Network Architecture

3.2 Network Architecture

Takayuki Kushida
IBM Research, Tokyo Research Laboratory

We made Global Digital Museum (GDM) prototype on the internet, which is becoming most public global network. GDM needs global search and interactive multimedia content editing on the network, which will become useful by high performance and serviceability of the network.

Network architecture is always a hot topic of network research, but the application of a new technology in the real world usually takes place long after it is created. The recent explosive growth of the Internet has lead to several major issues regarding network architecture. The major issue related to the current architecture of the Internet is the lack of QoS. Engineers and researchers always need to consider the issue to avoid performance degradation when an application is developed. This paper explores current issues related to the Internet, a cache technology to improve the performance of applications as an intermediate solution, QoS technology for the full guaranteed service as a near–term technology, and an active network architecture as a future technology. This paper shows how the issue can be resolved with those architectures by recent studies. This paper is structured as follows: In the next section, we describe the current architecture of the Internet. We then introduce a technology for improving the performance of the Internet. After that, we discuss QoS technology for the Internet, and finally we describe an active network architecture. This survey provides a basic outline of Internet–related networking research, allowing readers to grasp the overall trend of networks.

3.2.1 Packet–Switching Network

This section describes an architecture that uses what is called packet–switching technology. The basic mechanism of packet–switching technology is quite simple but powerful: by reducing the installation costs, it allows a network to be expanded, and by reducing the maintenance costs, it allows it to be scaled up to form a huge interconnected network (1).

In the architecture of a packet–switched network, when a host on the network sends data to another host on the network, the application data is divided into small chunks of discrete data called packets or datagrams, because a network component can easily handle these data chunks in network programs. This work is a normalization procedure of the application data for the network. Each packet has its own header so that it can be transferred autonomously at intermediate hosts. This packet header created at a source host contains self–sufficient control information for handling at intermediate hosts and for processing at a destination host.

A host is identified by a network address which is uniquely specified in the network. There are two kinds of a network address: a source and destination address. Although there are a large number of hosts connected to the network, the destination host can obtain information on which host sent it a packet by looking up the source address. In general, the network address is put into a packet header identifying the source host and the destination host; this header is also used for routing on intermediate hosts. The network address is checked at intermediate hosts so that the packet can be delivered to the next intermediate host on the same sub–network.

In general, a network consists of a source host, one or more intermediate hosts, a destination host, and unreliable media. This underlying media is unreliable for the transfer of packets because of bit errors, burst error and other failure of the media.

The packet is transferred from the source host to the destination host via several intermediate hosts. Each intermediate host, usually called a router, looks up the entry in the routing table which consists of network addresses and addresses of the next host that can handle the packet for the destination address. The router matches the destination address in the packet header, and delivers the packet to the next matched host on the same sub–network as soon as possible. This procedure is only packet delivery without any other additional work in the router. This is the best–effort service of the packet–switching technology.

We explain the basic mechanism of a packet–switched network in detail. To imagine how a packet–switched network works, we assume that the underlying computer network uses the Internet which is a set of networks and uses TCP/IP (Transmission Control Protocol/Internet Protocol) protocol suite, because many service providers offer it, and the operating systems of workstations and personal computers support this protocol in their basic configurations. The following description is based on the Internet; other protocol suites for packet–switched networks follow the same procedure.

Currently, the IP version 4 protocol (IPv4) is used for the Internet, and the IP version 6 protocol (IPv6) has been defined. The installation of the IP version 6 protocol (IPv6) in experimental networks has begun. Although the IPv6 has been defined and standardized, the IPv4 is still used at an operational level. The migration of the Internet from IPv4 to IPv6 takes a lot of time because of the number of installations of IPv4 hosts.

The IPv4 has 32–bit source and destination addresses, which are uniquely identified on the Internet. These IP addresses are assigned to local organizations by a central authority. The assignment needs to be carefully managed to avoid conflicts of addresses and to reduce address entries of the routing table by using a network mask, which is the method to aggregate the routing entries.

The router checks entries in the routing table, which is a large database for IP address entries. One routing technique is to use a default route: all packets that are not matched by any of the routing entries in the backbone router follow this route. Another technique is a network mask to hide part of bits in the address, so that these routing entries for the same route are aggregated. These techniques can decrease the number of routing entries and thus improve searching performance.

