Feb-20 Lecture

Putting Some of it Together

- Continue to probe the inter-workings of ISPs
  - Routing and peering
  - Route stability
  - IPSec then oh so briefly on firewalls

- We'll talk about additional services like content delivery next week

ISP/Core Impacts/Affects on TCP

- TCP behavior is heavily dependent on latency, jitter, and loss

ISP/Core Impacts/Affects on TCP

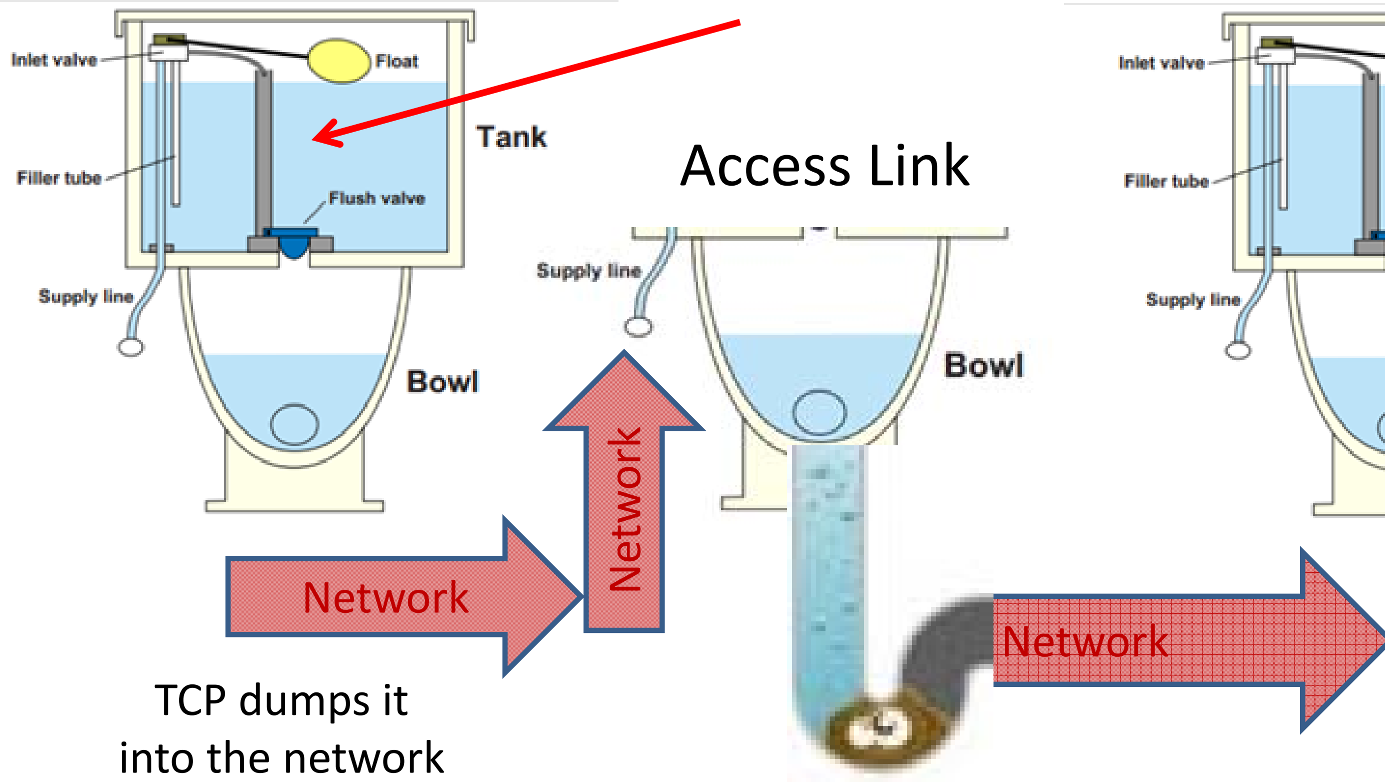
- TCP behavior is heavily dependent on latency, jitter, and loss
  - Latency: affects how quickly TCP gets through a "round" and therefore, add to its cong window
  - Jitter: also affects how responsive/accurate TCP can be in its timeout
  - Loss: more dramatic affect on cong window, but depends on the type of cong control strategy

