Nick’s Review topics

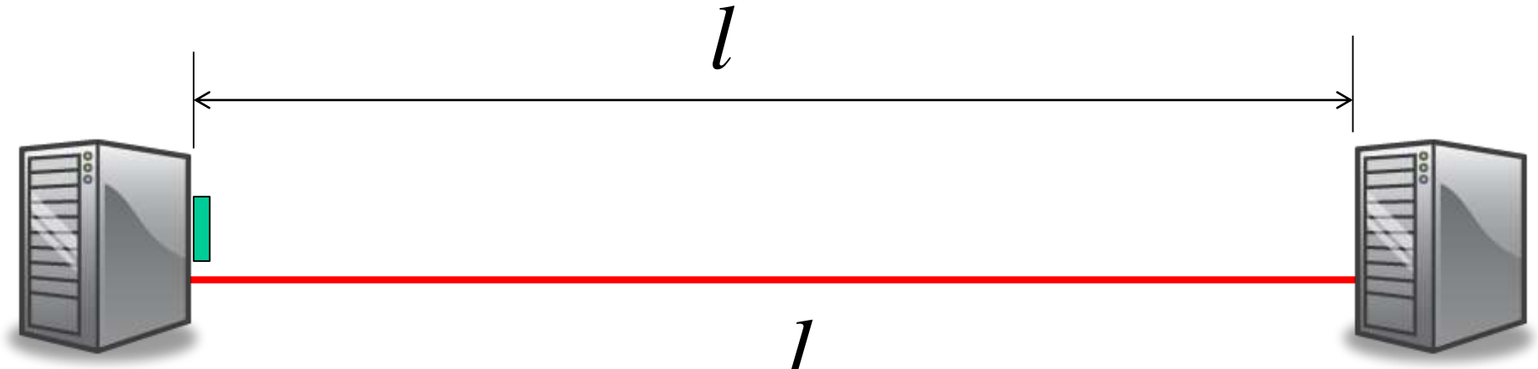
1. Packet Switching
2. Routing, spanning trees and Dijkstra

Packet Switching

Outline

1. End-to-end delay
2. Queueing delay
3. Simple deterministic queue model
4. Rate guarantees
5. Delay guarantees

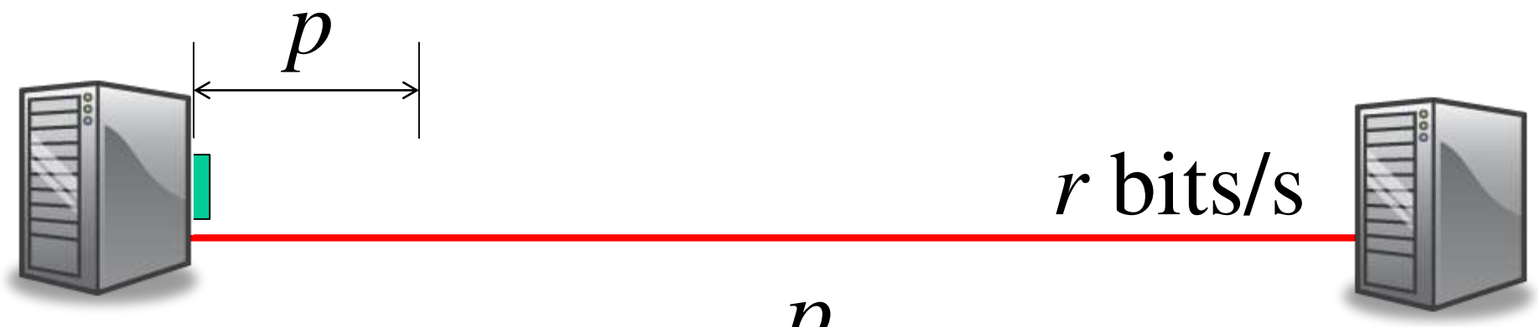
Propagation Delay, t_l: The time it takes a single bit to travel over a link at propagation speed c.



$$t_l = \frac{l}{c}$$

Example: A bit takes 5ms to travel 1,000km in an optical fiber with propagation speed 2 x 10^8 m/s.

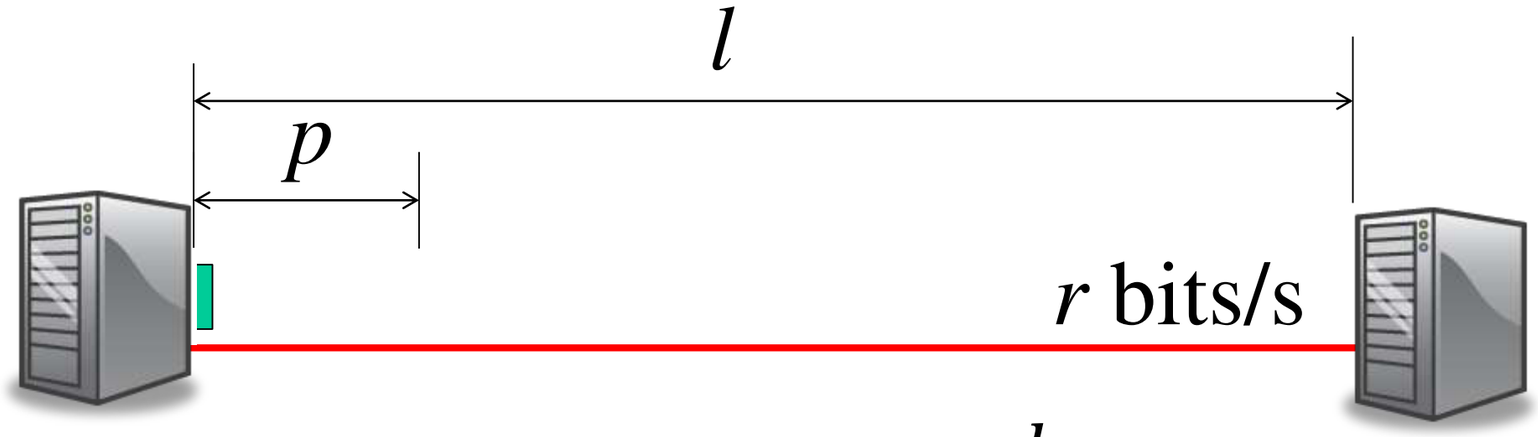
Packetization Delay, t_p: The time from when the first to the last bit of a packet is transmitted.



$$t_p = \frac{p}{r}$$

Example 1: A 64byte packet takes 5.12μs to be transmitted onto a 100Mb/s link.
Example 2: A 1kbit packet takes 1.024s to be transmitted onto a 1kb/s link.

Total time to send a packet across a link: The time from when the first bit is transmitted until the last bit arrives.

