Systems Network Architecture (SNA) has been enhanced to include features that address the topological, routing, congestion, reliability, and availability problems of networks. An important aspect of this new release of SNA is that it allows multiple active routes between network nodes. Multiple routing permits sessions between network users to use alternate routes in case of unexpected or planned route disruptions. In this paper, the multiple routing architecture of SNA is described.

An unrestricted data flow into the network can cause long delays and buffer depletion. Network congestion can be avoided by employing flow control mechanisms at both the local (node) and global (network) levels. This paper focuses on global flow control and describes the adaptive traffic-pacing "window" size algorithm that is the basis of the global flow control in SNA.

Routing and flow control in Systems Network Architecture

by V. Ahuja

IBM's Systems Network Architecture (SNA) formally defines the functions of various network system components, thereby allowing IBM to develop a unified set of products for distributed data processing systems. The architecture has grown from supporting a single-host configuration (Figure 1) to multiple host networks (Figure 2). These enhancements imposed new requirements on the network in the areas of data routing, flow control, and error recovery of message units. There are several publications that describe SNA.[1-5] This paper describes those functions of SNA that support multiple active routes and the flow control mechanisms that prevent network traffic congestion.

The general structure of an SNA network is shown in Figure 3. A network user (such as a terminal operator or an application pro-

Figure 1 A single-host SNA configuration

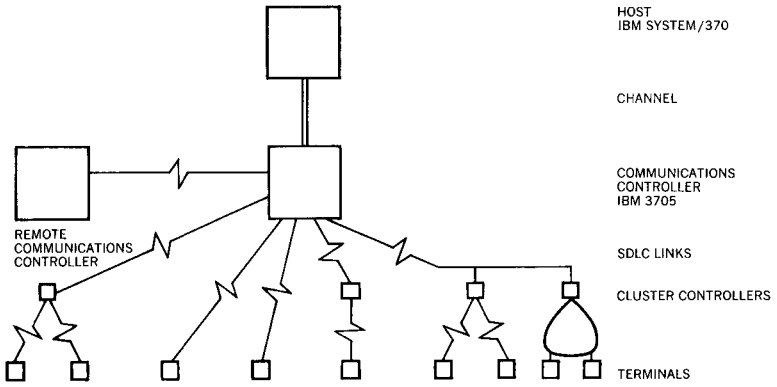
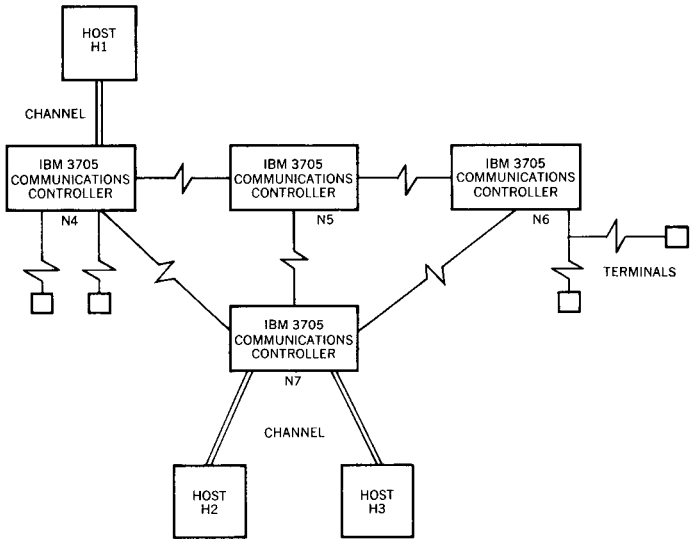
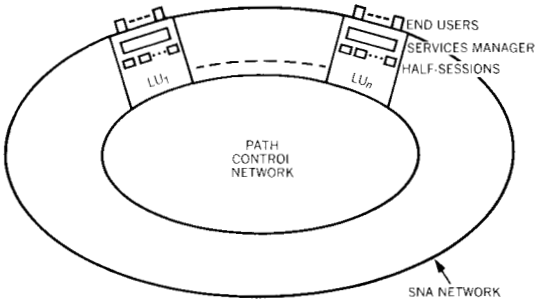


Figure 2 A multiple-host SNA configuration



gram) accesses the network through a *logical unit* (LU). Each LU is associated with a network address and contains a services manager and one or more *half-sessions*. Each half-session is connected by means of the path control network with a half-session at an LU elsewhere. The connection of two half-sessions is called a *session*. A session allows network users at different LUs to interact usefully with each other.[1]

Figure 3 General structure of an SNA network (See Reference 1 for more details)



LUs are one of three kinds of *network addressable unit* (NAU). The other two kinds are *system services control points* (SSCPs), which control a portion or a *domain* of a network, and *physical units* (PUs), which carry out commands from the SSCPs and control resources within their nodes. LUs, SSCPs, and PUs are all assigned network addresses. Like LU-LU sessions, there are SSCP-SSCP, SSCP-LU, SSCP-PU, and PU-PU sessions.

