Demand Priority Protocol

In multiple-hub networks, demand priority ensures fairness of access for all nodes and guarantees access time for multimedia applications.

by Alan R. Albrecht, Michael P. Spratt, Patricia A. Thaler, and Gregory C. A. Watson

Demand priority is the media access control protocol defined in the IEEE 802.12 draft standard. It is called 100VG-AnyLAN by HP. Various techniques are used to ensure fairness of access for all nodes and to guarantee access time for individual nodes. Round-robin selection procedures are used to give each node an equal opportunity to transmit data. Two priority levels are provided so that time-critical traffic such as interactive video, audio, and multimedia can be given priority service with guaranteed low delay. A bandwidth allocator can be introduced to control the amount of bandwidth each application can use.

These techniques make demand priority able to accommodate multimedia applications with guaranteed performance levels, while allowing normal traffic to use the remaining bandwidth in a fair manner.

Hubs

100VG-AnyLAN networks are centered around the concept of intelligent hubs (called repeaters in the IEEE 802.12 draft standard), each of which is at the center of a star configuration. A hub has several local ports, which are connected to individual nodes, which can be workstations, servers, bridges, routers, or lower-level hubs. A hub may also have one cascade port for connection to a higher-level hub.

Network Control

Control in a demand priority network is based on a request/grant handshake between the node and the hub. Each node needing to send a frame indicates its need to transmit by sending a request signal to the hub and waiting for the hub to grant it permission to transmit the frame. The concept can be seen as a sophisticated and flexible successor to the token in token ring networks; it ensures that collisions cannot occur in the network.

Hubs use a round-robin node selection procedure to ensure a fair opportunity for nodes to transmit data. The purpose is to ensure that no node can send two successive normal-priority frames until all other nodes have also had an opportunity to send a frame at normal priority.

Round-Robin Node Selection

The nodes are numbered in order around the entire network. The hub maintains a round-robin pointer that indicates which node is next to receive an opportunity to transmit. For example, if nodes 2, 3, 5, and 9 have indicated that they have packets to send and the round-robin pointer currently points to 3, node 3 gets to transmit. The pointer then moves on to 4. Assuming node 4 still has no data to send, node 5 will be selected next. The pointer will then move on to 6. When the pointer reaches the last node, it cycles back to 1.

The normal-priority selection cycle will be temporarily suspended when any node lodges a high-priority request. A separate round-robin pointer is maintained for high-priority requests, so that they too are treated fairly. The decision as to which node is next is made immediately after a transmission has finished. The hub looks at all requests that have been received since it last made a selection. Earlier requests are not stored so that spurious requests caused by noise or previous requests from a node that has changed its mind do not cause a false selection.

Typical Transaction

A typical transaction within a network using the demand priority protocol could run as follows. The example is for a network with a single hub, using 4-pair UTP cables. 4-pair UTP links require all four pairs to be used for data transmission in either direction, but dedicate two pairs in each direction for control when data is not being sent. Fig. 1 illustrates each step in the transaction.

While there is no data to be transmitted, all nodes send IDLE to the hub, and the hub returns IDLE to each node.

Suppose that node 1 has a frame to transmit to node 3 alone. It sends a request (REQ_N for normal priority, REQ_H for high priority) to the hub.

After referring to its round-robin pointer and checking for any other request, the hub decides to accept transmission from node 1. It indicates the selection by ceasing its IDLE transmission to node 1 (indicating a grant). This clears the outgoing signal from the hub and readies the link to receive data on all four pairs. Simultaneously, the hub also changes the control signal being sent to all other nodes from IDLE to INCOMING. (At this stage, the hub has no idea of the destination of the data packet.)

The other nodes respond to INCOMING by ceasing transmission to the hub. This clears their outgoing control signal so that they are prepared to receive the packet. The source node (node 1) starts transmitting its packet to the hub on all four pairs of wires as soon as it detects that the IDLE control signal has gone silent.

The hub decodes the beginning of the packet to determine the destination node or nodes. It finds that, in this case, the packet is unicast (destined for one node only) for node 3. It immediately starts sending the packet to node 3 and changes the control signal to all other nodes to IDLE.

Transmission continues until it is complete. Meanwhile, nodes not involved in the transfer can send REQ if they have packets to send. The hub notes such requests, but does not yet select the next node to transmit. When the source node (node 1) completes its transmission, it sends IDLE, or another REQ if it has another packet to send. After the hub has finished forwarding the frame, it selects the next node to transmit according to the round-robin selection procedure.