The Source of Latency

Server

Application drops
a load of data…

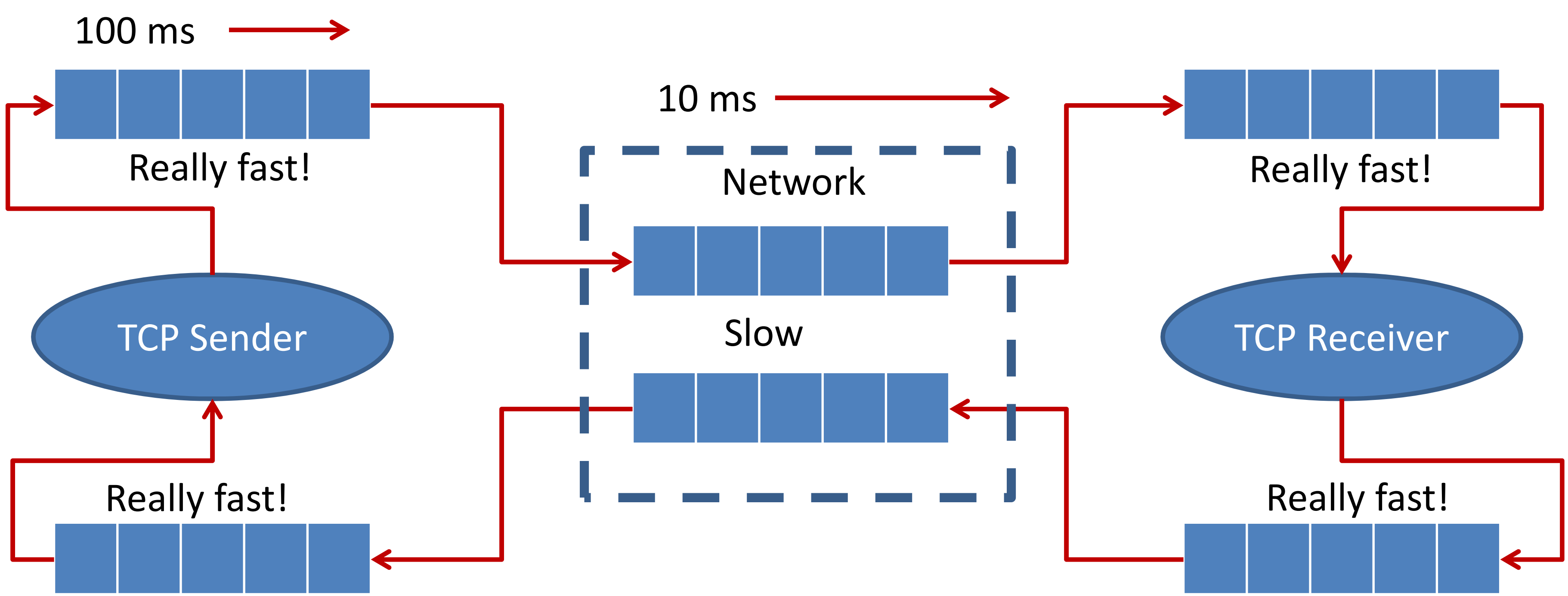
Client



TCP dumps it
into the network

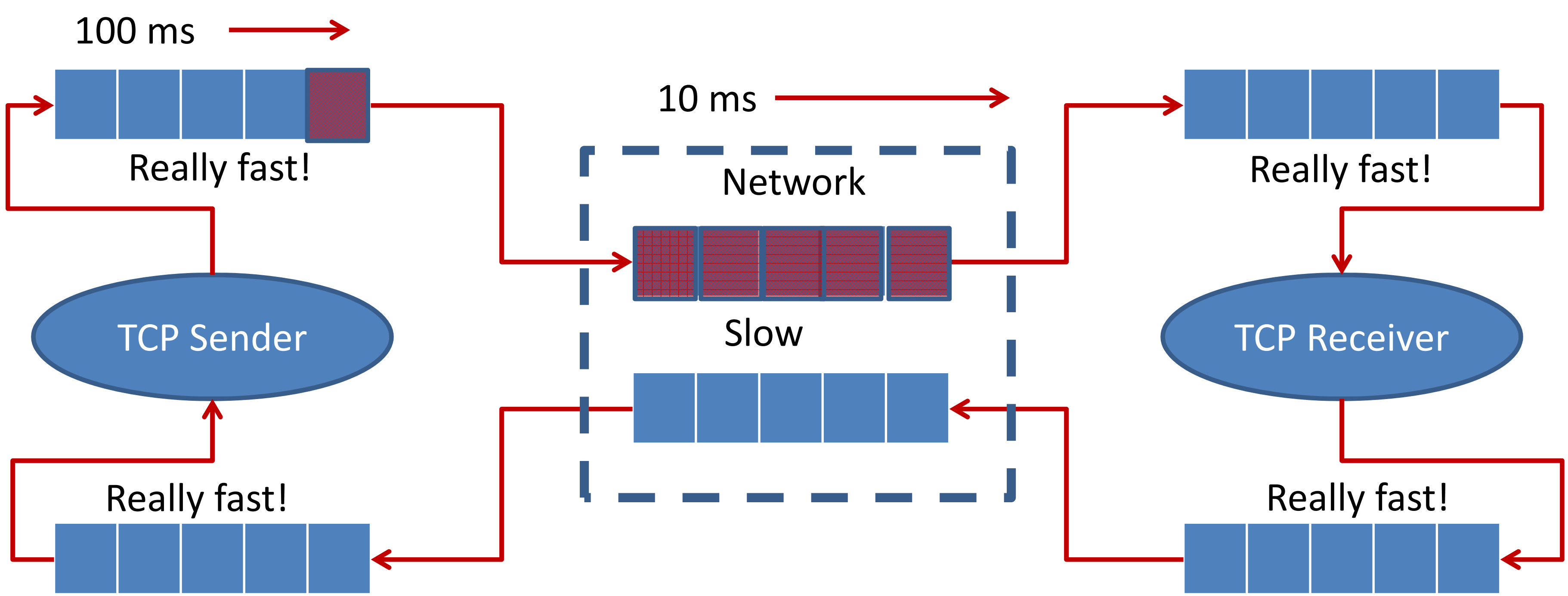
And it gets clogged!
This is latency!

