Homework1: Network Performance - Solutions

COM-208: Computer Networks

Transmission rate versus propagation delay

A network link is characterized by two important performance metrics:

- The transmission rate is the rate at which we can push bits into the link, and we measure it in "bits per second" (bps), "kilobits per second" (kbps), "megabits per second" (mbps), and so on. Note that 1kbps = 1000bps (not 1024bps), 1mbps = 1 million bps, etc.
- The propagation delay is the amount of time to transfer one bit from one end of the link to the other, and we measure it in time units (typically microseconds, these days).

These two metrics capture very different performance aspects: a link with excellent—for a given application—transmission rate may have inadequate propagation delay and vice versa.

You need to transfer 300TB of data from Lausanne to London. You have two options:

- Use an optical link of transmission rate 1gbps, length 800km, and propagation speed $2 \cdot 10^8$m/s, which directly connects Lausanne to London.
- Send a hard drive by post using a 48-hour delivery service. Assume that writing and reading from the hard drive is instant.

Which of these two "links" has a higher transmission rate? Which one has a lower propagation delay? Which one will transfer your data faster?

The post "link" has infinite transmission rate (negligible transmission delay), yet it has a larger propagation delay compared to the optical link (48hours vs. $8 \cdot 10^5$m/($2 \cdot 10^8$m/s) = 4ms).

Overall, the post will transfer the data faster (within 48hours) since using the optical link, it will take 300 TB/(10^9b/s) = $300 \cdot 10^{12} \cdot 8$b/($10^9$b/s) = 2400000s $\approx$ 28days.

Parallel paths and throughput

Sometimes, there exist multiple network paths between end-systems. Intuitively, adding paths should improve network performance, but which metric exactly? Clearly, adding paths (of the same type) cannot reduce the propagation delay between two end-systems, nor the transmission delay experienced by each single packet. What it *can* change is the overall rate at which A can send data to B: the throughput.

End-systems A and B are connected over M parallel network paths; in this context, "parallel" means that they do not share any links between them. Each path $k \in [1, 2, \ldots, M]$ consists of N links with transmission rates $R_1^k, R_2^k, \ldots, R_N^k$.

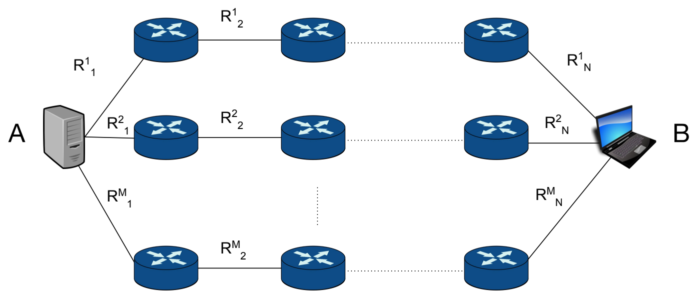

Figure 1: A network topology with multiple parallel paths.

- What is the maximum possible throughput from A to B, when they can use only one path at a time?

When allowed to use only one path at a time, the maximum possible throughput is the maximum throughput across paths, where for each path k, the throughput is equal to the rate of the slowest link on that path. So overall: $\max_{k=1}^{M} \min(R_1^k, R_2^k, \ldots, R_N^k)$.

- How does your answer change when A and B can use all M paths simultaneously?

When allowed to use all M paths simultaneously, the maximum possible throughput is the sum of the throughput over each path. So overall: $\sum_{k=1}^{M} \min(R_1^k, R_2^k, \ldots, R_N^k)$.

The many faces of delay

Computing the delay experienced by an individual packet is typically trickier than computing the average throughput experienced by a communication session. The maximum throughput between two end-systems is the transmission rate of the bottleneck link between them. In contrast, the delay experienced by an individual packet has many different components that may interact in non-obvious ways:

- The transmission delay encountered by a piece of data on a link is the amount of time to push the data into the link, and it is equal to the data size divided by the link transmission rate.
- The propagation delay of a link was defined above.
- The queuing delay encountered by a packet is the amount of time the data spends waiting in packet-switch buffers.
- The processing delay encountered by a packet at a packet switch is the amount of time for which the switch must process the packet in order to decide what to do with it.

Two end-systems, A and B, are connected by a single link of transmission rate R, length l, and propagation speed c. $R = 100$mbps, $l = 200$km, $c = 2 \cdot 10^8$m/s. What is the propagation delay of the link?

> The propagation delay of the link is $d_{prop} = l/c = 2 \cdot 10^5\text{m}/(2 \cdot 10^8\text{m/s}) = 10^{-3}\text{s} = 1\text{ms}$

A sends two back-to-back packets to B, the first one of size $L_1 = 1$kb, the second one of size $L_2 = 10$kb.

- What is the transmission delay experienced by each packet?