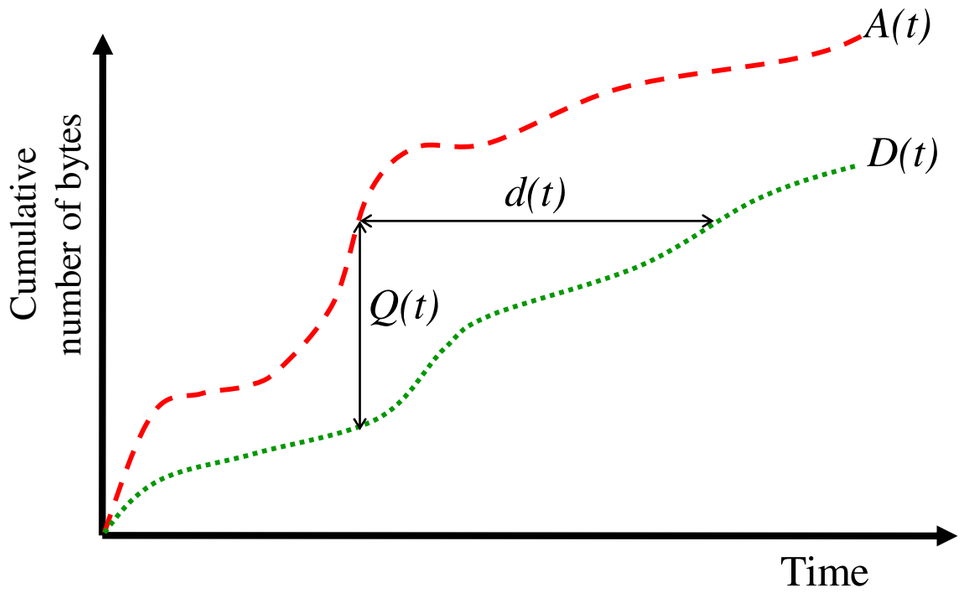


$$t = t_p + t_l = \frac{p}{r} + \frac{l}{c}$$

Example: A 100bit packet takes 10 + 5 = 15μs to be sent at 10Mb/s over a 1km link.

Simple model of a router queue

Simple model of a queue



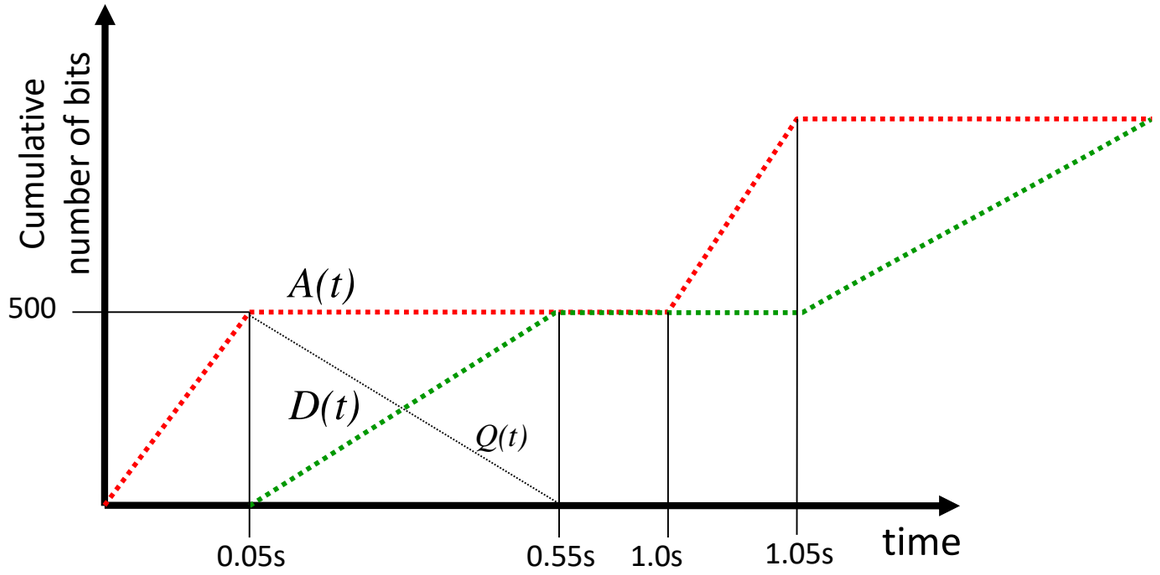
$A(t)$: Cumulative Arrivals. Non-decreasing.
$D(t)$: Cumulative Departures. Non-decreasing.

Queue occupancy: $Q(t) = A(t) - D(t)$.

Queueing delay, $d(t)$, is the time spent in the queue by a byte that arrived at time t, assuming the queue is served first-come-first-served (FCFS).

Example (store & forward)

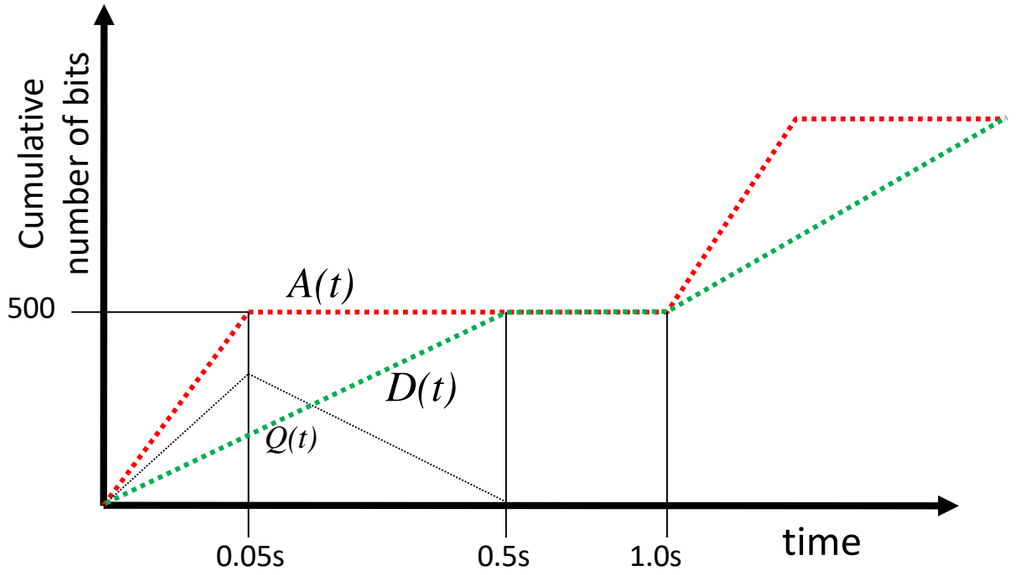
Every second, a 500 bit packet arrives to a queue at rate 10,000b/s. The maximum departure rate is 1,000b/s. What is the average occupancy of the queue?



Solution: During each repeating 1s cycle, the queue fills at rate 10,000b/s for 0.05s, then empties at rate 1,000b/s for 0.5s. Over the first 0.55s, the average queue occupancy is therefore 250 bits. The queue is empty for 0.45s every cycle, and so average queue occupancy is (0.55 * 250) + (0.45 * 0) = 137.5 bits.

Example (“cut through”)

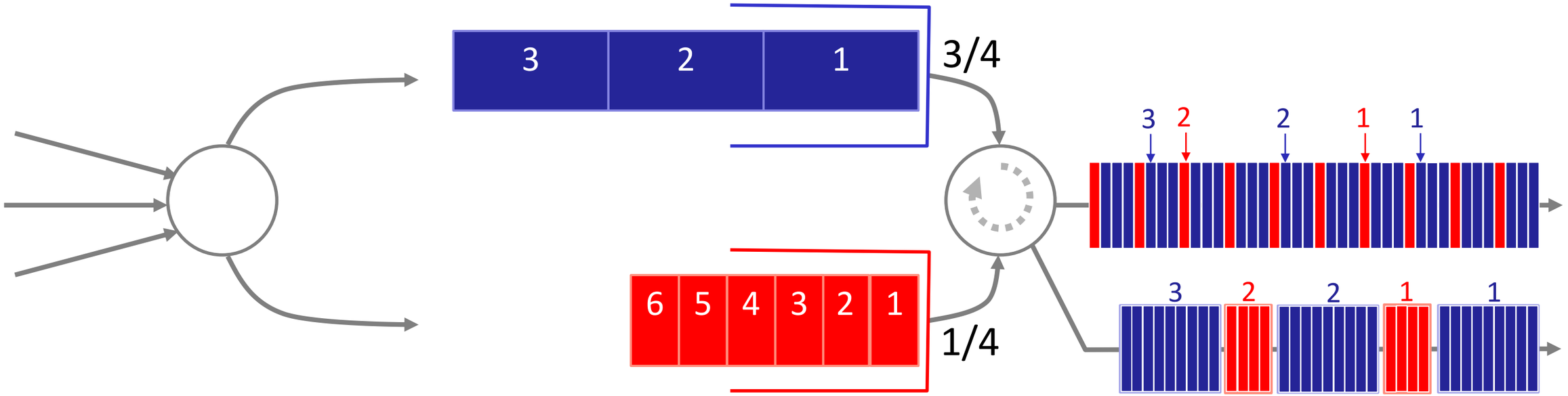
Every second, a 500 bit packet arrives to a queue at rate 10,000b/s. The maximum departure rate is 1,000b/s. What is the time average occupancy of the queue?