TCP Control Loop



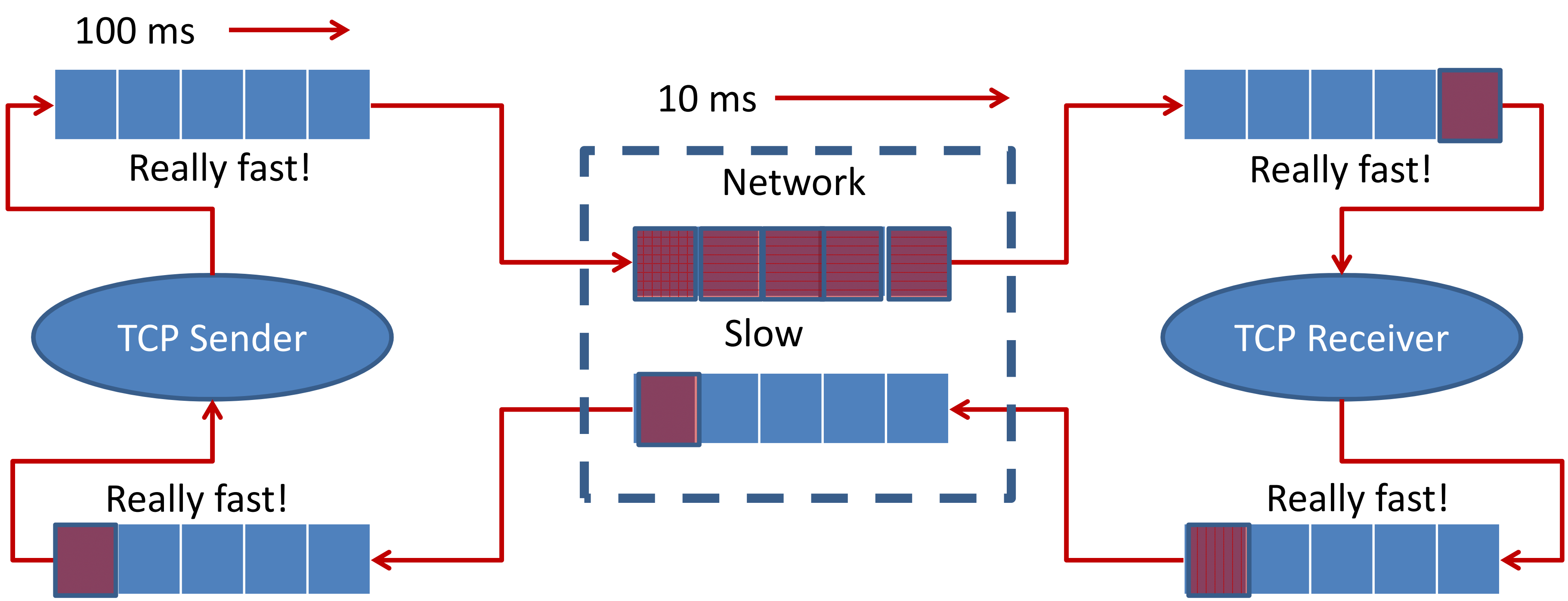
- Using Congestion Control and ACK clocking TCP will:
  - Calculate a SND.WND size that sends enough packets to keep the network busy for an RTT

TCP Control Loop



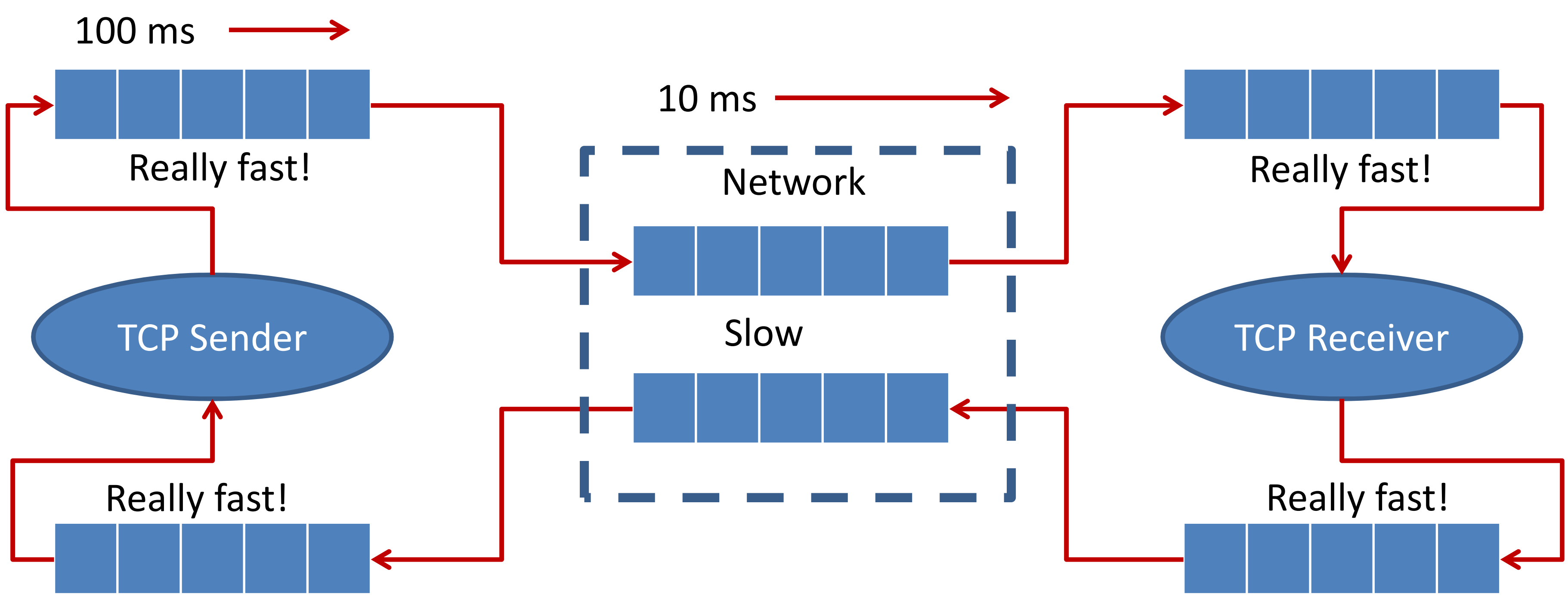
- Using Congestion Control and ACK clocking TCP will:
  - Calculate a SND.WND size that sends enough packets to keep the network busy for an RTT

