Determining Priority Flow Control Headroom

Norman Finn
Huawei Technologies Co. Ltd
nfinn@nfinnconsulting.com
cz-finn-pfc-headroom-0629-v01

Purpose of this presentation

- Lily Lv has made suggestions (e.g. new-lv-adaptive-pfc-headroom-0121-v03) proposing a new protocol for measuring the MAC control path timing in order to determine the “headroom” required for Priority Flow Control to ensure that no frames will be dropped due to input buffer overflow.
- This presentation suggests that one good way to deal with the problem is to:
  1. Use PTP to measure the link delay;
  2. Use non-standardized means for measuring the near-end PFC generation time and far-end PFC response time; and
  3. Use LLDP to convey one far-end PFC response parameter to the near end.

Explanation of a model for this problem

- The following model is presented in animation (.PPTX only) and in static steps (.PDF or .PPTX).

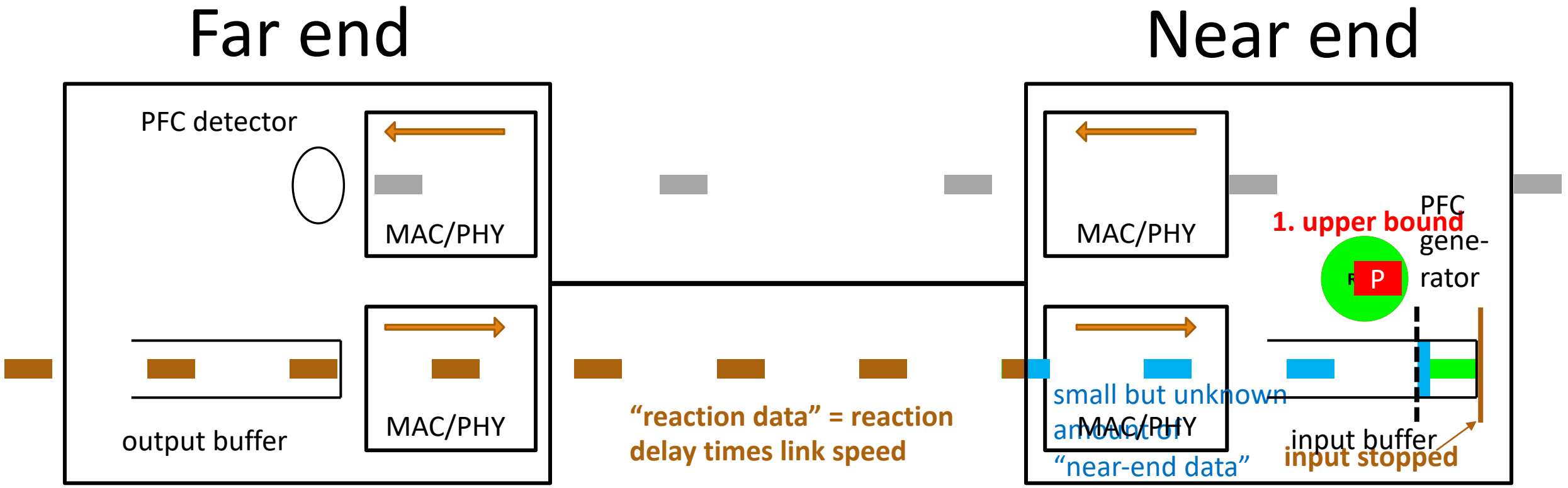
The problem

- Our model is that we have an input buffer in the near-end system into which received frames flow, that is drained sporadically.
- PFC allows the near-end system to transmit a pause message to the far-end system to tell it to stop transmitting frames.
- When the far-end system receives the PFC, it ceases transmitting further frames.
- But, it takes time to generate the PFC, to transmit it to the far end, and to receive it and shut off transmissions.
- In the meantime, data is still being transmitted, and after being shut off, there is data in the output port and on the medium that must be received.