After receiving a packet, each host processes it and sends the resulting information to other hosts. Under normal conditions, the destination host processes the received packet along with the protocol definition, and transfers it to applications.

A packet consists of its header and container. Since the packet header is used in the protocol process for controlling packets in the network, it is therefore part of the architecture. In general, a host connected to the network exchanges a large number of packets according to a exact procedure using a time line. This procedure and the control information in the packet header are defined as a protocol.

The protocol uses the concept of layers to divide up the functions for processing the packet. The protocol is divided into several software functions, and the software sometimes resides in the kernel of the Operating System for performance reasons. The layer processing in the source host is the same as that in the destination host to synchronize the packet processing. For example, the transport layer on the source host will synchronize one on the destination host.

The layers of the protocol facilitate understanding of network concept. Between the layers there are definitions, such as "socket" APIs, of how to pass data from one layer to another. Both the productivity of the protocol development and the ability of the maintenance for the protocol enhancement are increased by splitting the protocol into layers.

The name TCP/IP comes from two famous names in the Internet protocol stack: TCP is the Transmission Control Protocol and IP is the Internet Protocol.

The TCP/IP protocol suite is a set of incremental developed protocols, and many new protocols are now being developed as a part of the TCP/IP suite. Many applications have been developed on top of TCP/IP. The main reasons for the widespread use of TCP/IP networks are that the concept of the protocol is simple, and there are several implementations of the protocol instead of a set of complete standard documents. The architecture and implementation of the protocol are well documented, and all related documents can be easily retrieved from archives. Since there are implementations of the protocol suite with the complete source code, it is easy to understand the protocol suite and to identify issues related to improving the protocol.

The software component of the IP resides on the network layer responsible for the interconnection of the network. This protocol is responsible for the transparency of the entire network. This is a single network–wide architecture, and it is powerful for both local area networks and wide area networks. The packet header of IP has the source address and the destination address. The destination address is checked to deliver the packet to the next intermediate host, whose entry is kept in the routing table. The intermediate host directly transfers the packet to the next intermediate host along with underlying media. The intermediate host has only a function forwarding the packet so that the network can provide the best–effort service. No further processes are involved in IPv4.

The TCP is a transport protocol that guarantees a continuous, reliable, and streaming path between end–to–end hosts. Therefore, it has a function for recovering lost, corrupted, and incorrectly ordered packets on the network. Those lost and unacceptable packets are recovered only by the retransmission of the packet in TCP.

The TCP has a congestion control mechanism for probing the status of the network, and maintains high throughput at a steady rate. A number of studies have tried to improve TCP by using various types of networking environment such as slow speed, high speed, symmetry and asymmetry path. Since there is no evidence that the TCP behavior is optimal, improvement of TCP is a hot topic in network research. The TCP has the source and the destination ports in its header. These port numbers are directly mapped to application programs which use the TCP session at the end–to–end path. There are well–known port numbers which were reserved for the Internet server applications of TCP and UDP.

3.2.2 Issues related to Network Applications

This section describes issues related to network applications. While the expansion of the Internet offers many advantages, it also raises several issues concerning performance degradation. These issues need to be resolved in order to improve the Internet.

Packet loss is a common occurrence on the Internet. Packets on the Internet usually traverse from the source host to the destination host in several hops. The intermediate path transferred packets may suffer from traffic congestion or there may be no available buffer in the intermediate host; in such cases, incoming packets have to be dropped. As a result of this situation, there could be end–to–end packet loss. This causes serious performance degradation of the TCP session. In a study of end–to–end behavior on the Internet, approximately 2% to 10% of packets on the Internet were lost (2). The results of an experiment indicated that the packet loss was likely caused by either wrong implementation or configuration of routers, and by errors in router software. The probability of other causes such as bit error and traffic congestion is quite low.

One of the performance parameters for the Internet is packet delay. The definitions of the starting time and the ending time of a packet transfer are as follows: The starting time is the time at which a packet is sent from the source host. The ending time is the time at which it is received at the destination host. The difference between the starting time and the ending time is the transit delay of the transferred packet. The round–trip time is defined as the time needed for the packet to make a round trip between the source host and the destination host.

