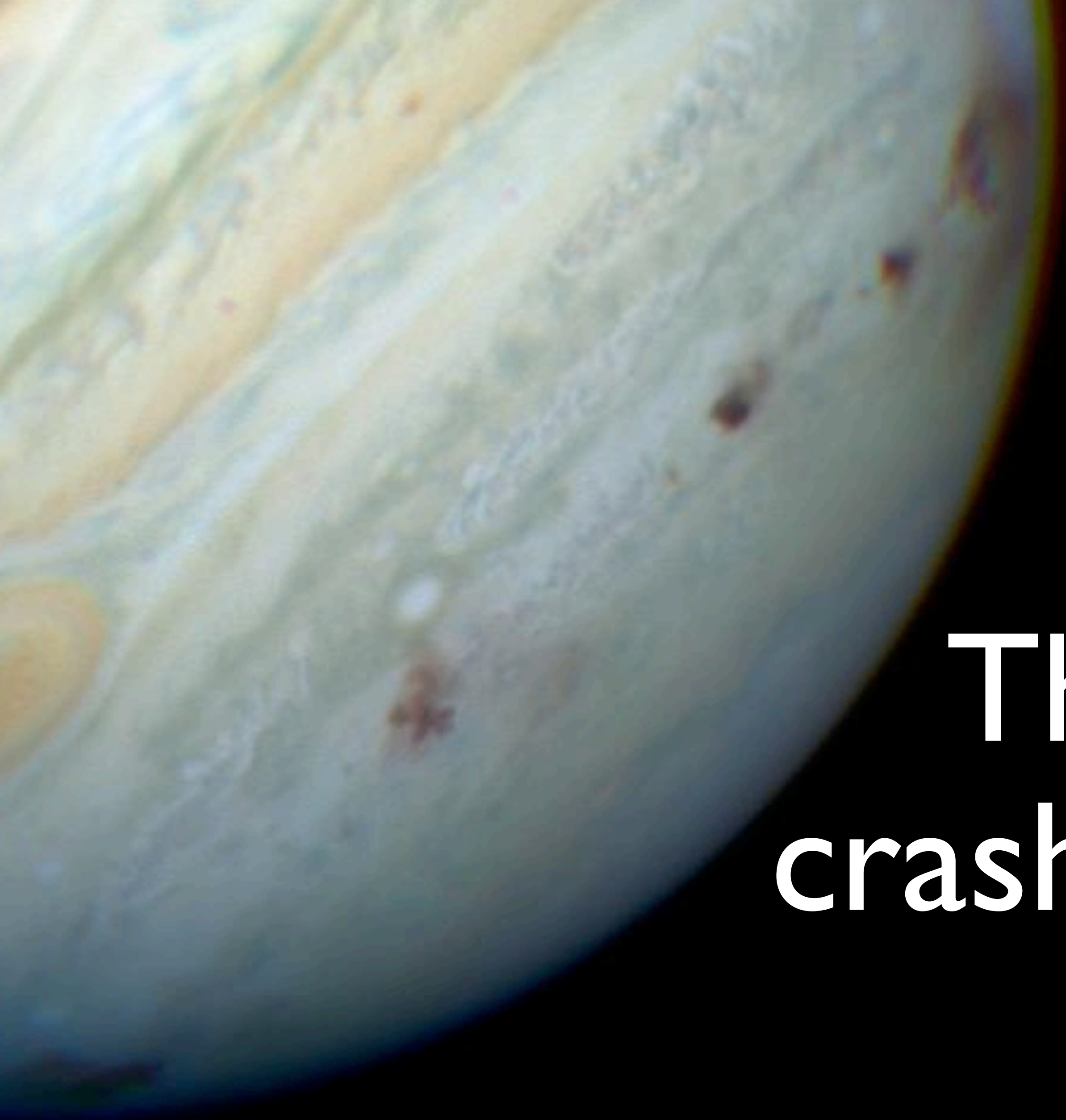


Things flying around, crashing into one another

Van Jacobson
van@parc.com

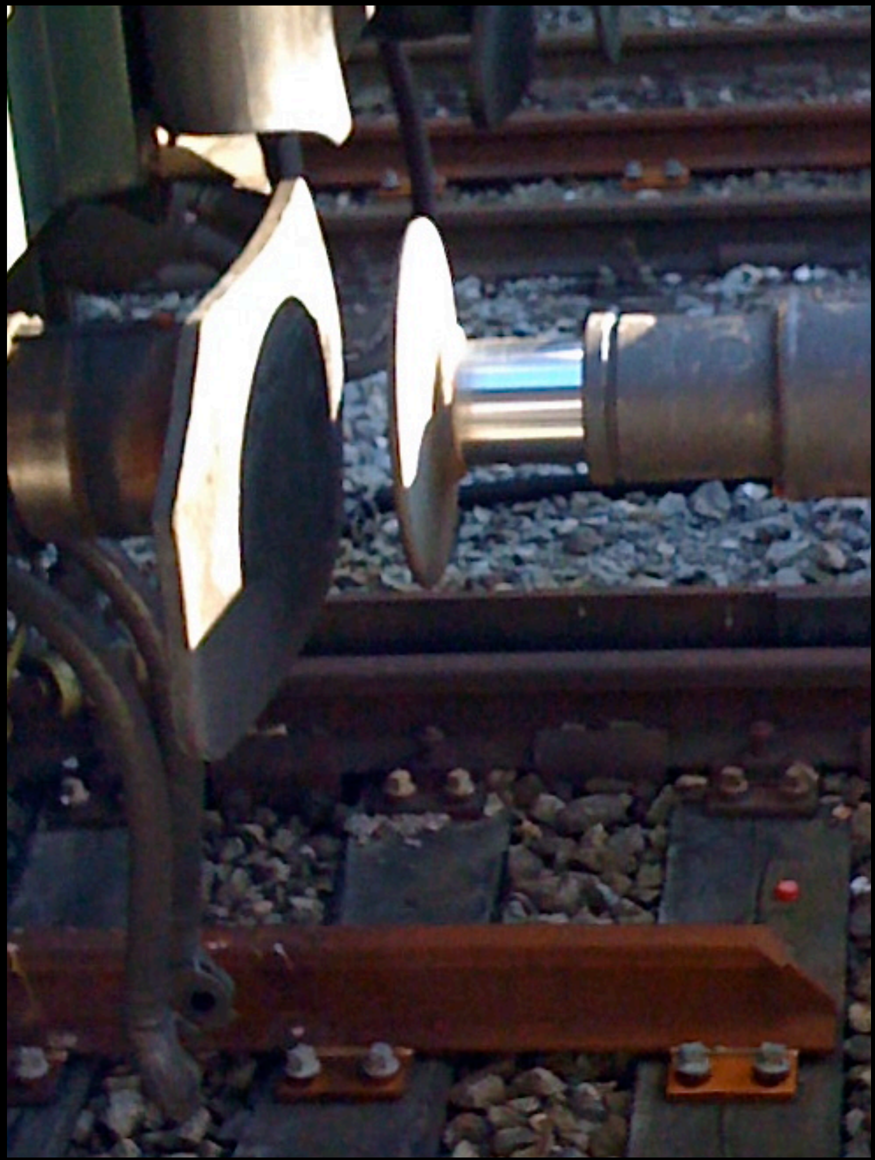
Netarch09
Ascona, Switzerland
16 March 2009

- The way we try to understand network behavior is strongly conditioned by a hundred years of well-developed teletraffic theory.
- But telephones were developed in a world without buffers so that theory is missing almost everything that drives current internet traffic.
- A more ‘kinetic’ view of traffic can lead to a better understanding of what we observe.

