The Beginnings of Packet Switching: Some Underlying Concepts

Paul Baran, Com21, Inc.

ABSTRACT

This article was written for a seminar held on the occasion of the Franklin Institute's 2001 Bower Award and Prize for the Achievement in Science to the author "for his seminal invention of packet switching — the foundation of modern communications networks and, in particular the Internet."

It describes the author's work 40 years ago focusing on the rationale creating the key concepts of packet switching. The article considers the development of each of a series of about 20 essential concepts. For example, it examines such subjects as the degree of redundancy to achieve any desired level of survivability; the necessity to chop data streams into small blocks; what information had to be appended to these blocks to allow the each block to find its own way through the network; why it was necessary for each network element to operate at an independent data rate; why all signals had to be converted to digital, and so on.

In brief it describes the "why" as well as the "how" of packet switching works.

INTRODUCTION

This article is a nostalgic trip, revisiting work I did 40 years ago. It describes why packet switching was invented, how it works, and why it seems to work so well. A personal recollection such as this tends to overemphasize the importance of one's own work. Throughout the design process, I borrowed freely using whatever technology best fit the objective. I mention this to avoid any inadvertent impression that all the ideas to be described are totally original. Furthermore, my work focuses on basic concepts, not implementation — which was done by others.

To keep matters in context it is desirable to describe where this work stood relative to the state of the art at the time it took place. In 1964 I wrote a highly detailed set of memoranda that describe this work down to the circuit level [1]. These memoranda are on the Internet and readily available to the reader [2]. I devoted an entire volume to the matter of history and alternative approaches considered [3], which I recommend to the serious reader to understand where this work stood in relation to other work at the time.

This activity was undertaken in 1960 at the RAND Corporation. RAND was established by the U.S. Air Force to preserve the operations research capability created by the Air Force in World War II, and to work on issues of national security. The freedom of the staff to choose projects, try novel approaches, and disagree with the bureaucracy along the way is difficult to imagine in the present environment. Today, proposals must be written, projects excessively monitored, and reports prepared whether or not there is anything worthwhile to report.[1] It was a different era then, and the remarkable degree of freedom I enjoyed that encouraged farout, and sometimes wild, thinking would be hard to duplicate today.

COLD WAR BACKGROUND

When I joined RAND in 1959, a glaring weak spot in our strategic forces command and control communications was a dependence on shortwave radio and the national telephone system, AT&T, both highly vulnerable to attack. H-bomb testing in the Pacific revealed that long distance shortwave (high-frequency) sky-wave transmission would be disrupted for several hours by a high-altitude nuclear blast. Computer simulations showed that weapons targeted at U.S. retaliatory forces would render long distance telephone communications service inoperative by collateral damage alone. While most of the telephone facilities would survive, the paucity of switching centers formed a dangerous Achilles' heel.

To cool tensions at this stage of the cold war, a retaliatory force capability was needed that could withstand a surprise attack, and survive sufficiently to return the favor in kind in a controlled manner. A survivable command and control communications infrastructure would be mandatory to get away from the guns loaded, hair trigger doctrine of the time.

RAND computer simulations showed that the telephone system would fail, while most telephone

[1] *I am particularly indebted to Frank Colbohm, John D. Williams, Frank Eldridge, Albert Wohlstetter, Paul Armer, Willis Ware and Keith Uncapher among others at RAND for continuing strong support while undertaking my highly controversial activity.*

0163-6804/02/$17.00 © 2002 IEEE

facilities survived. I believed that the problem was obviously in the topology of our communications networks, and there might be a solution. (I had worked on the subject of survivable networks while at Hughes Aircraft before coming to RAND, so I was not new to the subject.)

WHY NETWORKS ARE VULNERABLE

Let's consider three different communications network topologies in Fig. 1.

(a) *The centralized network* has all its nodes connected to a central switching node to allow simple switching, giving it a single point of high vulnerability.

(b) *The decentralized network*, representative of the AT&T Long Lines network at the time, is better. Instead of a single central switching node, the network comprises small centralized clusters, with most traffic going to nearby neighbors, and only the longer distance traffic routed to the longer links.

(c) *The distributed network* is a network without any hierarchical structure; thus, there is no single point of vulnerability to bring down much of the network.