> The transmission delay experienced by the first and second packet, respectively, is
>
> $d_{trans,1} = L_1/R = 10^3\text{b}/(10^8\text{b/s}) = 10^{-5}\text{s} = 10\text{us}.$
>
> $d_{trans,2} = L_2/R = 10^4\text{b}/(10^8\text{b/s}) = 10^{-4}\text{s} = 100\text{us}.$

- What is the total transfer time, i.e. the time that elapses from the moment A starts transmitting the first bit of the first packet until the moment B receives the last bit of the second packet?

From Figure 2 we can see that the total transfer time is $d_{prop} + d_{trans,1} + d_{trans,2} = 1.11\text{ms}$

- What is the packet inter-arrival time at B, i.e., the amount of time that elapses from the moment the last bit of the first packet arrives until the moment the last bit of the second packet arrives?

The packet inter-arrival time at B is equal to the transmission delay experienced by the second packet, i.e. $d_{trans,2} = 100\text{us}$.

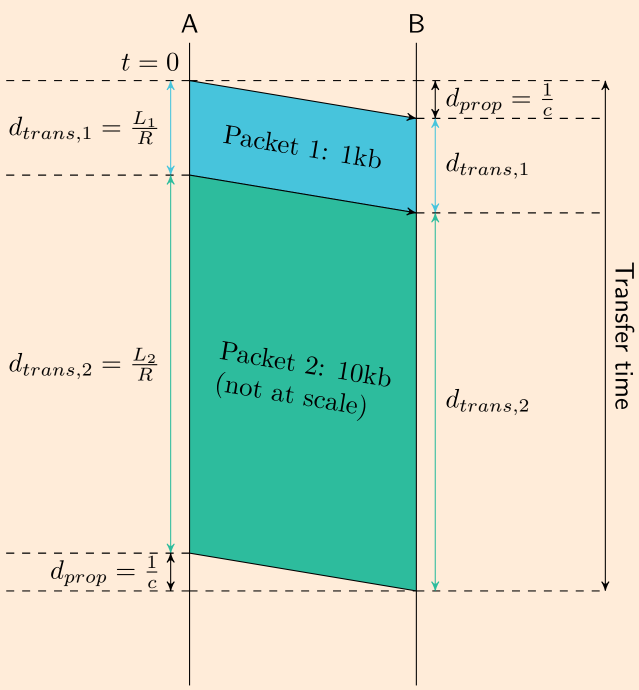

Figure 2: Timing diagram for transmitting two packets in sequence over a single link.

Circuit switching

Let's examine how transfer time changes as we start introducing packet switches between source and destination. Suppose end-systems A and B are connected by not one, but N links and $N-1$ packet switches. All the links have the same properties as the link in the previous problem.

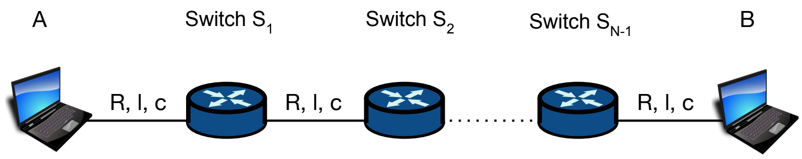

Figure 3: Two end-systems connected through multiple links and packet switches.

Suppose all these packet switches perform *circuit switching*, as in a traditional telephone network: before A starts "talking" to B, a physical circuit is established between them; this enables a packet switch to start transmitting each bit as soon as it receives it. A simple way to think of this scenario is that the packet switches "disappear": if every switch transmits each bit as soon as it receives it, then it is, in a way, as if the switch becomes part of the link, as if A is directly connected to B.

A sends two back-to-back packets to B, each of size $L = 1\text{kb}$. Assume zero processing delay.

- What is the total transfer time when N is equal to 2, then 3 links?

The timing diagram for $N = 2$ links is shown in Figure 4. Due to the fact that the switch uses circuit switching, it does not wait until receiving the full packet; instead it transmits every bit it receives immediately.

The transfer time is $2 \cdot d_{prop} + d_{trans,1} + d_{trans,2} = 2.02\text{ms}$.

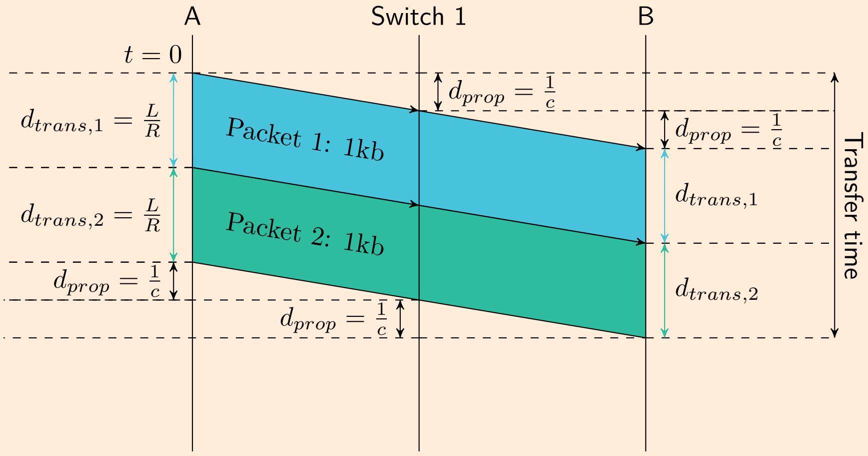

Figure 4: Timing diagram for a transmission over two links connected by a packet switch performing circuit switching.