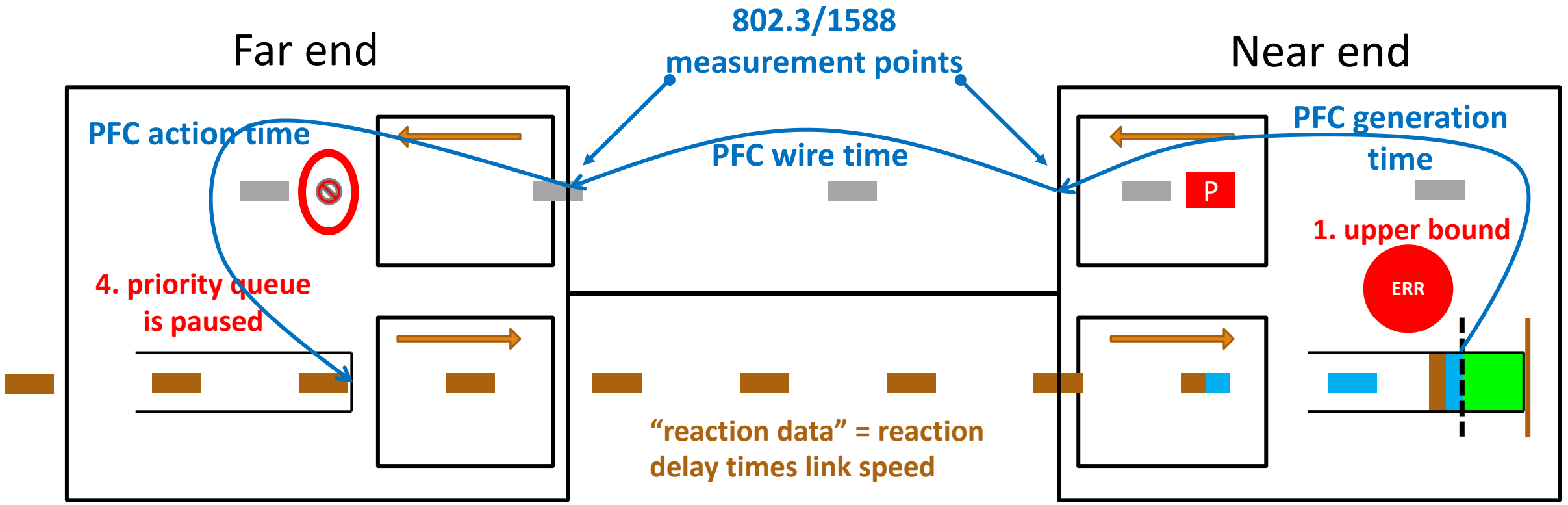
The other problem

- The same considerations apply to restarting the far-end system, again using PFC, when the local input buffer drops below a certain threshold, in order to ensure that a priority queue is not idled any more than necessary. The calculations are very similar to the ones presented here, and are left as an exercise for the reader.
- Furthermore, PFC is not a hard XON/XOFF protocol; it transmits a limit to the amount of data that the far end can send before it has to pause. Making use of this feature is not addressed, here.