Solution: During each repeating 1s cycle, the queue fills at rate 10,000b/s to 500-50=450 bits over the first 0.05s, then drains at rate 1,000b/s for 0.45s. Over the first 0.5s, the average queue occupancy is therefore 225 bits.
The queue is empty for 0.5s every cycle, and so average queue occupancy: $\bar{Q}(t) = (0.5 \times 225) + (0.5 \times 0) = 112.5$

How do I give flows a weighted fair share of a link? (instead of strict priority)

Weighted Fair Queueing

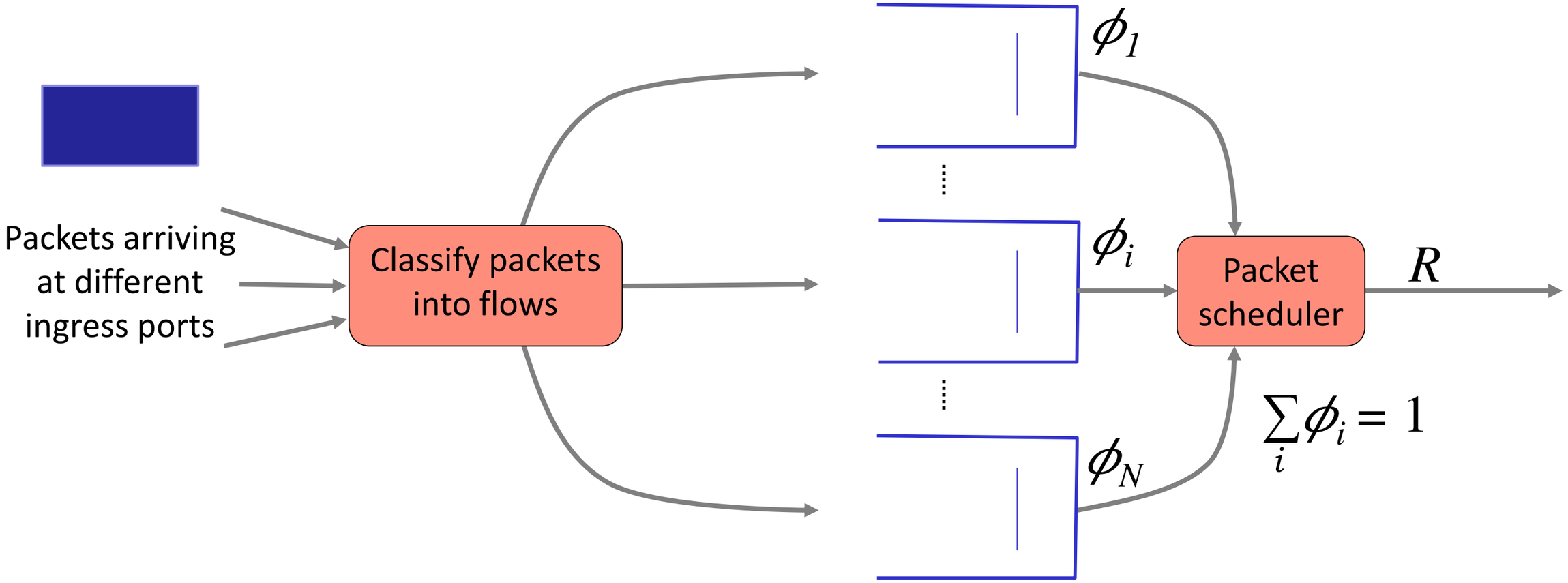


As before, packets are sent in the order they would complete in the bit-by-bit scheme.

Weighted Fair Queueing (WFQ)

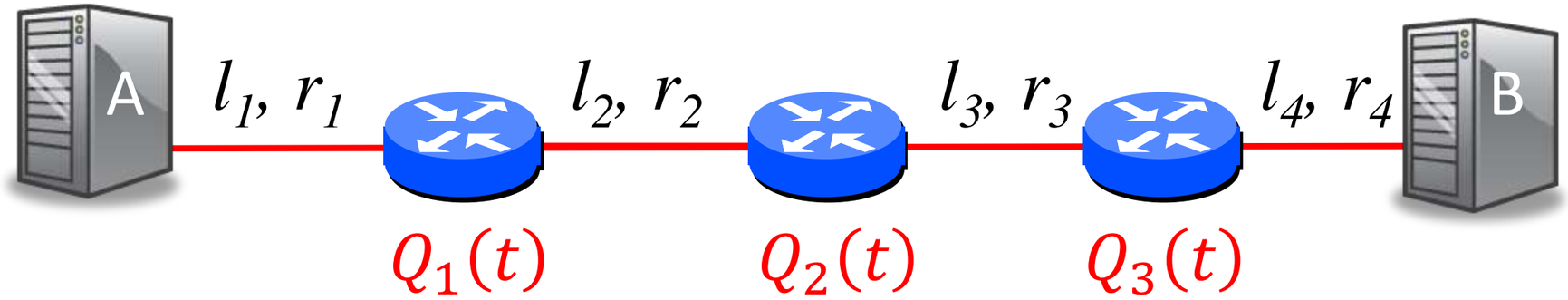
1. It can be proved that the departure time of a packet with WFQ is no more than L_{max}/R seconds later than if it was scheduled bit-by-bit. Where L_{max} is the maximum length packet and R is the data rate of the outgoing link.
2. In the limit, flows receive their weighted fair share of the link.