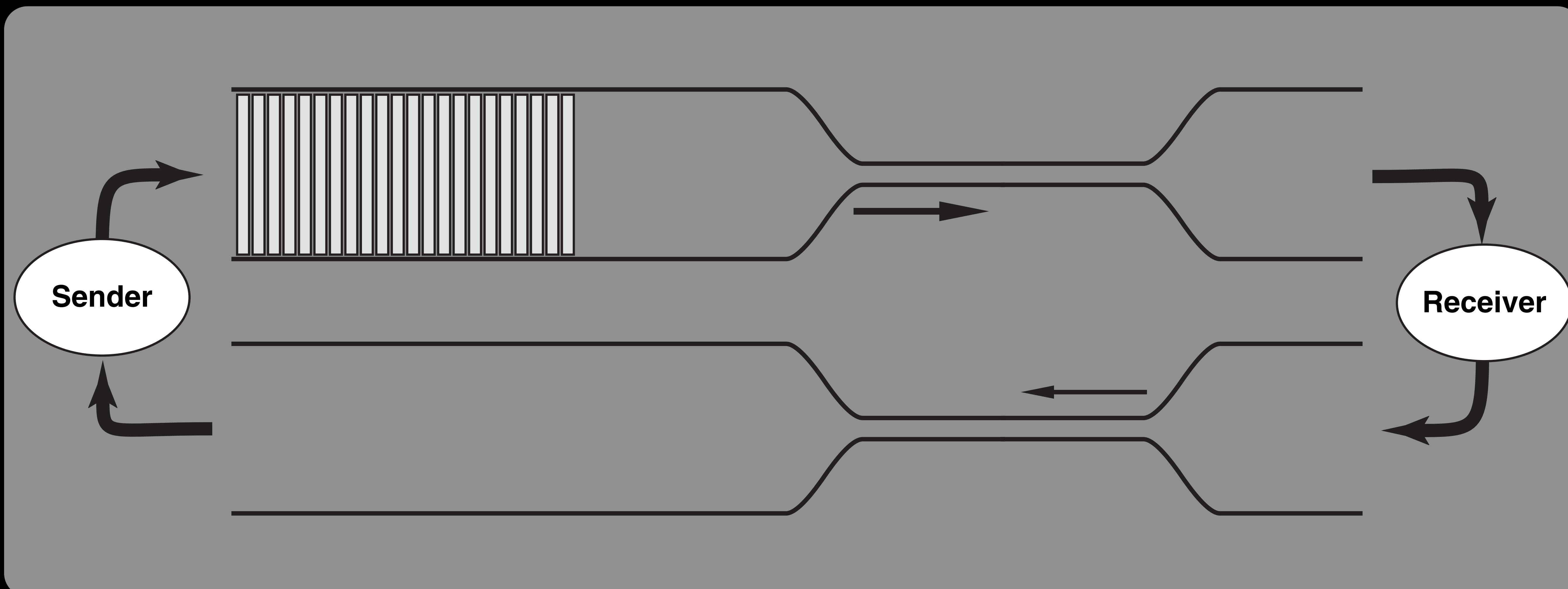
Buffers

To make a train, cars bang together to latch up their couplings. Buffers make the process non-destructive.

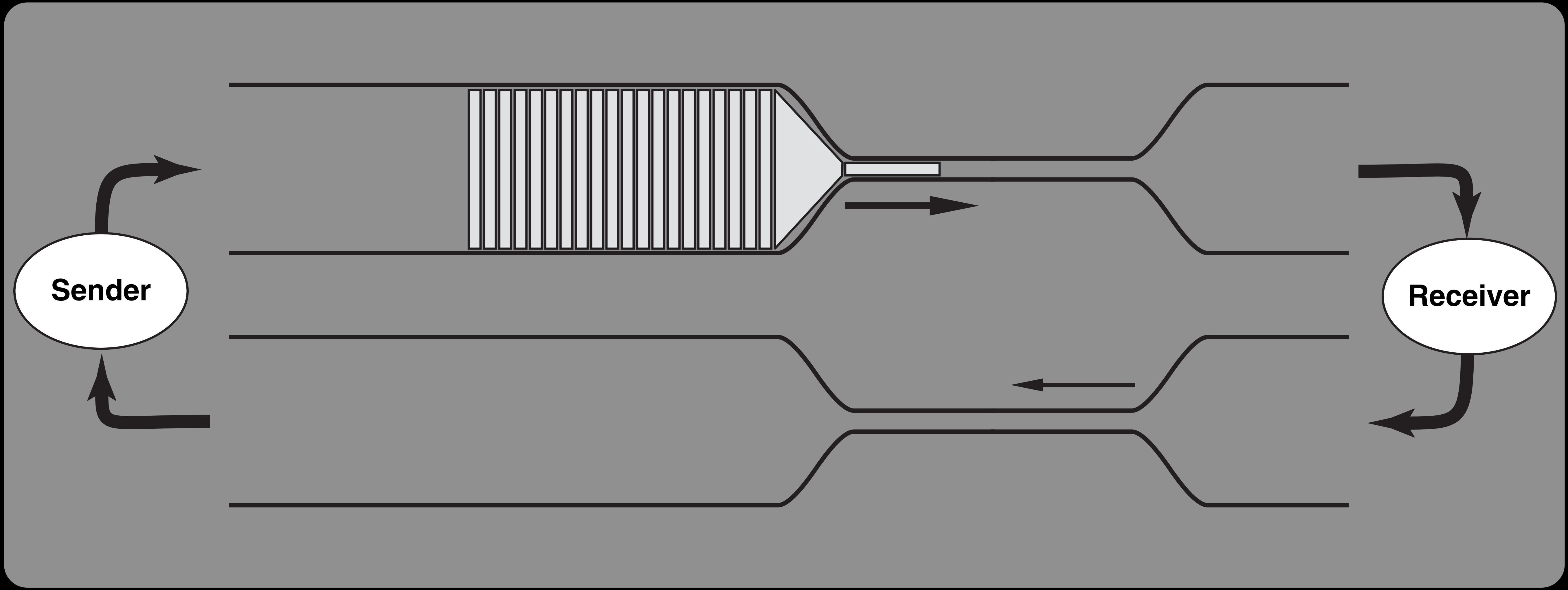
Packet conversations get started same way.



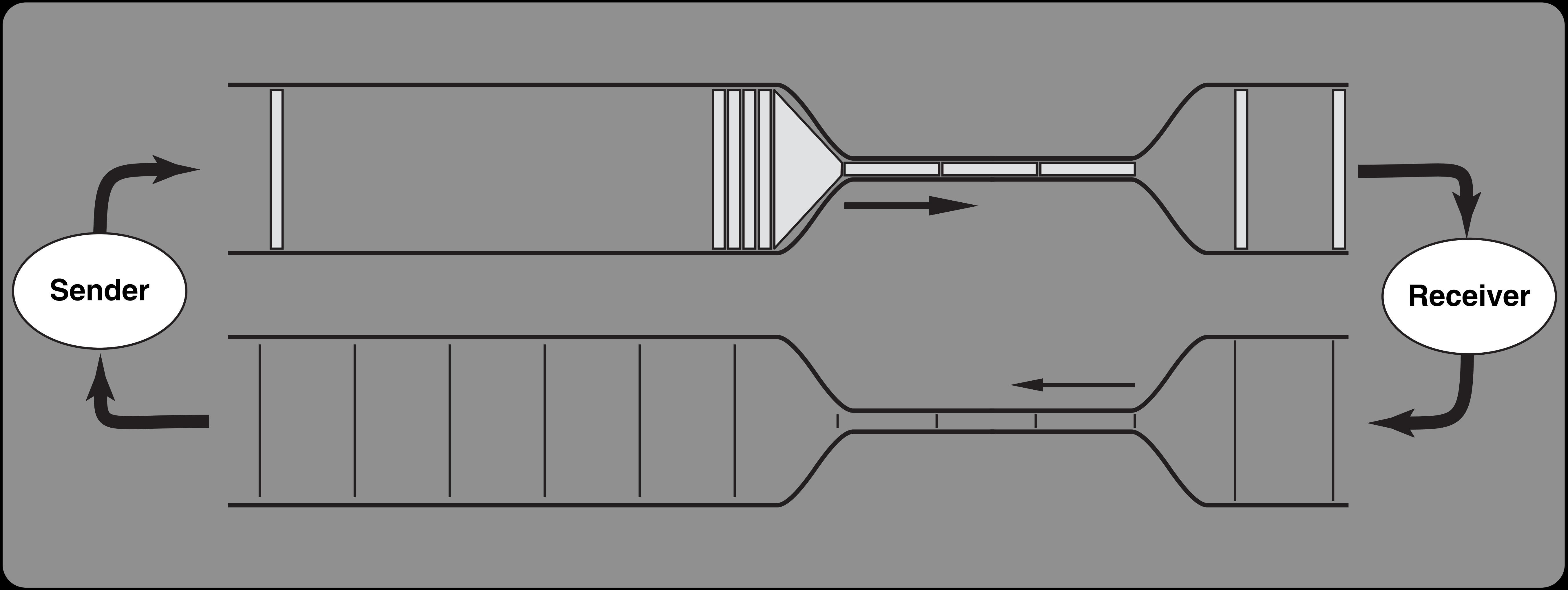
Sender injects a window's worth of packets