The association of an SSCP and the PUs, LUs, links, and link stations that it activates and deactivates form a domain of control for domain bring-up and take-down, dynamic address assignment, problem determination, maintenance statistics gathering, and other functions. SSCPs together control the cross-domain activities that allow LUs in different domains of the network to carry on sessions for their end users.[6]

PUs are divided into four types—Types 1, 2, 4, and 5—based on the further subdivision of domains into addressing *subareas* and on the capabilities of the nodes containing each PU. Network addresses have a *subarea* and an *element* address component. Each PU Type 4 and PU Type 5 is assigned a specific subarea address. All other PUs, LUs, and links in a domain are in the subarea of one or another of the Type 4 or 5 PU nodes, in which they reside or to which they are attached. The element address distinguishes a particular PU, LU, or link within a subarea.

Each node in the network is characterized by the PU type it contains. PU Type 4 and 5 nodes—IBM 3704 and 3705 Communications Controllers, the System/370, the 303X and 43X1 processors—provide full network address routing, as well as both local and global flow control capabilities. They also provide certain *boundary function* support, consisting of services relating to

IPLing (initial program loading), dumping, session activation and resetting, and address translation, for PU Type 1 and 2 nodes.

PU Type 1 and 2 nodes—terminals and cluster controllers—have somewhat less network awareness; primarily, they are free of network address awareness and of the responsibility of network routing and global flow control. They do participate, however, in local flow control. The PU Type 4 and 5 nodes provide network-to-local (short-form) address translation to and from the PU Type 1 and 2 nodes. This limits the sensitivity of the latter to changes in network address configuration, thereby enhancing dynamic network reconfigurability, because only nodes having network routing tables are affected by changes. PU Type 4 and 5 nodes differ only in that PU Type 5 nodes contain an SSCP in addition to the PU, while PU Type 4 nodes (such as the IBM communications controllers) do not.

In this paper, we refer to PU Type 4 and 5 nodes as *subarea nodes*, or in some cases, simply as *nodes*. A subarea node can be a terminus for, or a node within, "explicit" or "virtual" routes which will be described later. Subarea nodes and their routing and global flow control capabilities are the focus of this paper.

Message units are transported between NAUs through the path control network. The path control network consists of a path control component at each node, and data link control components for the links attaching the nodes. Path control provides routing and flow control of message units (or Path Information Units, PIUs). The data link control components manage the links between the nodes. Related details are described in the paper by Gray and McNeill in this issue.[6] In the remainder of this paper, the path control layer is addressed in greater detail.

Several releases of SNA have been developed and implemented in the past in various IBM products. To cite the major releases in sequence: SNA was originally announced for single-host networks (1974), then for multiple-host networks (1976), and then, problem determination and network management functions were added (1978). Cryptography and security features have also been added since the original SNA announcement. This paper describes the most recent release (1979), which enhances the flow control and routing support in SNA networks. This release consists of the program products ACF/NCP/VS (Network Control Program for the IBM 3705 with the Advanced Communication Function) Release 3, ACF/TCAM Version 2 (Telecommunications Access Method) Release 3, and ACF/VTAM (Virtual Telecommunications Access Method) Release 3, where both TCAM and VTAM are implemented in the System/370, 303X, and 43X1 processors.

The first section of the paper addresses the routing strategies of SNA. The second section describes the flow control mechanisms that limit congestion and minimize network response time.

Multiple active routes

The essence of multiple active routes in SNA can be explained by an example. Consider a network consisting of three hosts and four communications controllers, as shown in Figure 2. A NAU in host H1 requires a session with a NAU in communications controller N7. Clearly the shortest route in terms of the number of hops is H1 → N4 → N7. Now consider the case when the link between N4 and N7 is inoperative. This requires that some other route be used, such as H1 → N4 → N5 → N7 or H1 → N4 → N5 → N6 → N7. In earlier releases of SNA, only a single route could be defined between any two nodes. Thus, an alternate route could be used only after a new system definition and IPL. SNA now permits multiple routes between pairs of subarea nodes. In the multiple-route situation, the session can be reinitiated on another available route, such as H1 → N4 → N5 → N7.

Before describing the routing architecture, we review the various types of routing strategies. Strategies to provide routes may be "static" or "dynamic." Static routing assigns routes to a session at the time of session activation. Static routes are determined based on the network topology and average network loads. Dynamic routing encompasses adaptive techniques that determine the route for a message as it traverses the network, based on some criterion such as response time or resource utilization.

The routing strategy of SNA was developed to satisfy the following requirements:

1. Alternate routing should be possible for reliability and availability reasons.
2. Routing should be loop-free.
3. Routing should avoid protocol deadlocks, or indefinite waits, caused by two nodes waiting on each other for some request or response.
4. Routing should allow load splitting among more than one available route to improve resource utilization.