The underlying media between hosts is responsible for the trip delay, and it depends on what media it is used for. The router has a temporary buffer for storing transferred packets before they are sent to the next host. When a packet can be sent to the next host at the router, it is delivered to the next router using underlying media. The length of time the packet is stored at the router depends on how congested the underlying media to the next host is. The temporary storing time at each host and the packet transferring time are accumulated into a single value, which is the packet delay at end–to–end path. In general, there is variance of the packet delay because of the congestion, even when it is the same route between the source and destination. This delay causes an uncertain probability of receiving packets at the destination host.

In a study of the results from an experiment on the packet delay between source and destination hosts (2), the distribution of the round–trip time showed a variability that cannot be explained on the basis of cross traffic alone. It is assumed that there were numerous duplications and some reordering of packets. These phenomena occur even under light traffic conditions. The experiment shows that the Internet has characteristics of failure for network equipment.

3.2.3 Internet Performance

An approach of the statistical analysis is effective for determining the characteristics of the Internet. Analysis of the experiment results showed that the characteristics of delay on the Internet are sometimes similar, and that Internet delay is 'bursty' across multiple time scales (5). It has also been proved that the experimental results of the packet round–trip delay show Long–Range Dependence (LRD) behavior: they are either similar or asymptotically similar (4). The process for measuring the round–trip delay of packets takes samples at a certain sampling intervals. The probability distribution of round–trip delays decays more slowly than an exactly exponential line.

In a study of the Internet simulation (3), two key strategies were discussed: searching for invariants and judiciously exploring the simulation parameter space. This study describes the VINT project, which is a joint effort by USC/ISI, Xerox PARC, LBNL, and UCB to build a multi–protocol simulator.

One of the project goals is to build a multi–protocol simulator that implements unicast and multicast routing algorithms, transport and session protocols, reservations and integrated services, and application–level protocols.

The end–to–end performance should be considered during the design of an Internet application. The development of the application for the Internet requires the different assumptions from that of applications for intranets, which are connected by high–speed LANs for performance reasons. On the Internet, there is performance degradation, with the loss and delay of packets between end–to–end hosts. In addition, no common rule nor assumption regarding performance degradation on the Internet was developed. Therefore, studying the Internet characteristics is still an important topic of network research. We cannot estimate exact values for the performance degradation of the Internet. For application development, we need to make a basic assumption regarding degradation, and to optimize the behavior of an application for the conditions prevailing on the Internet. This optimization will be an adaptive approach for the Internet environment.

The working group IPPM in IETF published a document entitled Request For Comments (RFC), which is the de–facto standard (6). It describes a framework for the performance metrics on the Internet. The standardization work of IPPM is aimed at creating an official guideline to metrics for Internet performance, so that a common metric can be created. For example, if we can use the same parameter for different networks, we can compare their characteristics in identical conditions. In this way, it is possible to reach a common measure of Internet performance.

3.2.4 Application View Point of the Internet

Internet application programs can use the reliable streaming service with TCP and the unreliable datagram service with UDP.

Streaming data transfer is based on the TCP connection. If TCP retransmits a lost packet, the end–to–end transfer time is longer than in the normal state, in which does not occur the retransmission. Retransmission of the packet causes the performance degradation. If there is congestion on the end–to–end path, the application program is also unexpectedly delayed. There are several technologies for improving the performance of the application or hiding these disadvantages for end users.

On the other hand, if the unreliable datagram service based on UDP is used for the application, the application program always takes care of the status of the protocol. If the reliable transmission service is used for unreliable packets, a reliable protocol similar to TCP has to be implemented on the unreliable packet service.

The advantage of the unreliable packet service is that users can handle the protocol by themselves. For example, if one needs fast information delivery without any guarantee, the performance of the unreliable packet transmission is superior to that of the streaming data service with congestion control and reliable transmission because of the number of transferred packets.

We have to consider the degradation of the actual network performance that results from packet losses and packet delays. The length of a delay ranges from several milliseconds to several seconds, depending on how congested the Internet actually is whether the software or the hardware is wrongly configured.

Because of the lack of a mechanism for ensuring the delivery of packets on intermediate hosts, there is no guaranteed QoS on the Internet. The advantage of having no QoS mechanism is that router equipments are inexpensive because they are required only to support best–effort delivery on the network. The disadvantage is that there is no leased–line service on the Internet. This means that there is no guarantee at all for multimedia applications such as voice and video transfer. In addition, we can not use critical applications on the Internet because of the lack of guarantee.