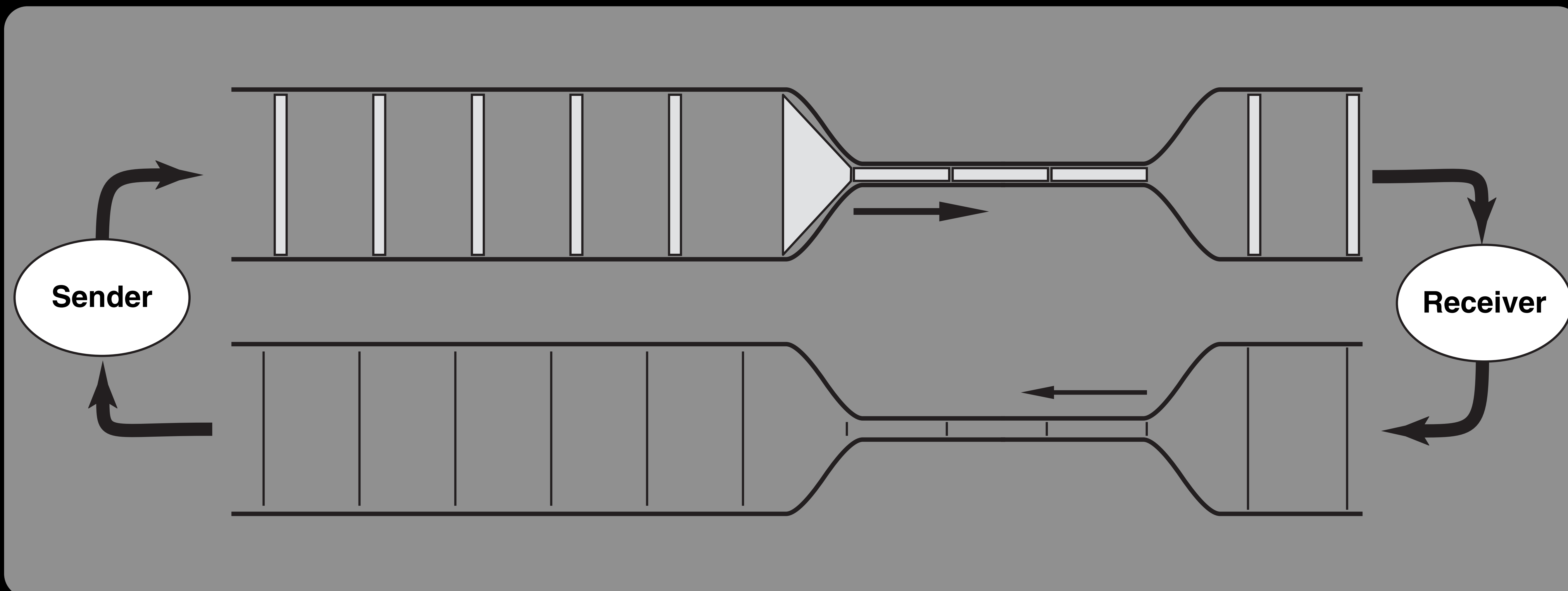
Packets reach high to low bandwidth transition



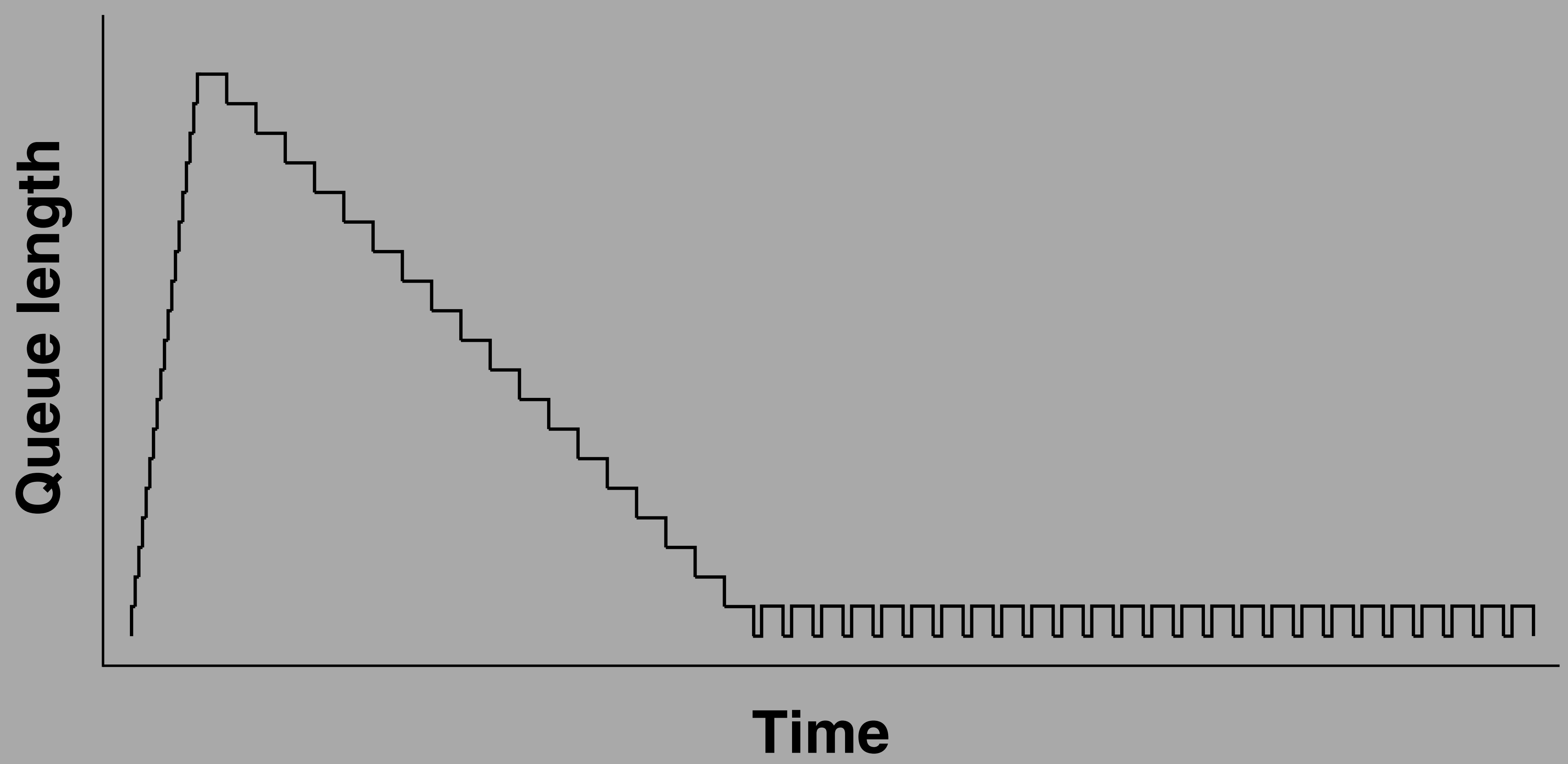
First ack returns and releases next data packet



