Extended Frame Sizes for Next Generation Ethernets

A WHITE PAPER

Alteon Networks

THE SERVER SWITCHING COMPANY

With Gigabit Ethernet technology, computer networks can now transfer over one billion bits of information every second. But Ethernet's maximum frame size of 1500 bytes isn't optimized for this new breed of ultra-high speed networks and can actually inhibit the ability of applications to take full advantage of the increased network capacity.

This is equivalent to traveling on a highway where only mid-sized cars are allowed. At each toll station where thousands of cars are processed by an attendant, each car must wait an inordinate amount of time to get through. And as traffic increases, so does the time it takes to get through the toll station. This time could be dramatically reduced if the passengers in cars could pass through the toll station on buses. This would reduce the number of vehicles processed and shorten overall delay for each passenger.

Today's large servers, whether offering Web-based services or traditional data services, may handle tens of thousands of packets per second and drive hundreds of Megabytes per second of application throughput. Collaboration between compute and data servers or backup of server data can result in massive, repetitive data transfers, generating steady, intense traffic.

Today, standard Fast Ethernet and Gigabit Ethernet still keep to the Ethernet frame sizes designed and optimized for the relatively modest data transfer needs of host computers and much slower Ethernet components of 20 years ago. That means these high-speed forms of Ethernet are still limited to exchanging data at no more than 1500 byte chunks, which is far from optimal for today's servers and networks.

Optimal frame size is a function of traffic, network and host characteristics. Bulk transfers and steady, intense traffic tend to benefit from larger frame sizes, as do high-bandwidth, high-reliability networks and more powerful server platforms. With the proliferation of servers and the increasing distribution of applications over multiple collaborating server appliances, continuous transfers of large blocks of data between servers are on the rise. Add to this the overhead fragmenting and reassembling of data into small packets and the cost to process each packet, and it's not surprising that most servers cannot keep up with the new generation of high-speed LANs. Who could have foreseen the massive advances in network and server capacity during Ethernet's early days?

Subsequently, other high-speed networking technologies have tended to better reflect the needs of today's servers and networks. FDDI, for instance, has a maximum transmission unit (MTU) of 4500 bytes. The default MTU for AAL5 over ATM is 9000 bytes, while Fibre Channel and the High Performance Parallel Interface (HIPPI) typically use a maximum MTU of 65280 bytes (theoretically Fibre Channel's MTU is unlimited). But the popularity and installed base of these technologies pales in comparison to Ethernet, putting users in a difficult spot. Ideally users would like marry the best attributes from these niche technologies with standard Ethernet.

Due to cost considerations and the desire to more easily manage homogeneous network topologies, designers favor Ethernet for end system connections, despite the fact that many of today's servers are hampered by a maximum Ethernet frame size that is substantially less than optimal for their needs. This creates a new opportunity to improve server performance and network throughput by extending the maximum frame size in high-speed Ethernet networks.

The Art of Handling Packets

Application performance depends largely on server and network throughputs. Server throughput is primarily a function of the server's processing power and load. Network throughput is directly driven by router and switch efficiency, in addition to the amount of raw bandwidth.

Server Considerations

In heavy traffic conditions, servers send and receive larger frames much more efficiently than smaller ones. The increased efficiency results from the fact that it takes fewer larger frames to transfer the same amount of data than with existing Ethernet packets. As there is a significant amount of fixed processing overhead per frame, processing overhead becomes proportional to the number of frames presented to the system.

Many of today's servers are hampered by a maximum Ethernet frame size that is substantially less than optimal for their needs

Sending and Receiving Packets

Much of the server overhead for transmitting a packet is independent of the size of the packet. For example, parsing and building the packet header takes the same amount of time for a large packet as a small one.

On the receive side, fewer frames means fewer "packet received" interrupts from the network interface card (NIC). In most implementations each time a packet is received by the adapter, it interrupts the host to inform it that: 1) it has received a packet and 2) to stop what it is doing and process the packet. Each of these interrupts consumes a significant number of host processor cycles.