As a result of no QoS service, each application has to handle its own guaranteed service. Since a part of responsibility of network service is in the application program, an incorrect configuration or is understanding during the development of an application may cause serious performance degradation and unexpected behavior in the transmission of the data.

There are several issues as regards the current Internet architecture, and the performance of the end–to–end path should be improved. Part of the current architecture of the Internet should be changed, or else the architecture of packet–switched networks should be changed entirely.

3.2.5 Performance improvement

One well–known technology for improving the performance of the Internet is caching of content data. In caching technology, recently accessed data in remote hosts is stored in local memory or a local file. If it is accessed again, the locally stored data is retrieved instead of accessing the remote hosts. If data is often accessed from an application, it might be stored for a long time. A basic assumption of the caching is that the processing time for retrieving the local data is shorter by a sufficient margin than that retrieving the remote data. In the results of this performance advance, retrieving the local data is improved for the response time. The main contribution of the shorter response time is the network transfer time and the time taken to access the local disk.

In this section, we explore a caching technology of the Internet to improve the performance as an intermediate solution. In the caching technology, the characteristics of the application behavior are critical to improving the performance, because the efficiency of the caching and retrieving methods usually depends on making an appropriate decision as regards the characteristics of the network traffic. On the Internet, the World Wide Web (WWW) is a notable application and there are many caching studies for the WWW.

In general, the performance of a distributed system is usually degraded between servers because of a failure of the underlying media. Since the WWW is a kind of a loosely coupled distributed system, it is subject to performance degradation between a client and a server. A local cache and a cache in a proxy server are needed to improve the performance in retrieving contents from servers.

The Internet Cache Protocol (ICP), which shares the contents of the Web among WWW caches, was defined to reduce the traffic (14). The purpose is to improve the performance of WWW access from WWW caches. The protocol is based on the UDP, which is unreliable and contains a small number of messages. The host using ICP sends the packet of this message to query the neighbors about the availability of contents. This is the de–facto standard, and the simplest method of improving the performance. There are several other methods for improving the share of caches.

A new method of improving ICP, called “Summary Cache” has been proposed (13). In the summary caching, each proxy server keeps a summary of URLs and checks it before sending a query directly to WWW servers. Comparative results for the performance of ICP with and without summary cache were given in the study. They show that it reduces the bandwidth consumption by over 50%, and eliminates between 30% to 95% of the CPU overhead, while at the same time maintaining almost the same hit ratio as ICP.

Prefetching is a mechanism for improving the performance of the WWW. The performance of the prefetching for the WWW was evaluated by carrying out a simulation study (7). The study showed that the latency of the content retrieval is reduced at the cost of a similar increase in the network traffic. It concludes that prefetching might be worthwhile when the increase in bandwidth for the prefetching does not degrade the service. The results shows that prefetching is particularly effective with non–shared serial lines such as those used in modem communication, or high–bandwidth, high–latency links such as those used in satellite communication.

In another study, prefetching for Web servers adopted both a prediction algorithm and a threshold algorithm (15). The prediction algorithm is used to obtain the access probability, and is defined as the conditional probability: There are two files, file A and file B. File A will be requested by a user viewing file B. This correlation function determines the request–view relation between file A and file B. The threshold algorithm measures of both the delay cost and the system resource cost. The costs for the system resource include the cost of processing the packets at the end hosts and that of transmitting them from the source to the destination. The simulation results show that using only the access probabilities at the client site yields successful prediction rates of around 70% even when the threshold is zero. In this study, an application program was developed together with the prediction algorithm.

A performance comparison of caching and prefetching was presented along with the results of the simulation (16). It showed that local proxy caching could reduce latency by at best 26%, prefetching could reduce latency by at best 57%, and a combined caching and prefetching proxy could provide at best a 60% latency reduction. The study concluded that caching offers moderate assistance in reducing latency, and that prefetching can offer more than twice the improvement, but is still limited in its ability to reduce latency.

If it is assumed that there will be future requests from end users, prefetching is an effective approach improving the response time of applications on the network. The prefetch method for retrieving the contents from servers basically increases the burst factor of individual sources, but it enabled Crovella and Barford to prove that the network performance for prefetch could be improved by using a simple transport rate control mechanism (9). They showed that even when prefetching adds no extra traffic to the network, the prefetching has a serious impact on the performance. The prefetching application can modify its behavior from a distinctly ON/OFF entity to one whose data transfer rate changes less abruptly, while still delivering all data in advance of the user's actual requests.