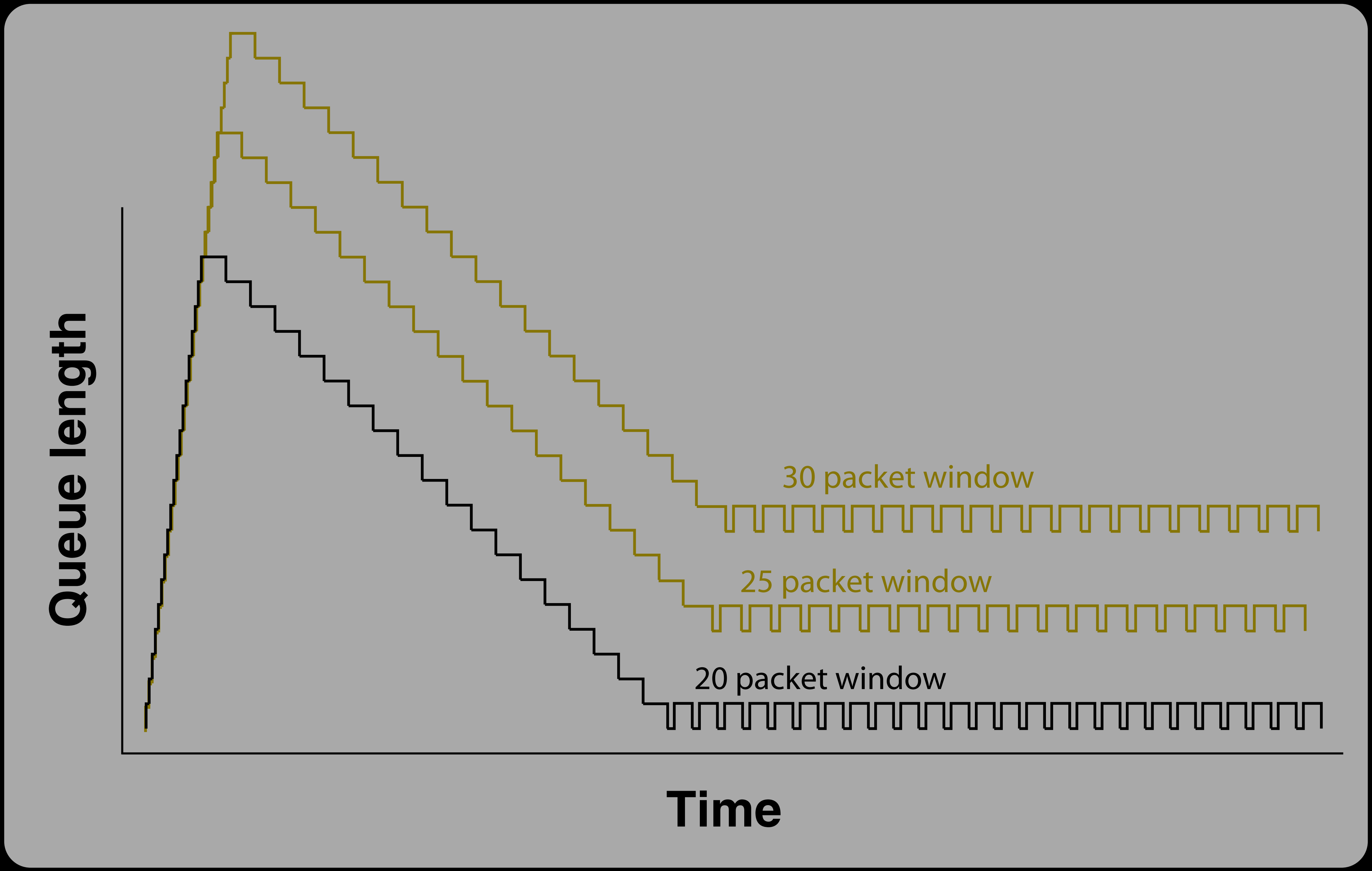
Steady-state reached



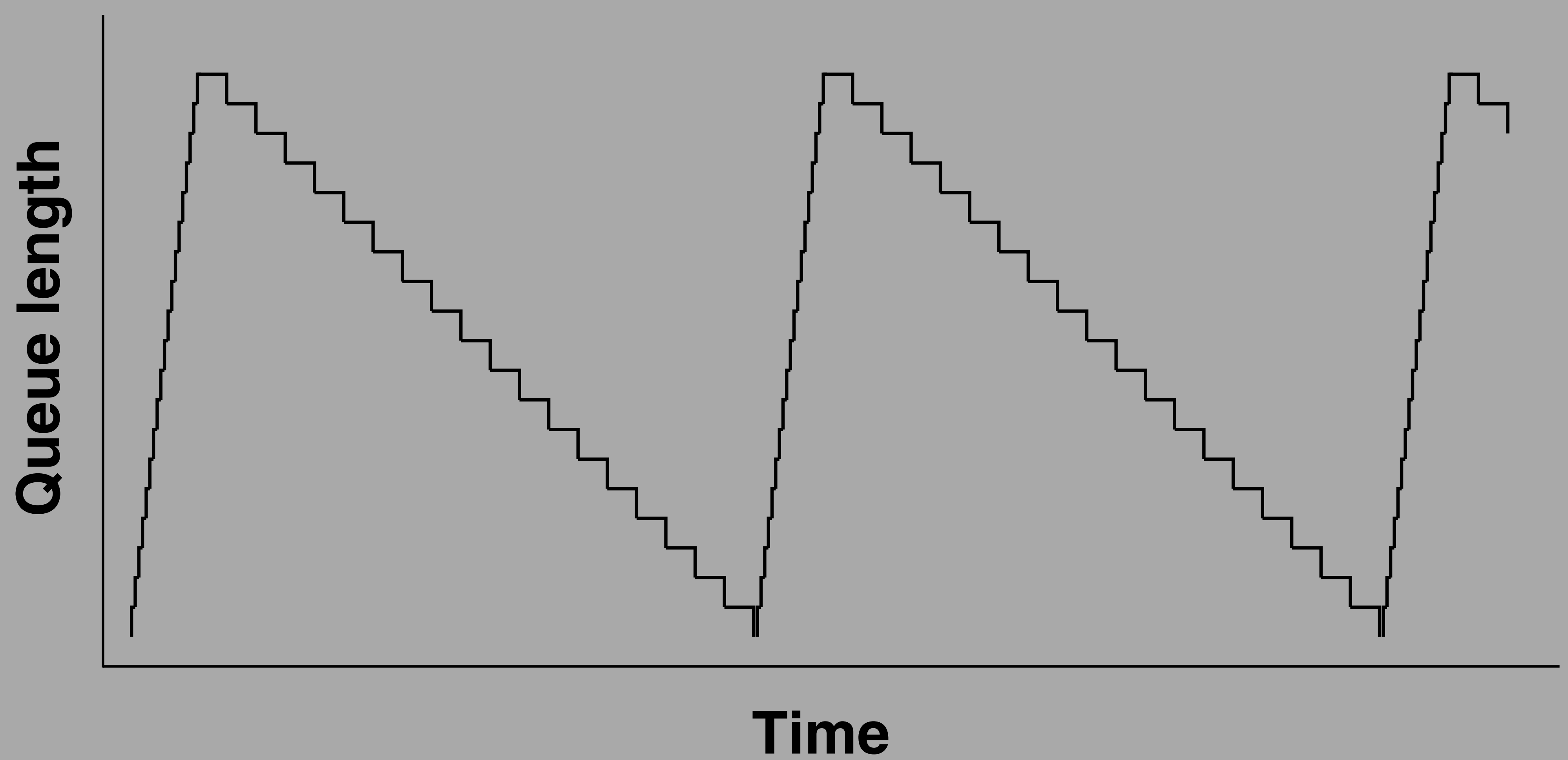
Queue behavior at the fast-to-slow transition