The timing diagram for $N = 3$ links is shown in Figure 5.

The transfer time is $3 \cdot d_{prop} + d_{trans,1} + d_{trans,2} = 3.02\text{ms}$.

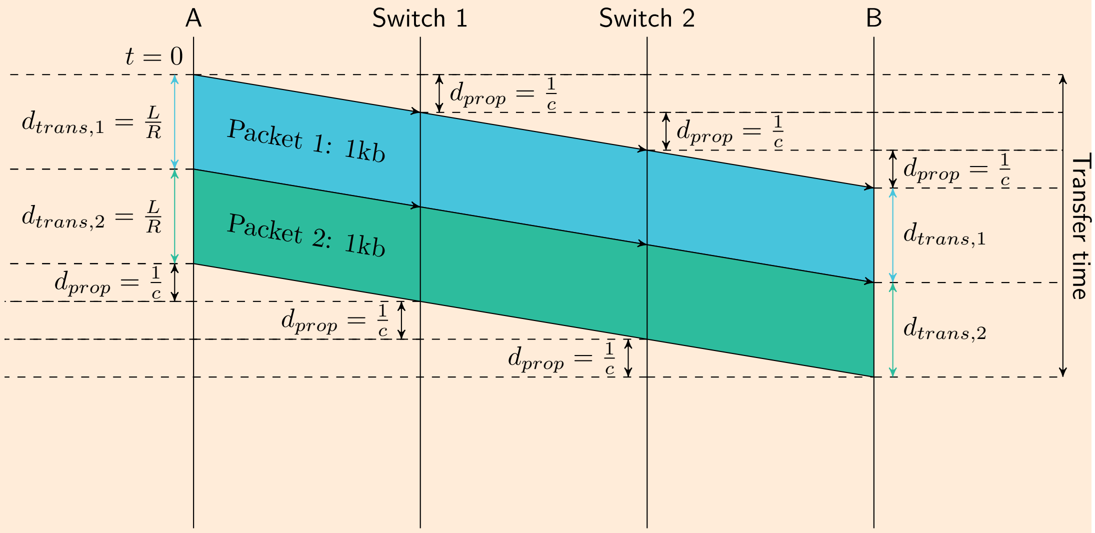

Figure 5: Timing diagram for a transmission over a path with three links, with packet switches performing circuit switching.

- What is the total transfer time as a function of an arbitrary number of links N?

> The circuit behaves like a physical link with rate R and length Nl. The transfer time is $N \cdot d_{prop} + d_{trans,1} + d_{trans,2} = N \cdot l/c + 2 \cdot L/R$

This is a good moment to introduce what we call *timing diagrams*: pictures that represent the various delays that a packet experiences as it travels through a network path. We have started such a diagram right below (Figure 6):

- There is a continuous vertical line for each end-system and packet switch; these lines represent time axes.
- A dashed horizontal line, that crosses the time axes, represents a point in time. For example, on the timing diagram below, the first dashed horizontal line represents $t = 0$, the second one represents $t = d_{prop}$, and so on.
- A continuous line that connects two time axes represents a bit as it travels between the corresponding devices; the beginning of this line represents the moment when the bit was transmitted, while the end of the line represents the moment when the bit arrived at the corresponding device. For example, on the timing diagram below, the first continuous line between the A and Switch1 time axes represents the first bit of the first packet as it travels from A to Switch1; the second continuous line between the same time axes represents the last bit of the first packet.

Using these diagrams, we can easily identify the various delay components that a packet experiences on each link and at each switch. We have marked these for the first packet sent from A to B, when $N = 2$ links. If you add the second packet, you will have solved the problem for $N = 2$. You can draw a new diagram, with 4 time axes, to solve the problem for $N = 3$. You cannot draw a diagram for an arbitrary number of links N, but 2 and 3 links are usually enough to give you the right intuition.