Measurement of the metric is another important subject for the performance of the Internet. Because the nearest host can be determined by this metric. How does an application program find the nearest host on the Internet? Some studies of Internet delay have shown that the characteristics of delay variation have a similar property. A simple definition of a neighbor on the Internet should be provided. The definition would be that a neighbor is a host with little wide bandwidth. Some studies have been made of the performance metrics of the Internet.

Because of the low accuracy of performance evaluation based on individual hosts, a system was studied in which the results were shared among client hosts on local area networks, and it is called SPAND (Sharing Passive Network Performance Discovery) (10). The purpose of the study was to provide a unified repository of actual end–to–end performance information for applications. SPAND provides a new service for measuring performance on the Internet.

Measuring the distance between two hosts on the Internet is a difficult task because there is no common definition for the parameter of the distance, and performance values change with time. A framework for defining distance measurements on the Internet in a project called IDMaps (11). In IDMaps, there are two architectures: Hop–by–Hop (HbH) and End–to–End (E2E). The goals of IDMaps are to provide distance metrics and accurate timely distance information, and to measure the total magnitude, total cost of that information.

A study of ways to improve the performance of the WWW describes an enhancement of HTTP, that adds the piggybacking protocol to HTTP (12). The piggybacking protocol is consistent with HTTP 1.1 and has transactions with a small amount of data in the per–server state. The method was proved statistically by using a log of WWW transactions.

3.2.6 Quality of Service

QoS is a topic of long–term network research, focusing on the Internet service. With the explosive growth in the population of the Internet users and increasing the demand of the Internet use for critical applications, the QoS has become a main focus for the next–generation of the Internet. Internet service providers (ISPs) provided QoS to their users. Many application services are required to support bandwidth reservation between end–to–end applications, and it can be used as a basis for leased–line services.

This section describes several research and development efforts to guarantee QoS on the Internet. Although many critical issues are involved in supporting QoS on the Internet, it has to be developed and made operational in the near term.

One simple question regarding priority is as follows: Is a service priority useful in a network? For an answer, the performance evaluation of a simulation for prioritized service is studied in (8). No definitive answer to the question is given, but a new approach is proposed for making a definitive evaluation of the network design. The results of the study showed that a single level of service degrades the performance to utilization levels of below 50%.

The high–speed Internet backbone vBNS uses a full–mesh Unspecified Bit Rate (UBR) in the ATM network. As a extension of vBNS, a QoS network was proposed with a reserved–bandwidth service in (20). In this extension, a Switched Virtual Connection (SVC) is set up through a Virtual Path (VP) with Variable Bit Rate (VBR). The study pointed out three issues related to the QoS networks: multicast, overall performance, and usage policy.

One IETF working group, "Diffserv," is focused on defining a differential service on the Internet, and provides an QoS architecture and a framework of QoS on the Internet. QoS on the Internet provides scalability of services by aggregating the classification of traffic. The architecture consists of a number of functional elements that were implemented in network hosts, including a small set of per–hop forwarding behaviors, packet classification functions, and traffic–conditioning functions including metering, marking, shaping, and policing (17). A Per–Hop Behavior (PHB) is defined to permit a reasonably granular means of allocating buffer and bandwidth resources at each host among competing traffic streams. The Differentiated Service (DS) field in the QoS is used to mark packets to select a per–hop behavior. A complete description of the DS field is given in RFC2474 (18). The architecture provides only one direction of traffic flow, which means that the traffic is asymmetric. Symmetric traffic is still being researched. The architecture of Diffserv is simple: traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different behavior aggregates.

The Internet2 project is focused on providing a full QoS network for both the research and education communities (19). The technology of Internet2 is based on the standard work by the IETF Diffserv WG. The requirements of QoS on the Internet are as follows: it should allow the use of advanced applications, it should be scalable, administrable, measurable (admit multiple, interoperable implementations of both individual pieces and sets of equipment and network clouds), it should have support from operating systems and middleware, and it is incrementally deployable starting in 1998.

In the Internet2 project, four different services are offered to end–hosts: premium service, assured service, class of service, and default service. The premium service is like a leased line that guarantees the peak bandwidth with low queuing delay, loss, and jitter. It is appropriate for the most intolerant locations and may appeal to an even broader community of providers, developers, and users on account of its easily understood service contract.