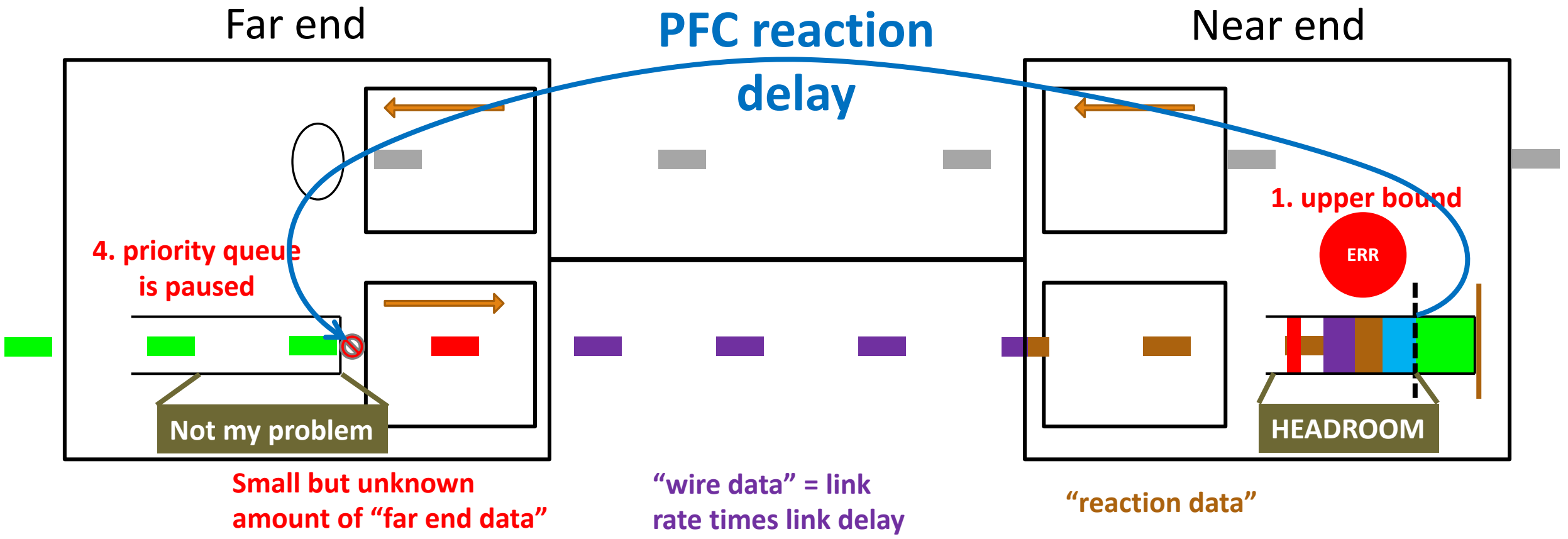
Priority Flow Control model animation 1 *(skip over in pdf)*



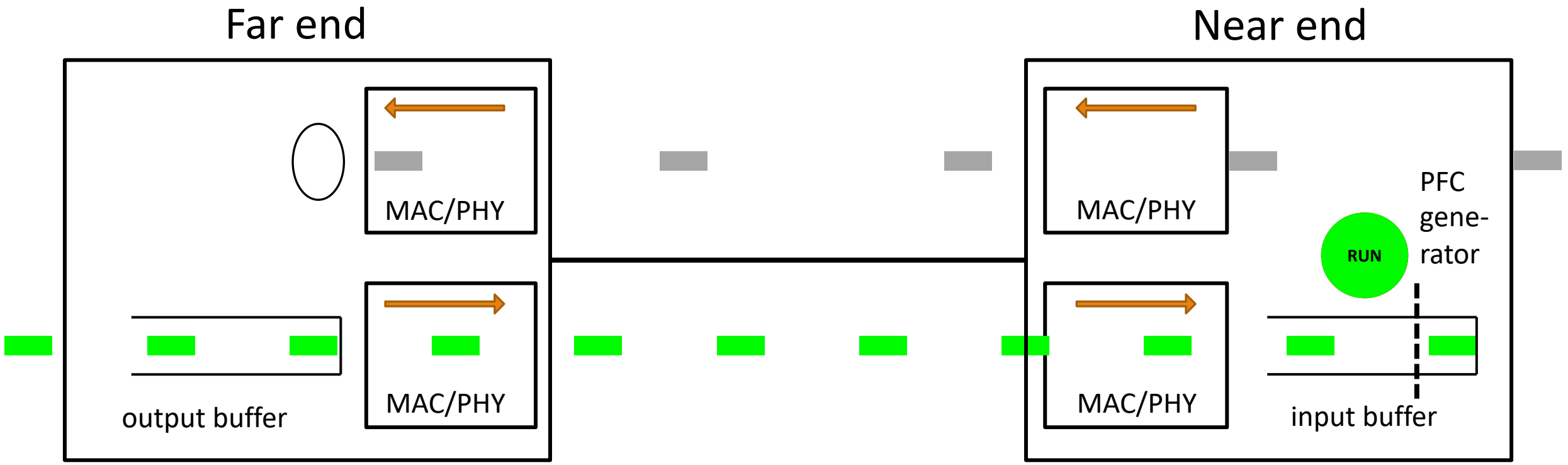
Priority Flow Control model animation 2 *(skip over in pdf)*