Shortly after I arrived at RAND I began to study the behavior of distributed networks with different levels of redundant connections.

In Fig. 2, we see a network with its nodes tied together with the minimum possible number of links. This is called a network of redundancy level 1. A network of redundancy level 2 looks like a fishnet, with horizontal and vertical links. When we reached redundancy levels on the order of 3 an interesting phenomenon occurred: the network became extremely robust. If a node survived physical damage, it would likely be connected to all other surviving nodes in the largest single group of surviving nodes. This meant that *it would be theoretically possible to build extremely reliable communication networks out of unreliable links, by the proper use of redundancy*.[2] In other words, if a redundantly connected node survived the physical attack, there is a high probability that this node, at least on paper, was somehow connected to all the other surviving nodes. "Somehow" was the issue, and was the motivation for packet switching.

STATE OF THE ART, 1960–1964

In those days (around 1960) we didn't know how to build communication switches where signals could traverse many serially connected nodes and operate reliably in the face of damage. The AT&T telephone system had a limit of five switched tandem links before a phone call was unacceptable.

A new way was needed to get usable signals through a large number of nodes, traveling via highly circuitous paths that could not be determined in advance. The new network would have to relay signals along without errors.

I considered several analog transmission approaches, but kept hitting a brick wall. The only way I could think of around this restriction was to transmit all signals digitally to avoid the distortion buildup, and the routing information would have to go along with the data itself.

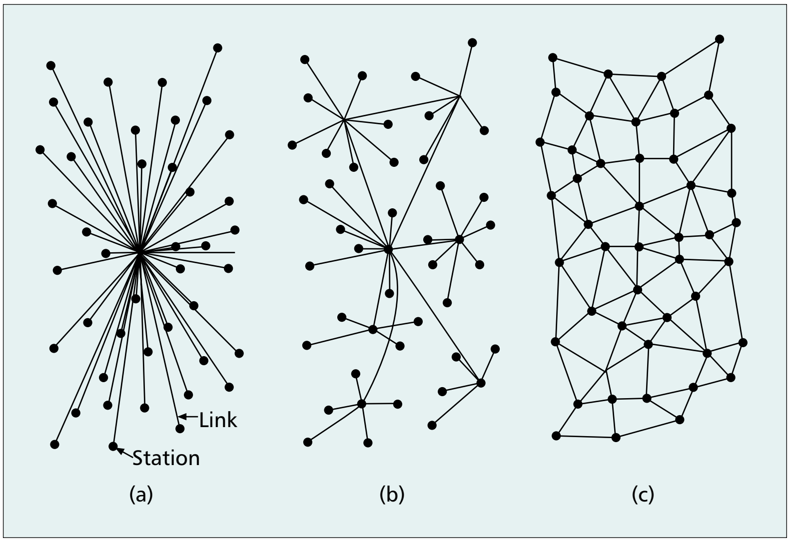

Figure 1. *Three forms of networks: a) centralized; b) decentralized; c) distributed.*

THE BROADCAST STATION DISTRIBUTED NETWORK

My first RAND distributed network proposal in 1960 was for a survivable teletypewriter network to carry what was then called "minimum essential communications." Carrying briefing charts and slides[3] around to the Pentagon and various military command centers, I found the term was unrealistic. Far, far more capacity was needed than was previously realized. So I went back to the drawing board and took on the challenge to come up with a scalable communications switching structure capable of dynamically routing high-bit-rate traffic among a large set of potential users, and where user requirements could not be predicted in advance.

(Meanwhile, the broadcast teletypewriter concept crept slowly through the Air Force process and was eventually assigned to the Rome Air Development Center in upper New York State for implementation. An experimental network was built to cover the northeast section of the US. Its only stress test was the massive Northeast power blackout in 1965 when it was said to have worked well.)[4] [4, 5]

NETWORK SYNCHRONIZATION

My interest was now focused on creating a new, very high-data-rate (in 1960s terms) network. I was able to start the design with a clean sheet of paper rather than fighting all the constraints of our then networks. For example, since the data flow in the network had to traverse many tandem nodes, I felt it would be impossible to synchronize all individual links in tandem to operate at the exact same data rate. Instead I proposed small computer-based switching nodes, to provide a small amount of buffering to eliminate the need for overall network timing, letting each link operate at its own "natural" data rate(Fig. 3).

