 4) also implemented by the Solaris kernel, above IP. TCP is responsible for reliable, end-to-end data transmission by acknowledging received data, performing retransmissions, and resequencing incoming IP datagrams to reconstruct the original data stream from the unreliable IP layer. In contrast, UDP is simply a "raw" interface to IP; UDP datagrams are as unreliable as IP datagrams. Applications access the TCP and UDP services of the kernel (and thus the IP services) through a set of kernel system calls using C; for example, the C functions `socket()`, `bind()`, `listen()` and `accept()`.

There are a number of important issues to consider when using the TCP/IP protocol over an ATM network. This document is not meant to be a comprehensive study of all the details involved in a TCP/IP-over-ATM implementation, but rather a survey of the most important issues from the practical point of view of the RCWS project. In Section 2, we will review some of the more important low-level ATM standards. Section 3 will consider various alternative standards available for encoding and sending IP datagrams on an ATM network. In Section 4, several important TCP/IP-over-ATM performance issues will be considered.

It should be noted that everything from the IP layer up to the top of the OSI stack remains completely unchanged regardless of the underlying network in use. The applications need not be changed because the API that allows applications to access TCP services remains unchanged.

2 ATM Standards

Before any higher-layer protocols such as IP can be implemented over ATM, it is first necessary to standardize the ATM layer on which those protocols are built. The task of coordinating the ATM standardization effort is handled primarily by the ATM Forum, an international non-profit organization formed with the objective of accelerating the use of ATM products and services through a rapid convergence of interoperability specifications [3].

When ATM was still very new, few standards existed at any level. ATM vendors developed proprietary hardware and software, making interoperability impossible. In the beginning, even the physical layer lacked standards—different vendors' ATM hardware worked at different clock speeds with different framing methods! However, the clock speeds and SONET/SDH framing standards stabilized several years ago and are now universally implemented.[1] Slowly, as the ATM Forum's higher-layer standards have evolved, true interoperability has become more and more of a reality.

2.1 Host–to–Switch Standards

ILMI (the Interim Local Management Interface) and UNI (the User–Network Interface) [5] are configuration and signaling protocols that allow ATM end-stations to exchange information with the ATM switch to which they are directly attached. ILMI is used for basic configuration functions; for example, it is used by ATM switches for informing ATM end-stations of their NSAP prefixes when the end-station boots. UNI is commonly used by ATM end-stations to request a virtual circuit (VC) in the ATM network from their local ATM switch. All switch vendors developed proprietary methods of creating VCs before the UNI standard existed, and these incompatible standards are still in use at some sites. However, in a multi-vendor network, adherence to the new standards is critical.

SynOptics CMS Version 1.1.x, the original CMS used on our SynOptics LattisCell switches, did not support UNI. PVCs could only be created from the SynOptics Network Management Application (NMA), and the ENI NICs were unable to request new VCs from the switch. However, UNI 3.0 is supported both by version 1.2 of CMS and Version 3.3.0 of the ENI NIC drivers. Using UNI, the ENI NICs can request VCs from the SynOptics switch, which is a necessary first step before IP-over-ATM can be run transparently.

2.2 Switch–to–Switch Standards

When an ATM end-station requests a VC from the ATM network using UNI, it specifies only the desired destination address, not the route through the network. Analogous situations exist on the Internet (the user only needs to specify the destination IP address, not the sequence of routers through which her data travels) and in the telephone system (the user only specifies the desired destination telephone number, not the route through the telephone network). It is the responsibility of the ATM network, not the end-station, to perform routing from the source to the destination switch. To extend our telephone analogy, UNI is like the keypad of a telephone—it specifies the language with which the user requests connections from the network, but has no bearing on the path through the network.

For interswitch routing to be possible, switches in an ATM network must both be able to request routing services from each other and exchange meaningful network topology information. Before standardized switch–to–switch protocols were established, vendors developed their own proprietary protocols to serve these needs. The resulting situation was that automatic routing of calls through a network of switches from the same vendor was possible, but routing in an internetwork of heterogeneous switches was not.

[1] While SONET framing is typically used in the United States, SDH is the standard in Europe. If the RCWS system stretches overseas, our overseas carrier will be responsible for ensuring SONET/SDH compatibility [4].