There is a possibility that a node that has received several consecutive packets and needs to make a transmission of its own might miss its turn in the round-robin cycle. This is because at the instant when the hub selects the next node to transmit, the receiving node is still receiving the end of the last packet. To overcome this, receiving nodes needing to transmit ignore the INCOMING control signal for a short period (called the *request window*) during which they submit their own request. The request window is shorter than the minimum gap between successive packets.

If the packet has a multicast address (it is destined for more than one node), the hub may need to wait until it has received the entire packet before transmitting it to the destination nodes. This is to avoid cross talk problems in 25-pair bundled cable. The article on page 18 provides more detail on 4-pair UTP links and the use of bundled cable.

In a network using 2-pair STP or fiber-optic cables, packets are transmitted down one fiber or pair of wires by multiplexing the four data streams. This means that the other fiber or wire pair is always available for lodging requests to the hub. See the articles on pages 18 and 27 for an expanded discussion of STP and fiber links.

Networks with Cascaded Hubs

A network can have several hubs in a tree structure known as a cascade (see Fig. 1 on page 8). One hub is designated the *root hub*. All other hubs will each have a connection to a next higher-level hub through their cascade port. They may also have lower-level hubs connected to some of their local ports.

Where there is more than one hub in a network, at any moment exactly one of them has *control* of the network activity. A hub is said to have control if it is selecting the next node to transmit. When the entire network is idle, the root hub has control.

If a network consists of a cascade of hubs, the round-robin pointers effectively treat all end nodes as if they were connected to a single hub, so that all nodes have an equal opportunity to transmit, however far they may be from the root hub. Fig. 2 shows the order of node numbering in a cascaded network.

A different control signal is needed when a hub receives a high-priority request while another hub is servicing normal-priority transactions. Suppose hub A is servicing normal-priority requests when hub B receives a high-priority request (see Fig. 3). Hub B sends the high-priority request up the cascade of hubs until it reaches hub X, which earlier passed control to hub A. Hub X then sends an ENABLE_HIGH_ONLY signal to hub A.

The ENABLE_HIGH_ONLY signal tells hub A to suspend its round-robin sequence at the end of the current transmission and return control to hub X. At the same time hub A tells hub X whether it had finished its normal-priority round-robin cycle. Hub X then passes control to hub B to service the high-priority request. When that is complete, assuming that hub A had not finished its portion of the round-robin normal-priority selection cycle, and assuming that no other high-priority requests have been received by hub X, control is passed back to hub A.