The assured service offers a lightly loaded network that is similar to a network with a controlled load, and is appropriate for tolerant and adaptive applications. The user's contract for a specific service profile is guaranteed so that in–profile traffic goes through on the "lightly loaded network" even if the network is congested. The class of service is relative and precedence–based. It is better than the best–effort service at meeting coarse user and international priorities. On the other hand, the default service is the best–effort service, which is exactly the same as that on the Internet.

For example, a university might give higher priority e–mail traffic than that of chat traffic. The lower priority is given for important e–mail among researchers. This service is not adequate for critical applications, but it is adequate for users who want coarse differentiation of network traffic. This example illustrates the service of the architecture. The default service is the best–effort service currently used for the Internet.

A QoS architecture for the Internet was studied as a response to the scaling imperative (21). In this study, first, the function of an edge router is only suggested to indicate the special and normal services transferring packets. Second, the administrative diversity and high speed forwarding both argue for a very simple semantics of the packet transfer indication method. Third, there is no central state, which means that a node on the network is only aggregated. The lack of an administrative center could cause fairness problems if the sharing of the network traffic load is not carefully controlled.

In the router, it makes sense to standardize forwarding behaviors. A few simple forwarding behaviors can be combined with rules enforced at the edge to provide a broad spectrum of services. The separation of the forwarding table and the behavior table in the router is a new method for building a flexible and adaptable network.

There are two methods of allocating the bandwidth and the resource: path allocation and bandwidth location. The path allocation is more difficult than bandwidth allocation because of resource management on the path.

QoS control of a network is studied in (22). There are several reasons for the demand to use QoS control. The work of RSVP and Integrated Services was originally focused on real–time and multimedia applications, which need a QoS capability for the end–to–end path. Recent demand of QoS networks has become broader, since a wider range of applications such as mission–critical applications, higher levels of assurance, and network traffic–allocation and management capabilities require QoS.

Ferguson and Huston discuss several future QoS research items: QoS Routing (QoSR), Multi-protocol Label Switching (MPLS), and RSVP extension (23). QoSR is described in (24), where it is defined as "a routing mechanism for which paths that we are determined based on some knowledge of resource availability in the network, as well as the QoS requirements of flows." QoSR supports optimal routing for the QoS on the network. If that network is operational, resource allocation and re–routing of packets will be mandatory for the resource management.

Label switching is a method for efficient routing of packets to their destinations, and can be applied to a QoS network. MPLS is focused on scalability of network layer routing, greater flexibility in delivering routing services, increased performance, and simplified integration of routers with cell–switching–based technologies. It holds various interesting possibilities in the realm of QoS, including Tag Switching, which provides direct mapping of the IP Type Of Service (TOS) field to the label CoS field.

In RSVP, the hop–by–hop signaling scheme is used. There is no explicit connection between RSVP and the underlying routing system. There is a proposal to include an interface to RSVP for routing in a QoS network. This interface is called Routing Support for Resource Reservation (RSRR), and provides for communication between RSVP and an underlying routing protocol similar in operation to any other type of APIs.

Although there are several issues as regards QoS activities, operational QoS networks will be installed for critical applications.

3.2.7 Active Networks

What will be the next step for network architectures? There is no clear answer to this question. In a survey of possible future networking architectures that we conducted, one of the candidates was the active network architecture, which has been studied in earlier work on new network architectures (25). The concept of active networks has been actively studied in the network research for several years (26). Active networks are focused on solving two issues in the current network architecture: deployment of new technology on all hosts and active participation of hosts in a network. In this section, we describe the architecture and technology of active networking in order to offer a view of the future networking technology.

In an active network, router behavior is defined by information in packet containers or pre–loaded data. A kind of programmable network router is used to construct the active network. The following example shows how the active network works. It is assumed that the sender explores the network performance and sends data to the network. At intermediate hosts, these data are subjected to an active process like a computer program, and modified the data along with the program. In addition, the actions taken there are done on a per–path or per–user basis in the network. In the current Internet architecture, packets behave passively. A router has only a simple procedure for transferring packets. The routers decide the destination of packets by looking up the entries in the routing table: there is no active processing at all. In contrast, an active network processes all packets by using a program included in the packet or loaded into routers previously.