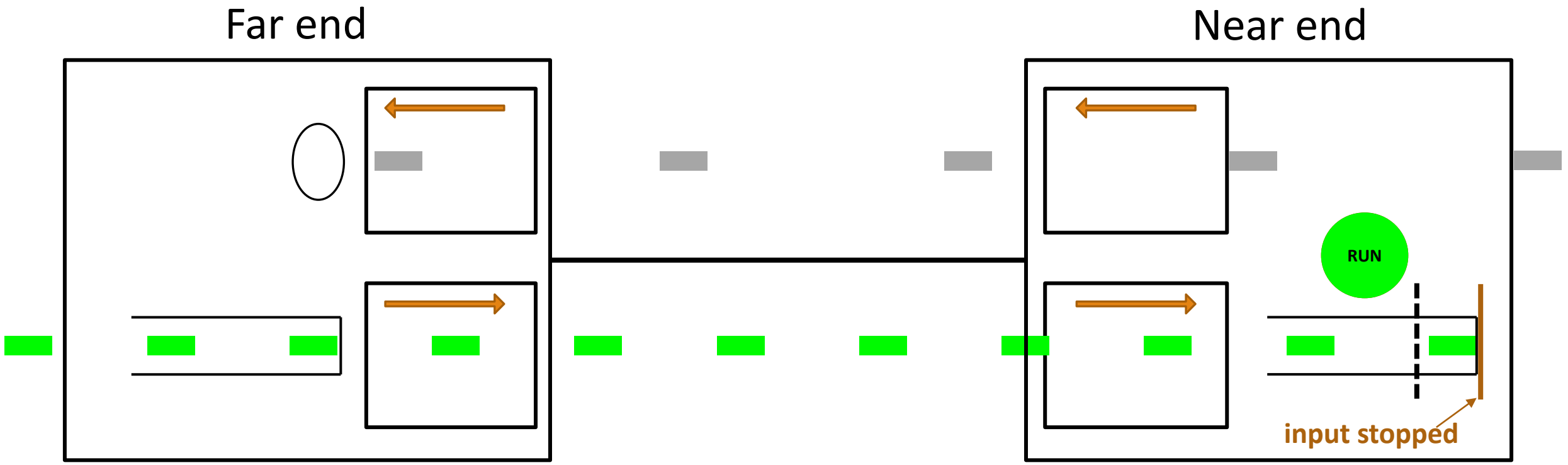
Priority Flow Control model animation 3 *(skip over in pdf)*