The primary advantage of dynamic routing is its adaptability to changes in network conditions, such as the traffic pattern and network topology, which can reduce the network transit time of message units and level the traffic over the links throughout the network. However, there are some inherent problems with dynamic routing. A distributed dynamic routing scheme, such as the one for the ARPA (Advanced Research Projects Agency) network,[7,8]

requires frequent updates throughout the network of tables defining the minimal delay routes to each destination node. This imposes additional processing overhead on the nodes and links. Because of frequent updates of routing tables, dynamic routing may also lead to traffic oscillations where links alternately experience high and low loads. Unless otherwise protected, some messages may traverse in a loop, strictly as a result of oscillating dynamic routing changes.

Finally, the requirements for a commercial network may be more complex because of additional routing constraints and requirements, such as for security considerations. Additional processing may be required to fulfill these needs in a dynamic routing environment. In a recent study, Rudin and Müller[9] have shown that fixed routing with end-to-end flow control provides better network performance (such as by reducing message delay) at high loads than adaptive routing with end-to-end flow control. They use certain analytic and simulation results for a ten-node network to support their conclusion.

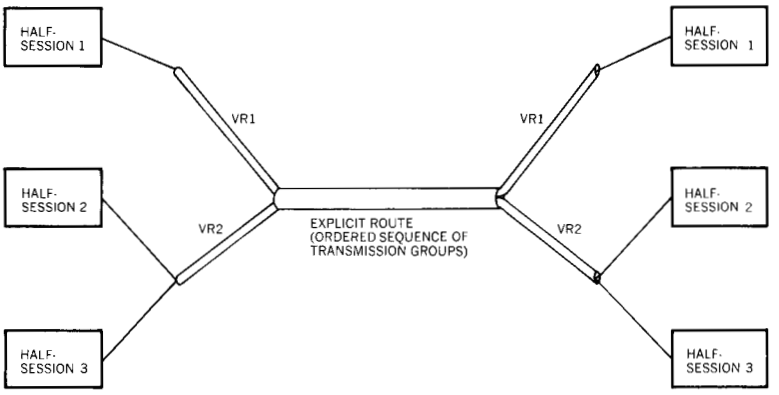
SNA routing satisfies the four requirements previously listed with a modified static routing strategy, based on explicit routing.[10] Up to eight routes are allowed between a pair of nodes. When a session is initiated, a list of virtual routes (described later) is specified. The session is assigned the first available (operational) route from the list. Should a session be disrupted due to failure of a node or link on the route, the user can reinitiate the session on another route. Routes are checked for loops at route activation time. Details of this checking and reinitiation, along with some observations on protocol deadlocks, are described next.

SNA routing

SNA internode routing functions are performed primarily in path control. (The PUs update and reinitialize the routing tables and exchange status information about the routes.) Path control also provides other functions, such as segmentation and assembly,[4] that are not discussed here.

In order to describe the routing architecture, several definitions are required. A group of one or more concurrently operating links between two adjacent subareas is called a *transmission group*. (See Reference 6 for details on transmission groups.) An *explicit route* (ER) is an ordered sequence of transmission groups. It is denoted by the subarea addresses at the two ends and an explicit-route number. There may be up to eight ERs between any two given subarea nodes. A *virtual route* (VR) is a full-duplex connection between two subarea nodes. It is denoted by the subarea addresses on the two ends of the VR, a virtual-route number, and a transmission priority. The transmission priority determines the order of message transmission on a transmission group (see Reference 6 for details). Each VR must use an underlying ER, and

Figure 4 Half sessions, virtual routes, and an explicit route



there may be one or more VRs using the same ER. Since there are eight virtual-route numbers and three transmission priorities, up to 24 VRs may exist between two subarea nodes. Several sessions may use the same VR simultaneously. Figure 4 illustrates the relationship of the above entities. VR1 and VR2 use the same ER.

The separate constructs of ERs and VRs provide flexibility in managing the routes. VRs provide an end-to-end control, whereas ERs provide the physical representation of the route. Furthermore, the network traffic flow control is exercised at the VR level. Thus, sessions using the same ER can be grouped into different VRs, thereby permitting different flow control and transmission priority options to be exercised on each VR.

A PIU is routed through the network as follows. The explicit-route number, a virtual-route number, transmission priority, and other fields are included in the *transmission header*. (The transmission header for this most recent release of SNA is called Format Identifier Type 4, or FID 4.) The routing table in a node is accessed by providing the destination subarea address and an explicit-route number. The table entries furnish the (adjacent) subarea address and the transmission group number to which the PIU is to be routed.