To make this idea concrete with an example, consider our RCWS network consisting of SynOptics ATM switches and the ATDNet consisting of Fore switches. Both networks use their own proprietary routing protocols. Therefore, calls that both originate and terminate within the RCWS network will be correctly routed through the SynOptics switches; similarly, calls entirely within ATDNet will be correctly routed through the Fore switches. However, in the absence of routing standards, a call cannot be routed across the boundary between the two networks. In such a situation, the only way for hosts on the RCWS network to communicate with hosts on ATDNet is to manually route a call through ATDNet using PVCs or PVPs (essentially using ATDNet as an "extension cord").

2.2.1 IISP

An intermediate and partial solution to this problem was IISP, the Interim Interswitch Signaling Protocol [6]. IISP was created as a temporary method of establishing very limited multi-vendor switch interoperability while the comprehensive routing standards were being written. As its name suggests, IISP is strictly a signaling protocol. This means that it can be used by switches *only* to request routing services from other switches, *not* to exchange network topology information; the topology of foreign networks must be manually configured by an administrator. Once the topology of foreign networks has been configured, IISP allows the local network to route calls through those foreign networks, even if the local and foreign networks use different (or proprietary) routing protocols internally. Of course, IISP can only be used in situations where the network topology is sufficiently simple—and changes sufficiently infrequently—that it can be manually configured by the network administrators.

To return to our example, the topology of our RCWS network and ATDNet is simple enough that IISP could be used to facilitate the transparent routing of calls across the network boundary.[2] Thus, IISP would allow our RCWS systems at NIH to place calls to hosts on ATDNet as easily as they place calls to other RCWS systems. Unfortunately, our version of SynOptics CMS (version 1.2) does not support IISP, so this is not possible.

2.2.2 PNNI

While IISP is a useful standard for interim use in small testbed networks, a much more comprehensive protocol is required for large-scale production networks. In March of 1996, the ATM Forum standardized Version 1.0 of PNNI [7], the Private Network–Network Interface, to facilitate both the exchange of routing requests *and* network topology information between switches. This important standard will allow automatic call routing through a heterogeneous ATM internetwork of arbitrary complexity (even on a global scale), through switches from different vendors.

PNNI is a very complex standard, and its implementation has required significant efforts from vendors. Ambiguities in the original PNNI specification resulted in incompatible implementations; initial PNNI interoperability tests between vendors were not entirely successful [8, 9]. The ATM

[2] This simplicity stems from the fact that ATDNet is made up entirely of Fore switches which appear to be a single, flat network.

Forum has been working to resolve the ambiguities in the standard, and plans to release an updated version of the PNNI v1.0 specification in June of 1997.

As successful PNNI implementations become more common, PNNI is sure to replace the existing proprietary routing protocols, eventually leading to seamless routing across virtually all public and private ATM network boundaries. While it may be possible for our RCWS network to survive using only IISP in the short term, PNNI will be a requirement of our network in the future, as RCWS workstations are distributed more widely throughout the emerging nationwide ATM infrastructure.

Unfortunately, our current version of SynOptics CMS (version 1.2) supports neither IISP nor PNNI. Because of the significant effort required to implement PNNI, and Bay Networks' unwillingness to invest any non-trivial effort in further development of the SynOptics LattisCell line of switches, PNNI will never be available for our current switches. Buying new switches at some time in the future seems inevitable.

3 IP over ATM

The purpose of the Internet Protocol (IP) [2] is to create a virtual network with virtual addresses, independent of any particular network hardware or physical transmission standard. IP is responsible for routing datagrams through this virtual network. However, in addition to the hardware-independent portions of the IP standard, an adjunct hardware-specific standard is required for each type of OSI layer 2 (datalink) network on which IP is implemented.

For example, at the majority of sites today, IP runs over Ethernet [10]. Internet RFC standards such as RFC 894 [11] and RFC 1042 [12] specify how IP-over-Ethernet is accomplished. They define how an IP datagram is encapsulated in an Ethernet frame, specify the dynamic resolution of IP addresses to Ethernet hardware addresses using the ARP protocol [13], and other similar issues. Similar standards exist for implementation of IP over FDDI [14], Token Ring [15], and many other datalink networks.

Standards now also exist for using IP over ATM. The same TCP and IP protocols are still used, and in our case are still implemented by the exact same code inside the Solaris kernel, but the IP datagrams that the kernel produces are encoded and transmitted over ATM instead of Ethernet. The purpose of this section is to examine some of the differences among the various standards that exist for accomplishing this.

It is important to reiterate that everything from the IP layer up to the top of the OSI stack remains completely unchanged when switching from an IP-over-Ethernet to an IP-over-ATM model. The same implementation of IP and TCP in the Solaris kernel is still used, the API for accessing TCP remains unchanged, and the applications do not need to be modified. It is only the "glue" used to stick IP onto the layer below it which requires modification.