Weighted Fair Queueing (WFQ)



Flow i is guaranteed to receive at least rate $\phi_i R$

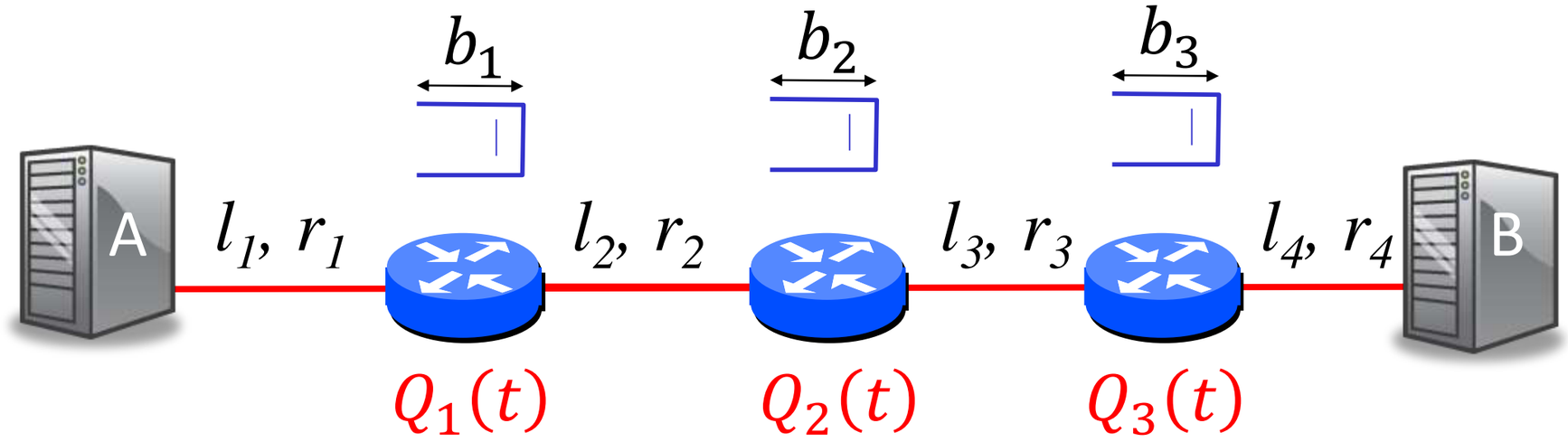
Delay guarantees: Intuition



End-to-end delay, $\tau = \sum_i \left(\frac{p}{r_i} + \frac{l_i}{c} + Q_i(t) \right)$

The following values are fixed (or under our control): p, c, l_i and r_i. If we know the upper bound of $Q_1(t)$, $Q_2(t)$, and $Q_3(t)$, then we know the upper bound of the end-to-end delay.

Delay guarantees: Intuition

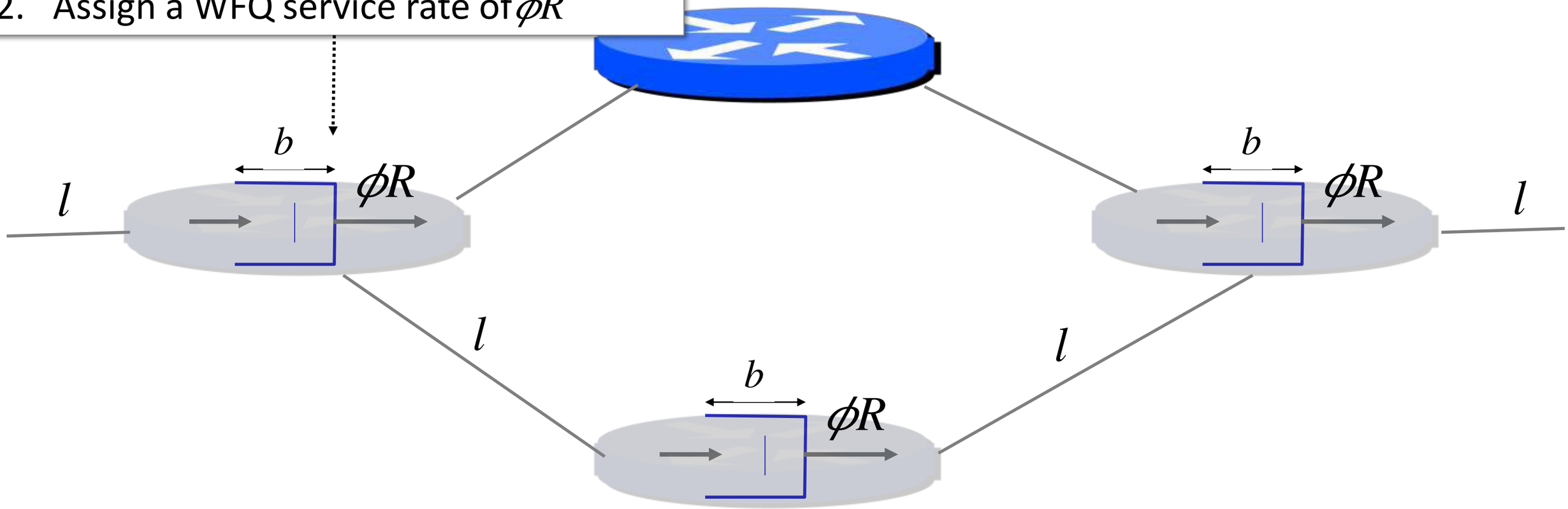


End-to-end delay for a single packet, $\tau = \sum_{i=1}^{4}\left(\frac{p}{r_i}+\frac{l_i}{c}\right) + \sum_{i=1}^{3} Q_i(t)$

$$\leq \sum_{i=1}^{4}\left(\frac{p}{r_i}+\frac{l_i}{c}\right) + \sum_{i=1}^{3}\frac{b_i}{r_i}$$