This choice meant that there would be no physical real-time connection between the transmitting and receiving ends. But I felt that would be okay; if the transmission data rate was high

[2] *The optimum degree of redundancy is a function of the sum of the loss of capacity by component reliability failures plus the damage anticipated. It doesn't make much difference if the damage is due to an attack or component reliability. Thus, it would be possible to build highly reliable networks out of unreliable links and nodes able to withstand high levels of damage at a redundancy level of about 3. Even a redundancy level of 1.5 would more than suffice for a network where no enemy attack is anticipated, just component reliability.*

[3] *The majority of the illustrations in this article were originally projected using glass lantern slides, reminding me how much technology has changed in 40 years.*

[4] *Extensive testing has taken place in the past on the simultaneous transmission of a commercial broadcast and submodulated teletyped signal. A number of commercial AM broadcast stations have been operated in a closed loop fashion to provide an extension beyond the range capability of one station.*

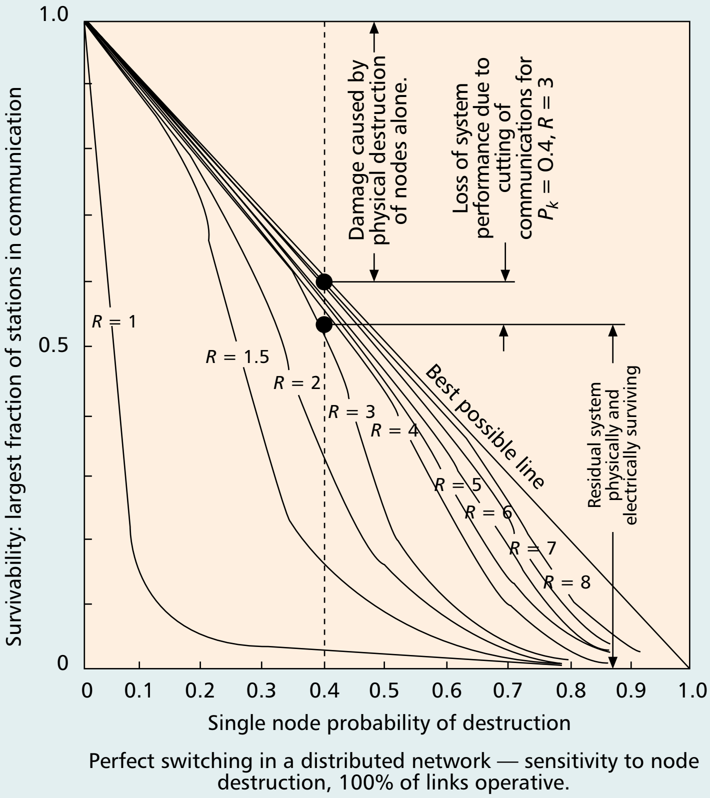

Figure 2. *Survivability in a distributed network with different levels of damage.*

enough, the user would be fooled by the illusion that a real-time connection existed.

Mix and Match

This breaking of the lock step nature of the circuit switch link meant that it should theoretically be feasible to build the network from a collection of different types of links, each operating at a different data rate if desired (Fig. 4).

From the earlier study of the effects of redundancy, high link reliability would not be needed in a distributed network anticipating heavy damage. This is unlike the case of circuit switching, where a single failed tandem element prevented end-to-end communications. This fundamental difference may seem obvious and even trivial today, but its statement tended to generate an undue number of livid words from otherwise competent communications transmission engineers. Those not versed in digital computer art tended to excessively strong objections. And most of those whose day-to-day occupation was caring for telephone lines thought that I must be crazy, a complete fraud who didn't understand how a telephone worked, or both. With some notable exceptions, the proposed ideas were not universally received with great joy.

Choice of Switching Data Rate

The next design choice was the approximate data rates for the switching node processing. At the time, there was some interesting early work underway at Bell Labs by John Mayo and others, on what would become the T1 multiplexing system. By replacing telephone loading coils, nominally at 1-mi spacing, with limiting amplifiers, 24 separate 64 kb/s digital voice channels could be multiplexed on existing copper telephone pairs at 1.54 Mb/s. The system was limited to a maximum range of about 150 mi before the jitter built up to make the link unworkable. But that was okay in my mind because I contemplated that the switching nodes would be retiming the digital signals anyway. So 1.54 Mb/s seemed like a good design data rate.