3.1 Classical IP and ARP over ATM

The first IP-over-ATM standard we will examine, "Classical IP and ARP over ATM," was published by the Internet Engineering Task Force in RFC 1577 [16]. The January 1994 standard, usually referred to as *Classical IP* or *CIP*, specifies a method of using IP over ATM networks. "RFC" documents are all Internet-related standards and information; thus, the standard does not (and should not) define ATM encoding of any protocol other than IP, the Internet Protocol.

Note that the term "Classical IP" is often used to refer to this protocol, and should not be confused with the colloquial term "classical IP" (lowercase c) or "the traditional layer-3 IP protocol." The choice of the term "Classical IP" is explained this way in RFC 1577:

> This memo describes the initial deployment of ATM within "classical" IP networks as a direct replacement for local area networks (Ethernets) and for IP links which interconnect routers, either within or between administrative domains. The "classical" model here refers to the treatment of the ATM host adapter as a networking interface to the IP protocol stack operating in a LAN-based paradigm.

3.1.1 The ARP Server

One of the primary issues surrounding any IP implementation is the method of resolving IP addresses into the hardware addresses of the underlying datalink network. In other words, when a host attempts to send an IP datagram to some destination IP address, it first needs a method of determining the hardware address that corresponds to host owning the desired IP address (or a gateway that is capable of reaching it). IP-over-Ethernet performs this function using ARP, the Address Resolution Protocol [13]. A host wishing to resolve an IP address into an Ethernet hardware address broadcasts an ARP Request to all hosts on the Ethernet; the host that owns the destination IP address then responds to the originating host with an ARP Reply.

Because ATM networks do not allow broadcasts, the standard ARP protocol cannot be used with CIP; RFC 1577 instead defines the concept of an *ARP server*. The ARP server is a host that maintains a canonical database of the ATM to IP address mapping for all hosts in the Logical IP Subnet (LIS). Each host gives its own ATM and IP addresses to the ARP server when it joins the LIS (usually once, when the host boots). Later, when a source host wishes to send IP datagrams to some destination host, the source host queries the ARP server to find the ATM address that corresponds to the desired destination IP address. Once the ARP server returns a reply, the source host creates an SVC to the destination host. This is usually accomplished by using UNI signaling to request an SVC between the source host and the ATM address returned by the ARP server. IP datagrams are then transmitted via that SVC, encapsulated in AAL5 SDUs as specified by RFC 1577. Note that this scheme requires a source host to simultaneously maintain individual SVCs for each host with which it is communicating. Once an SVC has been idle for some administratively configured amount of time, it is disconnected. In this way, the host implements a sort of "dial–on–demand" scheme of establishing individual point-to-point connections with hosts on an as-needed basis.

3.1.2 Scalability and Reliability

CIP was clearly meant to be only an intermediate standard, designed to allow early adopters to use ATM for their installed base of IP applications within small workgroups. The RFC even states explicitly:

> This memo details the treatment of the classical model of IP and ATMARP over ATM. This memo does not preclude the subsequent treatment of ATM networks within the IP framework as ATM becomes globally deployed and interconnected; this will be the subject of future documents.

The most significant factor limiting the scalability of CIP is that all hosts participating in the Logical IP Subnet (LIS) must agree to use a *single* ARP server to manage IP-to-ATM address mappings. The standard does not provide for, and in fact outlaws, the use of a distributed system of ARP servers such as the current Internet-wide DNS system for mapping alphabetic hostnames to numeric IP addresses. A single ARP server will quickly become a performance bottleneck as well as acting as a single point of failure for the entire network. Thus, CIP is appropriate only for workgroups (10's of hosts), and will fail if used enterprise-wide (1,000's of hosts) and certainly cannot be deployed Internet-wide (1,000,000's of hosts).

3.1.3 Classical IP Without an ARP Server

While a full-fledged Classical IP host dynamically creates and destroys point-to-point SVCs to other Classical IP hosts as needed, RFC1577 also allows IP datagrams to flow over manually created PVCs. In this simplified configuration, an administrator creates PVCs in the ATM network between each pair of hosts that could potentially exchange data, then configures the hosts with mappings of IP addresses to the VPI:VCI pairs of those PVCs. Because the hosts are no longer required to dynamically create SVCs, they no longer need the ATM to IP address resolution provided by the ARP server; this makes the ARP server unnecessary. Of course, this configuration is highly undesirable for all but the most trivial configurations because the administrator must maintain $O(n^2)$ PVCs in an LIS with n hosts.[3] However, it is a reasonably convenient method of setting up a point-to-point IP link between a pair of hosts, if one or both hosts are not capable of UNI signaling or communicating with an ARP server.