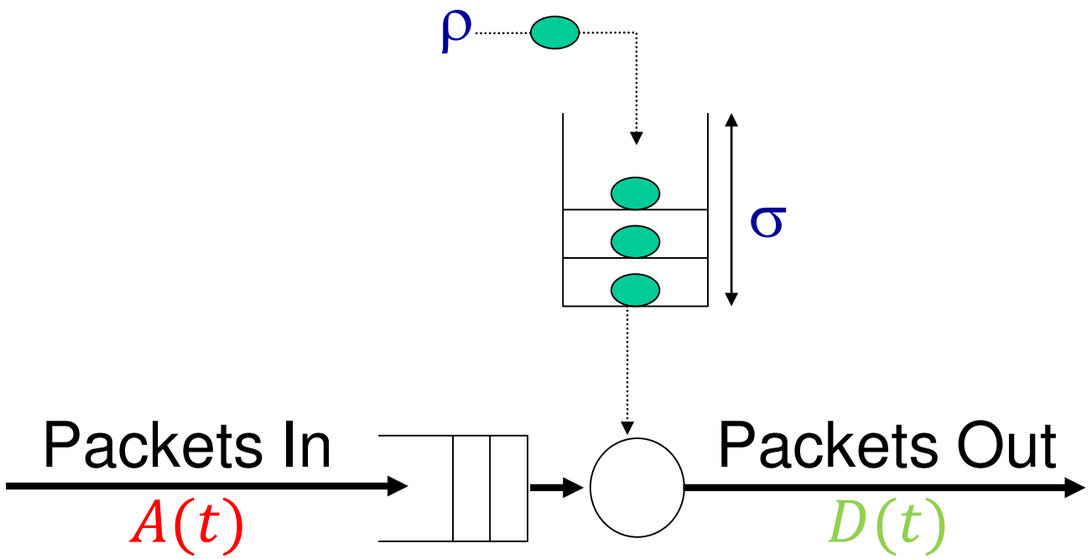
Bounding end-to-end delay

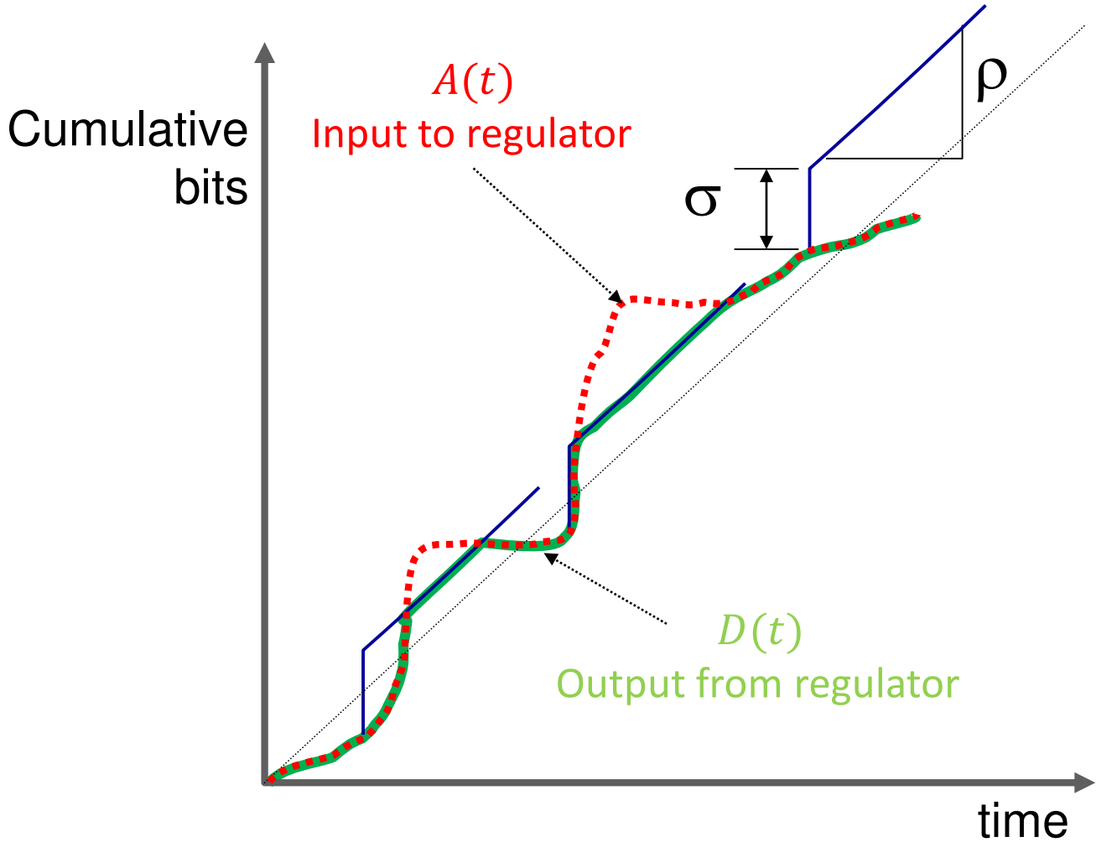
1. Allocate a queue of size b for this flow
2. Assign a WFQ service rate of ϕR



The end-to-end delay of a single packet of length p $\leq 4\left(\frac{l}{c}+\frac{p}{R}\right)+3\frac{b}{\phi R}$

The leaky bucket regulator



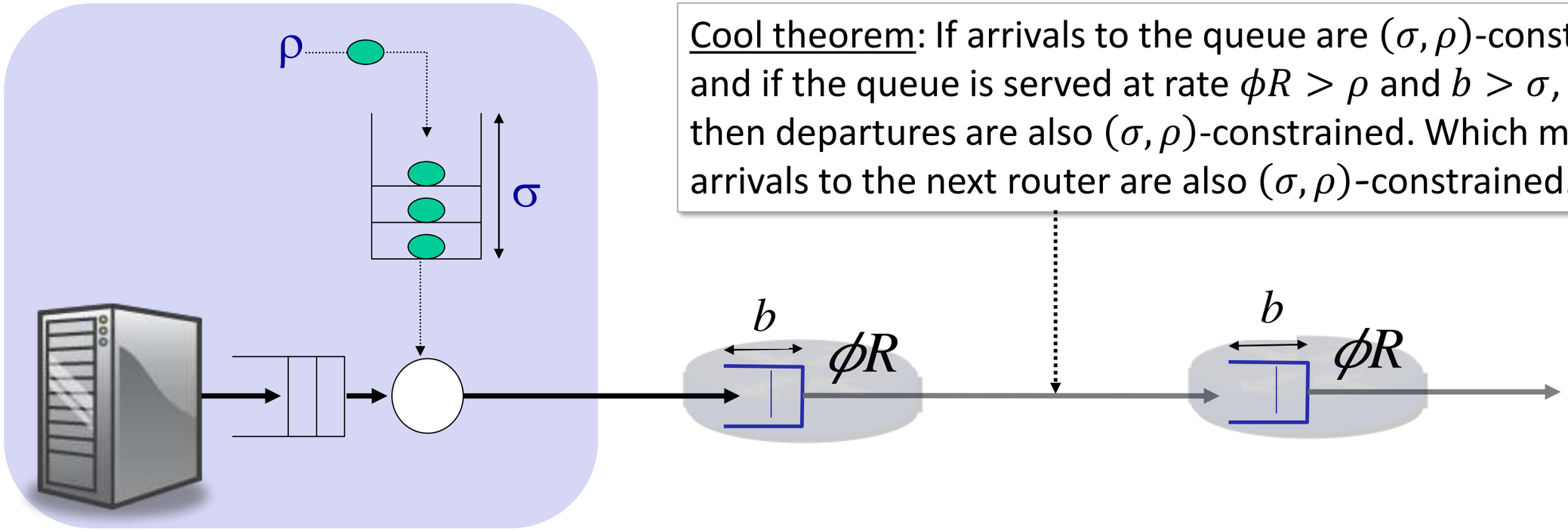


Number of bits that can be sent in <u>any</u> period of length t is bounded by: $\sigma + \rho t$

It is also called a "(σ, ρ) regulator"

The leaky bucket regulator

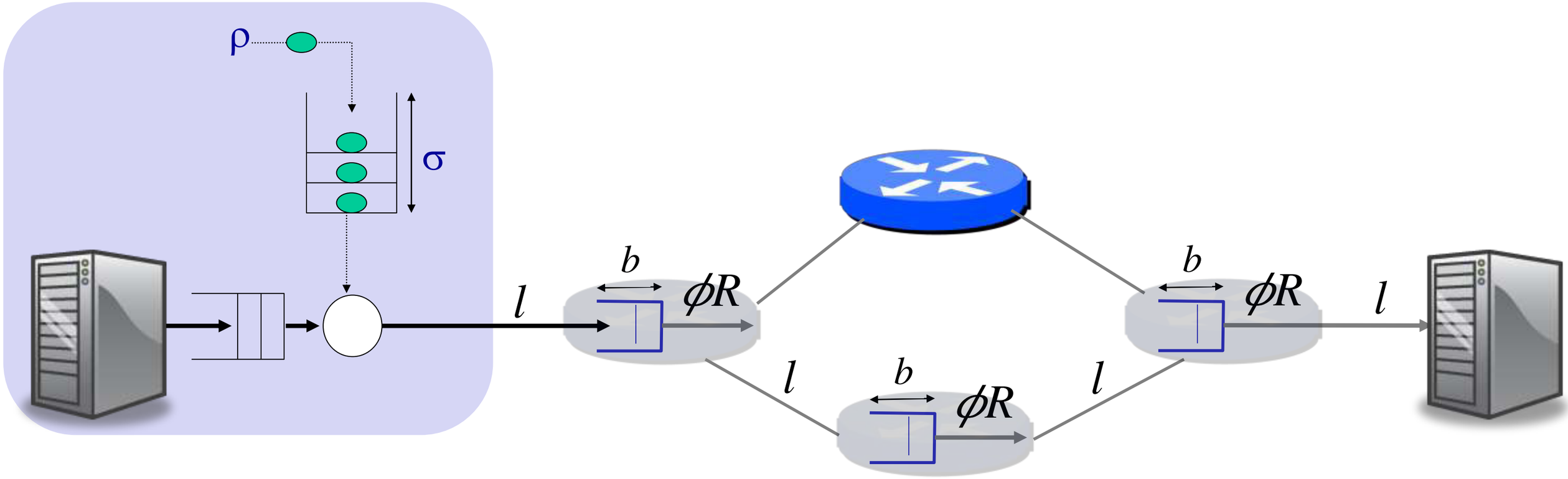
Limiting the "burstiness"



Cool theorem: If arrivals to the queue are (σ, ρ)-constrained, and if the queue is served at rate $\phi R > \rho$ and $b > \sigma$, then departures are also (σ, ρ)-constrained. Which means arrivals to the next router are also (σ, ρ)-constrained.