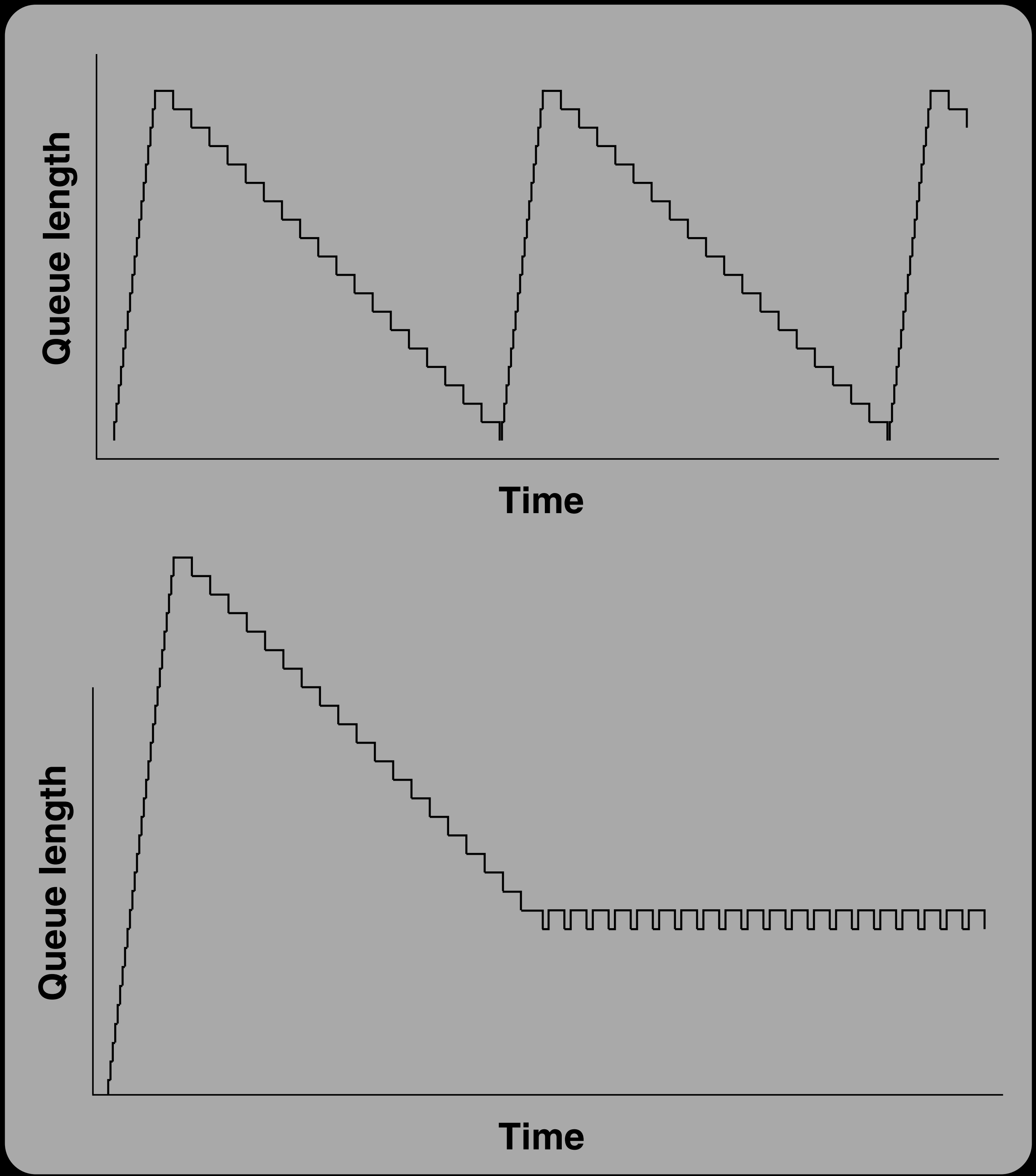


Queue behavior at the fast-to-slow transition



Queue behavior with ack-per-window receiver





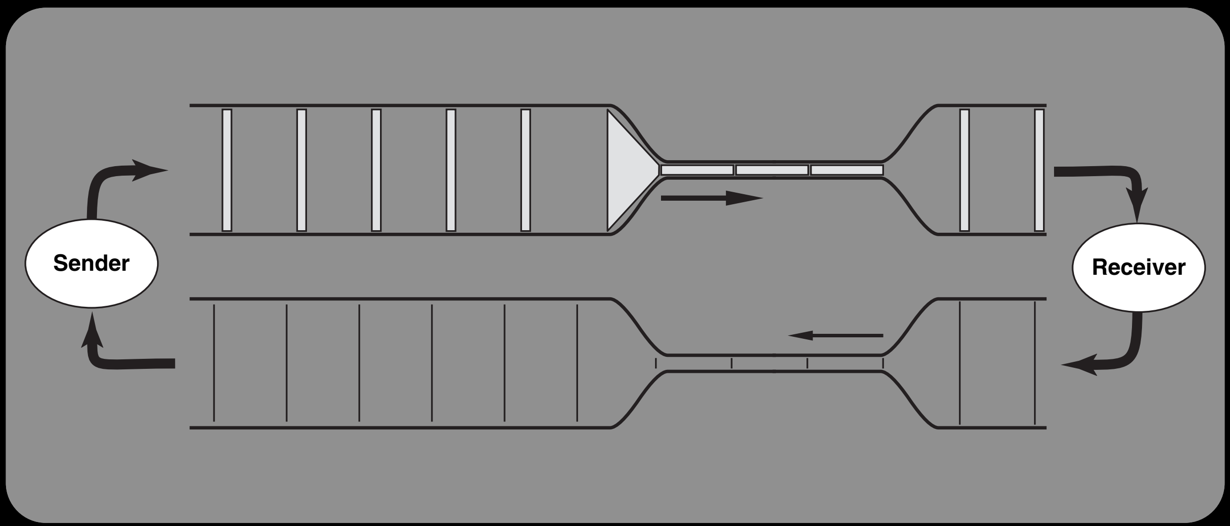
- Both of these queues have a long-term mean of ten packets.
- Upper queue is essential, lower is excess.
- Note that a long-term *min* correctly computes the excess queue.

Average queue length is *uncorrelated* with load

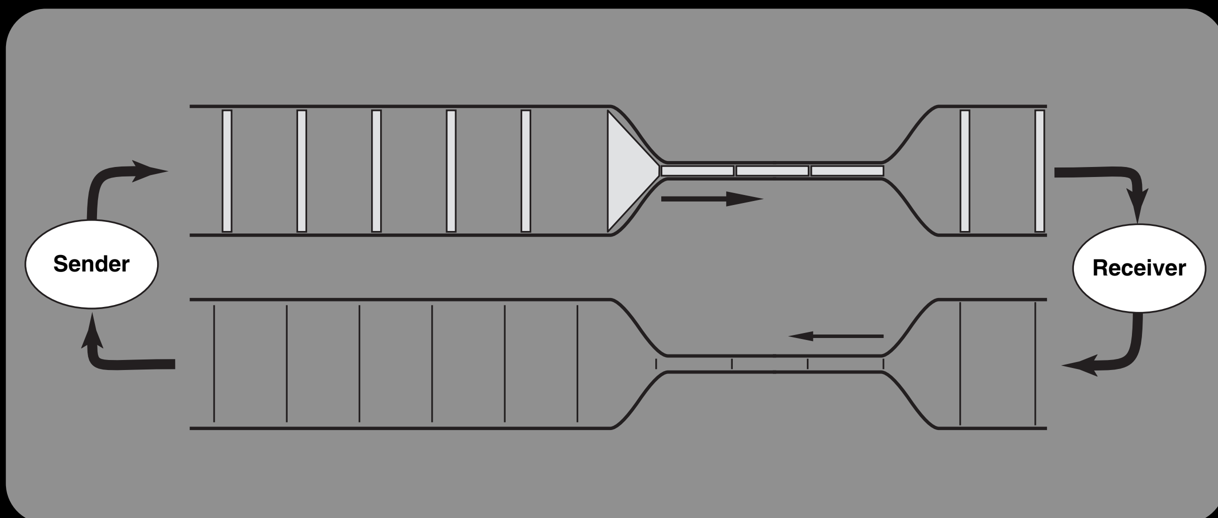
- Minor implementation variations give wildly different queue mean and variance for the same bottleneck load.
- Using mean queue length as a proxy for congestion is usually a performance disaster.
- Active Queue Management really needs to co-evolve with transport (Kelly, 2003; Briscoe, 2008).