Getting Through the Damaged Maze

The scheme I settled on to quickly find paths through a network of changing topology while it was being attacked was to route data through the network based on adaptive learning of past traffic. Intuitively it seemed that it should work. But of course I couldn't really be sure until after a computer simulation. My RAND colleague Sharla Boehm ran many simulations under different conditions confirming the network's behavior. The simple switching protocol exhibited remarkable intelligence, routing traffic efficiently, yet responding quickly to changes caused by damage. For example, under simulation we found that upon half the network being instantly destroyed, the remainder of the network reorganized itself and was routing traffic effectively within less than 1 s of simulated real world time.

The routing protocol was simple. Each message block, these days called a packet, had a *to* and *from* address field together with a *handover counter* field that was incremented every time the packet was sent from node to node. The value of the handover number was an estimator of the length of the path taken by each packet. Each switching node regarded recent handover numbers as better estimators than older measurements. The network not only learned, it also had to forget, and thus be able to respond to changes in link and node availability.

The Post Office Analogy

John Bowers, a RAND colleague, suggested that it was easier for him to visualize the concept by imagining an observant postman at each node (or post office). The postman could infer from the lowest received cancellation date (handover number) of the letters (packets) coming FROM any direction (link) the best direction to send traffic TO in the future to that address. By observing traffic passing through the node and by recording the handover numbers of the FROM station, together with the link number, the imaginary postman could determine the best TO link, the second best TO address, the third, and so on. When the shortest path link is busy or out of action, the next best path will be taken. Since the postman explanation was so easy to understand, I have used this explanation to this day.

Hot Potato Routing

To dramatize the need for speed in the switching nodes, I described the switching process by saying that each message block should be regarded as a hot potato, tossed from person to person, without gloves. You want to get rid of the hot potato as quickly as you can. If your first choice recipient is busy, toss it to your second choice recipient, and so on. If you have no better choice you are allowed to throw the hot potato back to the previous thrower. Everything had to be essentially instantaneous, if voice was to be transmitted, because voice is intolerant of delay.

This early routing scheme is called the *hot potato routing* algorithm, and has been reinvented the usual number of times, and now is most often called *deflection routing*.

SEQUENCE NUMBER

Since sequential packets can travel by different paths, they often arrive out of sequence. A short serial number in the header indicates the sequence of packets sent. The receiving unit notes the short modulo serial number and sorts the received packets into their correct sequence buffers, so packet after packet comes out in correct sequence even if some packets travel through a longer path than others.

CYCLICAL REDUNDANCY CHECK

Part of the housekeeping field in each packet is dedicated to error detection. The original RAND plan used a cyclical redundancy check (CRC), which is still the preferred error detection approach today. The CRC provides an efficient but not foolproof error detection test. If the error detection test fails, no acknowledgment is sent and the packet is retransmitted. Generally, an acknowledgment of the properly received packet is required before the responsibility for further relaying action is transferred.

An end-to-end error control measure, described below, ensures that the few distorted packets that get through the process will be caught and replacement packets requested. As a result, the system can be made arbitrarily error-free, even when using links with high error rates.

ON USING THE IDEAS OF OTHERS

All systems are built on top of the work of others, such as my use of the concept of CRC checking. Throughout this system design process I borrowed freely at the subsystem level, using whatever technology best fit the objectives. I mention this again to avoid any impression that all the ideas described are totally original.

CRYPTOGRAPHY

I proposed a two-level cryptography hierarchy. The first level would be a conventional node-to-node cryptographic transformation on each link with these local node-to-node keys known only by each of the two end nodes. The second level of cryptography would be a conventional end-to-end users' cryptography arrangement, with a separate key for each pair of affiliated users. The one new feature would be achieving an arbitrarily low overall error rate by requiring each previously received serial packet, after decoding, to form part of the key needed to decode the next packet. If an error slipped through the data stream the output would turn to garbage, and a packet request to resend the defective packet. Packets would be encouraged to intentionally travel by different routes. So after the first few packets, an eavesdropper having all the keys in the system would not be able to decipher the data stream since he would lack one or more of the critical packets needed to unravel the rest. Of course it is necessary to throw away the first few packets of a connection for this capability to take hold.