TCP Control Loop



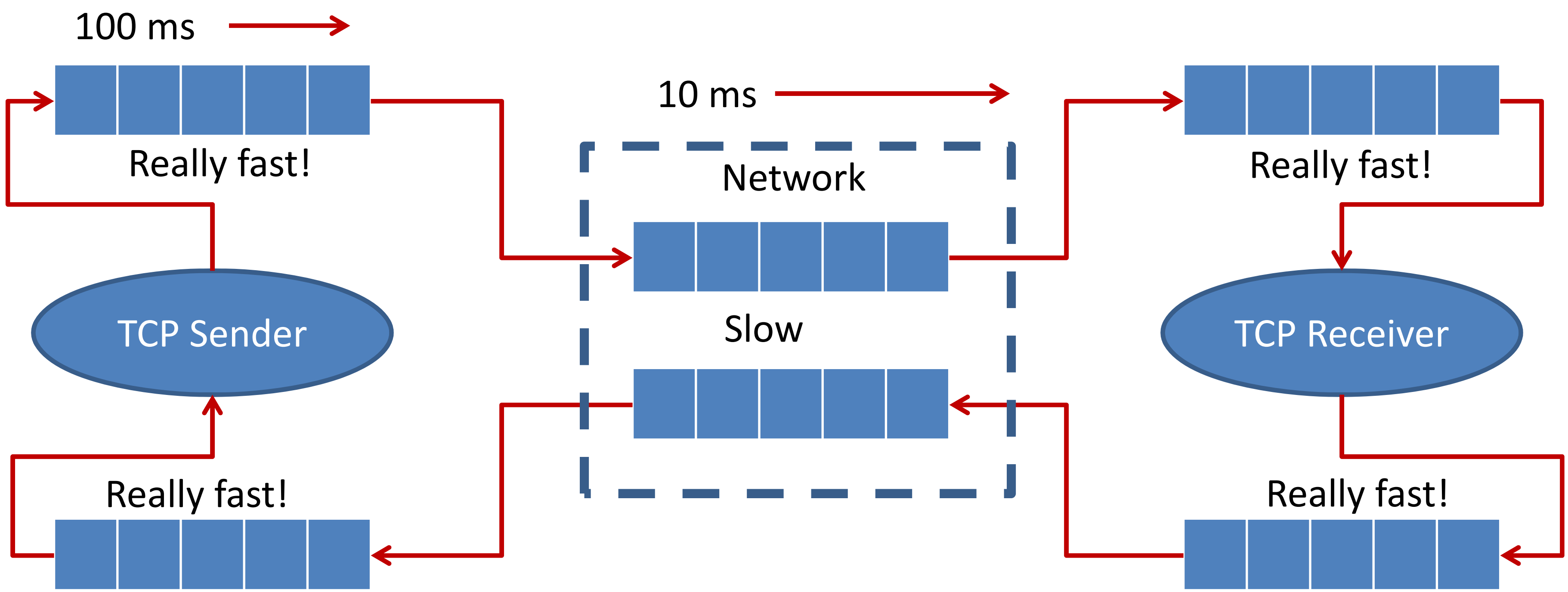
- Using Congestion Control and ACK clocking TCP will:
  - Calculate a SND.WND size that sends enough packets to keep the network busy for an RTT

TCP Control Loop



- By the time the first ACK arrives at the sender...
  - The sender queue will have drained
  - That's the goal at least

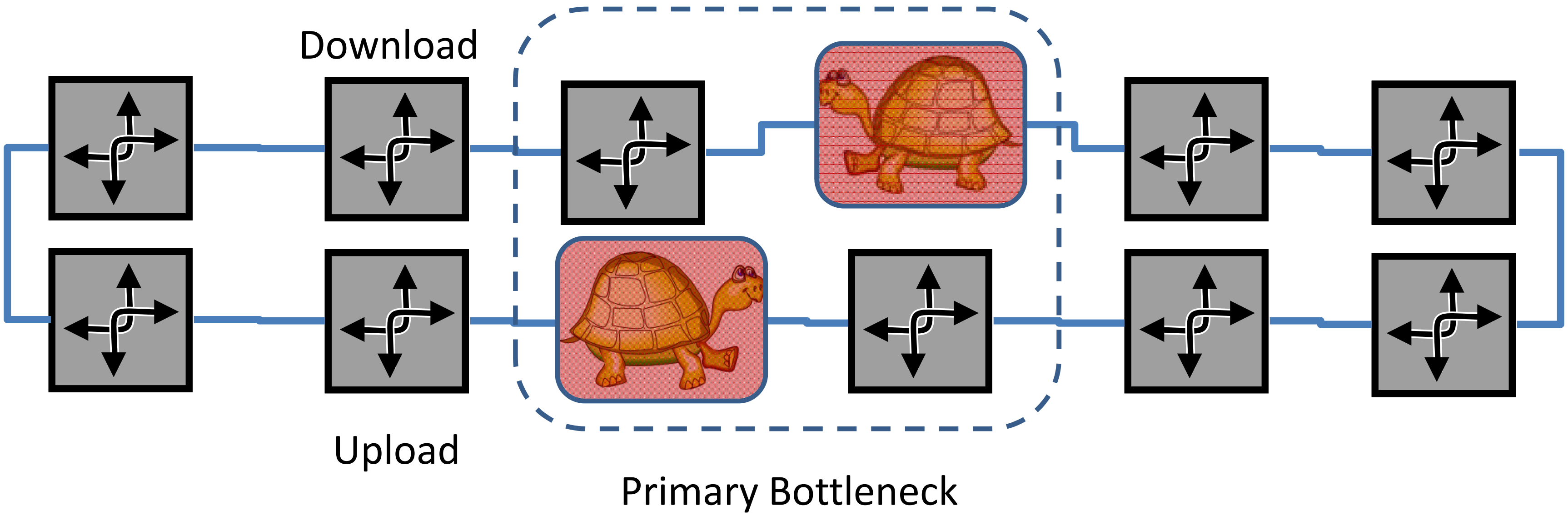
TCP Control Loop



- When packet loss does occur, it happens in one of two places:
  - On the ingress (packets arrive faster than can be processed)
  - On the egress (too many packets trying to go the same place(s))