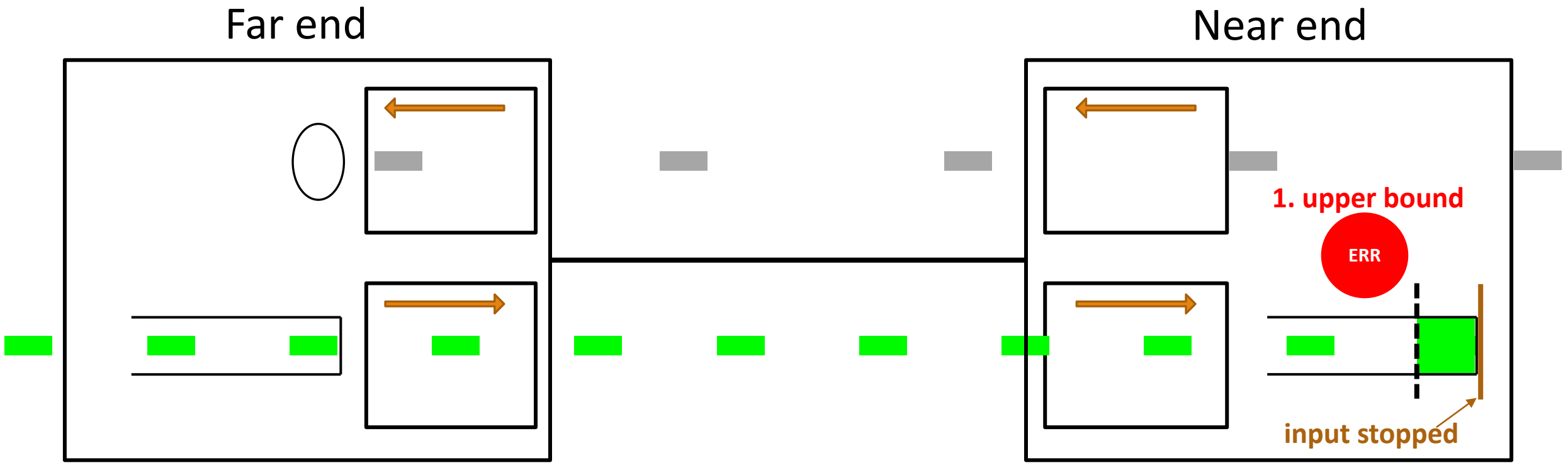
Priority Flow Control model step 1



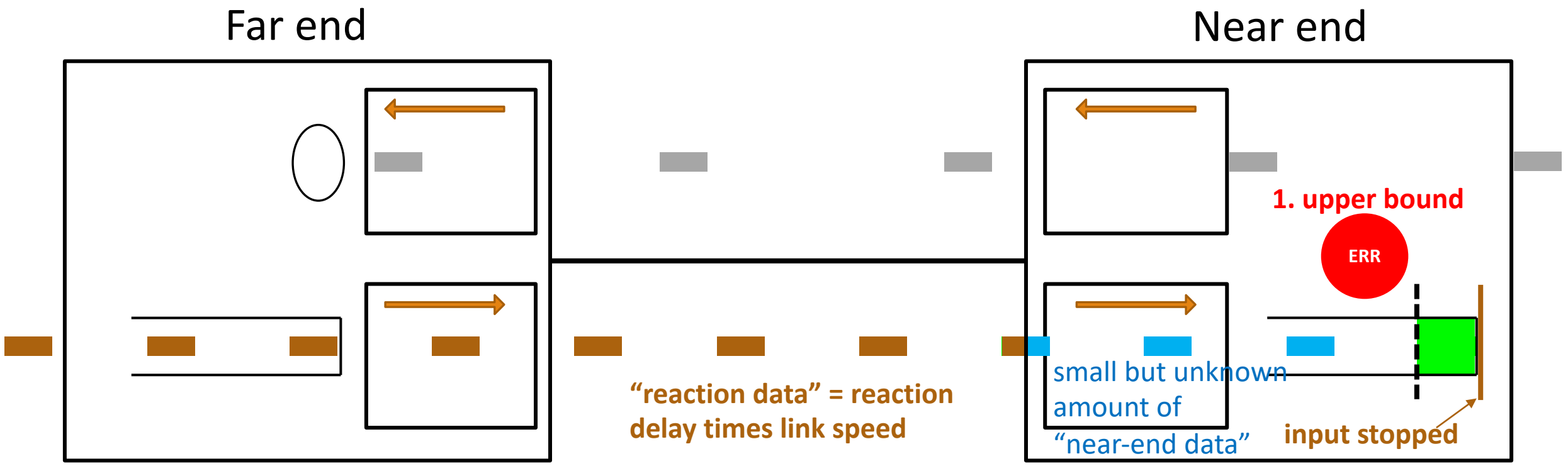
Priority Flow Control model step 2