SNA routing RUs

Several commands (SNA Request Units, or RUs) are required to activate, deactivate, test, and communicate the status of routes between the nodes. These are summarized in Table 1. Some requests are relayed by each PU along the route, rather than being sent directly from route end point to end point through the intervening path control elements. The Explicit-Route Operative (or ER_OP) and the Explicit-Route Inoperative (or ER_INOP)[11] requests are sent, respectively, after the bring-up or take-down of a transmission group in the network. ER_OP and ER_INOP identify the ERs using the transmission group, and flow from node PU to

Table 1 Routing commands in SNA

Command	*Flow**	*Action*
ER_OP/INOP	Node to adjacent node	Bring-up/take-down of links (or nodes) in ERs, to update the routing tables in the nodes on the ERs
ER_ACT	Node to adjacent node	Activates an ER
ER_TEST	Node to adjacent node	Tests an ER
ER_ACT_REPLY	Node to adjacent node	Confirmation of ER activation
ER_ACTIVATED	End node to end node	Three-way ER activation exchange
ACTVR	End node to end node	Activates a VR
DACTVR	End node to end node	Deactivates a VR

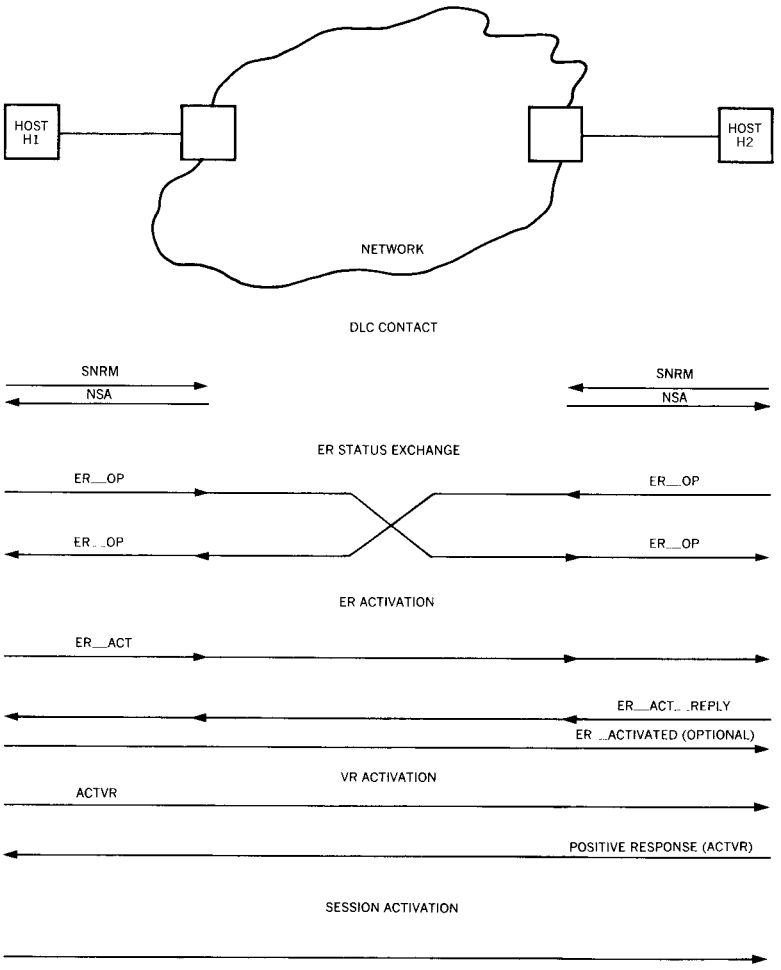
*"Node" in this column refers to "Subarea Node".

adjacent PU, to update the ER status. Given that the appropriate transmission groups are operational, an ER is activated using the Explicit-Route Activate, or ER_ACT, request. The ER_ACT request updates the routing tables of the nodes on the ER. The Explicit-Route Test, or ER_TEST, request is used to test an explicit route. An ER may be tested to determine its operational status by checking the entries in the routing tables of each node in the route. The ER_ACT and ER_TEST requests also ensure loop-free routes by verifying the routing tables at each node along the route. The Activate Virtual Route (or ACTVR) and Deactivate Virtual Route (or DACTVR) requests cause activation and deactivation, respectively, of a virtual route. A virtual route is activated only if its underlying explicit route is active. A virtual-route control block is allocated for the duration that the VR is active. (The control block maintains information on the VR, such as the explicit-route number of its underlying ER.) Once a VR is active, it can be assigned to sessions carrying PIU traffic.

sequence for route activation

We can now describe the sequence of requests to activate a virtual route. Consider two hosts connected through one or more nodes (Figure 5). First the links of the transmission group in the explicit route are brought up using a data link control contact procedure, such as the SNRM, NSA sequence shown in Figure 5.[2,3] When the first link of the transmission group has been activated, ER_OP exchange among the nodes is completed. Next, a node that requires the VR (say Host H1) sends ER_ACT to Host H2. (The ER_ACT flows from one PU to the next on the route.) Assuming a positive ER_ACT_REPLY is received, Node H1 sends ACTVR. If Node H2 has made a request for ER_ACTIVATED (described below), Node H1 sends the ER_ACTIVATED request to