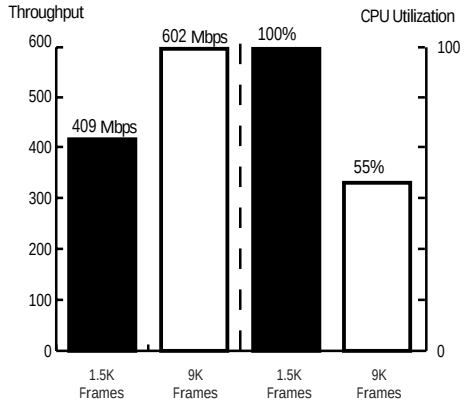
On a lightly loaded server, the added burden on the processor might not matter much. But on a heavily loaded system, dramatic performance improvement are seen when the processor is freed from this constant stream of interrupts. This is particularly important on Fast Ethernet and Gigabit Ethernet networks where servers may be receiving tens of thousands or even millions of packets per second.

Copying Data To/From Host Memory

Extended Ethernet frames also save a lot of host CPU cycles by reducing the number of times servers must move incoming data into memory. On both send and receive operations, memory transfers are more efficient with large packets, due to memory "paging" considerations. Computers organize their memory in "pages," most often of 4 Kbytes (4096 bytes), sometimes of 8 Kbytes or 16 Kbytes.

There is a fixed amount of overhead for transferring any amount of data up to a page. With a system supporting 4 Kbyte pages, an 8000 byte frame would incur only two operations to copy the data from the adapter to the appropriate host memory location. The equivalent amount of data sent using maximum length 1518 byte Ethernet frames requires six host copy operations and thus three times the host CPU cycles.

Figure 1. Extended Ethernet Frames vs. Standard Ethernet Frames*



* Using Gigabit Ethernet. Throughput on tests was limited to SBus capacity. TCP tests used dual 300 Mhz Sun servers running Solaris 2.5.1

Network Considerations

Router and switch efficiency is determined primarily by how much time they spend examining packet headers and determining how packets should be forwarded.

Examining Headers

Overhead for packet header parsing and making forwarding decisions is clearly proportional to the number of packets. Because routers examine many header fields and make complex decisions, larger frames dramatically increase their efficiency.

Headers Consume Network Bandwidth

Headers are the same size for all IP packets, whether big or small. Thus, headers consume proportionally less network bandwidth within larger packets. Though headers are generally small (no more than about a hundred bytes), they can consume a significant percentage of network bandwidth particularly under heavy load conditions where thousands or millions of small packets are being transmitted. Therefore larger packets significantly reduce the amount of raw network bandwidth being consumed.

Performance Improvements

The benefits of large frame sizes on a busy server had been demonstrated in several public performance tests. One of the tests showed a 50 percent reduction in server CPU utilization when using 9018 byte-sized Ethernet packets as opposed to 1518 byte frames while throughput increased by almost 50 percent from 409 Mbps to over 602 Mbps (see

Figure 1). This means not only increased throughput but also more server cycles available to process applications.

Though it's difficult to fully quantify CPU cycles savings associated with the use of extended frames, some approximations can be made.

For a typical server implementation, it takes approximately 1,200 CPU cycles to process the IP and TCP headers of a single Ethernet frame (1,000 machine instructions times 1.2 CPU cycles per instruction). A 9018 byte extended Ethernet frame can carry the payload of six standard Ethernet frames with the overhead of only one Ethernet frame. This saves the host from processing five packet headers and results in a savings of 6,000 CPU cycles (at minimum). For a 10 MB file transfer, this translates into a savings in excess of eight million CPU cycles.

How Large Should Frames Be?