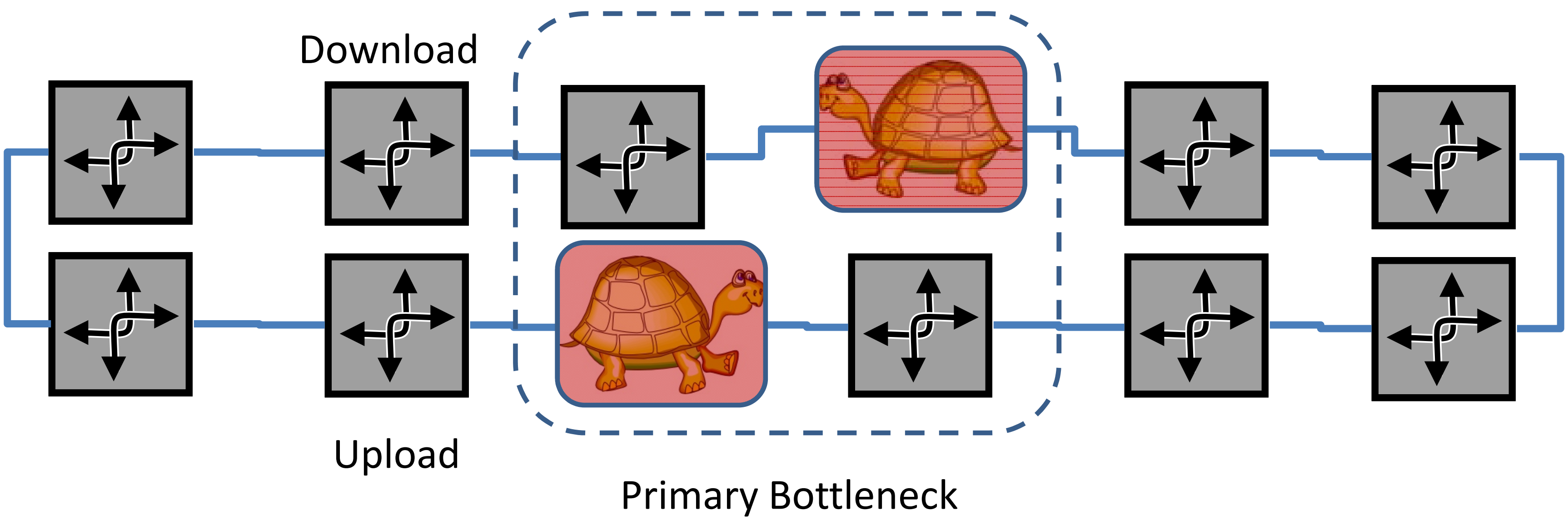
What Happens Next?

- Packets pile up at the slowest device (queue)
  - All the other queues are faster and will drain
  - Queue management must be applied at the slowest queue
- Why?



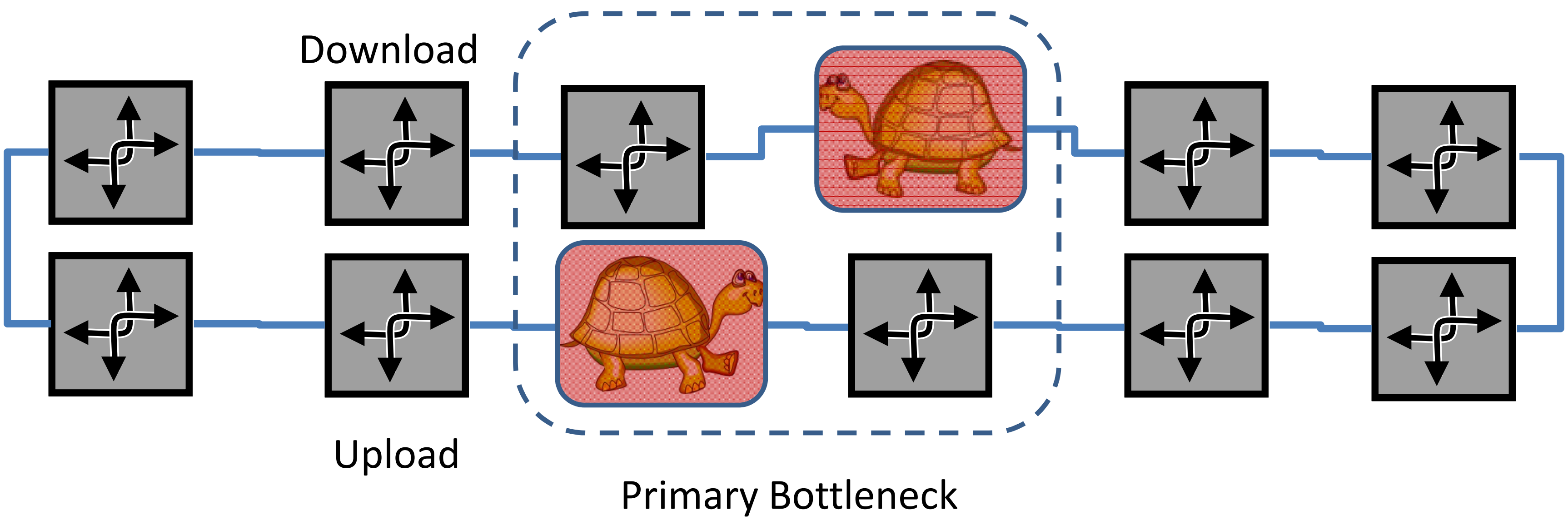
What is Bufferbloat?

- Memory is cheap
  - If packets are being lost on ingress/egress, just add more memory/buffer
  - …and more and more and more
- What are the consequences?