3.2 LANE

The LANE (LAN Emulation) standard [17] was released by the ATM Forum in January of 1995. LANE is a different, more recent standard (released one year after RFC 1577) which specifies protocols and methods for generically using ATM as a datalink-layer LAN. LANE has two variations: one to emulate Ethernet and one to emulate Token-Ring networks. Similar to CIP, LANE defines

[3] Because a PVC normally only traverses a single switch, the number of individual PVCs is actually $O(n^2 d)$ for an LIS consisting of n hosts on a network of diameter d. Even worse, all of the PVCs must be reconfigured every time the switch topology changes. Suffice it to say, this quickly becomes a logistical nightmare.

standards for encapsulating data that comes from higher up in the OSI stack into AAL5 SDUs. Unlike CIP, LANE provides additional features that more closely emulate traditional LANs. For example, LANE defines a "Broadcast Unknown Server" that receives data from individual clients and redistributes the data to other clients on the emulated LAN, emulating the functionality of an Ethernet broadcast operation.

The intent of LANE is to allow every protocol that has traditionally been implemented on top of Ethernet and Token Ring to be easily implementable over ATM. IP is one of those protocols, so LANE can be used to support IP-over-ATM. Additionally, since LANE is designed to generically replace Ethernet and Token Ring, it has the advantage of simultaneously supporting every other current and future layer 3 network protocol such as Novell's IPX, Apple's AppleTalk and AppleShare, and dozens of others.

Clearly, the motivations for CIP and LANE were different. CIP came from the Internet community, whose interest is promoting IP and thus developed a method for using IP over ATM. The ATM Forum, on the other hand, is interested in promoting ATM and therefore developed a method for using ATM as a LAN, on which all higher-layer legacy protocols can be transparently implemented.

3.2.1 Scalability and Reliability

LANE is not (and does not claim to be) anything other than a *local*-area networking protocol and thus will probably never be used in globally connected networks such as the Internet. However, depending on the implementation, it may be appropriate for enterprise-wide (1,000's of hosts) deployment. Quoting from the standard:

> The LE Service might be implemented in an ATM intermediate system or an end station (e.g., a bridge, router or dedicated work station). Alternatively it may be "part of the ATM network," namely, implemented in switches or other ATM specific devices. A possible implementation might be a single (centralized) LE Service. An alternative implementation could be a distributed one, e.g., where a number of servers operate in parallel and provide the redundancy required for error recovery. The LE Service could also be co-located with one or more LE Clients - potentially saving on hardware costs.
>
> [...]
>
> Any of the LAN emulation service components may be distributed over multiple physical entities or may be collapsed into fewer physical entities, even a single one. This document does not specify the means by which multiple physical entities cooperate to share the function of one LAN emulation service component.

The designers of LANE clearly had scalability in mind. CIP uses a single nondistributed, nonredundant ARP server; it is a single point of failure and may quickly become a bottleneck on a large network. In contrast, the LANE standard allows for distributed systems which are both scalable and more reliable.

As with much of computer science, generality comes at the price of complexity. CIP is a relatively simple protocol because it makes many simplifying assumptions—primarily that IP is the only supported protocol, that IP will only be used in certain "classical" configurations, and that exactly one ARP server will always be responsible for each LIS. LANE is more complex because it must emulate an Ethernet in every regard; for example, it must implement a method for broadcasting data to all stations on the emulated LAN. In addition, LANE is much more scalable, and can be configured much more flexibly, further adding to its complexity. RFC 1577 is a mere 17 pages while the LANE standard is 138.

3.3 What Should We Use?

In my opinion, LANE is the better way of implementing IP over ATM for the RCWS project. Its most attractive feature is the possibility of using a distributed LANE server. This is critical for the RCWS because our plan is to deploy RCWSs into geographically disparate regions—and more importantly, regions which will only have occasional ATM network connectivity to our site. If we install a LANE server in each part of the network that can potentially be isolated from us, every workstation network-wide will always be able to see an operational LANE server.

This exactly mirrors our intended strategy for EVE server development: to deploy a distributed EVE server in each geographic region so that those regions can have local conferences without connectivity to NIH (for example, so St. Louis researchers can use the RCWS even if the ACTS link is down), but have multiple EVE servers automatically elect a leader when they see more than one contending for control on a single network.

