GE Fanuc
Intelligent Platforms

Switched Ethernet Latency Analysis

Introduction

Ethernet has become perhaps the most ubiquitous communication technology in use today. It was originally used to connect computers in a communication network, but today Ethernet is finding a wide variety of new uses. Ethernet is replacing incumbent network technologies such as CAN bus, MIL-STD-1553, etc. because of its primary advantage: it typically offers higher data throughput rates, which are required in many newer application types, it comes at a fraction of the cost of many other technologies, and it is widely understood.

However, despite its throughput advantages, the latency of Ethernet is unpredictable. In fact, this may be Ethernet's key shortcoming: the actual time it takes for any single packet to traverse the network is unpredictable. The goal of this paper is to discuss aspects which influence latency, and provide an example of how Ethernet communication latency can be analyzed.

Ethernet Switch Architecture Overview

Definition of line rate switching

Figure 1 is a data flow diagram for a typical Gigabit Ethernet switch. Most of the switch silicon on the market today is designed to provide line rate switching performance for all size packets. This means that as long as the output port is not oversubscribed (ie. the expected egress data rate does not exceed the maximum line rate of the port) all packets destined to that port will be forwarded.

For example, Figure 1 shows a switch with twenty four ports. For this example, assume that each port has a line rate of 1Gbit/sec, and all packets entering Port 1 go to Port 2 and vice versa. Similarly all packets entering Port 3 will go to Port 4 and vice versa and so on continuing all the way to Port 24. In this case, the switch will be able to sustain a 1Gbit/sec input data rate and a 1Gbit/sec output data rate.