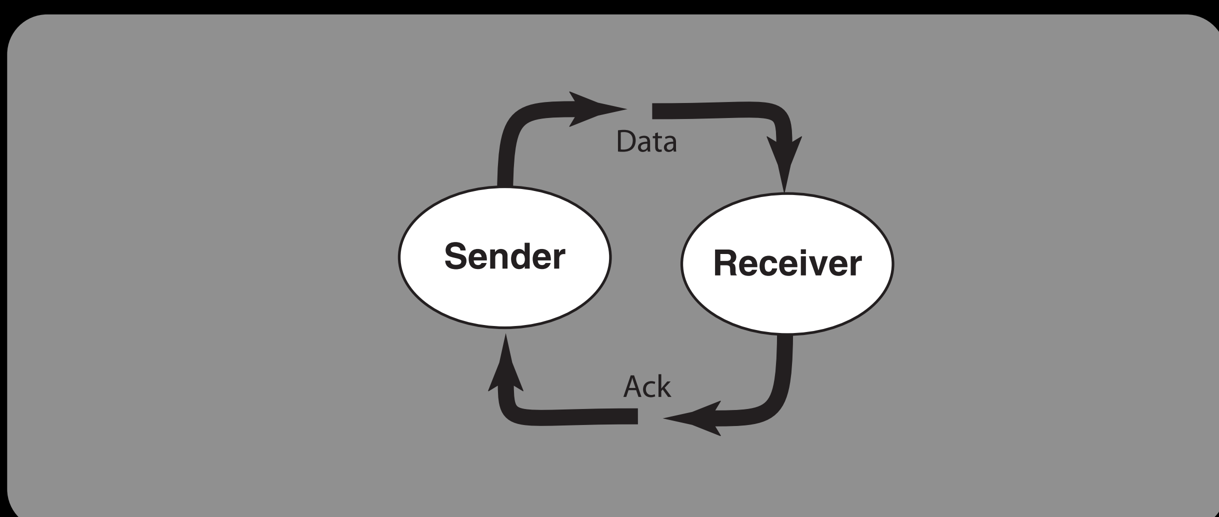
And what if everybody does it?

(Aggregated transport dynamics)



- One way to look at the transport dynamics is to think of the acks providing a ‘clock’ to the sender.





- One way to look at the transport dynamics is to think of the acks providing a 'clock' to the sender.

- A more general view is to ignore the details and just look at the sender and receiver's reciprocal exchange – their flow balance.

- In *Reversibility and Stochastic Networks* (Wiley, 1979) Frank Kelly explained the deep relationship between flow balance and stochastic process reversibility.
- From physics we know that reversibility (time-reversal symmetry) gives rise to most conservation laws (energy, momentum, angular momentum, ...).
- Thus flow balance gives a way to engineer composite and aggregated systems.

‘Self clocking’ helps you engineer at this scale:

‘Self clocking’ helps you engineer at this scale:

but only ‘flow balance’ helps you at this scale.

Digression: why end-to-end rather than hop-by-hop?

- It's been tried (X.25) and hop-by-hop works really poorly.
- The reasons are fundamental. Analytically, HbH behaves like a multiply jointed pendulum.

Digression: why end-to-end rather than hop-by-hop?

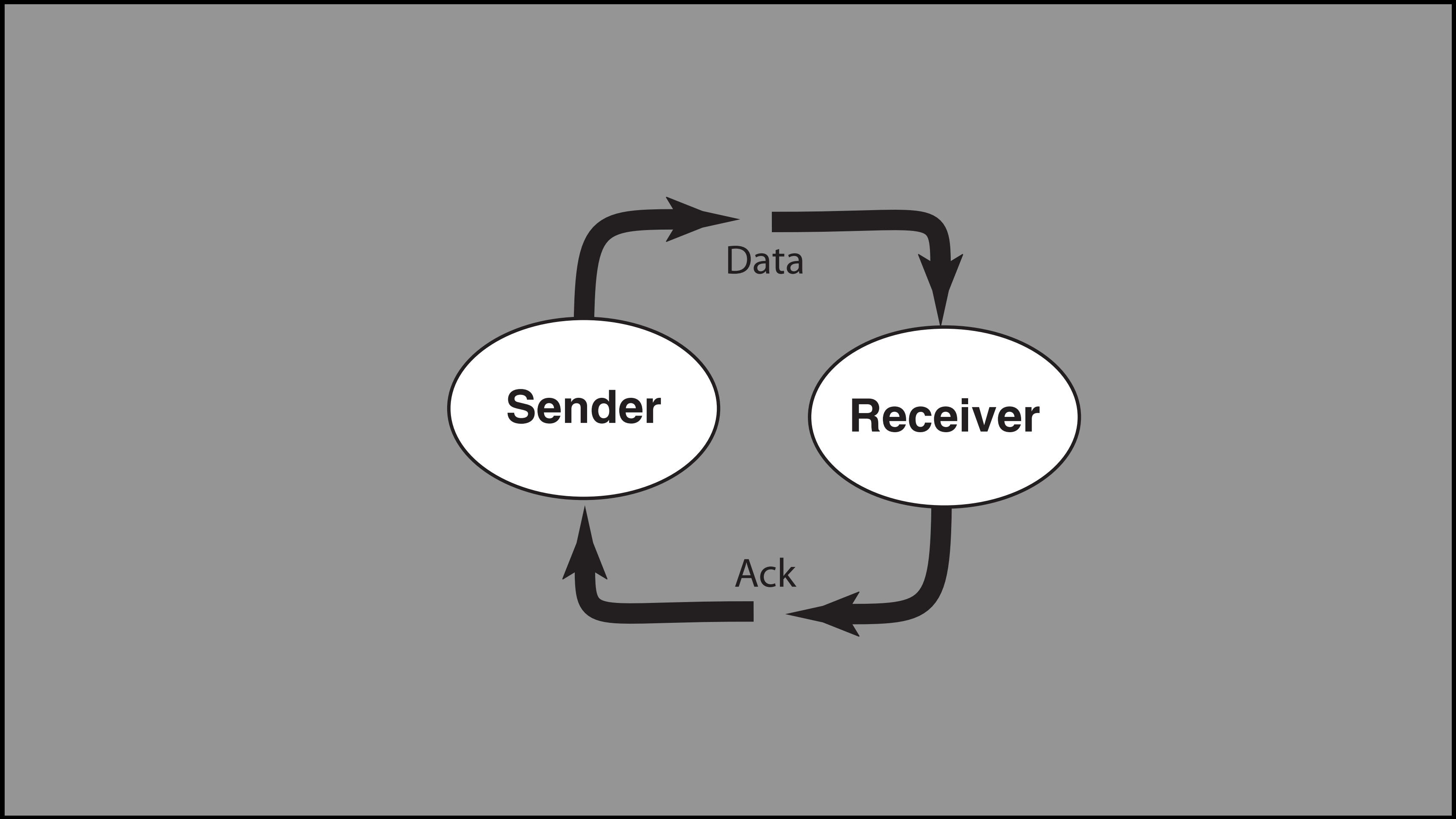
- It's been tried (X.25) and hop-by-hop works really poorly.
- The reasons are fundamental. Analytically, HbH behaves like a multiply jointed pendulum.

The Routh-Hurwitz stability criterion, one of the oldest results in control theory (E.J.Routh, 1877), says that one big servo loop will always be more stable than multiple small loops.