Also, if physicians start to use the RCWS as an actual clinical system for real patient treatment planning, it is dangerous to have the entire system hinge on a single host running our single RFC 1577 ARP server. A much safer plan is to have multiple, distributed LANE servers running on multiple hosts—preferably in different physical locations—for maximum resistance to failure.

In summary, LANE seems to be the clearly superior choice when considered independent of the implementation. However, implementation issues are significant.

3.4 Implementation Issues

Unfortunately, we live in a world of practice and not theory. ("In theory, theory and practice are the same. In practice, they're not.") Our choice of protocol must be influenced by the practical realities of which technologies are available.

ATM driver software, including a client or server for CIP or LANE, can only run on a NIC made by the same vendor as the driver; for example, Fore's LANE server cannot run on an ENI NIC. However, any standards-compliant LANE client should be compatible with any standards-compliant LANE server. Therefore, while it is not possible to mix vendors' software and NICs within a single host, it should be possible to use one vendor's NIC/software in one host, and a different vendor's NIC/software in a different host.

One important issue is that either LANE or RFC1577 is not available on some platforms. The Aruba driver suite provided by ENI with their ATM NICs comes bundled with an RFC 1577-compliant CIP client and server, but only a LANE client. ENI sells a LANE server package as a separate product, called Panama. Panama runs under SunOS version 4.1.3_U1 or higher, and Solaris 2.3 or higher. ENI normally sells Panama for $10,000, but NIH has been offered a 50% discount.

Fore's ATM NICs for UNIX workstations come bundled with both a client and server, for both CIP and LANE. Fore's CIP client and server, and LANE client, all work under SunOS and Solaris. However, the workstation version of their LANE server currently runs only under SunOS; they plan to release a Solaris version in 2Q97.

Fore also sells another version of their LANE server that runs on the CPU of their switches, as part of ForeThought. The switch-embedded LANE server must be purchased as a ForeThought add-on product. This option is currently not useful for us because not all of the RCWS workstations are directly connected to Fore switches at this time.

It is also possible that the latest version of SynOptics CMS has a built-in LANE server. However, Bay Networks has been consistently unable to provide me with any information on either their LattisCell products or supporting software such as CMS. In my opinion, we should not depend on them for support nor should we invest any more money into an unsupported product line unless absolutely necessary. Nevertheless, it may be useful to find out if CMS 2.x includes a LANE server, as it may turn out that we will be required to upgrade CMS for some other reason.

One problem is that the Mac ATM NICs sold by Fore only support LANE. (ENI does not sell Mac ATM NICs at all.) Currently, the RCWS workstations use CIP simply because we do not have a LANE server; therefore, they will need to be switched from CIP to LANE in order to allow the Mac to interoperate with them. On the other hand, the SNMP-manageable MMXs only support the simplified model of CIP that does not use an ARP server. This seemingly intractable problem can be resolved by running both a CIP and LANE client simultaneously on the RCWS NIC's single physical interface. Fore and ENI both claim their drivers are capable of this, which was confirmed by colleagues at the Department of Computer Science at the Washington University in St. Louis.

It should be noted well that although the LANE *standard* allows for distributed and redundant LANE servers, there is no guarantee that any particular vendor has *implemented* such a server. Before we buy any LANE server, we should investigate its scalability and redundancy features.

4 TCP/IP Performance

Although the protocols described above are enough to minimally implement TCP/IP over ATM, there are significant performance problems that arise when using ATM to carry such traffic. While these issues will not prevent TCP from operating over ATM, they do result in dramatic losses in throughput and efficiency. Several of these problems will be considered in this section.

4.1 TCP Window Scaling

The biggest problem seen in TCP-over-ATM implementations is a limitation of the TCP protocol itself. This problem is not specific to ATM, but manifests itself whenever TCP is used over a network with a large bandwidth-delay product. Such a network is known as a Long Fat Network, or *LFN* [18].

TCP is a *sliding-window protocol*, so the sending and receiving hosts must first negotiate a "window size," specified as a number of bytes, before transmitting any data. Subsequently, the sender cannot send more than a single window of data before receiving an acknowledgment from the receiver. An acknowledgment cannot possibly be received by the sender before a time equal to twice the network latency from the sending to the receiving host—also called the Round Trip Time (RTT). Therefore, the theoretical maximum TCP throughput is achieved when one window is sent every RTT. The original TCP standard assigns 16 bits of the header to represent the advertised TCP window size, limiting the window to 64 Kbytes. This limits TCP to sending 64 Kbytes per RTT.