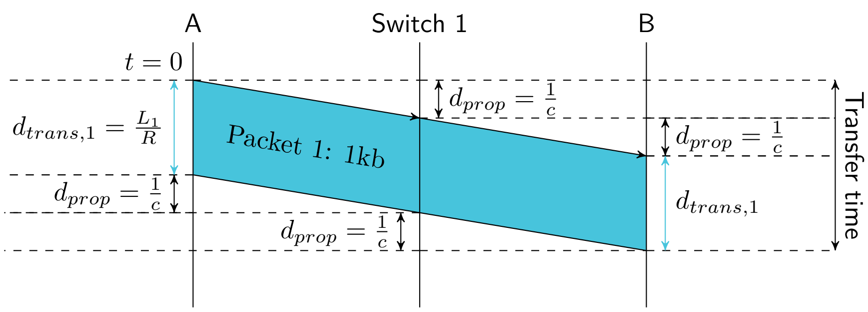

Figure 6: Timing diagram for the transmission of a single packet over two links connected by a packet switch performing circuit switching.

Store and forward

But, Internet packet switches do not typically perform circuit switching; many of them perform store-and-forward: when a switch receives a bit, it must store that bit in a buffer until the last bit of the corresponding packet has arrived; only then can the switch process the packet and start transmitting it over the next link.

Consider the same scenario as in the previous problem, except now, all the packet switches perform store-and-forward. Assume that the processing delay at each switch is 0. Still, the fact that every switch must buffer every bit until it has received the corresponding packet means that there will be some extra delay relative to the circuit switching scenario. (Picture a car rally where every time the first car arrives at an intermediate stop, it must wait for all the cars behind it to arrive before it can start again. Now replace the cars with bits of a packet.)

A sends two back-to-back packets to B, each of size $L = 1$kb.

- What is the total transfer time when N is equal to 2, then 3 links?
- What is the total transfer time as a function of an arbitrary number of links N?

> For this and the above question, the answer is practically the same as below, the only difference being that here $d_{proc} = 0$us.

- How do your answers change when the processing delay at each switch is $d_{proc} = 1$us per packet? Assume that the switch must finish processing and transmitting a packet before it starts processing the next packet.

> The timing diagram for $N = 2$ links is shown in Figure 7.
>
> The transfer time is $2 \cdot (d_{prop} + d_{trans,1} + d_{proc}) + d_{trans,2} = 2.032$ms.

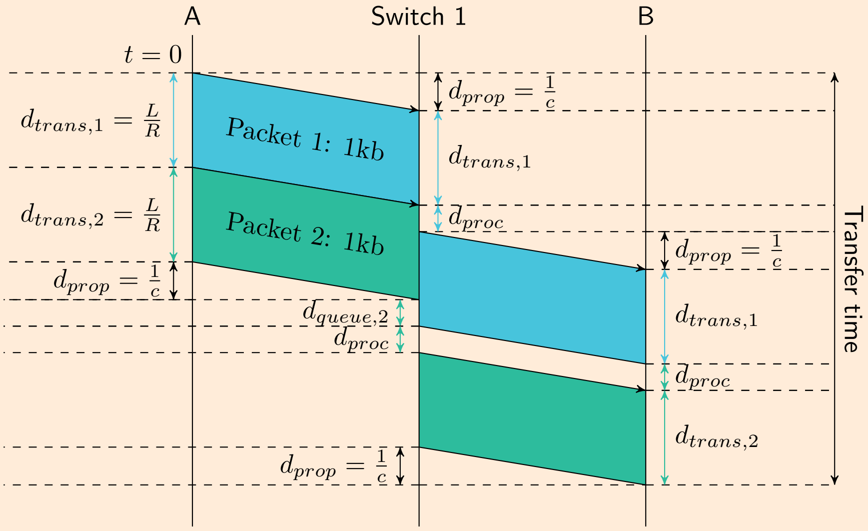

Figure 7: Timing diagram for a transmission over two links connected by a store-and-forward switch.

The timing diagram for $N = 3$ links is shown in Figure 8. The transfer time is $3 \cdot (d_{prop} + d_{trans,1} + d_{proc}) + d_{trans_2} = 3.043$ms.