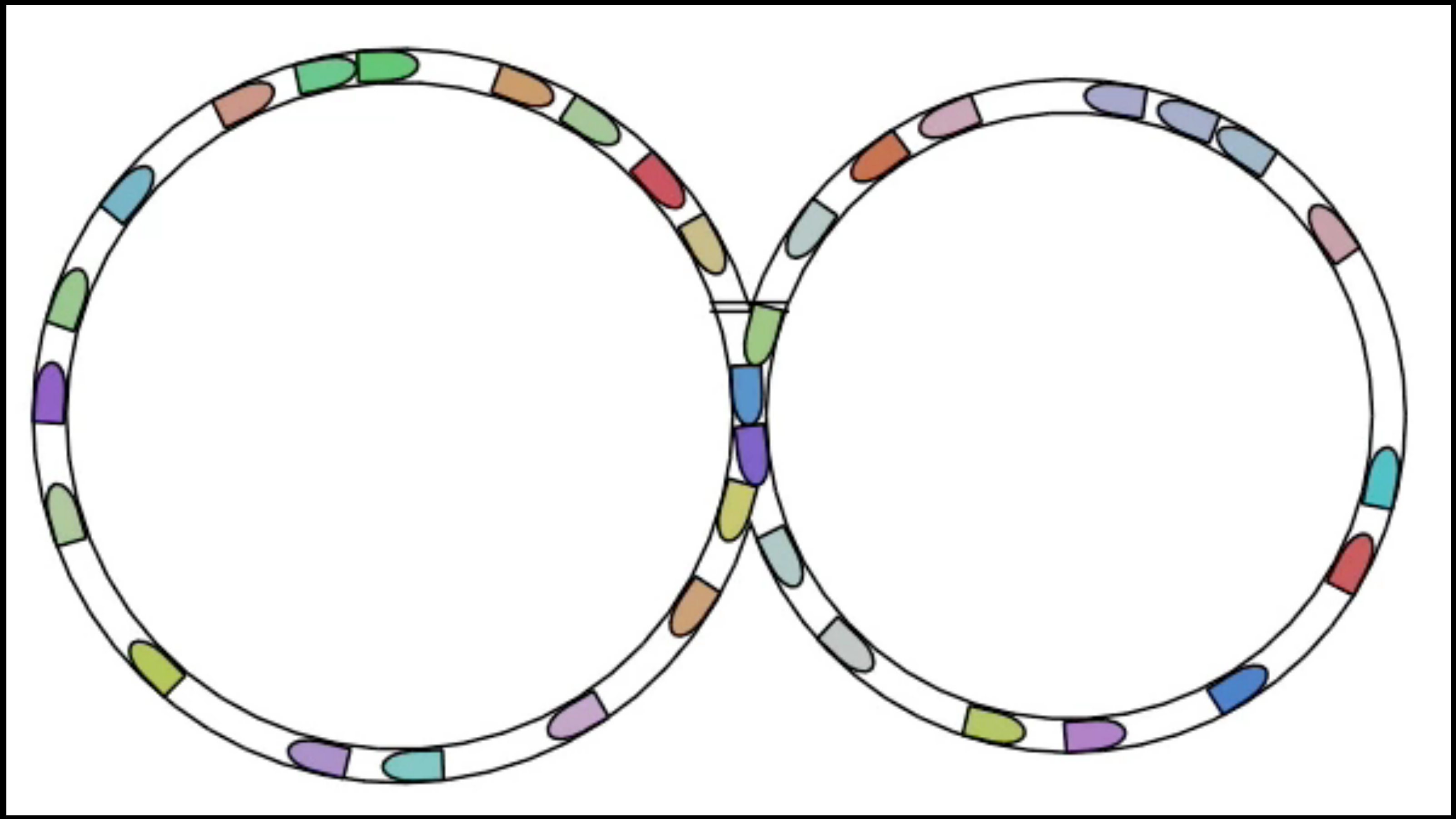
Traffic interactions

- Washboard roads or mogul fields are created by a co-evolution process between suspension and dirt or skis and snow.
- The nature of this process is to make things as bad as they can possibly be and to do so exponentially fast.

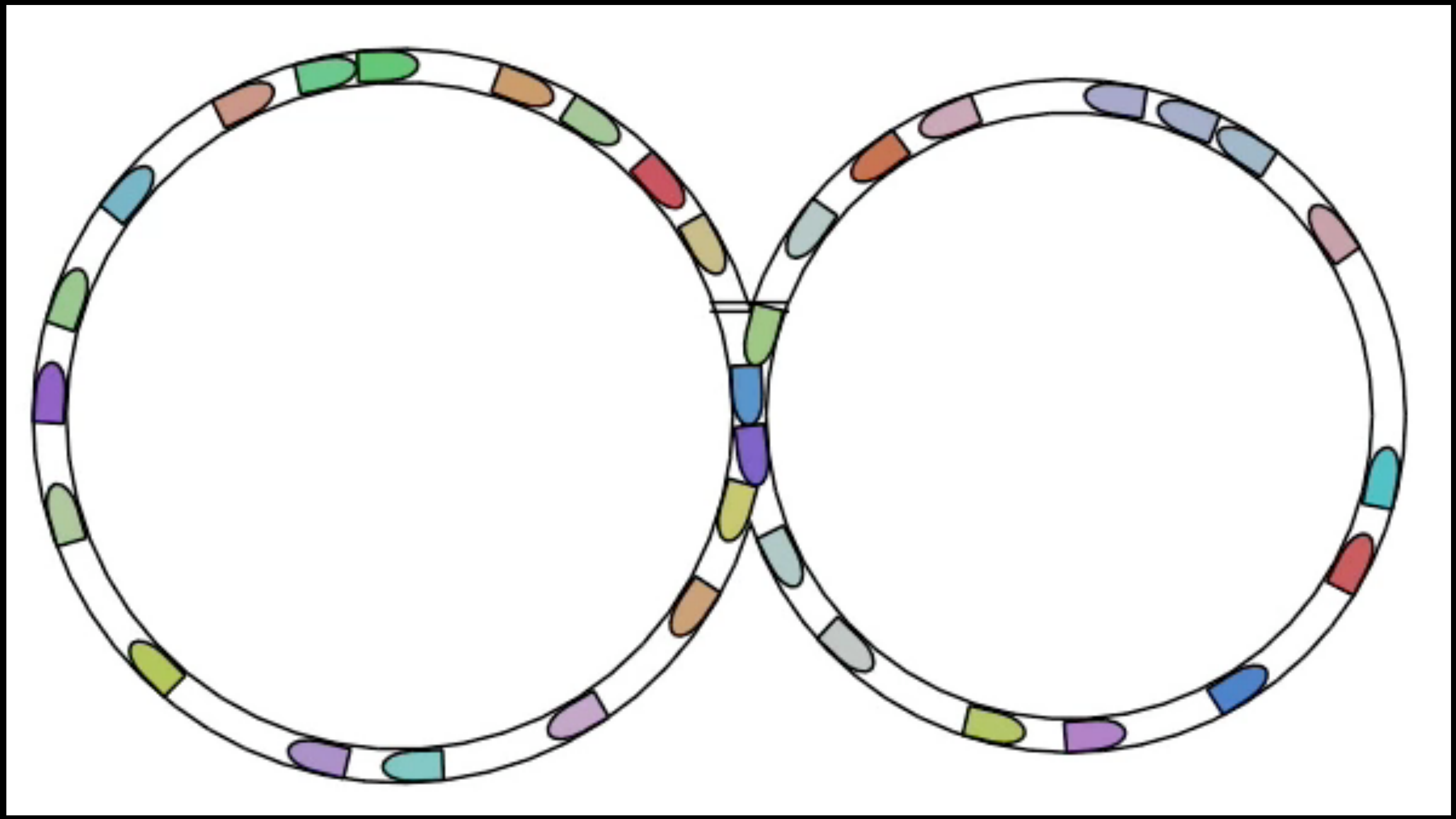
See *Washboard Road: The Dynamics of Granular Ripples Formed by Rolling Wheels*, Nicolas Taberlet, Stephen W. Morris and Jim N. McElwaine, Physical Review Letters, 99, 068003 (2007).
http://www.physics.utoronto.ca/nonlinear/papers_sand.html



- Flow balance and end-to-end flow control seem to be generically important to reliable transport.
- Both encourage us ignore end-point details and think in terms of aggregated traffic flowing in relatively big loops.
- Since economically viable networks are sparse, those loops interact. What is the effect of those interactions?