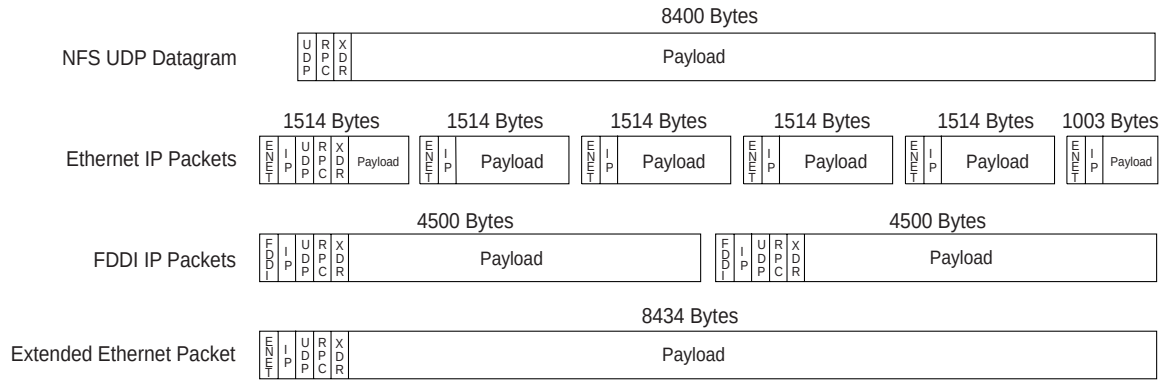
So, the larger the frame size, the better - given plenty of traffic and a very reliable network. Right? Not necessarily. Ethernet error detection techniques put a practical upper limit on frame size. Ethernet adapters, when transmitting frames, insert a 32-bit Frame Check Sequence (FCS) into every packet. The FCS is a number derived mathematically from all the other bits in the frame. When the frame is received, the receiving adapter performs the same mathematical operation on the frame. If this operation does not yield the same four-byte number in the FCS header field, the frame has been corrupted in transmission and is discarded.

With Ethernet, the FCS computation uses a 32-bit cyclic redundancy check (CRC-32). CRC-32 error checking detects bit errors with a very high probability. But as frame size increases, the probability of undetected errors per frame may increase. Due to the nature of the CRC-32 algorithm, the probability of undetected errors is the same for frame sizes between 3007 and 91639 data bits (approximately 376 to 11455 bytes). Thus to maintain the same bit error rate accuracy as standard Ethernet, extended frame sizes should not exceed 11455 bytes.

In addition, an efficient frame size can be derived based on memory page sizes in host computers. Memory page sizes (4 Kbytes, 8 Kbytes, 16 Kbytes) used by the majority of commercial systems make multiples of 4Kbytes plus total header space (1-200 bytes) attractive frame sizes from the point of view of minimizing copy operations. Lastly, an optimum frame size can be selected based on the block sizes used by the most popular applications.

For instance, a network file system (NFS) datagram is 8400 bytes. NFS is the most common file-sharing protocol in UNIX environments. A 9018 Kbyte frame size is attractive by accommodating a single NFS datagram in one Ethernet packet and staying comfortably within the standard Ethernet bit error rates. (see Figure 2).

Figure 2. Efficiency of Larger Ethernet Frames in NFS File Transfer



Compatibility Issues

A major concern in adopting extended frame sizes for high-speed Ethernet is backwards compatibility.

Theoretically, IP protocol stacks and routing entities can usually be configured to support MTUs of up to 64 Kbytes. Applications running on TCP over IP should not have any problems concerning MTU compatibility because the two end stations negotiate a common MTU when the TCP connection is established. The station with the larger MTU "throttles back" and uses the MTU of the other station.

The Pitfalls

However, there may be issues affecting intermediate IP routers. For instance, two end stations might both support extended frames, but an intermediate IP network might not. In that case, an IP router may have to fragment the packets. Fragmentation is undesirable because it places an additional burden on the router and on the receiving station, which has to reassemble the packets.