Figure 5 Exchange sequence to activate a virtual route



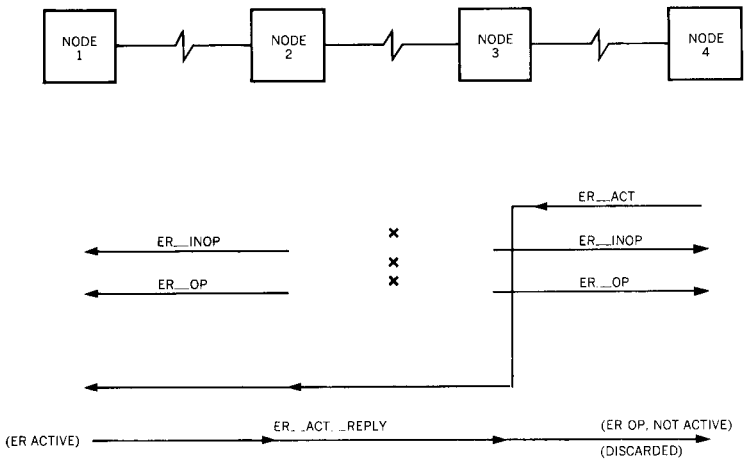
Node H2. A positive response to ACTVR results in activating the VR and allocating a virtual-route control block. One or more sessions can then be initiated on this VR.

prevention of protocol deadlocks

Network protocols should prevent deadlocks that result from two nodes waiting indefinitely for each other to send some request or response. Deadlock avoidance was a key consideration in developing multiple active route protocols. Here we describe an indefinite wait, or "hang" condition, and the resulting three-part exchange required to prevent it.

The synchronization of states at the two ends of a route may be incomplete because of network delays in exchanging requests.

Figure 6 ER_ACT, ER_ACT_REPLY: A hang condition (see Figure 7 for its resolution)

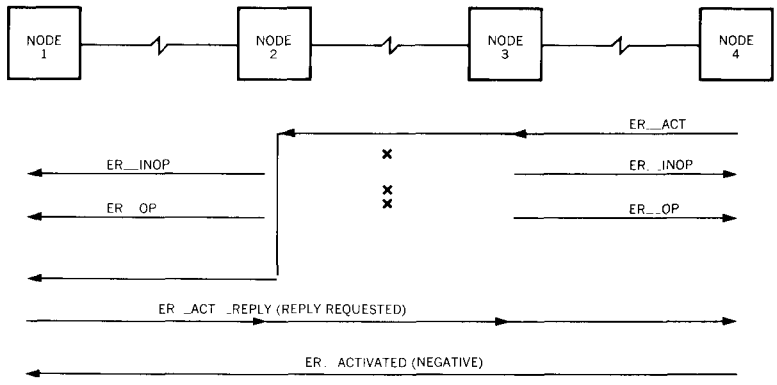


One problem for protocol design is to consider the network delays carefully, so as to avoid deadlocks that may result from other events (and requests) occurring during the interval that requests are being transmitted between the end points.

For example, an ER_ACT and ER_ACT_REPLY are exchanged to activate an ER between two end subarea nodes. However, the exchange may lead to an indefinite wait as shown in Figure 6. Node 4 sends ER_ACT to Node 1. In the meantime, the transmission group between Nodes 2-3 fails, and ER_INOP and (assuming that the link recovers quickly) ER_OP are sent. ER_ACT from Node 4 reaches Node 1. Node 1 sends a positive ER_ACT_REPLY, since it has received an ER_OP. On receiving ER_ACT_REPLY, Node 4 determines that it has not sent an ER_ACT, since its earlier ER_ACT request has been reset by the ER_INOP and ER_OP sequence. So Node 4 discards the ER_ACT_REPLY. Now the ER in Node 1 is hung, since Node 1 assumes that the ER is active and Node 4 does not. This condition has resulted from the intervening ER_INOP and ER_OP, while an ER_ACT was in transmission on the explicit route. In this case, a three-part exchange is needed so that Node 4 should also, besides discarding the ER_ACT_REPLY, send an additional request informing Node 1 of the state of Node 4's ER. This is shown in Figure 7 and described below.

The node building the ER_ACT_REPLY requests that ER_ACTIVATED be sent by Node 4. Node 4 receives ER_ACT_REPLY. Since its ER is in an operational state (and is not expecting an ER_ACT_REPLY), it sends a negative ER_ACTIVATED. So, Node 1 resets its ER to the inactive state. This exchange prevents the deadlock or hang condition shown in Figure 6.

Figure 7 Resolution of ER_ACT, ER_ACT_REPLY hang condition of Figure 6



class of service

The preceding description addressed the requests and corresponding flows that provide multiple routing in SNA networks. Next, we briefly describe the class-of-service interface to users of virtual routes.