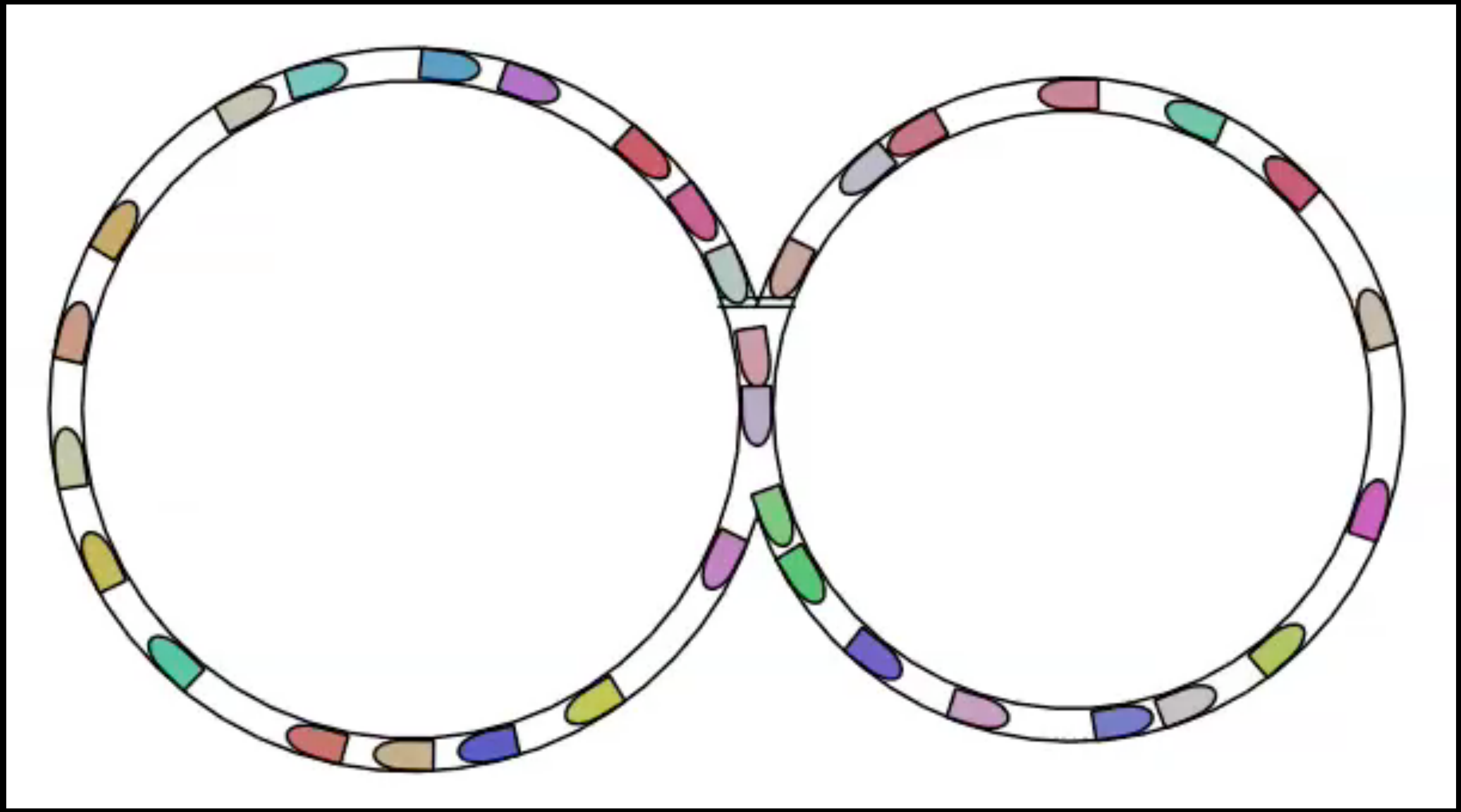


- Consider two disjoint loops with a single common segment.
- Each loop is half filled with randomly distributed packets. (Note that the only randomness in this system is the initial positions of the packets.)
- At the entry to the common segment is a merging rule which determines which side gets service in the case of conflict. The rule is ‘alternate’ for this example.



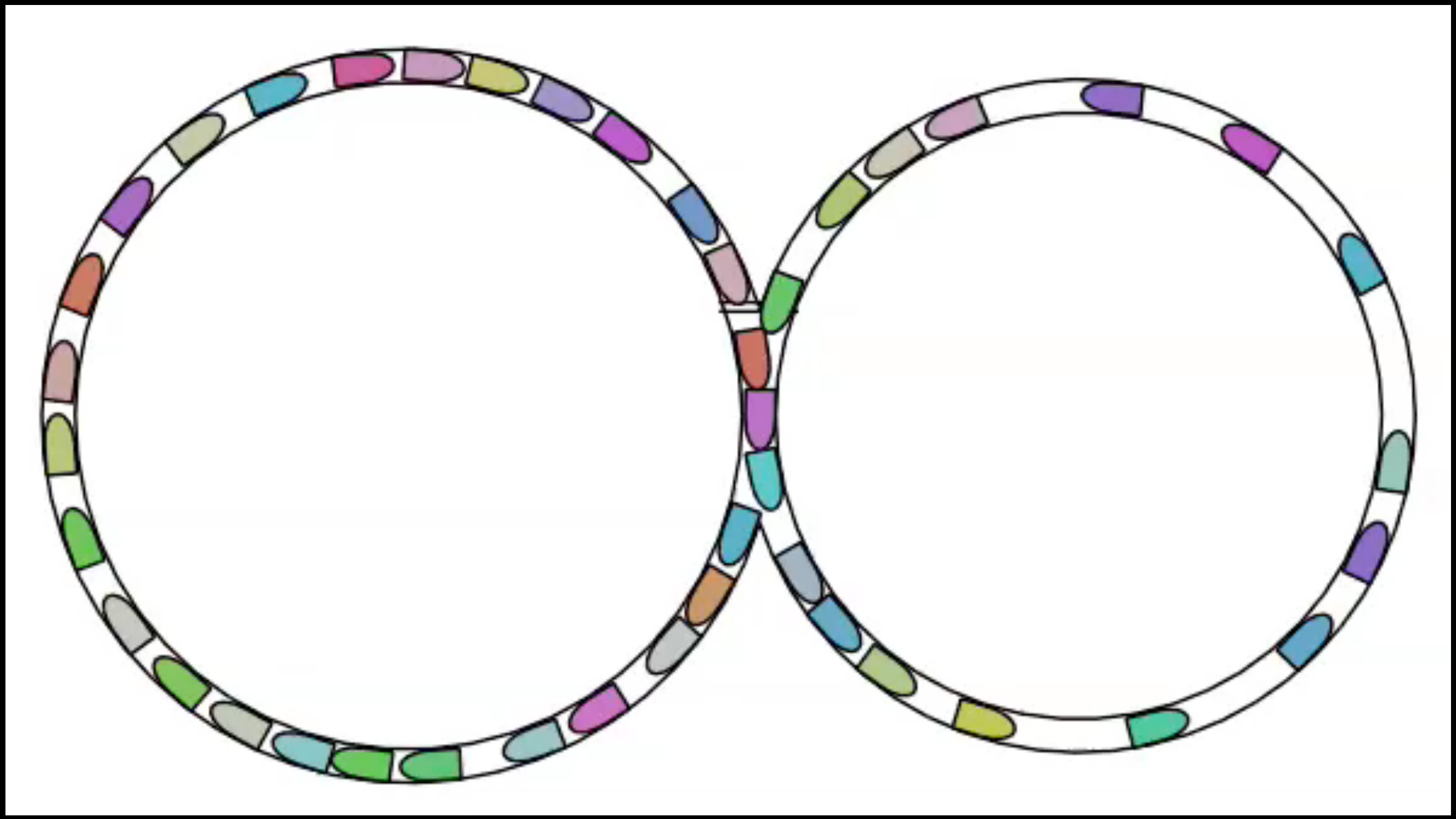
- We end up with perfectly uniformly distributed traffic (after quadratic time) with minimum delay that's robust against perturbations.
- Unfortunately, no commercial router behaves like this... they all have a bias in favor of serving the previously served input because that dramatically improves local performance.

- Same situation as before but now the merging rule is to service a queue until it’s empty.
- Since a clump can’t be broken up once created and a clump on one side causes a clump to form on the other, this scenario creates a few big clumps exponentially fast.
- And, unlike the previous case, queues and delays are close to the maximum possible for this traffic intensity.



- Same situation as before but now the merging rule is to service a queue until it's empty.
- Since a clump can't be broken up once created and a clump on one side causes a clump to form on the other, this scenario creates a few big clumps exponentially fast.
- And, unlike the previous case, queues and delays are close to the maximum possible for this traffic intensity.

- The fixed-point is a function of the maximum traffic density in either loop.
- If we increase the density in the left loop to 75%, things always congeal to one big lump and do so in at most two cycle times.
- Since a clump can't break up, a brief traffic pulse that clumps one side causes everything to stay clumped forever after.



- The fixed-point is a function of the maximum traffic density in either loop.
- If we increase the density in the left loop to 75%, things always congeal to one big lump and do so in at most two cycle times.
- Since a clump can't break up, a brief traffic pulse that clumps one side causes everything to stay clumped forever after.

- A kinematic view of traffic helps us visualize and understand both aggregation and interaction, things that just don’t exist in telco netwoks.

- It’s not enough just to remove complexity from the core of the network with things like end-to-end flow control.

- We also have to actively generate simplicity (good mixing) to combat the nasty structure that’s always created by interaction.