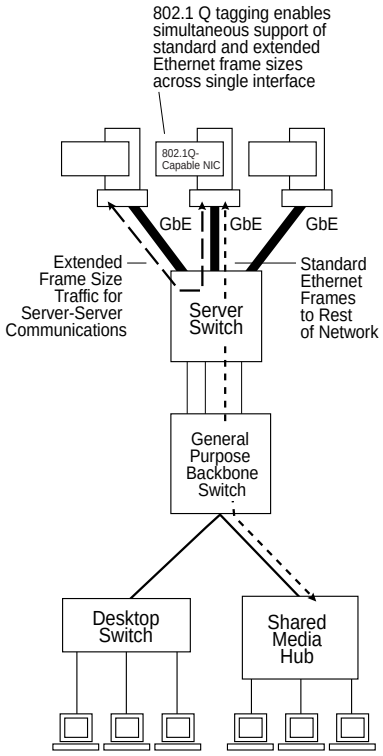
Even worse, if the DON'T_FRAGMENT bit is set in the IP header of the packets, the router will drop the packets instead of fragmenting them. The router usually sends an ICMP DESTINATION UNREACHABLE - FRAGMENTATION NEEDED message to the sending station. This causes the station to reconfigure its IP protocol stack for a smaller MTU. In this case, the sender has to "throttle back" by sending smaller packets. Newer IP protocol stacks have per route or per path MTUs which make it easier to deploy extended frames for a particular application.

UDP protocol stacks may send datagrams up to 64 Kbytes, as requested by the application. But there is no mechanism to negotiate UDP MTU with the peer station. If UDP datagrams that exceed the MTU of any intermediate network are sent, IP fragmentation will automatically occur.

The Solution

The most important issue in the extended Ethernet frame discussion is the ability to have seamless operation between devices that support larger frames and those that do not. Extended Ethernet frames can be easily implemented and controlled within legacy Ethernet environments by partitioning the reach of these frames, either physically or through the use of standard tagging techniques such as IEEE 802.1Q.

Figure 3. Integrating Extended Ethernet Frames Into Existing Ethernet Infrastructures



Since extended frame sizes yield the most benefit in large transfers such as backup and data replication, their application can frequently be confined to a server farm or power workgroup. The simplest way to partition large frames is to build separate back-end server LANs or workgroup LANs over which extended frames flow freely among devices that support them. This is an expensive scheme as it requires a separate adapter on each host. However, the adapter cost often pales in comparison to the savings in valuable server cycles and increased productivity.

A more cost-efficient way to achieve the same result is via the use of VLANs. The IEEE 802.1Q specification is an emerging standard for tagging Ethernet frames with a VLAN ID. Devices that implement such a scheme will support multiple VLANs or IP subnets on a physical port.

Using the 802.1Q mechanism, large frames are tagged and partitioned in a VLAN in which all equipment (i.e. switches and adapters) support extended frame sizes. Compared to the physical partitioning method, this allows an Ethernet adapter to support both standard Ethernet frames and extended frames over the same physical link. This effectively eliminates interoperability problems resulting from forcing extended frames on devices that only support standard frames (see Figure 3).

There are other schemes which would allow transparent use of extended frame sizes for both new and legacy Ethernet equipment without requiring any partitioning schemes. Once available, they would allow for the seamless integration and operation of extended frames within traditional client-server networks regardless of device support.

Conclusion

Higher speed networks are driving the need for more efficient packaging of data. With new Gigabit-class networks comes the requirement to maximize the efficiency of the attached end systems. In heavily-loaded networks or server LANs where continuous data transfer is required, current Ethernet frame sizes can actually degrade performance - negating many of the initial benefits of high-speed Gigabit networks.

Extended frames significantly enhance the efficiency of Ethernet servers and networks by reducing host packet processing by the CPU and increasing end-to-end throughput. A 9018 maximum frame size fits well with server page sizes of 4Kbytes and 8 Kbytes, accommodates the NFS protocol, and stays comfortably within the standard Ethernet bit error checking limits. By partitioning extended frames to application-specific VLANs or physical LANs compatibility with legacy equipment becomes a “non-issue.”