Certain sessions (such as those for interactive traffic) may require faster network service than other sessions (such as those for batch traffic). The service should depend on the transmission groups (or bandwidth) and transmission priority (which influences network delays) through the network. So, an ordered list of virtual routes is provided for each class-of-service name. To begin with, at session activation time, the user specifies by name (to avoid awareness of details of the underlying physical routes), explicitly or implicitly, the class of service for the session. The node receiving the session activation request attempts to assign a VR, beginning from the first entry in the list of virtual routes defined by the class-of-service name. If the VR is not active, its activation is accomplished by sending (if not already sent) ER_ACT and receiving a positive ER_ACT_REPLY. An ACTVR is then sent, as described earlier. The session activation request from a user is rejected if no VR can be activated from the virtual-route list for the specified class-of-service name.

Several sessions may simultaneously use the same VR. The VR is deactivated when there is no session using it, thereby freeing up the VR control block storage.

Flow control

Networks consist of finite resources. An unrestricted flow of PIUs into the network can create a condition such that its resources are

overloaded. A network is said to be congested when it has unusually long delays due to the imposition of more traffic than it can handle. A mechanism that regulates network flows to avoid congestion has generally been called *flow control*.

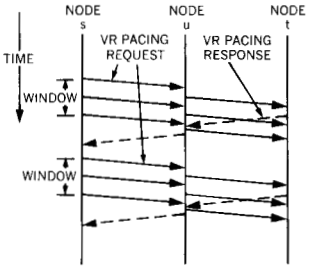
The subject of flow control has been addressed by several authors.[8,12-16] Flow control algorithms can operate on either global or local levels. A global flow control algorithm provides a coordinated, network-wide mechanism to prevent congestion. Local flow control algorithms operate within individual nodes, managing the network traffic through each node separately, based on information local to the node. We confine our discussion to the global flow control mechanisms.

A global flow control must:

1. Evenly distribute traffic among subarea nodes and transmission groups to avoid some nodes getting overloaded relative to others.
2. Restrict PIUs that cannot be delivered within an expected time from entering the (path control) network.

Global flow control algorithms are classified as centralized or distributed.[12] Centralized flow control operates by collecting network-wide traffic information at a central node. The central node then determines flow assignments for each node, and transmits them accordingly. A centralized flow control scheme imposes network overhead for various control messages to and from the central node. It also requires providing a backup central node for availability. An important drawback of centralized flow control is the inherent delay in providing information on the new flow assignments. Such information may be outdated by the time it arrives at the affected nodes.

Figure 8 End-to-end flow control

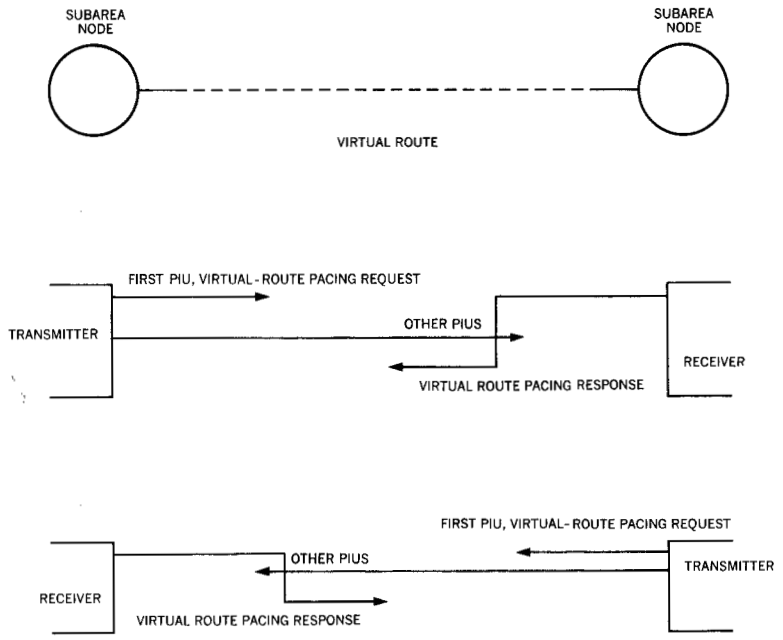


Two types of distributed flow control strategies are in use. The isarithmic flow control, as described by Davies,[14] attempts to restrict the total number of message units in the network at any time. The end-to-end flow control[17] attempts to restrict the number of message units in a "virtual connection" (similar to a virtual route in SNA) by employing control at its two ends. Both schemes are essentially similar since each attempts to enforce a limit on the total number of message units in the network. For SNA, an adaptive end-to-end flow control is exercised on each virtual route, as described next.