The 64K window limit is usually not a problem on local-area networks with very small latencies ($\leq$1 ms), but quickly becomes the performance-limiting factor as the latencies become larger and the desired network bandwidth increases. On a large LAN with an RTT of 5 ms, sending a 64K window every RTT yields a maximum TCP throughput of 102.4 Mb/sec—still relatively good, but already less than the theoretical 135 Mb/sec available to an OC3 ATM network.[4] Even worse, a cross-country path with a typical RTT of $\approx$70 ms allows only 7.31 Mb/sec [19]. A particularly pathological case is using TCP over a satellite such as the ACTS where the RTT is 500 ms, making the ceiling on TCP throughput just over 1 Mb/sec.

Assuming Einstein was right, the latency of ground-based communication will never significantly improve due to the constraint of the speed of light [20]; therefore, increasing the size of the TCP window is the only way to increase TCP throughput. A solution developed by the IETF is a new TCP option called "TCP Window Scaling" that allows TCP to advertise windows much larger than 64K. RFC 1323 [21] defines TCP Window Scaling and several other improvements to TCP that are required to operate over LFNs. The modifications are also designed to maintain compatibility with earlier TCP implementations.

Vendors have been slow to adopt the changes proposed in RFC 1323. SunSoft provided an unsupported alpha-test implementation of RFC 1323-complaint TCP for Solaris 2.4 in November of 1995. There was no general release of this implementation for Solaris 2.5 or 2.5.1, although it can be ordered via a Sun Consulting special. SunSoft has promised that it will be part of Solaris 2.6, slated to be released in 3Q97.

It should be noted, however, that since this limitation is part of the TCP protocol itself, another option to achieve better performance is simply to use a different protocol. One possibility that fits the RCWS application very well is *Commedia*, a distributed message-passing system written by Yair Amir of Johns Hopkins University. Commedia is implemented using UDP/IP, with its own algorithms for ensuring the end-to-end data integrity usually provided by TCP. However, unlike

[4] The base transmission rate of a SONET network is 155.52 Mb/sec. After SONET framing, 149.76 Mb/sec is available to the ATM layer. ATM cell header overhead leaves 135.631 Mb/sec available at the adaptation layer [4].

TCP, Commedia was originally designed to work well in LFNs, and does not have limitations like TCP's window size.

4.2 Selective Acknowledgment

Current implementations of TCP use a cumulative acknowledgment scheme. This means that a TCP receiver can only acknowledge the longest contiguous prefix of the data stream it has received from the sender, even if it has received additional noncontiguous data. For example, consider a TCP sender sending a data stream consisting of n bytes. If a single TCP packet consisting of bytes $i \dots j$ is lost, the receiver will successfully receive bytes $1 \dots i-1$ and $j+1 \dots n$; however, the cumulative acknowledgment scheme of TCP dictates that the TCP receiver may only report that bytes $1 \dots i-1$ have been received. The TCP sender will then retransmit bytes $i \dots n$, even though the only bytes missing are $i \dots j$. The bandwidth used to retransmit bytes $j+1 \dots n$ is entirely wasted. On LFNs, this effect causes a particularly catastrophic loss of efficiency [18].

Selective acknowledgment (SACK) allows the TCP receiver to report non-sequential data that it has received, preventing wasted bandwidth caused by unnecessarily retransmitted data. RFC 1072 [22] first proposed a SACK extension to TCP, but the feature was omitted from RFC 1323 [21] because the authors felt several technical problems needed to be resolved before including SACK in TCP [18]. Vendor support of SACK has been extremely slow despite studies showing that SACK can drastically reduce unnecessary retransmissions [23]. As of this writing, SunSoft has no plans to support SACK under Solaris.

4.3 Cell Loss from a Datagram

All datalink networks have a Maximum Transfer Unit (MTU) which defines the largest unit of data that can be sent at once over the network. RFC 1626 [24] defines the MTU of IP over ATM to be 9180 bytes, so a typical IP datagram will be fragmented by AAL5 into at least 192 ATM cells. If a single cell of a datagram is dropped (for example, due to congestion), the datagram is corrupted because AAL5 has no provision for retransmission of lost cells. Therefore, any bandwidth used to transmit the remaining cells of the already-corrupted datagram is entirely wasted.