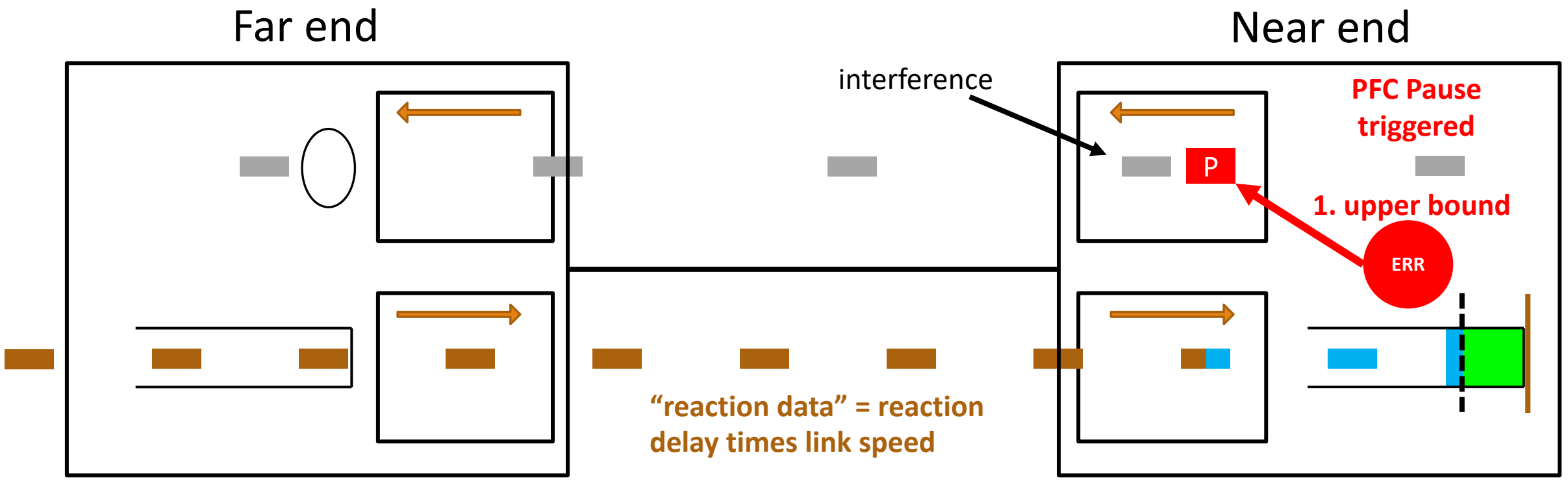
Priority Flow Control model step 3



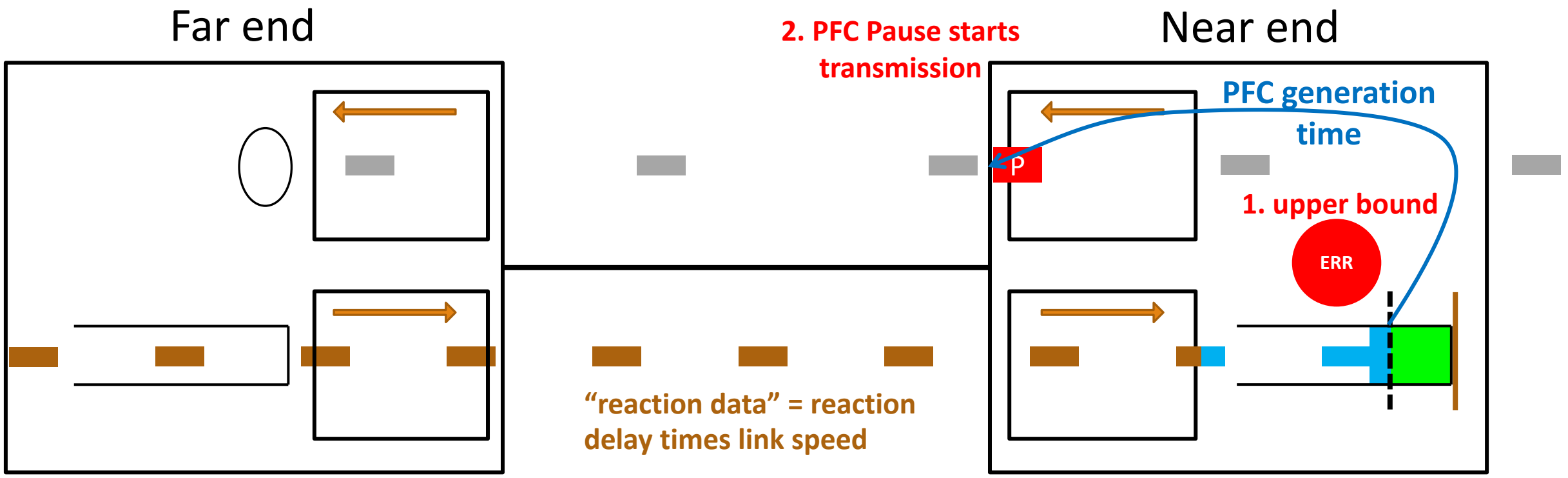
Priority Flow Control model step 3 relabeled