VR flow control architecture

The VR end-to-end flow control mechanism is illustrated in Figure 8. Nodes s and t are the two end points of a virtual route; the window size is the maximum number of PIUs that the sender can transmit in a group, or "window," of PIUs, following permission to send the window. Node s seeks permission to send the next

Figure 9 PIU flow on a virtual route



window by setting a pacing request bit in the transmission header of the first PIU of a window destined for Node t. A pacing response from Node t permits Node s to send another window of PIUs after completing the current window. Node s repeats this process for each window that it sends: requesting permission in the first PIU of the current window to send the next window, and awaiting the pacing response before sending the next window.

An adaptive scheme exists for dynamically adjusting the window size within specified limits. On a given virtual route, pacing control and a corresponding window size exist for each direction of flow, as shown in Figure 9. The window size may vary between a minimum of h to a maximum of $3h$, where h is the number of transmission groups in the underlying ER. The minimum and maximum limits of window size, k, are communicated by including these values in the ACTVR request. To facilitate description of the adaptive window algorithm, the following terms are defined:

1. VR Pacing Response: Permission from the receiver to send k more PIUs. A prior request for permission was received by the receiver from the transmitter. The receiver withholds the VR pacing response if there are no buffers to receive PIUs.

The following bits are included in the transmission header and turned on by the window receiver to control the PIU flow from the transmitter.

2. CWRI (Change Window Reply Indicator): A bit to indicate a change in window size, k, by an amount of one without violating the minimum or maximum limits specified by ACTVR. If this bit is on, the window size may be decreased; otherwise it is increased.
3. RWI (Reset Window Indicator): A bit to indicate a reduction in window size to the minimum window size specified by ACTVR.

The adaptive window algorithm:

algorithm

1. Initially, k is set to the minimum value, h. After ACTVR exchange, start with a window size of h.
2. The window size is changed at the transmitter as follows.
   (a) $k \leftarrow h$, if RWI is on. Any node on the route can turn this bit on if severe congestion exists in the network. On receiving this bit set to one, the transmitter resets the window size to the minimum window size of h.
   (b) $k \leftarrow k + 1$, $k \leq 3h$, if the bit CWRI is off and there are PIUs waiting for transmission.
   (c) $k \leftarrow k - 1$, $k \geq h$, if CWRI is on. Any node on the route can make a request for this bit to be turned on if it is moderately congested. ("Moderate" and "severe" congestion are implementation-defined. "Severe" congestion would result in the RWI bit being set on.) On receiving the CWRI bit set to one, the transmitter decreases the window size by an amount of one, without going under the minimum window size. Setting CWRI or RWI is the means of adaptively reducing window sizes in a network that is experiencing moderate or severe congestion.

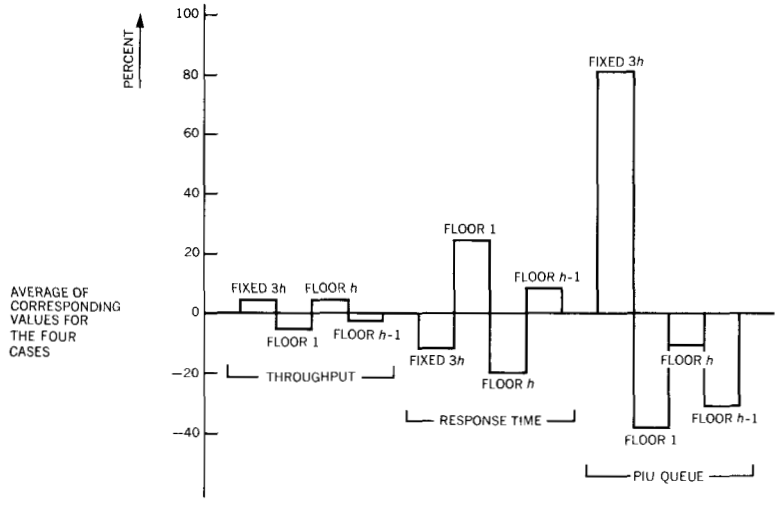
   The criterion for setting CWRI or RWI is based on two factors: (1) the number of free buffers and (2) the size of the message unit queue in the node on this route.

performance

Networks without any virtual-route flow control are exposed to poor performance as reported by Deaton.[18] Where flow control is achieved via end-to-end VR pacing, the window size may be fixed or dynamically changed according to an adaptive algorithm. In the case of an adaptive window size, various window size limits are possible.

Torrey and Ledford[19] considered a network consisting of host processors and communications controllers. Virtual routes with up to four hops were considered. Traffic was equally distributed over all virtual routes, and PIU sizes of 25 or 2048 characters were selected. Figure 10 provides the comparative performance for four cases (h is the hop count):

Figure 10 Comparative performance of various window size algorithms



1. Window size fixed at $3h$.
2. Dynamic window size within the limits of 1 and $3h$.
3. Dynamic window size within the limits of h and $3h$.
4. Dynamic window size within the limits of $h - 1$ and $3h$.

