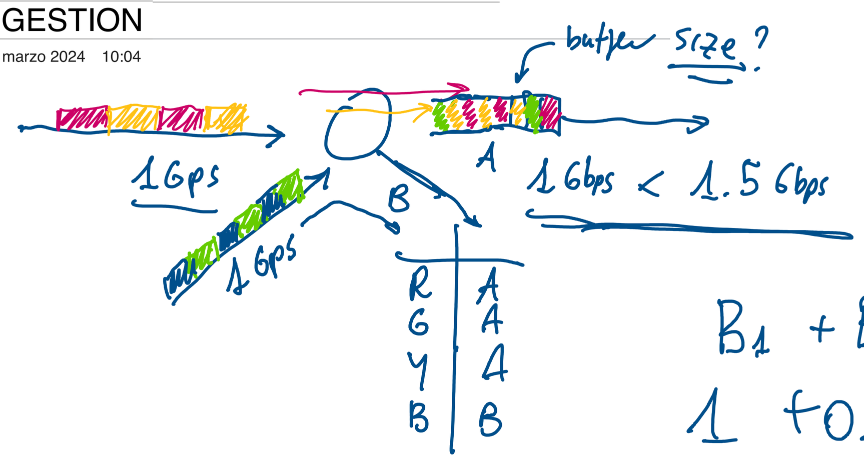
3.5 STATISTICAL MULTIPLEXING AND CONGESTION

martedì 5 marzo 2024 10:04



$T = 10\,ms$

$B_1 + B_2$

$1 + 0.5$

$0.5\,Gbps \cdot T <$

$$1 \begin{cases} > OUT & \boxed{[(B_1+B_2) - OUT] \cdot T < Buffersize} \\ \leq OUT & \end{cases}$$

$$\boxed{0.5 \cdot 10^9 \cdot 1 \cdot 10^{-3} = 500\,kbit}$$

Let's consider an example where two incoming data streams arrive at a router and are both directed towards the same outgoing link. Imagine that the first stream is from a video conferencing application requiring a steady flow of data to maintain a high-quality video call, and the second stream is from a file download, which can vary significantly in its data rate depending on network conditions and server response.

The outgoing link has a fixed capacity, let's say 100 Mbps (Megabits per second). During a particular time interval, the combined data rate of the two streams increases to 120 Mbps due to a surge in the file download speed. Since the outgoing link cannot handle more than 100 Mbps, the router must deal with the excess 20 Mbps of data.

This situation leads to buffering, where the router temporarily stores the excess data in its memory (the buffer) until the outgoing link has enough capacity to handle it. Buffering helps manage short-term discrepancies between incoming data rates and the outgoing link's capacity, ensuring that data is not lost when the network is congested.

However, the duration of this buffering period is crucial. If the combined incoming data rate continues to exceed the outgoing link's capacity for an extended period, the buffer may become full. Once the buffer is full, any new incoming data packets will be dropped until space becomes available in the buffer again. This condition is known as buffer overflow.

The size of the buffer plays a significant role in determining how long the router can accommodate excess incoming data before dropping packets. A larger buffer can store more data, allowing the router to handle longer periods of congestion but at the cost of increasing the latency for the buffered data packets. This is because packets in the buffer must wait longer before being transmitted over the outgoing link.
The phenomenon where large buffers cause long delays is referred to as bufferbloat, which can severely impact applications that are sensitive to latency, such as real-time video conferencing or online gaming. These applications may experience lag or jitter as a result.

Conclusions: while buffering is a necessary mechanism for dealing with short-term network congestion, the size of the buffer and the duration of congestion can significantly impact the flow of data. If the buffering interval is prolonged due to sustained high data rates that exceed the link's capacity, it can cause increased latency, potentially degrading the quality of real-time applications. The challenge for network designers is to balance the size of buffers to manage congestion effectively while minimizing latency for time-sensitive data streams.

To express the relationship between the incoming data flows, the buffer size, and the duration of the overflow period in a network scenario, we can define several parameters and formulate this relationship mathematically. Let's assume we have two incoming data flows, F1 and F2, measured in bits per second (bps), and an outgoing link with capacity C, also in bps. The buffer size is denoted as B, measured in bits.

1. Combined Incoming Data Rate: The total incoming data rate at any given moment is the sum of the individual data rates of the two flows: F_total = F1 + F2.

2. Rate of Buffer Fill: When F_total > C, the excess data rate that cannot be immediately transmitted over the outgoing link starts to fill the buffer. The rate at which the buffer fills is the difference between the total incoming data rate and the capacity of the outgoing link: R_fill = F_total - C.

3. Buffer Overflow Duration: The duration T_overflow for which the buffer can accommodate the excess incoming data before overflowing is determined by the size of the buffer B and the rate of buffer fill R_fill. Assuming the buffer is initially empty and the incoming rate remains constant: T_overflow = B / R_fill.

This explanation shows that the overflow duration is directly proportional to the buffer size and inversely proportional to the rate at which the buffer fills. It is important to note that this formula assumes a simplified scenario where the rates of the incoming flows and the outgoing link capacity are constant, which may not always be the case in real-world networks.
If F_total <= C, then R_fill = 0, indicating that the incoming data rate does not exceed the capacity of the outgoing link, and thus, there is no overflow.

This computation highlight the importance of appropriately sizing buffers and managing data flow rates to prevent buffer overflow while minimizing latency.