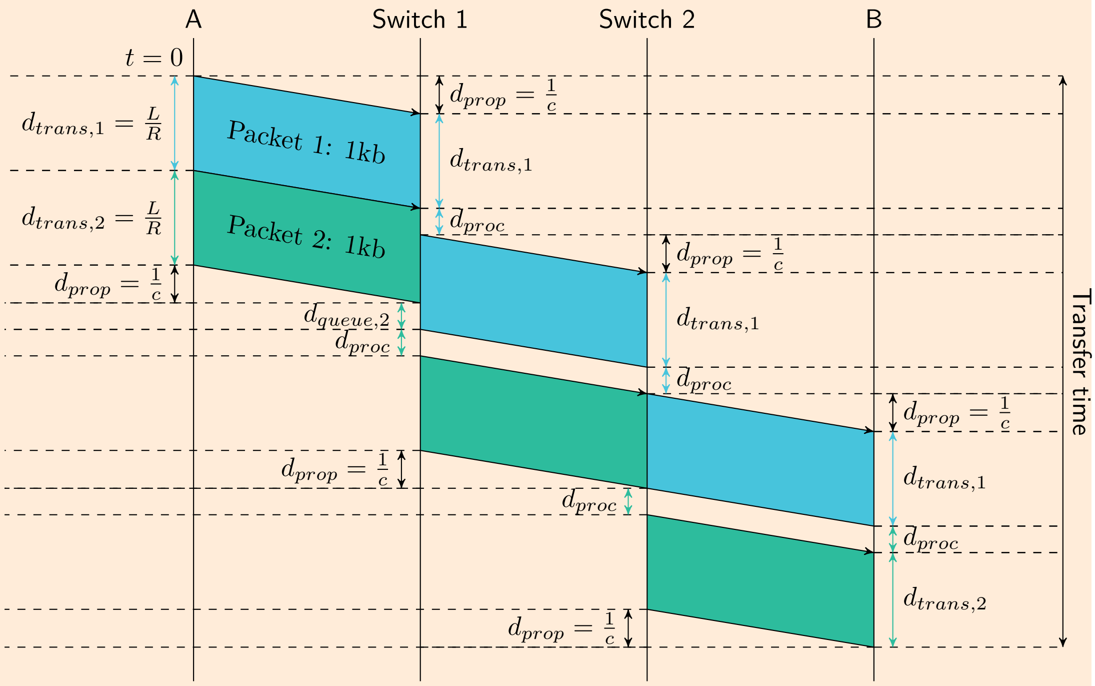

Figure 8: Timing diagram for a transmission over three links connected by store-and-forward switches.

We can see from the previous two cases that as we add one more link, the transfer time increases by $(d_{prop} + d_{trans,1} + d_{proc})$. Thus for $N \geq 2$ links, the transfer time is $N \cdot (d_{prop} + d_{trans,1} + d_{proc}) + d_{trans_2}$.

- Does the second packet experience any queuing delay at the first packet switch? To answer, focus on the first packet switch and compute the difference between: (a) when the switch transmits the last bit of the first packet and (b) when the switch receives the last bit of the second packet. If this is greater than 0, it means that the second packet must wait for the switch to finish transmitting the first packet, i.e., it experiences queuing delay.

Yes, it does.

As shown in Figure 7 and Figure 8, at Switch1, the second packet experiences queuing delay $d_{queue,2} = (d_{prop} + d_{trans,1} + d_{proc} + d_{trans,1}) - (d_{prop} + d_{trans,1} + d_{trans,2}) = d_{proc} = 1\text{us}$.

- Does the second packet experience any queuing delay at any other packet switch?

No, because as soon as the last bit of the second packet arrives at any of the subsequent switches, the first packet has already been fully transmitted over the next link.

The easiest way to solve this problem—and most delay problems—is through timing diagrams. We have started one right below that represents the first packet sent from A to B when $N = 2$ links. Notice what is happening at the switch: because it is a store-and-forward switch, it cannot transmit the first bit of the first packet as soon as it receives it, it must wait for the last bit of the first packet to arrive; only then it can process and transmit the first packet.

To solve the problem, add the second packet, then draw another timing diagram for $N = 3$.

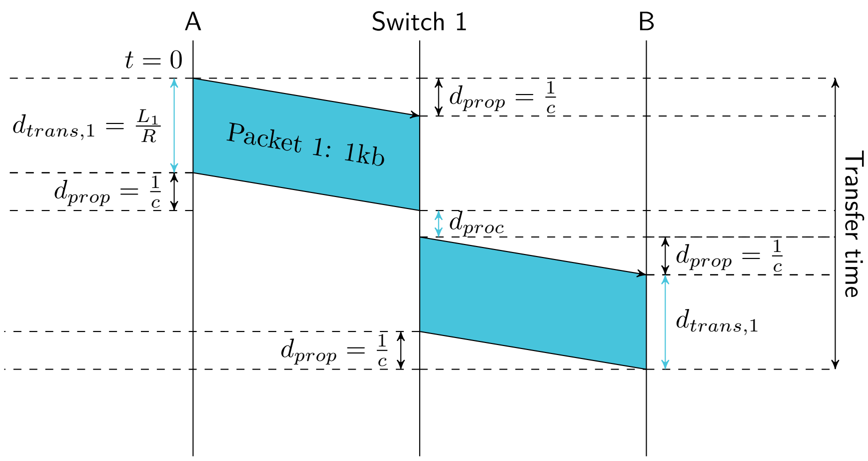

Figure 9: Timing diagram for a transmission over two links connected by a store-and-forward switch.

Now with lots of packets!

When two end-systems communicate over the Internet, they often exchange $P > 2$ packets. Intuitively, that should increase the total transfer time, but by how much? A factor of P? Does it depend on N?