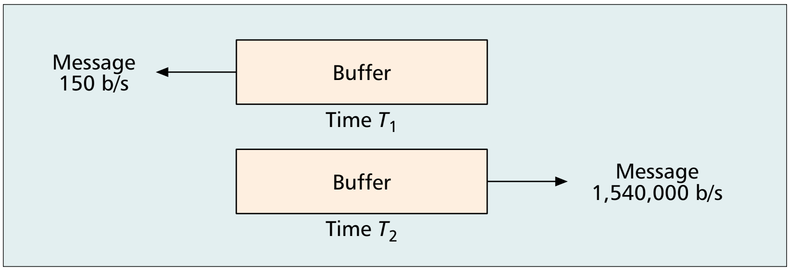

Figure 3. *The time transformer concept in packet switching.*

SUPER-RELIABILITY

This cryptographic scheme automatically ensures that all received traffic is totally error-free, since any error would cause a backup and retransmission, generally via a different path. End-to-end errors simply cannot occur; the length of each packet, 1024 bits, in effect constitutes a very large key space. I do not believe any system has yet been built in this manner, but I still like the idea.

SWITCHING NODES AND MULTIPLEXING STATIONS

There are two parts of the system. To this point we have been considering the switching nodes that get packets from one point in the network

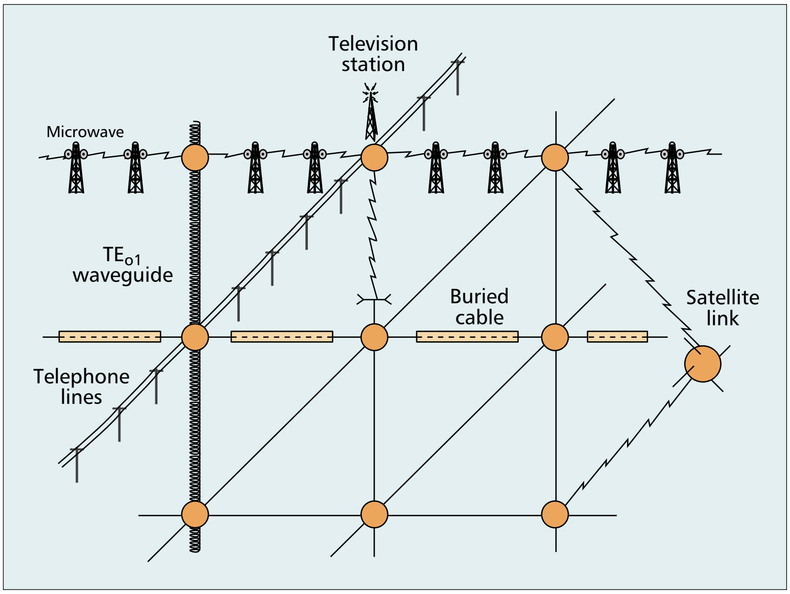

Figure 4. *A network composed of a heterogeneous combination of different media.*

Wherever you move, in essence you take your telephone number with you. This is unlike the telephone system at the time where the telephone number referred to a specific physical central office pair of wires. This is one of the characteristics of today's Internet we take for granted.

to another. A second subsystem is needed to terminate conventional circuits from many users. This second unit, called a *multiplexing station* provides functions such as filling in missing (blank) packets during silence periods and the end-to-end functionality. It provides the concentration of a large number of lower-data-rate users sharing a few high-speed channels. Each sent packet must have the local address of the end user as well as the address of the multiplexing station. On the transmitting end, the functions include chopping the data stream into packets, adding housekeeping information and end-to-end error control information to the outgoing packets. On the receiving end, each multiplexing station uses terminating buffers temporarily assigned to each end addressee to unscramble the order of the arrived packets, and buffer them so that they come out as an error-free stream, only slightly but not noticeably delayed.