There are two approaches to studying active networks: the programmable switch approach and the capsule approach. The programmable switch approach maintains the packet format. In addition, it provides a discrete mechanism that supports the downloading of programs for the active node. This is similar to automatically downloading a micro code from a server before the switch is operational. In the capsule approach, active miniature programs are encapsulated in transmission frames and executed at each host along their paths. The capsule approach needs more research than a programmable switch approach because of the performance issue for the packet transfer.

The activities of the active network was summarized (29). A general architecture was defined for the active network. The architecture consists of multiple execution environments (EEs). The execution environment is the running environment of the active program. Projects for the execution environment are Smartpackets, Active Network Transport System, Liquid Software, SwitchWare, Netscript and Liane. In this report, the NodeOS on which multiple EEs run, was explained. Security issues and applications of the active network are described. The target applications are active reliable multicast, protocol boosters, active congestion control, auctions and Internet enhancement.

The work on active networks is motivated by technology "push" and user "pull." The technology "push" is the emergence of active technologies, compiled and interpreted, supporting the encapsulation, transfer, interposition, and safe and efficient execution of program fragments. The user "pull" comes from the ad–hoc collection of networked applications that perform user–driven computations at active hosts. These networked applications include firewalls, Web proxies, multicast routers, mobile proxies, and video gateways.

The capsule approach has been adopted at the group of MIT, which is studying "active storage," NACK fusion, and traffic filtering for firewalls. This approach uses a built–in interpreter or compiler of programming languages. The SwitchWare project, which is studying the programmable switch approach, is led by a group at the University of Pennsylvania (27). The active network has to balance programming flexibility against the security requirements. SwitchWare achieves balance by using three layers: active packets, active extensions, and active router infrastructure. Active packets contain a small program that replaces traditional data packets, active extensions provide services on new network elements that are dynamically loaded, and the active router infrastructure provides a high security integrity for active nodes.

The implementation of an active network architecture called PLANet was reported in (28). PLANet is a pure active network architecture, which means that all packets contain programs written in a special–purpose packet language called Packet Language for Active Networks (PLAN). The active node was extended to change the function by loading an additional code dynamically. This is the first implementation of a first purely active network that does not have any underlying networking protocol such as TCP/IP.

The active network has created a major paradigm shift from ordinary network architectures. Many researchers participate in this area, and a lot of studies are in progress.

3.2.8 Conclusion

This paper has described the network architecture that is currently used for the Internet and related issues that need to be solved. We described three new types of network architecture to solve these issues. Studies of next–generation architectures for the Internet have a major impact on all network applications. The main contribution of the new architectures is in improved performance between applications. Further improvement in the behavior of applications is expected to result from changes in the architecture. For example, when we can reserve the level of performance on the Internet, we will be able to build APIs on top of the network software. Application programs can use these APIs for their applications. The assurance of an end–to–end path can be expanded to an application area. As a result, the network software will have take responsibility for part of the service. In the next generation Internet, both the development and the usage of applications will change along with the architecture. Introduction of a new architecture will greatly expand the area of network applications. GDM will become useful on the useful global network.

Notes

1) D. E. Comer, *Internetworking with TCP/IP*, (Prentice–Hall, NY, 1995).
2) D. Sangi, A. K. Agrawala, and O. Gudmundsson, Experimental Assessment of End–to–End Behavior on the Internet, In *Proc. of IEEE INFOCOM '93* (1993) pp. 867–874.
3) V. Paxson and S. Floyd, Why we don't know how to simulate the Internet, In *Proc. of the Winter Simulation Conference '97* (December 1997).
4) Q. Li and D. L. Mills, On the Long–range Dependence of Packet Round–Trip Delays in Internet, In *Proc. of IEEE ICC '98* (1998) pp. 1185–1191.
5) M. S. Borella and G. B. Brewster, Measurement and Analysis of Long–range Dependent Behavior of Internet Packet Delay, In *Proc. of IEEE INFOCOM '98* (1998) pp. 497–504.
6) V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, Framework for IP

Performance Metrics, *IETF Request for Comments*: 2330 (May 1998).

7) V. N. Padmanabhan and J. C. Mogul, Using Predictive Prefetching to Improve the World Wide Web, ACM SIGCOMM Comp. *Comm. Rev.* 3, (No. 26, 1996) pp. 22–36.