If $\phi R > \rho$ and $b > \sigma$ then delay through the first router for all packets in the flow $\leq \frac{b}{\phi R}$

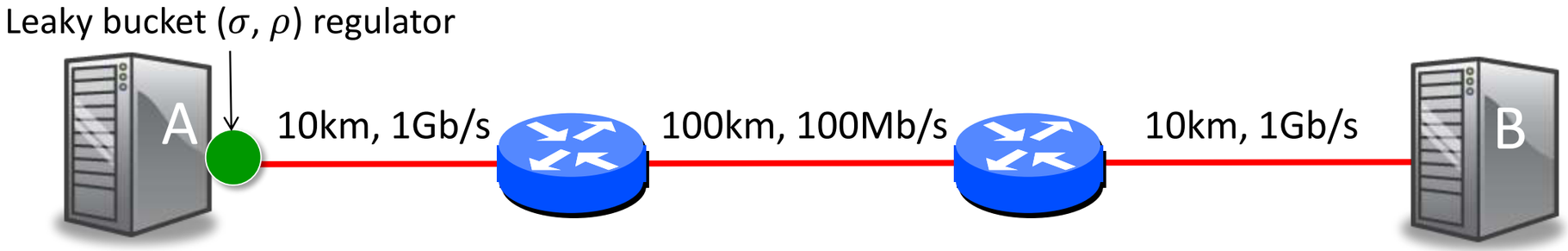
Putting it all together



If $\phi R > \rho$ and $b > \sigma$ then the end-to-end delay of <u>every</u> packet of length p $\leq 4\left(\frac{l}{c} + \frac{p}{R}\right) + 3\frac{b}{\phi R}$

An Example

Q: In the network below, we want to give an application flow a rate of 10Mb/s and an end to end delay of less than 4.7ms for 1,000 byte packets. What values of σ and ρ should we use for the leaky bucket regulator? And what service rate and buffer size do we need in the routers? (Assume speed of propagation, $c = 2 \times 10^8$m/s).

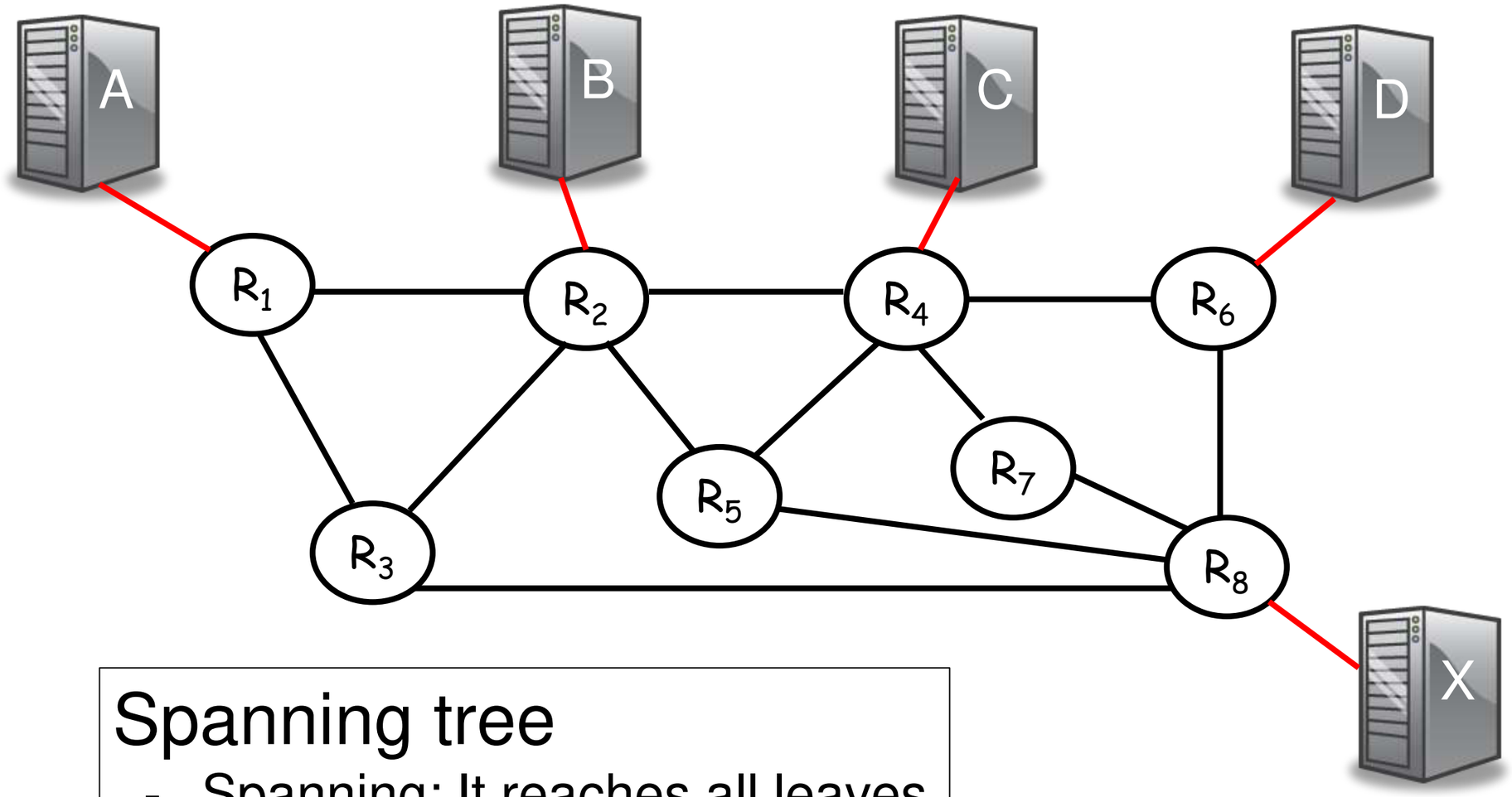


A: The fixed component of delay is $(120km/c) + 8{,}000bits(\frac{1}{10^9} + \frac{1}{100\times10^6} + \frac{1}{10^9})$ = 0.7ms, leaving 4ms delay for the queues in the routers. Let's apportion 2ms delay to each router, which means the queue in each router need be no larger than $2ms \times 10$Mb/s $= 20{,}000$bits (or 2500bytes). Therefore, the leaky bucket regulator in Host A should have $\rho = 10Mb/s$ and $\sigma \leq 20{,}000bits$. WFQ should be set at each router so that $\phi_i R \geq 10Mb/s$ and the flow's queue should have a capacity of at least 2500bytes.