SEPARATING LOGICAL FROM PHYSICAL ADDRESSES

An interesting difference from circuit switching practice is the concept of separating the physical address from the logical address. This came about in part from the requirements for a system designed for command and control communications, which sought to avoid single points of failure. Since the military command structure is composed of individuals who are unique and can't be replicated very easily, their vulnerability would violate the no single target being better than any other concept. The idea here was to play the old shell game of a single pea and multiple walnut shells. Think of the pea being the commander and the various points of entry into the network as the walnut shells. The commander could appear at any location on the network and start operating, and the network would quickly learn the new location and begin routing traffic accordingly. By removing any connection between physical and logical addresses, a new freedom is created that is useful in many ways. Wherever you move, in essence you take your telephone number with you. This is unlike the telephone system at the time where the telephone number referred to a specific physical central office pair of wires. This is one of the characteristics of today's Internet we take for granted.

SUPPRESSION OF SILENCE

In most circuit-switched communication applications, silence is the usual message since no information is transmitted most of the time (remote computer terminals, voice, two-way video, etc.). There is an economy to be gained by not sending long strings of 0s or 1s that contain no information, necessary in conventional circuit-switched networks. The magnitude of this economy can be large, because the common facilities are so effectively shared. In packet switching, we avoid sending packets unless there is information to be sent. If there is no change in the data stream relative to the content of the last packet, why bother sending a packet?

VIRTUAL CIRCUITS

The multiplexing station has the responsibility to provide the connected user with the missing nontransmitted packets to maintain an illusion for the user that his or her computer is always connected. This concept is called a *virtual circuit*. There is no limitation to the number of virtual circuits that can be simultaneously maintained; in brief, the process fakes out the connection. The virtual circuit creates the illusion of a circuit always being there when you need it, but consuming no resources when not instantly needed. It is the high speed of transmission and switching that allows this sleight of hand to create an illusion that a physical connection is always present.

When considering the network as the user at the multiplexing station, the network appears as a fuzzy cloud. The user does not particularly care which instantaneous path his or her traffic uses to get to its destination. The user need not be concerned about the transportation portion of the network. Rather, all the user sees is a virtual circuit to the chosen end destination. The user cannot tell the difference between a physical and a virtual circuit, but the economics permits "selling" the same "circuit" many times over, legally.

RELIABILITY

There are two components of reliability: the probability that a path exists between two users, and the probability of no errors when using that virtual circuit. The factors that combine to provide super-reliability possibly include:

- The redundancy of the routes allowed
- The policy of keeping a "carbon copy" of the transmitted packet until each node is certain that the packet sent has been correctly received by the next recipient
- End-to-end control to replace any lost packets

If in doubt, the packet is retransmitted and the sequence number used to clean up the duplicates.

WHERE DID THE NAME PACKET SWITCHING COME FROM?

I used the term *message block* in the early 1960s. In 1965 Donald W. Davies of the British National Physical Laboratory, unaware of my earlier work, independently came up with the same basic concept — and chose the same data rate, 1.54 Mb/s, and the same packet length, 1024 bits. Davies called his system *packet switching*, a far better choice of words, and it has become the name that stuck. Davies said he specifically chose the term packet switching to distinguish it from message switching, an earlier technology dating from the telegraph and later the teletypewriter era. Davies wrote a paper shortly before he died in June 2000 [6] describing his contribution to the field, followed by a careful analysis of a 1962 doctoral thesis recently cited by another highly regarded early worker in the field, who recently began claiming priority for the invention of packet switching. It is Davies' position, on detailed examination, that the cited reference dealt not at all with packet