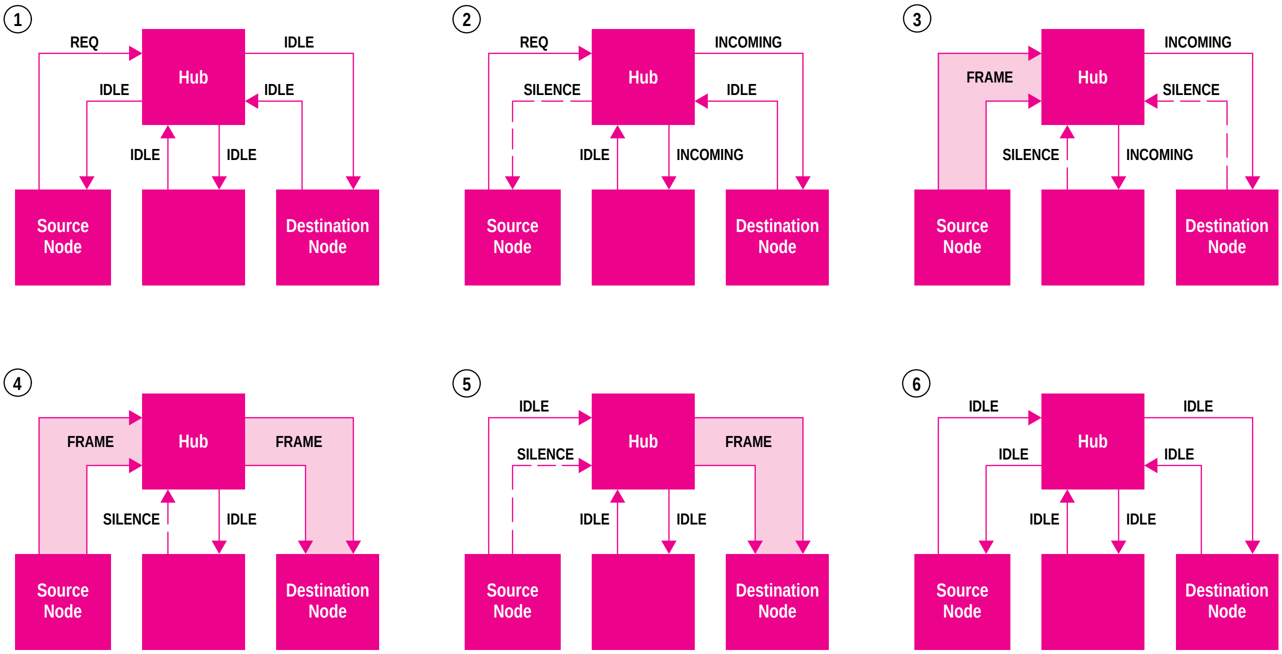

Fig. 1. The stages in a typical demand priority transaction.

Network Protocol Layers

The Open Systems Interconnection (OSI) Reference Model of the International Organization for Standardization (ISO) describes the structure of networks. It defines the seven layers shown at the left side of Fig. 2 on page 9. The lower three layers are relevant to the demand priority protocol.

Layer 3: Network Layer (NL). The network layer is responsible for passing a packet of data through an internetwork, which can consist of many individual LANs and even wide area links.

Layer 2: Data Link Layer (DLL). The data link layer provides the transmission of data between two nodes on the same network. It receives a packet from Layer 3, the network layer, and adds source and destination addresses and other information to it. The DLL consists of two sub-layers: the logical link control (LLC) sublayer and the media access control (MAC) sublayer.

The LLC sublayer links the network layer to the MAC. There are standard LLCs, enabling different protocols to link successfully. In 100VG-AnyLAN, the LLC can be either an IEEE 802.2 Class I LLC, supporting Type 1 unacknowledged, connectionless-mode transmission, or a Class II LLC, supporting Type 2, connection-mode transmission.

When a frame is ready for transmission in an end node, it is sent from the LLC sublayer to the MAC sublayer where the appropriate Ethernet or token ring MAC frame is built. The frame is then passed to the physical layer's PMI (physical medium independent) sublayer.

When a data packet is received from the physical layer, the MAC sublayer reassembles the MAC frame and performs various checks for errors in the received frame. Only valid frames are sent on to the LLC sublayer in an end node.

Frames received by the hub's RMAC sublayer are forwarded to the addressed destination (if it can be determined) and to all promiscuous ports regardless of error condition (there is no LLC in a hub). Frames containing errors are marked with an invalid packet marker.

Layer 1: Physical Layer (PHY). The physical layer defines the procedures and protocols associated with the physical transmission of bits (such as cable interfaces, data signal encoding, and connector types and pinouts). It consists of two sublayers: the PMI sublayer and the physical medium dependent (PMD) sublayer.

The PMI sublayer includes provisions for quartet signaling, data ciphering, 5B/6B encoding, the addition of preambles, and start and end frame delimiters (SFD and EFD). See the articles on pages 18 and 27 for details.

During transmission, the PMI sublayer accepts data from the MAC sublayer and prepares the packet for transmission. It converts the octet data into quintets which are separated into four streams. In each stream, each quintet is ciphered and then encoded as a 5B/6B sextet. The PMI adds physical layer headers and trailers to each data stream.

When receiving data packets, the PMI removes physical layer headers and trailers and then passes the packet onto the MAC layer. It decodes each received 5B/6B sextet, deciphers the resulting quintet, merges the four deciphered quintet streams, and converts the result into a single octet stream for delivery to the MAC sublayer.

The PMI sublayer connects with the PMD sublayer through the medium independent interface (MII).

The PMD sublayer includes channel data packet multiplexing (for 2-pair STP and fiber-optic cabling only), NRZ encoding, link medium operation, and link status control. It connects with the physical medium (the cable) through the medium dependent interface (MDI).

Link Initialization—Joining the Network

When a node first joins the network, a handshake training sequence occurs between it and the hub, during which the node sends the hub its 48-bit MAC address. It also sends other information, such as what type of frame it will use (IEEE 802.3 or IEEE 802.5) and if it wants to receive all packets of data whether addressed to it or not (promiscuous mode). The hub accepts or rejects this information. For example, the hub might be configured by the network administrator to reject promiscuous mode to preserve a high level of data privacy.

Guaranteed Performance

The two priority levels make it possible to guarantee bandwidth to applications and to keep the access delay (the time for which nodes may have to wait before being allowed to transmit) within bounds.