Consider the same scenario as in the previous problem: A and B are connected through $N - 1$ store-and-forward switches, each introducing processing delay 0. Except now, A sends P back-to-back packets to B, all with the same length L. Assume zero propagation delays.

- What is the total transfer time?

Since the timing diagram becomes a bit more complicated when considering multiple links, we may think in the following way:

At time $N \cdot L/R$ the (last bit of the) first packet reaches end-system B, the second packet is ready to be transmitted over the last link, the third packet is ready to be transmitted over the second-to-last link, etc. At time $N \cdot (L/R) + L/R$, the second packet reaches B, the third packet is ready to be transmitted over the last link, etc. Continuing with this logic, we see that all packets will have reached B by time

$$d_{tranfer} = N \cdot L/R + (P - 1) \cdot L/R = (N + P - 1) \cdot L/R$$

Queuing and bottlenecks

You may have noticed the artificial uniformity in the previous problems: all packets were of the same length, all links were of the same transmission rate... Reality is not like this, of course. Packets have different lengths, and links have different transmission rates, and both of these things lead to queuing. Even if A and B are alone in the Internet—there is no other traffic to interfere with theirs—when A sends back-to-back packets to B, a packet may have to wait at a packet switch for the previous packet to be transmitted, either because the previous packet is longer, or because the next link is slower.

End-systems A and B are connected through two links with one store-and-forward packet switch in the middle, introducing processing delay 0. Both links have propagation delay d_{prop}. The first link has transmission rate R_1, while the second one has transmission rate R_2.

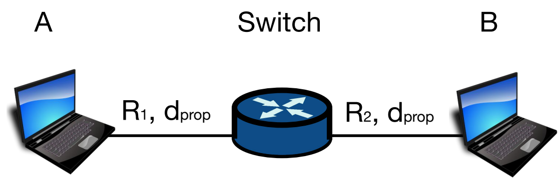

Figure 10: Two end-systems connected through two links.

Assume that $R_1 = R_2$. A sends two back-to-back packets to B, the first one of size L_1, the second one of size $L_2 \neq L_1$.

- In which scenario does the second packet experience queuing delay at the switch? How much?

In case the first packet is longer than the second packet, i.e., $L_1 > L_2$.

As shown in Figure 11, the queuing delay experienced by the second packet is given by $d_{queue,2} = d_{trans,1} - d_{trans,2}$.

- What is the total transfer time (in this scenario)?

As shown in Figure 11, the transfer time is $2 \cdot (d_{prop} + d_{trans,1}) + d_{trans,2}$.

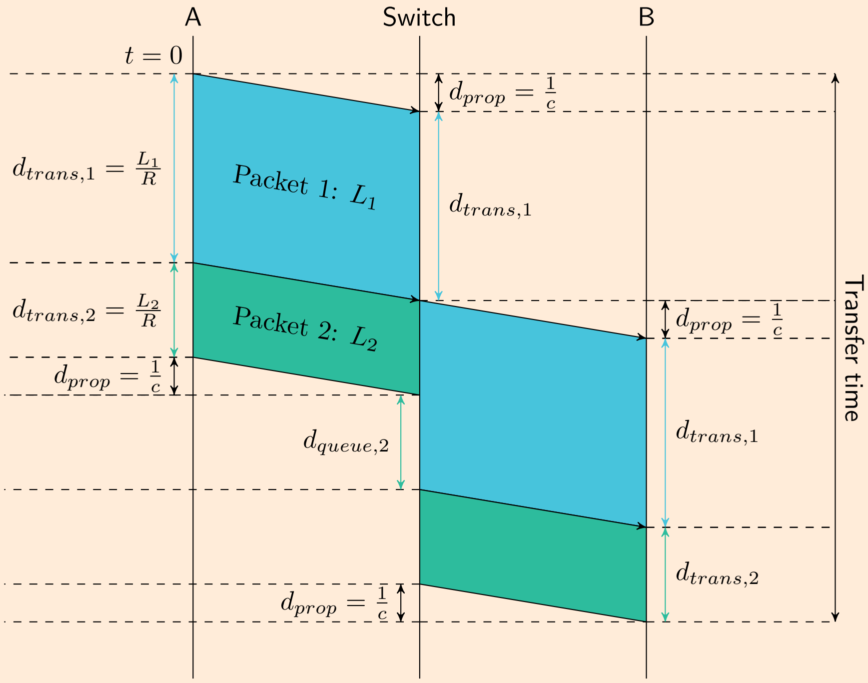

Figure 11: Timing diagram for a transmission over two links connected by a store-and-forward switch.

[This one is if you want to challenge yourself.] A and B are connected though N links and $N-1$ store-and-forward packet switches, each introducing processing delay 0, all links have propagation delay d_{prop} and transmission rate R, and A sends P back-to-back packets to B, of different lengths, $L_1, L_2, ..., L_P$. Assume infinite packet-switch buffers (no packet drops).