Function	Reason	1960 P-1995	1962 P-2626	1964 RM-ODC
1. Distributed network, level of redundancy.	Highly survivable structure to withstand possible enemy damage and element unreliability.	√	√	I, p. 10
2. Digital transmission	Allows many tandem connected links required, without irreversible analog distortion buildup.	√	√	I
3. High-speed data links.	Far more economic to share a sharing a high-speed link than using many low speed trunks.		√	VI
4. Adaptive digital switching	Allows almost infinite path choices to get around damaged nodes and links.		√	I
5. Packets	Allow time transformation for multiple real-time data streams, and rapid data handling and verification.	√	√	I, p. 20
6. Handover number	Required to determine optimum path choice and to remove lost packets.		√	II, p. 6
7. Sequence number	Required to reorder packets that arrive by variable length paths.		√	VIII, p. 16
8. Cyclical redundancy check	Catches most transmission errors and prevents buildup of errors.		√	II, p. 23
9. Hot potato routing	Assures generally shortest path taken while rapidly dynamically adapting to network changes		√	II, p. 6 I. p. 26+
10. Virtual circuits and blank suppression	Creates the illusion that many full-time circuits exist by taking advantage of statistics — mostly nothing is being sent.		√	III, p. 14
11. Congestion control	Required to prevent local network overloads		√	IV, p. 14
12. Trace back	Allows pinpointing network fault location.			VIII, p. 66
13. Switching nodes separate from user terminations	Move the high complex logic portions of the system to the edges of the network.			VIII, p. 4
14. Physical and logical addresses decoupled	Allows users to move around and have their communications follow.		√	I, p. 32
15. Dual level universal cryptography	Allows use of network without excessive concern about eavesdroppers.			IX, p. 19
16. Learning and forgetting	Required for network adaptation to damage.		√	VII, p. 48
17. Priority, precedence, and overload prevention	Now called quality of service (QoS).			IV
18. Error-free operation	Combination of link-by-link CRCs combined with end-to-end crypto that removes remaining errors.			VII, p. 23
19. Common user type system	Greater redundancy possible, more survivable and lower-cost if shared among largest number of users.		√	X
20. Universal modality	Better economics of a single network handing all types of services, voice data, teletypewriter, etc.		√	X
21. Record of previous history	Objective: to avoid reinventing the wheel.			V

Table 1. *Basic concepts of packet switching to be found described in RAND P-1995 (1960), P-2626 (1962) and in the* On Distributed Computers *series of RAND Research Memoranda (1964).*

switching, but solely on the older message switching art. This has recently become a public issue [7]. The issue is then exactly what is packet switching, and how is it defined? A brief search on the Internet finds close consistency in all the dozens of definitions that can be found, for example:

"In packet-switching, a message is divided into packets, which are units of a certain number of bytes. The network addresses of the sender and of the destination are added to the packet. Each network point looks at the packet to see where to send it next. Packets in the same message may travel different routes and may not arrive in the same order that they were sent. At the destination, the packets in a message are collected and reassembled into the original message." [8]

Another example is:

"…a data transmission technique whereby user information is segmented and routed in discrete data envelopes called packets, each with its own appended control information for routing, sequencing and error checking; allows a commu-

Davies called his system "packet switching," a far better choice of words, and it has become the name that stuck. Davies said he specifically chose the term "packet switching" to distinguish it from "message switching," an earlier technology dating from the telegraph, and later the teletypewriter era.

nication channel to be shared by many users, each using the circuit only for the time required to transmit a single packet; describing a network that operates in this manner." [9]

The new argument seems to be that it is unimportant whether the communications stream is chopped into packets or not, while Davies' position is that all the wonderful things that occur with packet switching simply do not occur unless you chop the data stream into packets.

THE UNDERLYING CONCEPTS: REFERENCE

The title of this article calls for describing the underlying concepts of packet switching, but I have run out of space while only partway through. I added a chart, Table 1, that lists the concepts that came out of the RAND work. Essentially all the work was defined by 1961, and fleshed out and put into formal written form in 1962. The idea of hot potato routing dates from late 1960. The detailed series of RAND Research Memoranda were essentially completed in 1961–1962, but held up to allow the entire series to be released at one time. The dates of the concepts listed are the dates of publicly available literature, that is, technical papers available in depository libraries around the world. Three reference points, 1960, 1962, and 1964, are shown with the relevant references of 1) RAND Paper P-1995, "Reliable Digital Communications Systems Using Unreliable Network Repeater Nodes," May 29, 1960; 2) RAND Paper P-2626, "On Distributed Communications," November. 1962; and 3) the August 1964 series of RAND Memoranda op cit. (The drawings in the present article were taken from the 1962 paper.) All references are available from the RAND Publications Department, and the multivolume RAND RMs are available online as described earlier.