Bandwidth and access delay depend on the size of the network and the way it is configured. Nodes using high-priority traffic may need to be configured so that the amount of bandwidth they can use is restricted, but nodes using only normal-priority traffic will operate totally unaware that a high-priority service is being provided to other nodes.

Bandwidth

Without any other form of control, nodes wishing to send high-priority traffic will automatically be allowed to send an equal number of high-priority frames. Any bandwidth not used by this high-priority traffic is then automatically shared

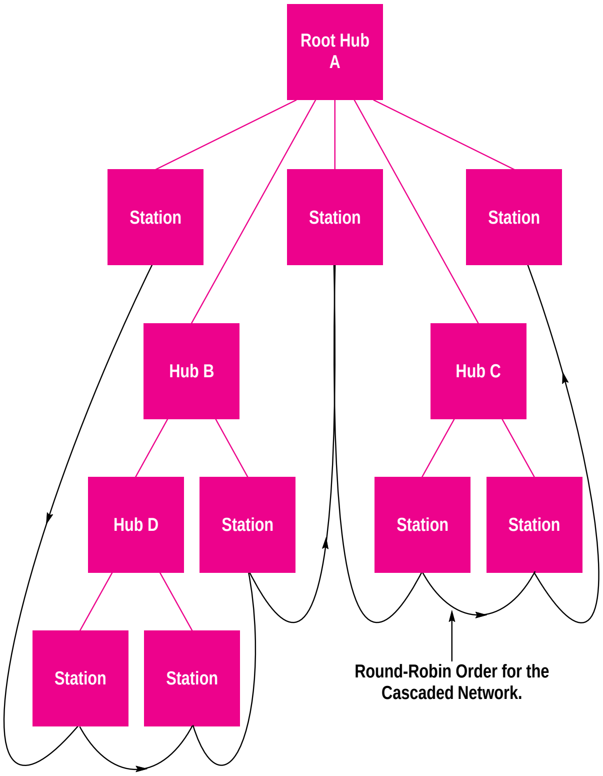

Fig. 2. The numbering of nodes in a cascade of hubs.

between all nodes wishing to send normal-priority traffic. Clearly, where there is excessive high-priority traffic, the bandwidth allowed for high-priority requests needs to be restricted in some way so that normal-priority traffic is never completely stopped. Ways of doing this are described in the article on page 33.

Access Delay

The deterministic behavior of the round-robin selection procedure means that it is easy to establish a reasonably accurate estimate of the maximum access delay that a high-priority packet will experience. For a hub with n nodes using high-priority traffic, the worst-case delay will be nT, where T is the time it takes to transmit the largest possible frame. It is not (n − 1)T as might be expected, because the worst case occurs when a high-priority request is received from all nodes just at the moment when the hub starts to service a normal-priority request. For example, the worst-case delay for a hub with 32 nodes forwarding 1500-byte IEEE 802.3 frames is 4 ms, or for 4500-byte IEEE 802.5 frames, 12 ms.

Access delay for a normal-priority request is more difficult to calculate, but the performance data presented in the next section shows what delays might be expected under heavy load.

Performance in a Simulated Example

Various simulations have been run to examine the effect on high-priority traffic as normal-priority traffic increases. Each simulation ran for 100,000 frames, and the results were taken from the central portion of the run to avoid anomalies that occur at startup and shutdown.