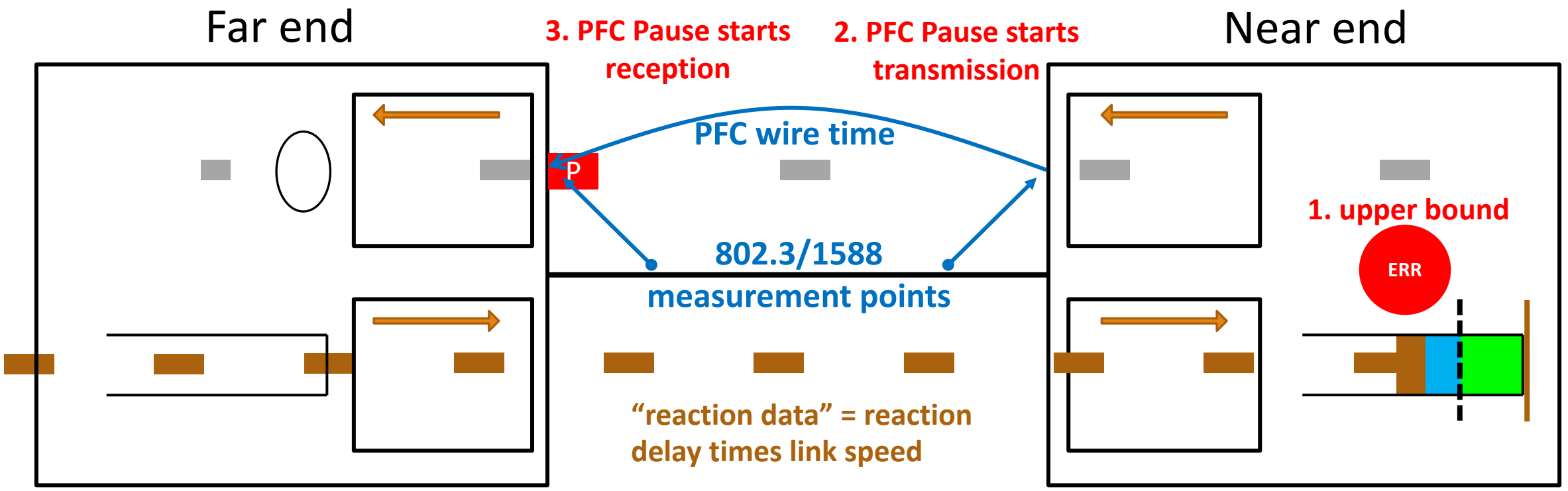
Priority Flow Control model step 4



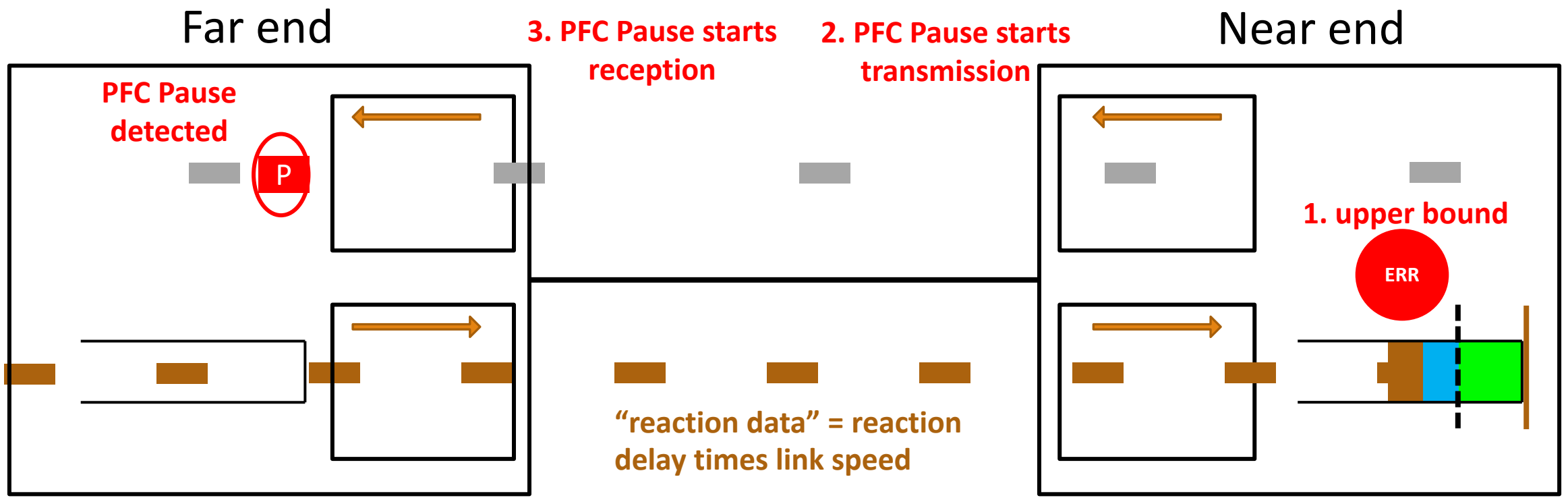