Nick’s Review topics

1. Packet Switching
2. Routing, spanning trees and Dijkstra

Spanning Tree



Spanning tree

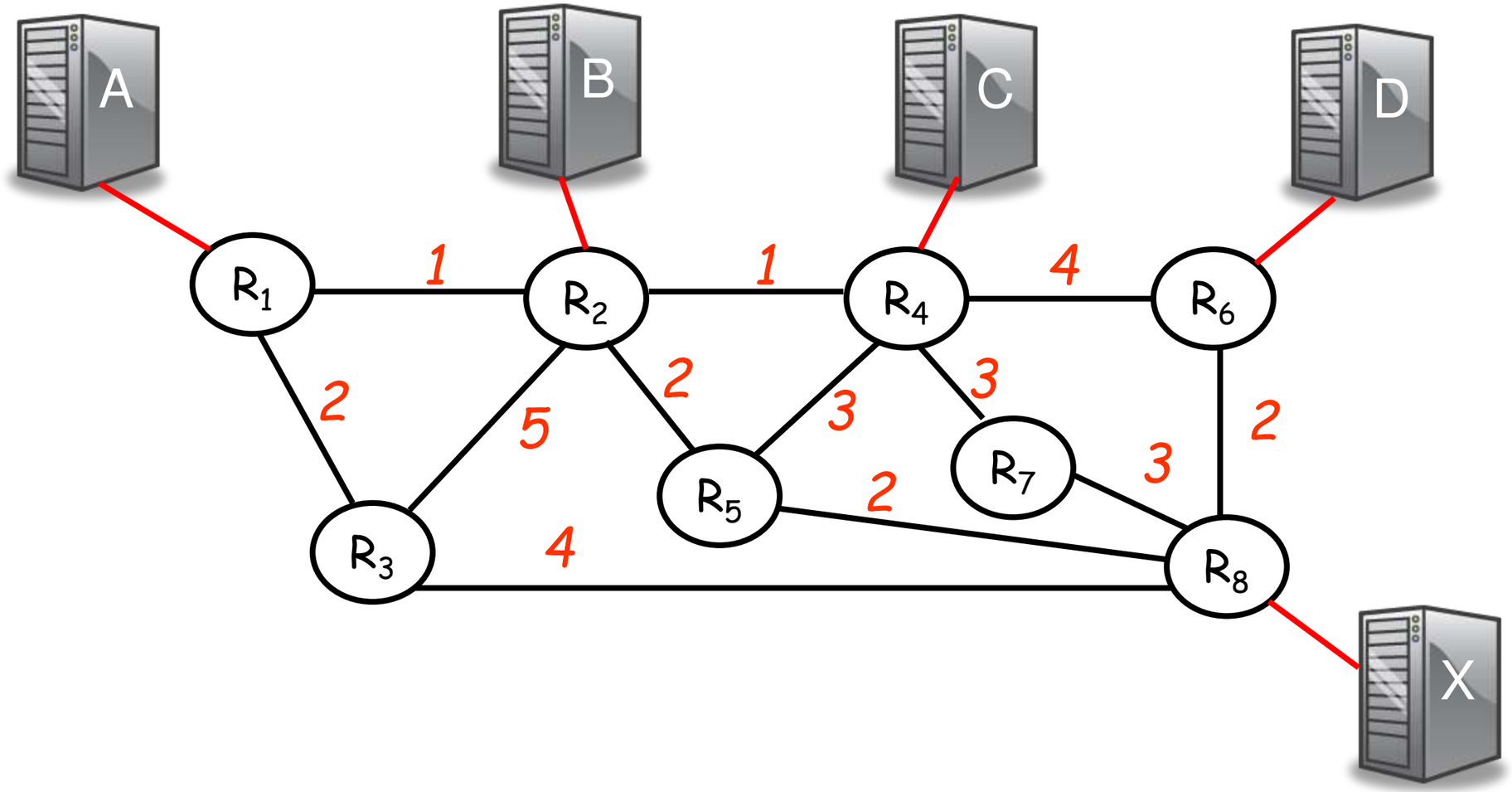
- Spanning: It reaches all leaves
- Tree: It has no loops

Dijkstra's shortest path first algorithm
(example of a "Link State Algorithm")

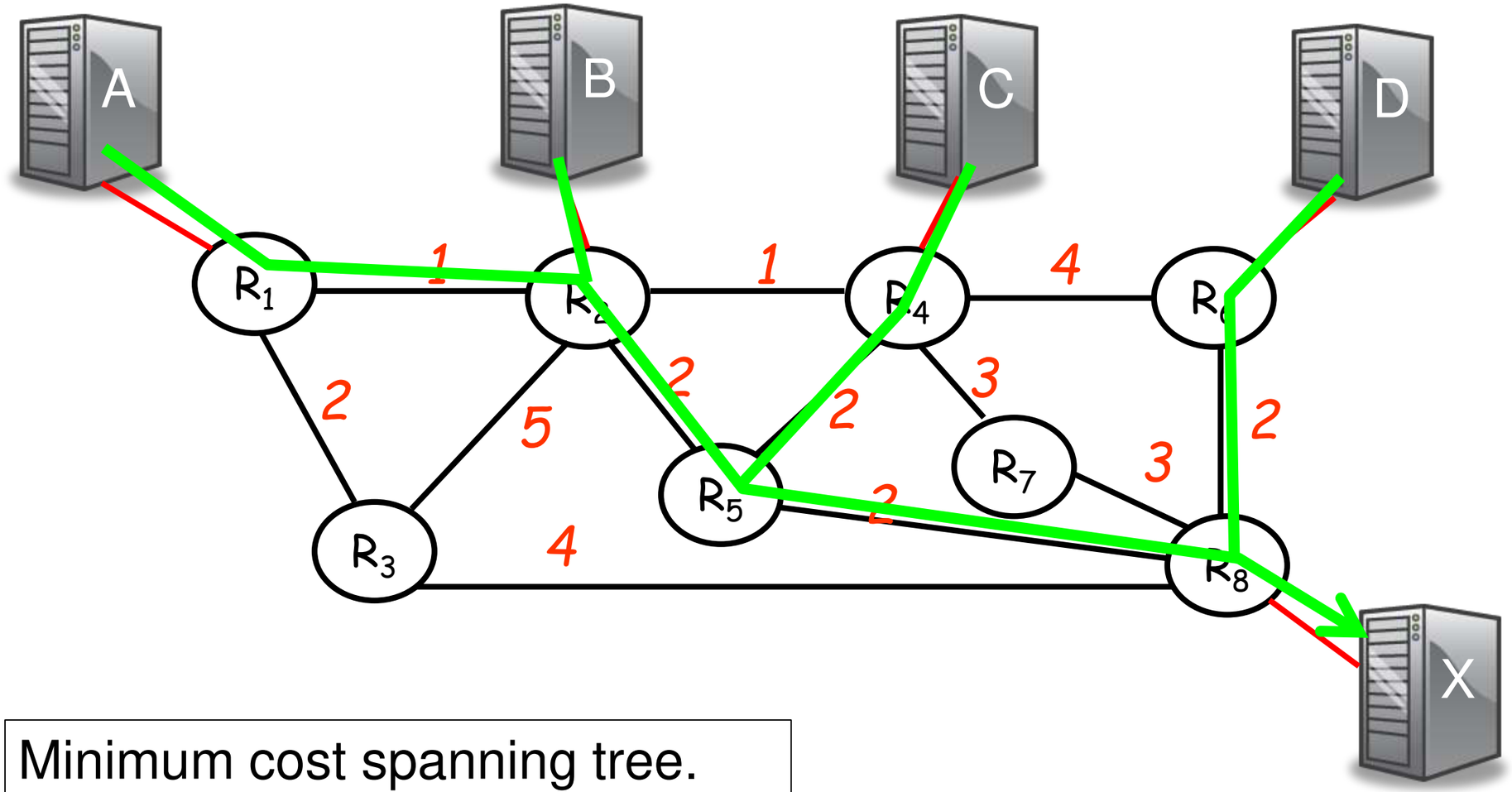
1. Exchange link state: A router floods to every other router the state of links connected to it.
   - Periodically
   - When link state changes
2. Run Dijkstra: Each router independently runs Dijkstra's shortest path first algorithm.