This problem can only be solved by modifying the cell-transmission algorithm of the ATM switches; no fix is possible through modification of the TCP or IP code on the sending host. Two algorithms that attempt to alleviate this problem proposed by Romanow and Floyd [25] are Partial Packet Discard (PPD) and Early Packet Discard (EPD). With PPD, once a switch drops an AAL5 cell, all subsequent AAL5 cells on the same VC are also dropped until the switch sees the end of the AAL packet. Although PPD helps throughput, all cells transmitted *before* the dropped cell still waste bandwidth. With EPD, the switch drops an entire AAL5 SDU (i.e., every cell in an IP datagram) after its port buffers exceed some predefined "safe" threshold but before they overflow. This causes an IP datagram to be dropped in its entirety, preventing lost bandwidth from the transmission of "dead cells."

Romanow and Floyd's proposal includes simulations of TCP throughput with and without PPD and EPD; their conclusion is that EPD has the effect of making TCP throughput over ATM close to optimal. However, there is a major flaw in their analysis: the simulator assumes that the transmitting and receiving hosts are both connected to the same switch; in my opinion, EPD and PPD methods will break down in a network of switches.

Consider two hosts attached to two different switches on an ATM network, and that the best route from Host A to Host B traverses n switches, all of which are using EPD. Suppose that some switch i along the path from A to B experiences congestion, and begins to drop cells from the VC connecting A to B according to the EPD algorithm. "Dead cells" will now be transmitted from host A to switch i, only to be dropped by switch i; the bandwidth used to transmit the cells from A to i is entirely wasted.

Unfortunately there is not yet a good solution for the problem of losing cells within an AAL5 SDU, or the essentially equivalent problem with the IP protocol of losing a single fragment of an IP datagram fragmented at the IP layer. Both problems are areas of ongoing research [26].

5 Acknowledgments

Personal communication with a large number of people was of invaluable help in the preparation of this report. Richard Verjinksi of Fore Federal Systems answered a large number of questions about Fore ATM products; Daniel Proch, Paul Reisinger, and Jose Vela of Fore Systems provided additional technical information. James Johnson and Larry Phillips of Efficient Networks, Inc. helped a great deal in understanding the operation of ENI's NIC drivers and ATM software. John DeHart of the Washington University in St. Louis' Department of Computer Science and Scott Chaney of STS Technologies, Inc. illuminated how the MMXs perform in-band IP over ATM. Excellent information about TCP window scaling and vendors' support of RFC 1323 was provided by Curtis Villamizar from ANSNET (Advanced Network and Services, Inc.), Jamshid Mahdavi of the Pittsburgh Supercomputing Center, and Richard Stevens of NOAO (National Optical Astronomy Observatories). A number of people from Sun Microsystems and SunSoft answered questions about TCP window scaling support in Solaris, including Jerry Chu and Elizabeth Konesky.

References

[1] K.M. Kempner, D. Chow, P. Choyke, J.R. Cox, J.E. Elson, C.A. Johnson, P. Okunieff, H. Ostrow, J.C. Pfeifer, and R.L. Martino. The development of an ATM–based Radiology Consultation WorkStation for radiotherapy treatment planning. In Yongmin Kim, editor, **Image Display:** *Medical Imaging 1997 Conference Proceedings*, volume 3031, pages 500–511. Society of Photo-Optical Instrumentation Engineers, 1997.

[2] Douglas E. Comer. *Internetworking with TCP/IP—Principles, Protocols, and Architecture.* Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1988.

[3] ATM Forum. Internet home page, `http://www.atmforum.com`.

[4] D. Ginsburg. *ATM Solutions for Enterprise Internetworking.* Addison–Wesley, 1996.

[5] ATM Forum Technical Committee. ATM User–Network Interface Specification Version 3.0. Specification af-uni-0010.001, ATM Forum, Mountain View, CA, September 1993. Available on the Internet at ftp://ftp.atmforum.com/pub/approved-specs/af-uni-0010.001.ps.

[6] ATM Forum Technical Committee. Interim Inter–Switch Signaling Protocol. Specification af-pnni-0026.000, ATM Forum, Mountain View, CA, December 1994. Available on the Internet at ftp://ftp.atmforum.com/pub/approved-specs/af-pnni-0026.000.ps.

[7] ATM Forum Technical Committee. P-NNI Version 1.0. Specification af-pnni-0055.000, ATM Forum, Mountain View, CA, March 1996. Available on the Internet at ftp://-ftp.atmforum.com/pub/approved-specs/af-pnni-0055.000.ps.

[8] J. Caruso. PNNI put to the test—MCNC checks vendor compliance with ATM Forum spec. *Communications Week*, December 23, 1996. Available on the Internet at http://-www.techweb.com/se/directlink.cgi?CWK19961223S0040.