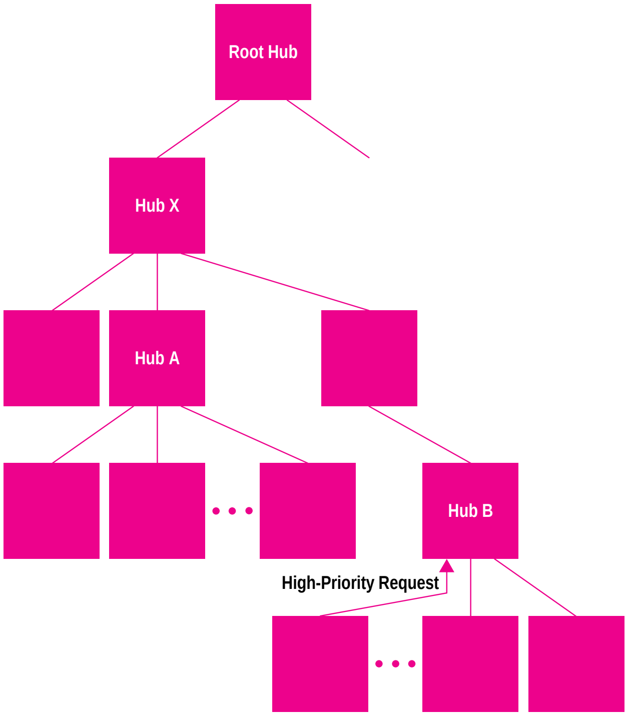

Fig. 3. In a cascade of hubs, a hub (hub B) may receive a high-priority request while another hub (hub A) is servicing normal-priority transactions. Hub B sends the high-priority request up the cascade of hubs until it reaches hub X, which earlier passed control to hub A. Hub X then sends an `ENABLE_HIGH_ONLY` signal to hub A.

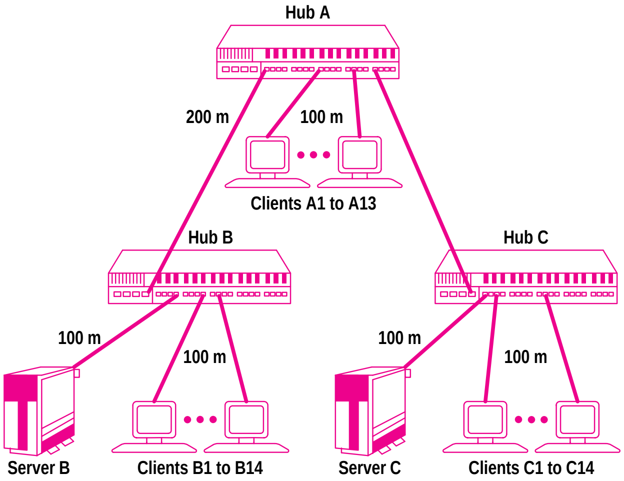

Fig. 4. Network configuration used in the simulated example.

The network configuration used for one of the simulations is shown in Fig. 4. All fifteen ports are used in each of the hubs. The hubs are connected to each other by 200-m links and the end nodes are connected to the hubs by 100-m links. Hubs B and C each have three nodes that are allocated high-priority bandwidth so that each node can send a block of eight maximum-sized frames every 10 ms. The bandwidth assigned to each node is equivalent to about 9.7 Mbits/s, and is substantially more than needed for MPEG-encoded video. The total high-priority traffic is therefore about 58 Mbits/s.

All stations, including the six with high-priority traffic, send some maximum-size frames at normal priority. The simulation measured the time for which nodes waited for access as the number of these normal-priority frames increased.

Fig. 5 shows the mean and maximum access delays that were observed for both high-priority and normal-priority traffic. Access delay is defined as the time a frame spends at the head of its node's transmission queue waiting for access to the network.

It is clear from the graph that the delay that high-priority traffic encountered was almost independent of the amount of normal-priority traffic. Even when the total load on the network was 93 Mbits/s (35 Mbits/s at normal priority in

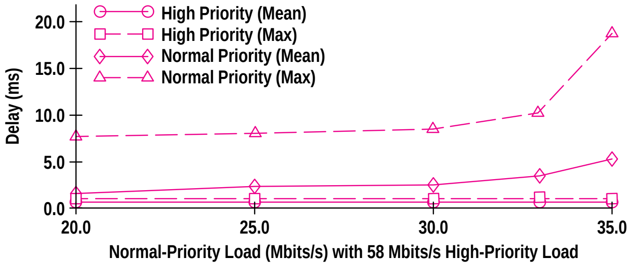

Fig. 5. Access delays in the simulation.

addition to the 58 Mbits/s of high-priority load), high-priority traffic still had a mean access delay of less than 0.5 ms and a maximum access delay of less than 0.8 ms.

These delays are, of course, dependent on the number of stations that have high-priority traffic, but it is clear that a number of high-priority streams can be supported with a guaranteed low delay.

Acknowledgments

We would particularly like to acknowledge the contributions of our fellow team members from Roseville Networks Division: Joe Curcio, Dan Dove, and Steve Goody.

Bibliography

Additional information on the demand priority protocol is contained in the following papers:

1. A. Albrecht, J. Curcio, D. Dove, S. Goody, and M. Spratt, "An Overview of IEEE 802.12 Demand Priority," *Proceedings of GLOBECOM*, 1994.
2. G. Watson, A. Albrecht, J. Curcio, D. Dove, S. Goody, J. Grinham, M. Spratt, and P. Thaler, "The Demand Priority MAC Protocol," *IEEE Networks Magazine*, January 1995.