8) S. Bajaj, L. Breslau, and S. Shenker, Is Service Priority Useful in Networks?, In *Proc. of ACM SIGMETRICS '98* (Madison, WI, June 24-26, 1998).

9) M. Crovella and P. Barford, The Network Effects of Prefetching, In *Proc. of IEEE INFOCOM '98* (San Francisco, CA, 1998).

10) S. Seshan, M. Stemm, and R. H. Katz, SPAND: Shared Passive Network Performance Discovery, In *Proc. of USENIX UIST '97* (Monterey, CA, December 1997).

11) P. Francis, S. Jamin, V. Paxson, L. Zhang, D. F. Gryniewicz, and Y. Jin, An Architecture for a Global Internet Host Distance Estimation Service, In *Proc. of IEEE INFOCOM '99* (NYC, NY, March 22–25, 1999).

12) E. Cohen, B. Krishnamurthy, and J. Rexford, Improving the end–to–end performance of the Web using server volumes and proxy filters, In *Proc. of ACM SIGCOMM '98* (Vancouver, BC, Canada, Aug. 31–Sep. 4, 1998).

13) L. Fan, P. Cao, J. Almeida, and A. Z. Broder, Summary Cache: A Scalable Wide–Area Cache Sharing Protocol, In *Proc. of ACM SIGCOMM '98* (Vancouver, BC, Canada, Aug. 31–Sep. 4, 1998).

14) D. Wessels and K. Claffy, Internet Cache Protocol (ICP) version 2, *Requests for Comments* No. 2186 (September, 1997).

15) Z. Jiang and L. Kleinrock Prefetching Links on the WWW, In *Proc. of IEEE ICC '97* (Montreal, Quebec, Canada, June 1997).

16) T. M. Kroeger and D. D. E. Long, Exploring the Bounds of Web Latency Reduction from Caching and Prefetching, In *Proc. of USENIX USITS '97* (Monterey, CA, USA, December 1997) pp. 22–32.

17) S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, *IETF Request For Comments*: 2475 (December 1998).

18) K. Nichols, S. Blake, F. Baker, and D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, *IETF Request For Comments*: 2474 (December 1998).

19) B. Teitelbaum, J. Sikora, and T. Hanss, Quality of Service for Internet2, In *Proc. of First International Joint Applications Engineering QoS Workshop* (Santa Clara, CA, May 21–22, 1998).

20) C. Song, L. Cunningham, and R. Wilder, Quality of Service Development in the vBNS, *IEEE Communications Magazine* (May 1998) pp. 128–133.

21) V. Jacobson, Differentiated Services for the Internet, In *Proc. of First International Joint Applications Engineering QoS Workshop* (Santa Clara, CA, May 21–22, 1998).

22) J. Wroclawski, Evolution of End–to–End QoS: A design philosophy, In *Proc. of First International Joint Applications Engineering QoS Workshop* (Santa Clara,

CA, May 21–22, 1998).

23) P. Ferguson and G. Huston, *Quality of Service*, Prentice–Hall, NY, 1998.

24) E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, A Framework for QoS–Based Routing in the Internet, *IETF Request For Comments*: 2386 (August 1998).

25) D. D. Clark and D. L. Tennenhouse, Architectural Considerations for a New Generation of Protocols, In *Proc. of ACM SIGCOMM '90* (Philadelphia, PI, September 24–27, 1990).

26) D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J Minden, A Survey of Active Network Research, *IEEE Communications Magazine*, (January 1997) pp. 80–86.

27) D. S. Alexander, W. A. Arbaugh, M. W. Hicks, P. Kakkar, A. D. Keromytis, J. T. Moore, C. A. Gunter, S. M. Nettles, and J. M. Smith, The Switch Ware Active Network Architecture, *IEEE Network, Special Issue on Active and Programmable Networks* (May 1998).

28) M. Hicks, J. T. Moore, D. S. Alexander, C. A. Gunter, and S. M. Nettles, PLANet: An Active Network, In *Proc. of IEEE INFOCOM '99* (NYC, NY, Mar. 21–25, 1999).

29) J. M. Smith, K. L. Calvert, S. L. Murphy, H. K. Orman and L. L. Peterson, Activating Networks: A Progress Report, *IEEE Computer* (32) 4 (April 1999) pp. 32–41.