[9] M. Cooney. ATM put to compatability test. *Network World*, January 13, 1997.

[10] IEEE Project802. Local area network standards: CSMA/CD access method and physical layer specifications. *IEEE Standard 802.3*, 1985.

[11] C. Hornig. RFC 894: Standard for the transmission of IP datagrams over Ethernet networks, April 1984. Available on the Internet at ftp://ftp.internic.net/rfc/rfc894.txt.

[12] J. Postel and J. Reynolds. RFC 1042: Standard for the transmission of IP datagrams over IEEE 802 networks, February 1988. Obsoletes RFC0948 [27]. Available on the Internet at ftp://ftp.internic.net/rfc/rfc1042.txt.

[13] D. Plummer. RFC 826: Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit ethernet address for transmission on Ethernet hardware, November 1982. Available on the Internet at ftp://ftp.internic.net/rfc/rfc826.txt.

[14] D. Katz. RFC 1390: Transmission of IP and ARP over FDDI networks, January 1993. Obsoletes RFC1188 [28]. Available on the Internet at ftp://ftp.internic.net/-rfc/rfc1390.txt.

[15] T. Pusateri. RFC 1469: IP multicast over token-ring local area networks, June 1993. Available on the Internet at ftp://ftp.internic.net/rfc/rfc1469.txt.

[16] M. Laubach. RFC 1577: Classical IP and ARP over ATM, January 1994. Available on the Internet at ftp://ftp.internic.net/rfc/rfc1577.txt.

[17] ATM Forum Technical Committee. LAN Emulation over ATM Version 1.0. Specification af-lane-0021.000, ATM Forum, Mountain View, CA, January 1995. Available on the Internet at ftp://ftp.atmforum.com/pub/approved-specs/af-lane-0021.000.ps.

[18] W. R. Stevens. *TCP/IP Illustrated—The Protocols*. Addison–Wesley, Reading, MA, 1994.

[19] C. Villamizar and C. Song. High Performance TCP in ANSNET. Technical report, Advanced Network and Services, Inc., September 1994. Available on the Internet at ftp://ftp.ans.net/pub/papers/tcp-performance.ps.

[20] A. Einsten. On the Electrodynamics of Moving Bodies. *Annalen der Physik*, 1905.

[21] D. Borman, R. Braden, and V. Jacobson. RFC 1323: TCP extensions for high performance, May 1992. Obsoletes RFC1185 [29]. Available on the Internet at ftp://ftp.internic.net/-rfc/rfc1323.txt.

[22] R. Braden and V. Jacobson. RFC 1072: TCP extensions for long-delay paths, October 1988. Available on the Internet at ftp://ftp.internic.net/rfc/rfc1072.txt.

[23] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. Technical report, Lawrence Berkeley National Laboratory, 1996. Available on the Internet at ftp://ftp.ee.lbl.gov/papers/sacks_v3.ps.Z.

[24] R. Atkinson. RFC 1626: Default IP MTU for use over ATM AAL5, May 1994. Available on the Internet at ftp://ftp.internic.net/rfc/rfc1626.txt.

[25] A. Romanow and S. Floyd. Dynamics of TCP traffic over ATM networks. *IEEE Journal on Selected Areas in Communications*, May 1995. Available on the Internet at ftp://ftp.ee.lbl.gov/papers/tcp_atm.ps.

[26] R. Jain. Congestion control and traffic management in ATM networks: Recent advances and a survey. In *Computer Networks and ISDN Systems*, February 1995.

[27] I. Winston. RFC 948: Two methods for the transmission of IP datagrams over IEEE 802.3 networks, June 1985. Obsoleted by RFC1042 [12]. Available on the Internet at ftp://ftp.internic.net/rfc/rfc0948.txt.

[28] D. Katz. RFC 1188: A proposed standard for the transmission of IP datagrams over FDDI networks, October 1990. Obsoleted by RFC1390 [14]. Obsoletes RFC1103 [30]. Available on the Internet at ftp://ftp.internic.net/rfc/rfc1188.txt.

[29] R. Braden, V. Jacobson, and L. Zhang. RFC 1185: TCP extension for high-speed paths, October 1990. Obsoleted by RFC1323 [21]. Available on the Internet at ftp://-ftp.internic.net/rfc/rfc1185.txt.

[30] D. Katz. RFC 1103: Proposed standard for the transmission of IP datagrams over FDDI networks, June 1989. Obsoleted by RFC1188 [28]. Available on the Internet at ftp://ftp.internic.net/rfc/rfc1103.txt.