1.3-38

Christopher Zúñiga, C28730

March 31, 2025

Problem

We want to transfer a 10 Mb file from "Host 1" to "Host 2," through 3 intermediate routers (so there are 4 links in total). We compare the total time (from the very start of sending until the last bit arrives at Host 2) under:

- Circuit switching
- Message switching
- Datagram switching

The following parameters are given:

- **File size**: 10 Mb (megabits).
- **Number of routers**: 3 (hence 4 links total).
- **Distance** between Host 1 and Host 2: 2000 km (each of the 4 links is 500 km).
- **Propagation speed**: 200,000 km/s.
- **Transmission bandwidth**: 1 Mbps (i.e. 1 Mb/s).
- **Node (router) processing delay**: 100 ms per *router* for each arriving unit (message or packet).
- **Datagram size**: 75 kb (assume 1 kb = 1000 bits, so 75 kb = 75 000 bits).

1 Key Quantities

Per-Link Propagation Delay

Each link is $\frac{2000}{4} = 500\,\text{km}$, and the propagation speed is 200,000 km/s. Hence the one-way *propagation* delay per link is:

$$t_{\text{prop}} \;=\; \frac{500\,\text{km}}{200{,}000\,\text{km/s}} \;=\; 0.0025\,\text{s} \;=\; 2.5\,\text{ms}.$$

Transmission Times

For the entire $10\,\textbf{Mb}$ **file** at $1\,\text{Mbps}$:

$$t_{\text{tx(file)}} \;=\; \frac{10\,\text{Mb}}{1\,\text{Mb/s}} \;=\; 10\,\text{seconds}.$$

For a single $75\,\textbf{kb}$ **datagram** at $1\,\text{Mbps}$:

$$t_{\text{tx(datagram)}} \;=\; \frac{75{,}000\,\text{bits}}{1{,}000{,}000\,\text{bits/s}} \;=\; 0.075\,\text{s} \;=\; 75\,\text{ms}.$$

Router Processing Delay

Each router introduces a $100\,\text{ms}$ processing time for a *message* or *datagram* after it fully arrives but before forwarding.

2 Circuit Switching

Basic assumptions:

- We first perform an *end-to-end circuit setup*, incurring:

  $$(4 \text{ links of propagation}) \;+\; (3 \text{ routers' processing}).$$

  If we assume the setup message is short and bandwidth is high enough, the main times are $4 \times t_{\text{prop}} + 3 \times 100\,\text{ms}$.

- Once the circuit is established, data flows as a continuous stream (no store-and-forward at intermediate routers).

- Finally, the last bit traveling across the 4 links experiences (roughly) $4 \times t_{\text{prop}}$ of propagation once it is sent.

Setup Time

$$t_{\text{setup}} \;=\; (4 \times t_{\text{prop}}) \;+\; (3 \times 0.1\,\text{s}) \;=\; 4 \;\times\; 0.0025 \;+\; 3 \;\times\; 0.1 \;=\; 0.01 + 0.3 \;=\; 0.31\,\text{s}.$$

Data Transmission + Final Propagation

Once the circuit is up, we send $10\,\text{Mb}$ at $1\,\text{Mbps}$ over 4 links (pipelined). The *time to pump out* the file is $10\,\text{s}$. The last bit, after being transmitted, still must propagate to the receiver over $4 \times t_{\text{prop}} = 0.01\,\text{s}$. Hence,

$$t_{\text{data}} + t_{\text{last\_bit\_prop}} \;=\; 10\,\text{s} + 0.01\,\text{s} \;=\; 10.01\,\text{s}.$$

Total (Circuit Switching)

$$t_{\text{total(circuit)}} \;=\; t_{\text{setup}} + t_{\text{data}} + t_{\text{last\_bit\_prop}} \;=\; 0.31 \;+\; 10.01 \;=\; 10.32\,\text{s}\,.$$

3 Message Switching

In *message switching*, the **entire file** (treated as one large message) must be fully transmitted to each router, stored (and processed for 100 ms), then forwarded to the next router, and so on. We have 3 routers = 4 hops.

