**Packet Switching (Delay)**

**Processing delay:**

Time requires examining the packet header and determining where to direct the packet is a part of processing delay.

**Queuing delay:**

The packet experiences a queuing delay as it waits to be transmitted into the link. Queuing delay varies packet to packet. If there is no packet in the queue there will be no queuing delay. If there is heavy traffic there will be a huge delay.

**Transmission delay:**

Packet transmitted in first come first serve method. Our packet can be transmitted if all the packets before that have already arrived are transmitted. Packet length L bits, transmission rate R.

dtrans=length of the packet/ rate of the link

dtrans= L/R

Suppose the packet is 150 bytes and transmission rate is 10 Mbps,

dtrans=150/10\*106

**Propagation delay:**

The time requires a packet to start from the beginning to reach the next router d/s, here d is the distance between two routers and s is the propagation speed of the link, Propagation speed depends of the medium of the link. Usually the speed is 2\*108 m/s to 3\*108 m/s.

Suppose the distance between two routers is 1200 km and speed is 3\*108 m/s,

Dprop = distance/speed = 1200\*103/3\*108

***d*total = *dproc* + *d*queue + *d*trans + *d*prop .**

**Packet Switching (Delay) Calculation**

Consider a highway which has a toll booth every 100 kilometers. You can think of the highway segments between toll booths as links and the toll booths as routers. Suppose that cars travel (i.e., propagate) on the highway at a rate of 100 km/hour (i.e., when a car leaves a toll booth it instantaneously **accelerates to 100 km/hour** and maintains that speed between toll booths). Suppose that there is a caravan of 10 cars that are traveling together, and that these ten cars follow each other in a fixed order. You can think of each car as a bit and the caravan as a packet. Also suppose that each toll booth services (i.e., transmits) a car at a **rate of one car per 12 seconds**, and that it is late at night so that the caravan's cars are only cars on the highway. Finally, suppose that whenever the first car of the caravan arrives at a toll booth, it waits at the entrance until the nine other cars have arrived and lined up behind it. (Thus the entire caravan must be "stored" at the toll booth before it can begin to be "forwarded".) The time required for the toll booth to push the entire **caravan onto the highway is 10/(5 cars/minute) = 2 minutes**. This time is analogous to the transmission delay in a router. The time required for a car to travel from the exit of **one toll booth to the next toll booth is 100 Km/(100 km/hour) = 1 hour**. This time is analogous to propagation delay.

Therefore the time from when the caravan is "stored" in front of a toll booth until the caravan is "stored" in front of the next toll booth is the sum of "transmission delay" and "the propagation delay" - in this example, 62 minutes.

In the above scenario propagation delay>transmission delay (1hr>2 min)

Let's explore this analogy a bit more. What would happen if the toll-booth service time for a caravan were greater than the time for a car to travel between toll booths? For example, suppose cars **travel at rate 1000 km/hr** and the toll booth services cars at rate one car per minute. Then the traveling delay between toll booths is **6 minutes** **((100/1000)\*60)** and the **time to serve a caravan is 10 minutes**. In this case, the first few cars in the caravan will arrive at the second toll booth before the last cars in caravan leave the first toll booth. This situation also arises in packet-switched networks - the first bits in a packet can arrive at a router while many of the remaining bits in the packet are still waiting to be transmitted by the preceding router.

In the above scenario transmission delay> propagation delay (6 min<10 min)

If we let *dproc*, *d*queue, *d*trans and *d*prop denote the processing, queuing, transmission and propagation delays, and then the total nodal delay is given by

*d*nodal = *dproc* + *d*queue + *d*trans + *d*prop .