DISCLAIMER

From time to time I have been assigned credit for all sorts of things that I haven't done. For example, I am not responsible for the ARPANET. Its initiator was Robert Taylor, and it was a project managed by Larry Roberts who provided the high-level conceptual specifications with the design detailed and implementation by Bolt, Beranek and Newman, Inc. My role was very minor, as described by Abbate:

"Paul Baran, too became directly involved in the early stages of planning the ARPANET. Roger Scantlebury had referred Lawrence Roberts to Baran's earlier work. Soon after returning to Washington from Gatlinburg, Roberts had read Baran's *On Distributed Communications*. Later he would describe this as a kind of revelation: 'Suddenly I learned how to route packets.' (Norberg and O'Neil 1996, p. 166). Some of the ARPANET contractors, including Howard Frank and Leonard Kleinrock, were also aware of Baran's work and had used it in their research. In 1967 Roberts recruited Baran to advise the ARPANET planning group on distributed communication and packet switching. Through these various encounters, Roberts and other members of the ARPANET group were exposed to the ideas of Baran and Davies. And they became convinced that packet switching and distributed networks would be both feasible and desirable for the ARPANET." [10]

The ARPANET grew and flourished through the effort of many, including graduate students around the country who turned the basic BBN-designed packet switching network into a computer communication network by the work of many others in the research community.

It is appropriate to give credit to those who had a major role in that early activity. That list would include, among others, Vint Cerf, Danny Cohen, Steve Crocker, Howard Frank, Frank Hart, Bob Kahn, Len Kleinrock, John Melvin, Severo Ornstein, John Postel, Larry Roberts, Elmer Shapiro, and Bob Taylor.

I can cover only a small part of the story of packet switching in this limited space, and from a limited point of view on how the ARPANET was incorporated into the larger concept of a network of networks, each preserving its own uniqueness but adding to the whole. And it then grew and grew. And we haven't touched on other applications of packet switching, or that essentially all new communications networks being built today are based on the use of packets.

REFERENCES

[1] P. Baran *et al.*, "On Distributed Communications," RM 3420, 3103, 3578, 3638, 3097, 3762-7, The RAND Corp., Santa Monica, CA, Aug. 1964.

[2] http://www.rand.org/publications/RM/baran.list.html

[3] P. Baran, "On Distributed Communications," vol. V, *History, Alternative Approaches, and Comparisons*, RM-3097-PR, The RAND Corp., Santa Monica, CA, Aug. 1964.

[4] E. E. Selover, "HF Survivable Communications System," *IEEE Mohawk Valley Communications Symp.* (NATCOM), Oct. 1965 (AFSC Project 9680), pp. 35–40."

[5] J. Costa, "RADC Develops Logic Design for Emergency Broadcasting," *Electronic News*, Jan. 10, 1966, p. 32.

[6] D. W. Davies, "An Historical Study of the Beginnings of Packet Switching," *Brit. Comp. Soc. J.*, vol. 44, no. 3, 2001, pp. 151–62.

[7] K. Hafner, "A Paternity Dispute Divides Net Pioneers," *New York Times*, Nov. 8, 2001.

[8] http://www.searchnetworking.techtarget.com

[9] Data Communications, Mar. 1988, glossary compiled by the staff of Data Communications, p. 270.

[10] J. Abbate, *Inventing the Internet*, MIT Press, 1999, pp. 36–37.

ADDITIONAL READING

[1] J. Naughton, *A Brief History of the Future*, 1999 and 2000, Overlook.

[2] G. Dyson, *Darwin Among the Machines*, 1997, Addison Wesley.

[3] K. Hafner, and M. Lyons, *Where Wizards Stay up Late*, 1996, Simon & Schuster.

[4] A. Norberg and J. O'Neil, *Transforming Computer Technology*, 1996, Johns Hopkins.

BIOGRAPHY

PAUL BARAN [F] (Paul@Baran.com) received a B.S. in E.E. from Drexel in 1949 and an M.S. in E.E. from UCLA in 1959. He started seven Silicon Valley companies of which five became public companies. He presently serves as chairman of the board for Com21, Inc., and as a trustee/director for several not-for-profit organizations including the Charles Babbage Foundation, the IEEE History Center, and the Marconi International Fellowship. He is a fellow of the AAAS and a member of the National Academy of Engineering. His past awards include the IEEE Alexander Graham Bell Medal, the Japanese (NEC) Computers and Communications Award, the ACM SIGCOMM Award, the IEEE Internet Award, an honorary Doctor of Science degree in engineering from Drexel University, and an honorary Ph.D. in policy analysis from the RAND Graduate School.