- What is the total transfer time?

In general, when the packets have different lengths, the order at which the packets are sent may affect the total transfer time.

However, this is not the case in this exercise where all links have the same processing and propagation delays, the same transmission rate, and packet-switch buffers are infinite (no packet drops). Thus, the total transfer time is the same as when sending the *longer* packet, say L_1, first and then the rest of the packets. The total transfer time is given by

$$d_{tranfer} = N \cdot \left(\frac{L_1}{R} + d_{prop} \right) + \sum_{i=2}^{P} \frac{L_i}{R}, \quad \text{where} \quad L_1 \geq L_i, \quad \forall i \in [2, 3, \ldots, P]$$

Now let's go back to the simpler scenario where A and B are connected through two links with one store-and-forward packet switch in the middle, introducing processing delay 0. Assume that $R_2 \neq R_1$. A sends two back-to-back packets to B, each of size L. Is it possible that the second packet experiences queuing delay at the packet switch? Let's consider all the possibilities:

- Suppose $R_1 < R_2$, i.e., the first link has a lower transmission rate than the second one, in other words, the first link is the bottleneck. (Picture a narrow local street, followed by a wide highway. Would you expect cars to queue up at the highway entrance?)
  - How much queuing delay does the second packet experience at the packet switch?

As shown in Figure 12, the second packet experiences 0 queuing delay at the switch.

  - What is the packet inter-arrival time at B?

As shown in Figure 12, the packet inter-arrival time at B is given by $d_{interarr} = d_{tranfer} - d_{tranfer_1} = d_{trans,2} + d_{trans',2} - d_{trans',1}$.

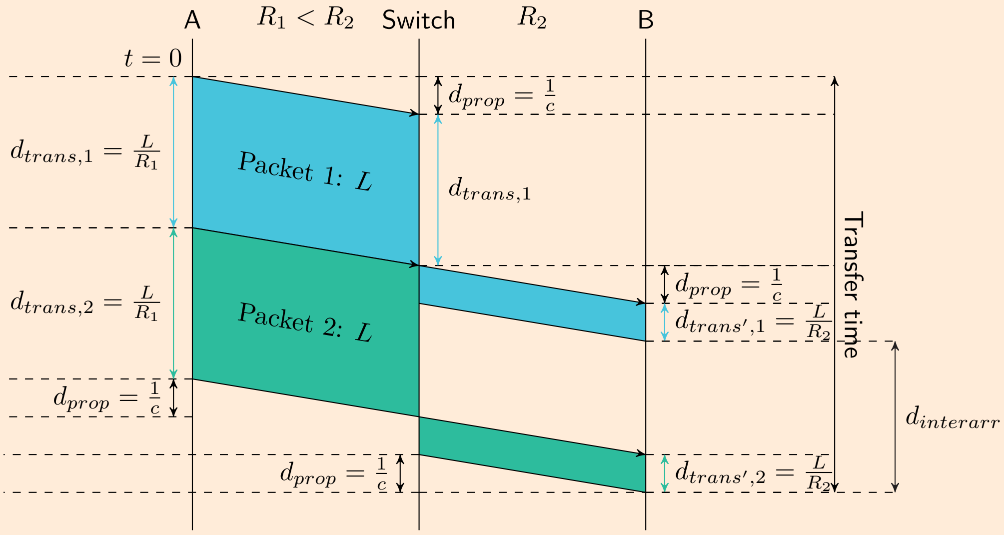

Figure 12: Timing diagram for a transmission over two links connected by a store-and-forward switch.

- Now suppose $R_1 > R_2$, i.e., the second link is the bottleneck. (Picture a wide highway, followed by a narrow local street. Would you expect cars to queue up at the highway exit?)
  - How much queuing delay does the second packet experience at the packet switch?

As shown in Figure 13, the queuing delay of the second packet at the switch is given by $d_{queue,2} = d_{trans',1} - d_{trans,2}$.

  - Suppose that A sends the second packet T seconds after sending the first one. How large must T be to ensure no queuing at the switch?

$T \geq d_{queue,2}$ to ensure no queuing delay at the switch.