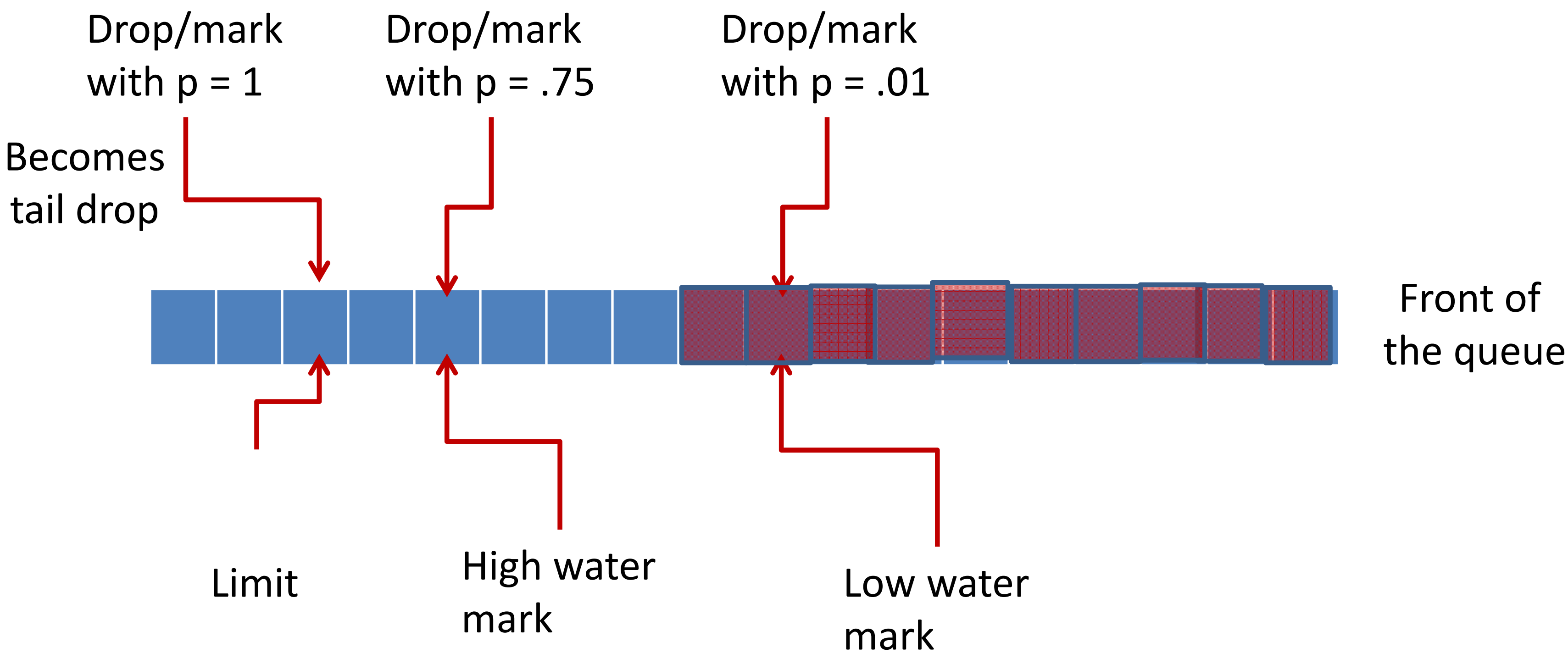


How To Solve Bufferbloat?

- Instead of trying to prevent packet loss, allow it to happen
  - TCP is good at adjusting
- Better yet, use packet loss to “signal” to TCP that congestion is occurring

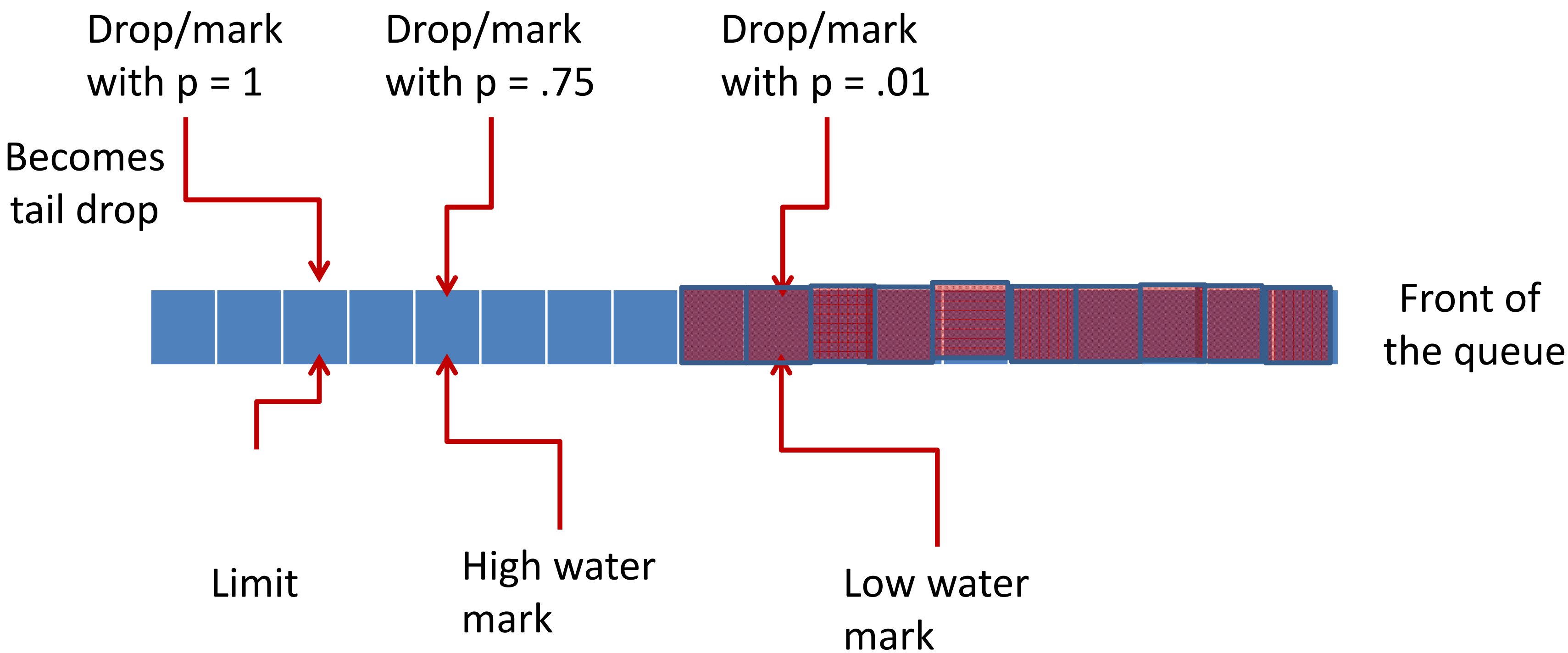


Random Early Detection (RED)