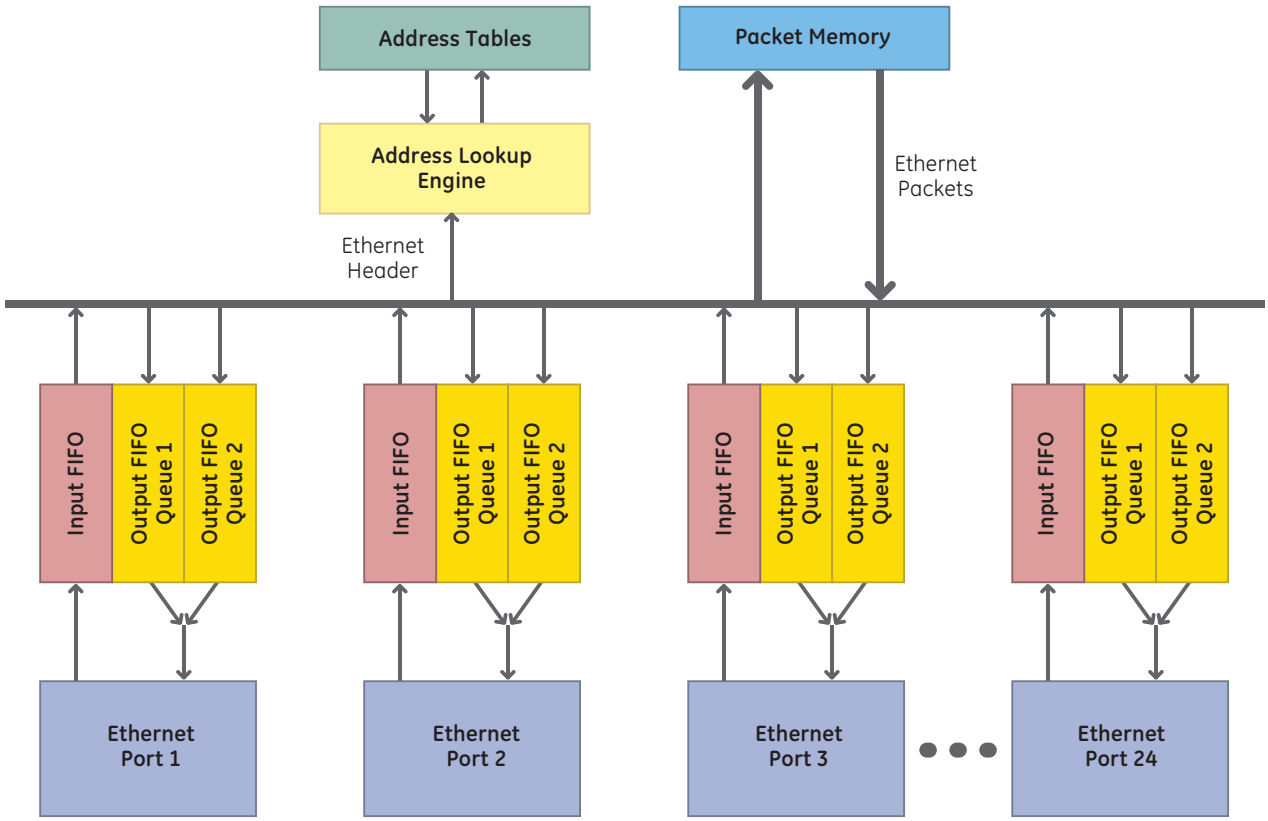

Figure 1: Ethernet Switch Data Flow View

On the other hand, consider an example where all packets entering Port 1 and Port 3 are destined for Port 2. In this case, if the ingress data rate on Ports 1 and 3 is 1Gbit/s, then the total expected egress traffic to Port 2 would be 2 Gbits/s. This is clearly not possible since the Port 2 line rate is only 1 Gbit/s and Port 2 is oversubscribed, resulting in packet drop. These concepts apply to all packet sizes.

Switch Performance Calculations

Most Ethernet Switches are designed with Shared Memory architecture. In this architecture, two bottlenecks exist within the switch silicon:

- Packet Memory throughput
- Packet Address Lookup capacity

In order to provide line rate switching on all ports, memory throughput needs to be such that all packets arriving on all ports simultaneously at line rates can be written to memory. In addition, all packets departing on all ports simultaneously at line rates need to be read from the memory. This means that in our example in Figure 1, combined memory read/write throughput shall be roughly 48 Gbits/s.

From a Packet Memory throughput perspective, the worst case scenario is when packets are maximum size (1518 Bytes), hence overhead due to inter-packet gaps is minimal. From the Address Lookup capacity perspective, in order to sustain line rate throughput the Address Lookup Engine needs to be able to process all packet headers arriving on all ports simultaneously. In this case minimum packet size of 64 Bytes is the most critical case.

Considering an inter-packet gap (12 Bytes) and a Preamble (8 Bytes), the maximum packet rate that can be achieved on 1Gbit Ethernet port is about: 1Gbit/(84B * 8bits) = 1.48 Mpackets/s. Considering the 24 port example above, the Address Lookup Engine will be able to do: 24 x 1.48 Mpackets/s = 35.52 Mpackets/s

These calculations show the minimum performance levels within the Ethernet switch that are implied by the claim that the switch supports the line rate packet switching on all ports. These calculations and assumptions will be used later to calculate worst case packet latency.

Packet Flow and Quality of Service

Most common Ethernet Switches today use Store and Forward architecture. This means that first a packet is fully received and stored in the packet memory. Then, the packet is read back from memory and forwarded to the output queue. Switches typically have more than one output queue, 4 or 8 queues being common. Each output queue can be associated with a different Quality of Service (QoS) level. Quality of Service can be indicated by using a 3 bit field within the VLAN tag. The whole idea behind QoS is that packets with higher priority will be sent out first. One thing to keep in mind is that once a packet transmission has started, it will continue even though a higher priority packet shows up to be transmitted on the same port. This characteristic of Ethernet will become important in latency calculation.

Switched Ethernet Latency Calculation Example

Let's assume the network topology shown in Figure 2. The idea behind this example is to come up with the worst case scenario that would result in the highest latency. Let's assume there are four devices, A – D, communicating via this switch. The following assumptions will further define this example:

- Device A is a main controller
- Device B and C send large (1518 Byte) frames at high data rates to Device A, assume they are video cameras streaming data. Traffic from B and C to A has a low priority.
- Device D is a latency critical controller, which sends 1000 Byte packets every 100ms to Device A. Traffic from D to A has a high priority.