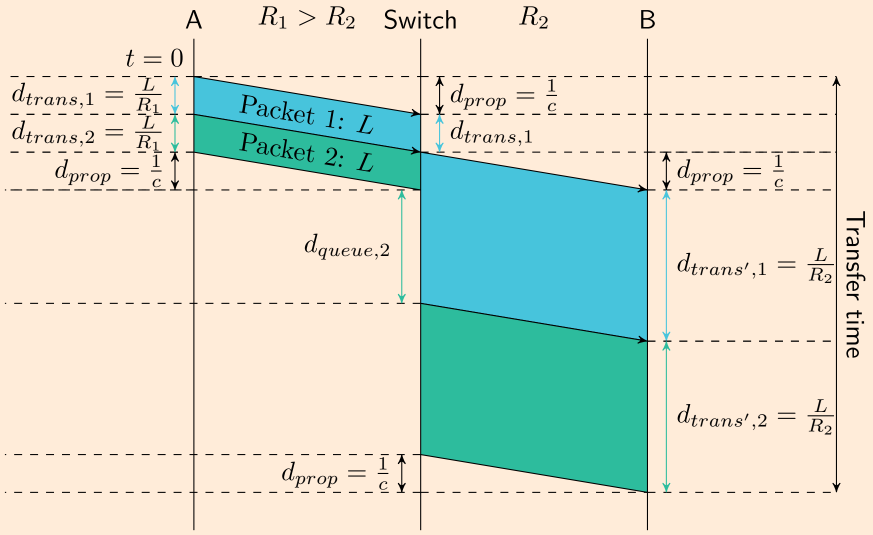

Figure 13: Timing diagram for a transmission over two links connected by a store-and-forward switch.

The bandwidth-delay product

Now that you have a good grip on the concepts of transmission rate, delay, and throughput, we can explore a crucial networking concept, called the "bandwidth-delay product".

End-systems A and B are directly connected by a link of length $l = 20000$km, propagation speed $c = 2.5 \cdot 10^8$m/s, and transmission rate $R = 2$mbps.

- First, what is the propagation delay of the link, d_{prop}? (Just to remind you that the propagation delay is determined in a straightforward way by the link's physical properties.)

$d_{prop} = l/c = 2 \cdot 10^7\text{m}/(2.5 \cdot 10^8\text{m/s}) = 80\text{ms}$

- What is the maximum number of bits that can be in transit on this link at any point in time? To answer, compute the time that elapses from the moment A transmits the first bit until that first bit reaches B; what is the maximum number of bits that A may have transmitted in this time? This is the bandwidth-delay product of the link, which captures, in a sense, the link's capacity in bits.

The bandwidth-delay product of the link is $BDP = R \cdot d_{prop} = 2 \cdot 10^6\text{bps} \cdot 80 \cdot 10^{-3}\text{s} = 16 \cdot 10^4\text{b}$.

- Now visualize the link as a pipe filled with bits. What is the "length" (in meters) of one bit in the link, i.e., what fraction of the link's length corresponds to one bit? Which link properties does this length depend on?

The "length" of a bit is given by $l/BDP = 2 \cdot 10^7\text{m}/(16 \cdot 10^4\text{b}) = 125\text{m}$ per bit, so 1 bit is 125 meters long.

The length of a bit depends on the propagation speed and the transmission rate of the link as $l/BDP = l/(R \cdot d_{prop}) = l \cdot c/(R \cdot l) = c/R$.

Prelude to Internet security

We haven't explored Internet security at any depth, but we would like you to start thinking about what security could mean in different situations.

You are responsible for the security of the following applications:

- An online banking system that enables customers use their private accounts and perform financial transactions (just like ATM banking).

> Impersonation attack should be prevented to ensure legitimate customers' transactions.
>
> There are fraudsters who have stolen/duplicated some customer's ATM card and somehow (often done by pretending on the phone to be bank staff) obtained information about the PIN. In this way, fraudsters are able to impersonate the legitimate bank customers when using the ATM.
>
> A solution to this is a fingerprint check.

- A weather website that should be available 24 hours per day.

> Denial-of-Service (DoS) attacks should be prevented to ensure seamless availability of the website.
>
> A solution to this is: Once being under attack the server could identify the source of the attack through the sender IP in the packets that she receives. Then it could set a firewall rule that blocks incoming traffic from that IP, in order to make the website available again. Of course there are much more sophisticated DoS attacks like Distributed DoS (DDoS) attacks with source IP spoofing, that are more difficult to prevent.
>
> For further knowledge, you may always take a short look at DDoS prevention techniques online, even if it is a bit early to understand all the details. A good reference is http://www.cisco.com/c/en/us/support/docs/security-vpn/kerberos/13634-newsflash.html.

- A chat application (similar to Snapchat) that provides ephemeral private messaging: once the recipient of a message has viewed its content, the message is automatically deleted; the recipient may make a message copy, but in that case the sender is explicitly notified.

Eavesdropping attacks should be prevented to ensure that the sender is always notified when someone has copied her message.

For example, imagine that the recipient does not keep a copy of the sender messages. Yet, an eavesdropper sitting between the sender and the receiver is able to capture the sender's messages. In this case, the application running at the legitimate recipient will not generate any copy notifications, and as a result, the sender will continue to believe that her messages are being deleted, although the eavesdropper has copied them.

A solution to this is the use of the HTTPs (HTTP-Secure) protocol, which would provide encrypted message exchange between the sender and the receiver.

Which type of vulnerability should you seek to avoid in each case? Explain why and outline a potential solution to address the attack.