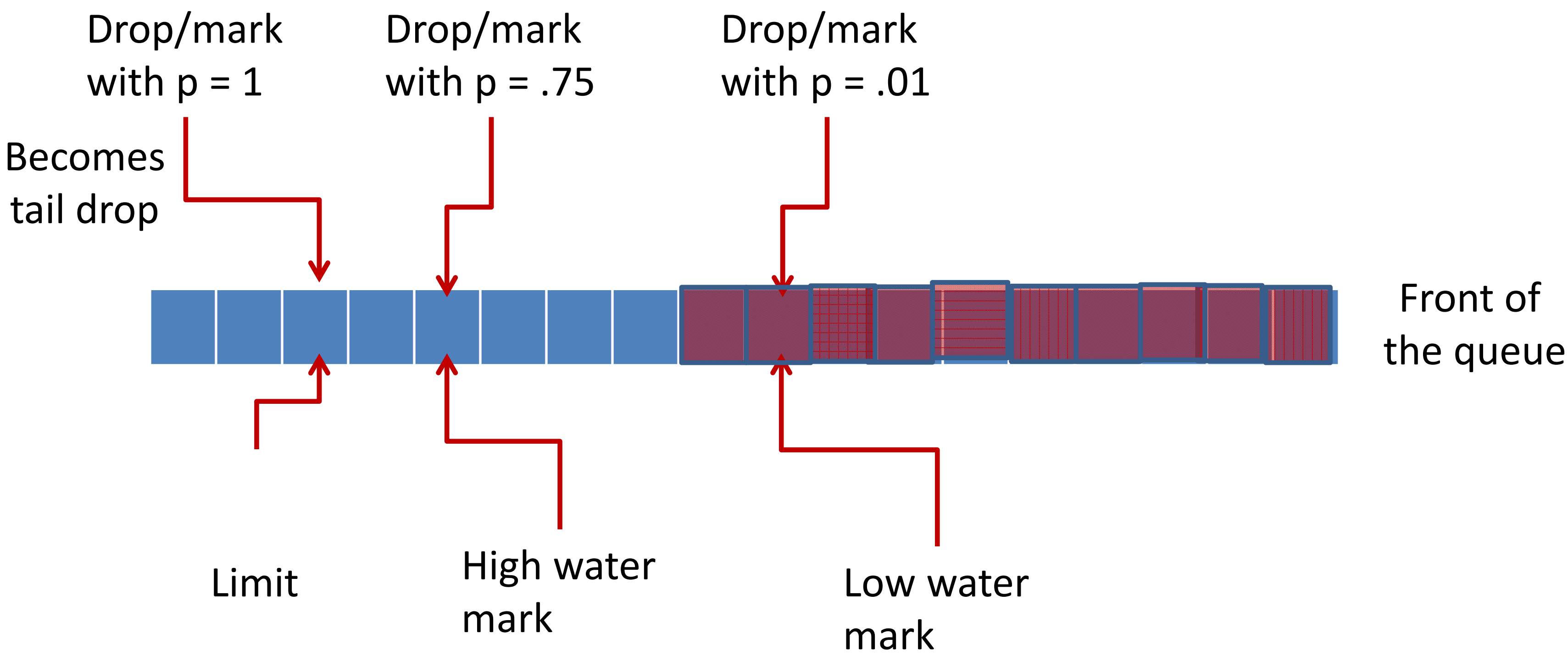
- When queue grows beyond "low water mark" start to drop packets with Prob=p (low p)
  - Can drop or just mark them

Random Early Detection (RED)



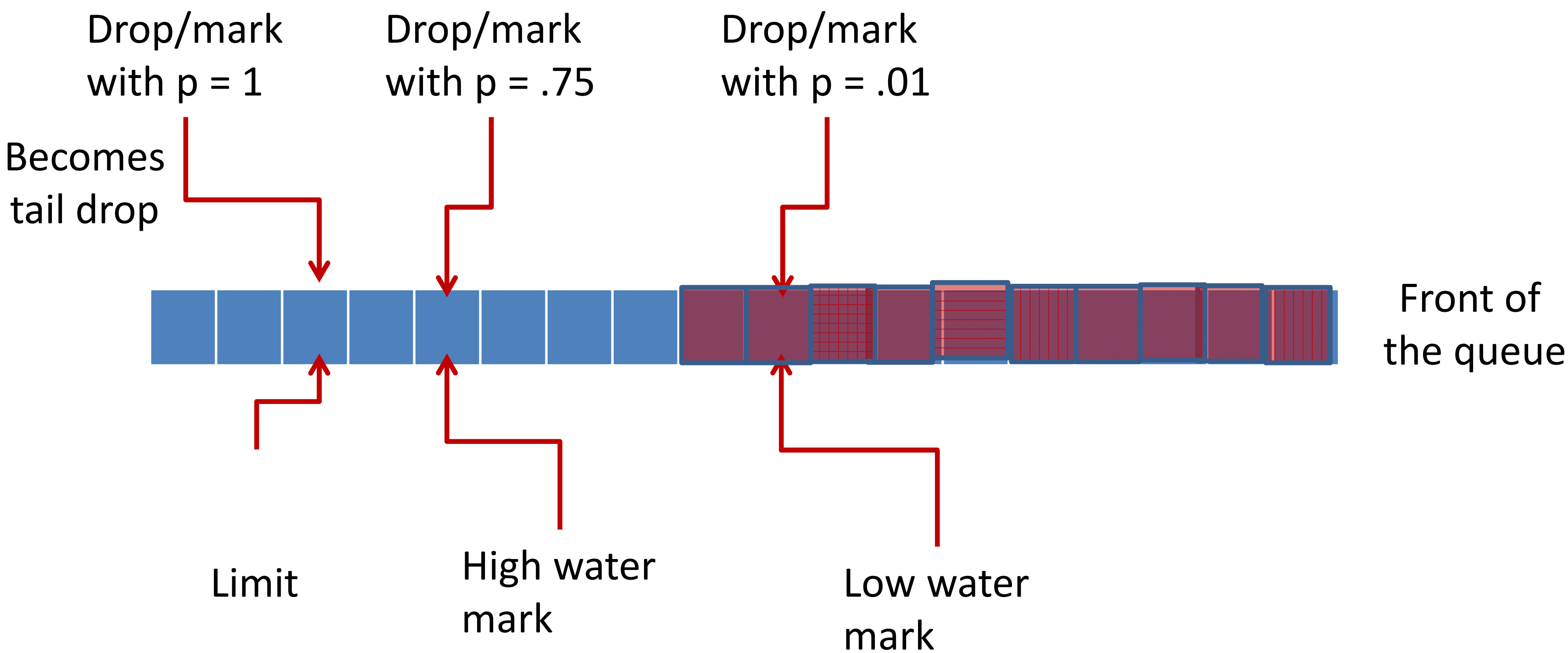
- As queue fills, become more aggressive on dropping probability
- When queue is full, nothing left to do, but drop everything

Random Early Detection (RED)



- As always, there's a challenge!
  - What to set queue thresholds to?
  - What to set p values at?