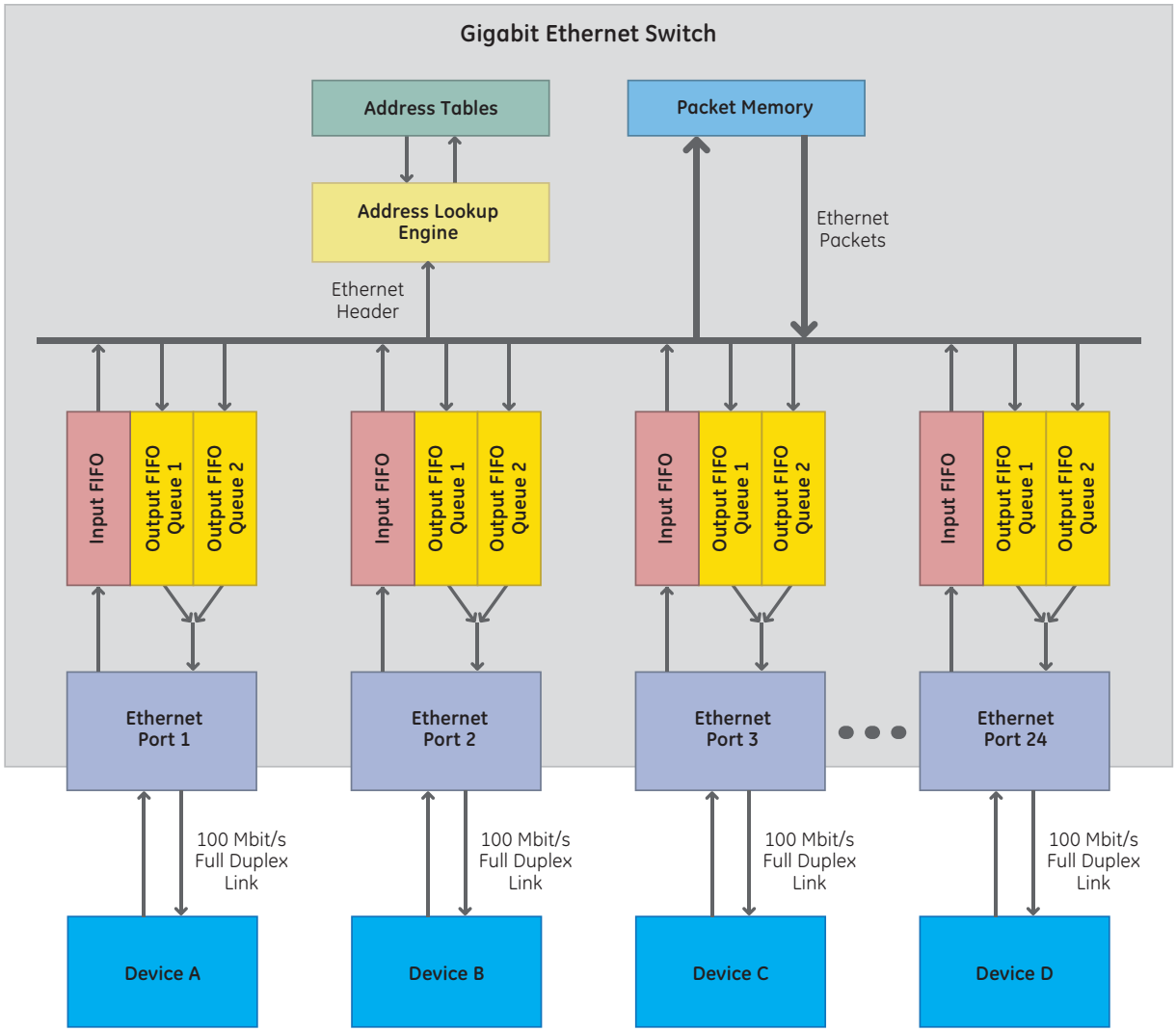

Figure 2: A Latency Analysis Scenario

Note: In this example, the switch is a Gigabit Ethernet switch, but the links are established at 100Mbit/sec. This will influence the internal switch capacity (as discussed earlier), and the maximum data/packet rate each port can sustain. This example applies equally to redundant networks, as long as they are isolated from each other.

First, let's give a latency definition: For the purpose of this example, latency is the time from when the first packet bit is clocked out on the Ethernet transmit link of Device D until the last bit of the same packet is clocked in on the receive link of Device A.

This model does not include the time required by the operating systems on both Device D and Device A to prepare the packet and to move it into the Ethernet controller. Calculations will use these abbreviations:
B=Byte, b=bit, ms=millisecond, µs=microsecond.