The above four cases are respectively labeled Fixed $3h$, Floor 1, Floor h, and Floor h-1 in Figure 10. The throughput is measured in number of PIUs transmitted per unit of time. The response time is the round-trip delay through the network. The PIU queue is the average queue size of PIUs in the nodes. The results are normalized to an average value in each case. The throughput for the Floor h case is the same as that for the fixed window size case. However, the Floor h case offers the minimum response time. Although Floor 1 and Floor h-1 offer less congestion, Floor h is generally preferable because of its significant reduction in response time, a result that seems to hold for a range of reasonable settings of the network parameters.[19]

Summary

Networks with multiple, interconnected nodes have imposed specific routing and flow control requirements on SNA. Such networks also provide the opportunity to use multiple active routes. The routing architecture of SNA employs the concepts of physical paths (explicit routes) and logical connections (virtual routes) to exploit this opportunity.

Network congestion is reduced by local and global flow control mechanisms. An end-to-end VR pacing algorithm is used to pro-

vide global flow control; the window size is changed to adapt to network traffic and available buffer space.

ACKNOWLEDGMENTS

Numerous individuals have contributed to the routing and VR flow control architecture of SNA. I acknowledge the contributions of S. D. Dilley, F. McGriff and G. Young of NCP; J. H. Benjamin, J. G. Knauth, and B. J. Heldke of VTAM; F. D. George and W. B. Jackson of TCAM; K. Maruyama and M. Reiser of Research; and J. P. Gray, M. W. Doss, D. P. Ledford, G. A. Deaton, E. G. Huff, and J. C. Torrey of the Communications Systems Architecture group. I also acknowledge the contributions of G. D. Schultz, J. P. Gray, and E. H. Sussenguth in reviewing this paper and suggesting improvements.

CITED REFERENCES AND NOTE

1. T. F. Piatkowski, D. C. Hull, and R. J. Sundstrom, "Inside IBM's Systems Network Architecture," Special Report, *Data Communications*, 34-48 (February 1977).
2. J. P. Gray, "Network Services in Systems Network Architecture," *IEEE Transactions on Communications* **COM-25,** No. 1, 104-116 (January 1977).
3. R. J. Cypser, *Communications Architecture for Distributed Systems*, Addison-Wesley Publishing Co., Reading, MA (1978).
4. *Systems Network Architecture, Format and Protocol Reference Manual: Architecture Logic*, SC30-3112; available through the local IBM Branch Office.
5. J. P. Gray and C. R. Blair, "IBM's Systems Network Architecture," *Datamation* **21,** No. 4, 51-56 (April 1975).
6. J. P. Gray and T. B. McNeill, "SNA multiple-system networking," *IBM Systems Journal* **18,** No. 2, 263-297 (1979), this issue.
7. J. M. McQuillan and V. G. Cerf, *A Practical View of Computer Communication Protocols*, Notes distributed at the ACM Professional Development Seminar, Washington, DC (April 1978).
8. R. E. Kahn and W. R. Crowther, "Flow control in a resource-sharing computer network," *IEEE Transactions on Communications* **COM-20,** No. 3, 539-545 (June 1972).
9. H. Rudin and H. Müller, "On routing and flow control," *Proceedings of International Symposium on Flow Control in Computer Networks*, Versailles, France (February 12-14, 1979).
10. R. R. Jueneman and G. S. Kerr, "Explicit path routing in communication networks," *Proceedings of Third International Conference on Computer Communications*, Toronto, 340-342 (August 1976).
11. ER_INOP replaces a similar SNA 3 request called Lost Subarea (or LSA).
12. V. Ahuja, "On congestion problems in communication networks," *Proceedings of Trends and Applications: 1978, Distributed Processing*, Washington, DC, 40-46 (1978).
13. E. Raubold and J. Haenle, "A method of deadlock-free resource allocation and flow control in packet networks," *Proceedings of the ICCC*, 483-487 (1976).
14. D. W. Davies, "The control of congestion in packet-switching networks," *IEEE Transactions on Communications* **COM-20,** No. 3, 546-550 (June 1972).
15. G. A. Deaton and D. J. Franse, "A computer network flow control study," *Fourth International Conference on Computer Communications*, Kyoto, Japan, 135-140 (1978).
16. V. Ahuja, "Algorithm to check network states for deadlock," *IBM Journal of Research and Development* **23,** No. 1, 82-86 (January 1979).

17. L. Kleinrock, *Queuing Systems, Volume 2: Computer Applications*, John Wiley & Sons, Inc., New York (1976).
18. G. A. Deaton, "Flow control in packet-switched networks with explicit path routing," *Proceedings of International Symposium on Flow Control in Computer Networks*, Versailles, France (February 12–14, 1979).
19. J. C. Torrey and D. P. Ledford, unpublished work.

Reprint Order No. G321-5097.