Each router finds min-cost spanning tree
to every other router.

Example Annotated Graph

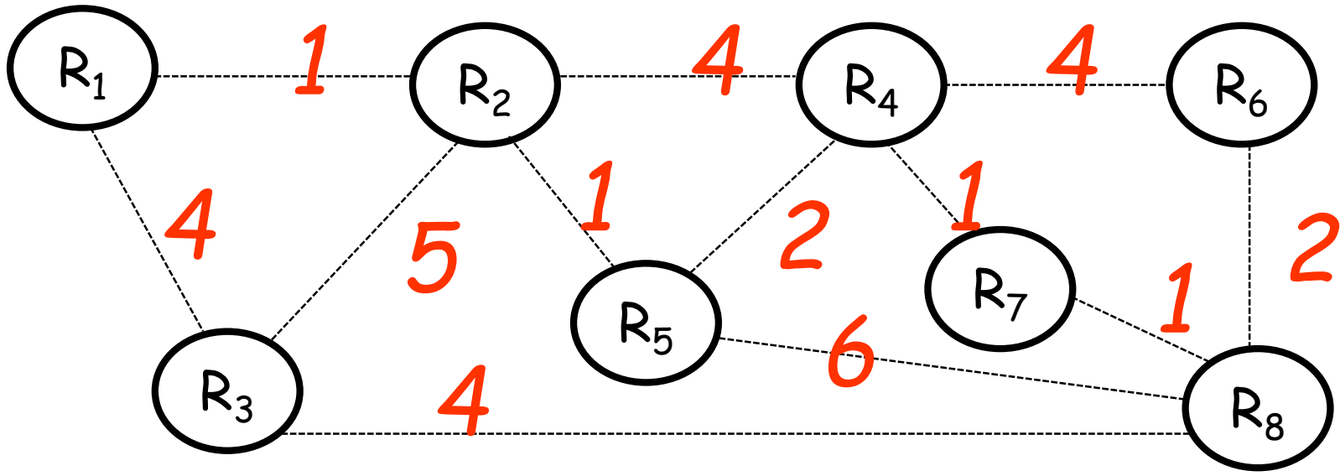


Example Annotated Graph

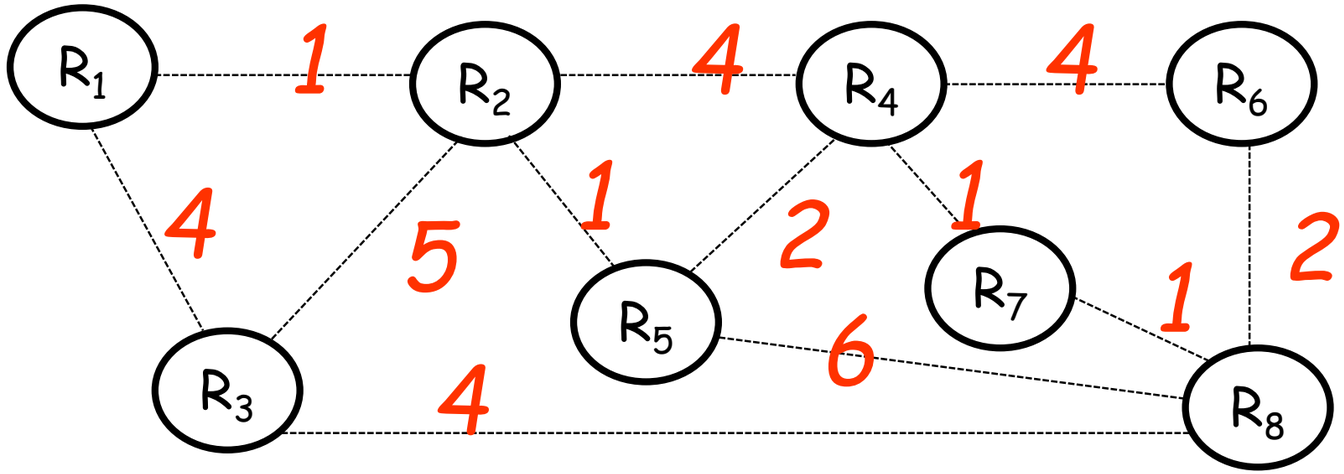


Minimum cost spanning tree.
In this case, simple.

An example: spanning tree rooted on R_8

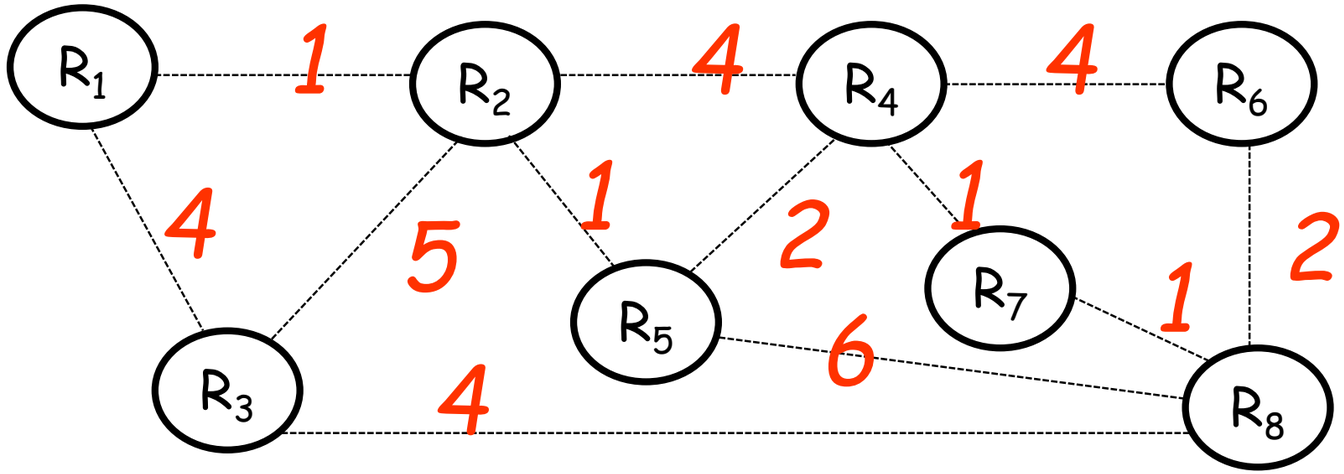


The algorithm



	0	1	2	3	4	5	6	7
Shortest Path Set	R_8							
Candidate Set	R_3R_5 R_6R_7							
Add	R_7							

The algorithm



	0	1	2	3	4	5	6	7
Shortest Path Set	R_8	R_8R_7	R_8R_7 R_6	R_8R_7 R_6R_4	$R_8R_7R_6$ R_4R_3	$R_8R_7R_6$ $R_4R_3R_2$	$R_8R_7R_6$ $R_4R_3R_2R_1$	$R_8R_7R_6R_4R_3R_2R_1R_5$
Candidate Set	R_3R_5 R_6R_7	R_3R_5 R_6R_4	R_3R_5 R_4	R_3R_5 R_2	R_5R_2 R_1	R_5R_1	R_5	Empty
Add	R_7	R_6	R_4	R_3	R_2	R_1	R_5	Done