1. **At hop 1**: Send 10 Mb at 1 Mbps $\Rightarrow$ 10 s, plus 2.5 ms propagation. Once it arrives, router processes for 100 ms.
2. **At hop 2**: Same 10 s send, 2.5 ms prop, 100 ms process.
3. **At hop 3**: Same ...
4. **At hop 4**: Same 10 s send, 2.5 ms prop, but *no* extra router processing since the destination is a host.

Putting this together:

$$\text{Time per hop} = 10\,\text{s} \; + \; 0.0025\,\text{s} \; \approx \; 10.0025\,\text{s}.$$

We do that 4 times (4 links), but we have 3 router-processing delays in between:

$$t_{\text{total(message)}} \; = \; (4 \times 10.0025)\,\text{s} \; + \; (3 \times 0.1)\,\text{s}.$$

Numerically,

$$4 \times 10.0025 \; = \; 40.01\,\text{s}, \quad 3 \times 0.1 \; = \; 0.3\,\text{s}.$$
$$t_{\text{total(message)}} \; = \; 40.01 + 0.3 \; = \; 40.31\,\text{s}.$$

4 Datagram (Packet) Switching

Here, the file is split into many 75 kb (kilobit) datagrams (store-and-forward *per* datagram). We have:

$$\text{File size} = 10{,}000{,}000\,\text{bits}, \quad \text{Datagram size} = 75{,}000\,\text{bits}.$$
$$\text{Number of datagrams} \; = \; \frac{10{,}000{,}000}{75{,}000} \; = \; 133.\overline{3} \approx 134\,.$$

(We typically round up to 134 datagrams.)

One Datagram's End-to-End Delay

For a *single* datagram:

- Transmit time per link: 0.075 s.
- Propagation per link: 0.0025 s.
- Processing at each router: 0.1 s.

Each datagram goes over 4 links (thus 3 routers in between). The store-and-forward model says each router must fully receive the datagram before forwarding it, incurring a 100 ms process time at *each* intermediate router. For each of the **first 3 links**, we have (tx + prop + process). On the **4th link**, we have tx + prop but *no extra router process* at the end.

Hence,

$$t_{1\,\text{datagram}} = 3 \times (\,0.075 + 0.0025 + 0.1\,)\,\text{s} \;+\; \big(0.075 + 0.0025\big)\,\text{s}.$$

$$=\; 3 \;\times\; 0.1775 \;+\; 0.0775 \;=\; 0.5325 + 0.0775 \;=\; 0.61\,\text{s}.$$

Total Delay for 134**134** Datagrams (Pipelined)

In datagram switching, the *sending host* transmits the 134 datagrams back to back. Each datagram has a 0.075 s transmit time on the *first* link. So after the first datagram, the second starts 0.075 s later, etc. Because of pipelining, while the nth datagram is traveling through the next stages, the $(n+1)$th datagram can be on the earlier link, etc.

A common formula for store-and-forward pipelines is:

$$t_{\text{total(datagram)}} \;=\; t_{1\,\text{datagram}} \;+\; (\text{NumberOfDatagrams} - 1) \;\times\; t_{\text{tx}\,(\text{onfirstlink})}.$$

$$=\; 0.61\,\text{s} \;+\; (134 - 1) \;\times\; 0.075\,\text{s}.$$

$$=\; 0.61 \;+\; 133 \times 0.075 \;=\; 0.61 \;+\; 9.975 \;=\; 10.585\,\text{s}.$$

So once the first datagram has fully traversed all 4 links (0.61 s), each subsequent datagram finishes about 0.075 s after the previous one *in the steady pipeline*. By the time we have sent all 134, the total is about $\boxed{10.59\,\text{s}}$.

Final Numerical Results

- **Circuit Switching**:

$$\boxed{t_{\text{total}} \approx 10.32\,\text{s}}$$

- **Message Switching**:

$$\boxed{t_{\text{total}} \approx 40.31\,\text{s}}$$

- **Datagram Switching**:

$$\boxed{t_{\text{total}} \approx 10.59\,\text{s}}$$