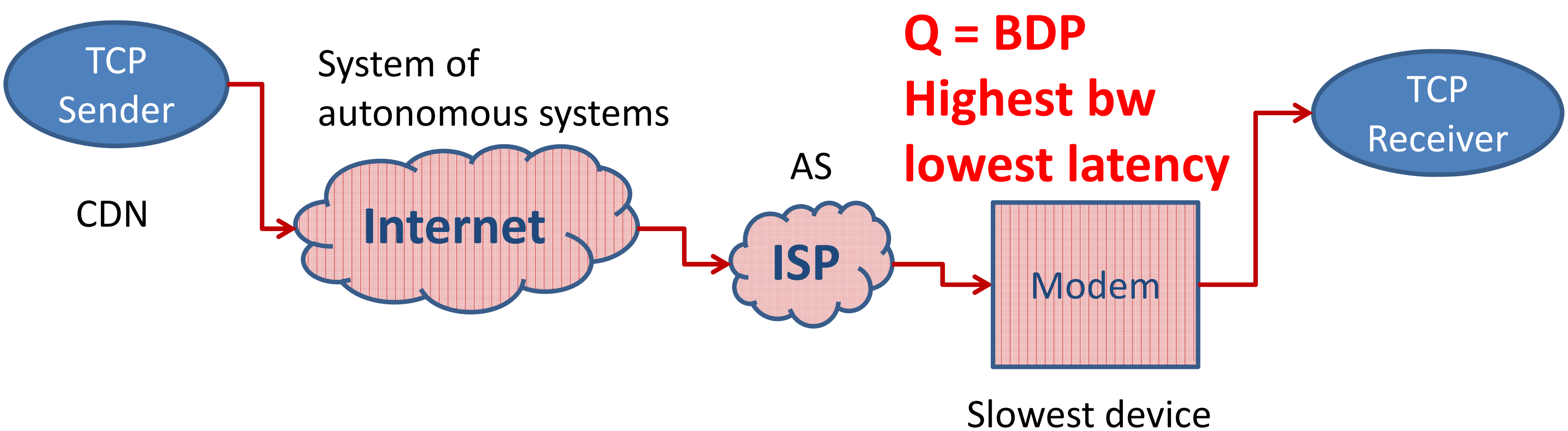
Random Early Detection (RED)



- The answer is based on the Bandwidth Delay Product

Bandwidth Delay Product (BDP)

- TCP assumptions:
  - Queue size = BDP
    - Highest bandwidth
    - Lowest latency
- What is BDP?
  - Bandwidth = The output rate of the slowest device in the path
  - Delay = Round Trip Time (RTT)



Queue Size Equations

- Queue Size = Bandwidth * Delay
  - Bytes = bytes / sec * sec

Queue Size Equations

- Queue Size = Bandwidth * Delay
  - Bytes = bytes / sec * sec
- Q = BDP
  - If Q = BDP → full bandwidth, lowest latency
  - If Q < BDP → lose bandwidth, lowest latency
  - If Q > BDP → full bandwidth, high latency **(Bufferbloat)**
- However, we do not know the bandwidth or the delay in advance.
  - So how do we set the queue size?

How do you configure RED?

- **Yes**, but tuning is supposed to be difficult…

1. Red parameters:
2. **Bandwidth** 2 Mbps
3. **Delay** 200 ms
4. Avpkt 1000 bytes
5. Limit 8 * max

1. Min = BDP
2. Max = 2 * Min
3. Burst = (min+min+max) / (3 * avpkt)
4. Limit = 8 * Max

5. Min 50,000 Bytes
6. Max 100,000 Bytes
7. Burst 67 segments
8. Limit 800,000 Bytes

- Just set some “reasonable” values and see how it works

Some Experiments

- When you plug in numbers, the queue size is a simple calculation (and fixed)

- Q = BD
- Q = 2 Mbps * 200 ms
  - Q = 400,000 bits (or 50kB)

- But what happens if the selected queue size isn't right?

Some Experiments

- When you plug in numbers, the queue size is a simple calculation (and fixed)

- Q = BD
- Q = 2 Mbps * 200 ms
  – Q = 400,000 bits (or 50kB)

- But what happens if the selected queue size isn't right?

Queue Size Equations

- Q = BDP
  - If Q = BDP → full bandwidth, lowest latency
  - If Q < BDP → lose bandwidth, lowest latency
    - Queue is too small and more packets are dropped than it should be
    - TCP responds by reducing how much data is sent
  - If Q > BDP → full bandwidth, high latency **(Bufferbloat)**
    - Queue is too large
    - TCP doesn't see any drops
    - But RTT grows to unreasonable lengths

Next Steps

- RED was one of the first solutions
- Many others since RED
- They create complex mathematical models that can use network information to estimate queue size
  - Packet pairs, ping times, statistical distributions

The Evolution Continues

- Where once ISPs were just conduits for traffic, now there is much more engineering
- Early efforts were abysmal failures
  - See, e.g., integrated services (int-serv), the Resource Reservation Protocol (RSVP), and ATM
- More recent efforts have taken lessons from the past and been less processing-intensive
  - MultiProtocol Label Switching (MPLS)
  - Now Software Defined Networking