Based on Figure 2, latency can be calculated as follows:

1. Time to transmit all the bits from Device D to the input FIFO of Switch Port 24:

   (1000B(packet) + 20B(preamble, inter-frame gap))*8b / 100Mb/s = 81.6 µs

2. Time to write packet data into switch memory:

   Let's assume worst case scenario where all the other ports (assuming all 23 ports) have just received large 1518B packets and are ahead of us in writing it into the memory. To make things really bad, let's assume that all 24 ports are waiting to get a max size 1518B packet from the memory. Using previously calculated memory throughput:
   (23*1518B*8b + 24*1518B*8b) / 48Gb/s = 11.89 µs

   To write our critical packet into memory will take an additional:

   1000B*8b / 48Gb/s = 0.167 µs

   So the total time to store the packet will be:

   11.89 µs + 0.167 us = 12.06 µs

   NOTE: Being conservative, we have assumed that all pending packets are also of high priority, which is typically not the case. In a more realistic case, the waiting time for memory access should be reduced to the time it takes to write in one packet.

3. Time to perform a lookup:

   The address lookup engine is operating in parallel to packet data storage and therefore will not be adding extra time to the packet processing.

4. Time to read packet data from memory:

   This calculation is similar to Step 2. In a very conservative case, we assume that all other packets being read/written are of the same high priority resulting in waiting time:

   (23*1518B*8b + 24*1518B*8b) / 48Gb/s = 11.89 µs

   Then the time for read our critical packet will be:

   1000B*8b / 48Gb/s = 0.167 µs

   So the total time to read the packet and write it into the Port 1 output queue will be:

   11.89 µs + 0.167 µs = 12.06 µs

5. Time to transmit the packet:

   In this step, we have to consider how full the output queues are for normal and higher priority traffic. Switches typically have a number of algorithms that allow applying weights to different priority queues. A sound network implementation would use the highest priority queues only for time critical low bandwidth and low burstiness control traffic. In this case, most of the time, the high priority queue will be empty. On the other hand, it is still possible that even if devices send high priority packets only once every 100ms or so, there are occurrences when all those packets show up at the switch at the same time, filling up the output queue. Therefore, being conservative, let us assume that there are 23 other high priority packets waiting in the queue.

   Then, time to wait: 23*(1000B + 20B (preamble, IPG))*8b / 100Mb/s = 1876 µs

   And, time to transmit: (1000B + 20B (preamble, IPG))*8b / 100Mb/s = 81.6 µs

   Total time being: 1876 µs + 81.6 µs = 1957 µs

   NOTE: Waiting time in the output queue could be by far the highest contributor to the overall latency, hence it is important to accurately establish how many independent devices can be potentially sending high priority packets to the same destination device. If, for example, there are only 10 such devices, instead of 24 as assumed in the calculation above, waiting time can get reduced down to: 9(1000B + 20B (preamble, IPG))*8b / 100Mb/s = 734 µs*

   Putting it all together we get:

   81.6µs + 12.06µs + 12.06µs + 1957 µs = 2062 µs
   or 2.06 ms

Conclusion

The calculations in this paper give an example of how worst case latency in a Switched Ethernet network can be estimated. There are a few other things, however, that are important to note:

1. Ethernet is a Best Effort technology. Switches will drop packets as soon as their memory and FIFOs fill up, hence it is important to plan for lower than maximum network utilization.
2. TCP and UDP protocols used over Ethernet are notorious for generating bursty traffic patterns. Even when a network is underutilized, if every network device starts sending a large number of packets at once, the switch packet memory might overflow. Therefore, network design should look into how much packet memory is available and allocate memory based on packet priority, starting to drop low priority packets first if overflow occurs.
3. Another aspect of network design is the end nodes themselves. The operating system needs to support latency expectations. If the operating system is tied up with some task for an extended amount of time without servicing Ethernet packets, that in turn might create FIFO congestion on the device itself. As a response, the Ethernet controller might invoke Ethernet flow control to throttle down packet transmission from the switch, moving congestion further upstream in the network. Therefore network design shall insure that devices are capable of handling the expected amount of traffic, and that they are able to assign packet priorities and differentiate their handling.

GE Fanuc Intelligent Platforms Information Centers

Americas:
1 800 368 2738 or 1 703 263 1483

Asia Pacific:
+81 3 5544 3973

Europe, Middle East and Africa:
Germany: +49 821 5034-0
UK: +44 1327 359444

Additional Resources

For more information, please visit the GE Fanuc Intelligent Platforms web site at:

www.gefanuc.com

02.09 GFT-737