Priority Flow Control model step 5



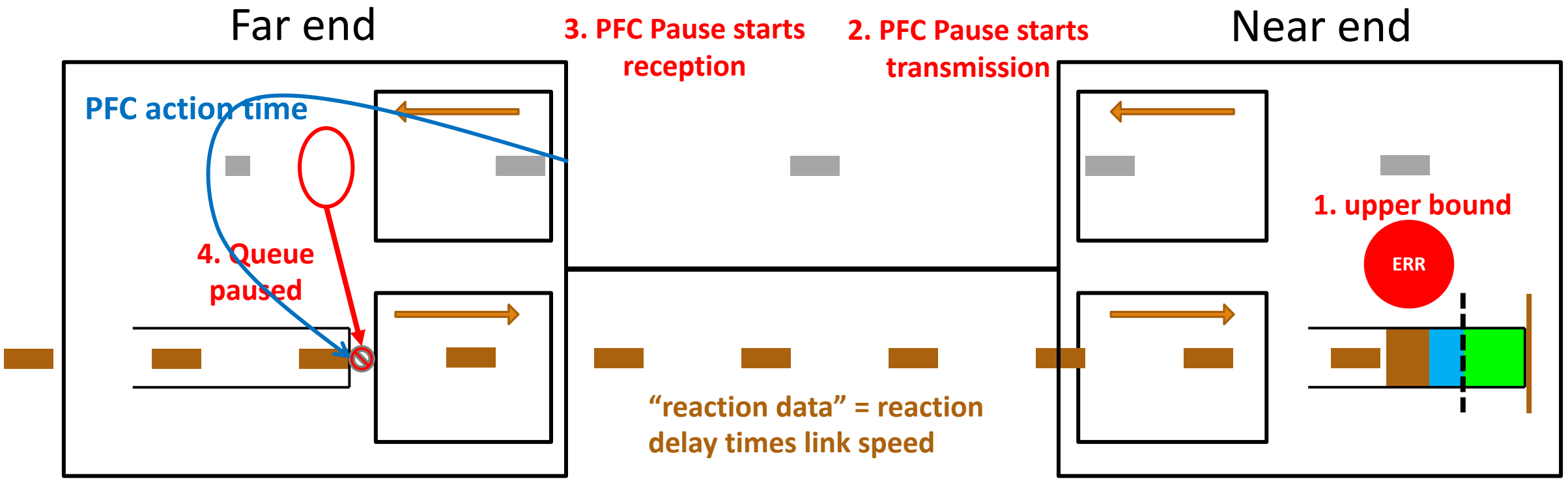
Priority Flow Control model step 6



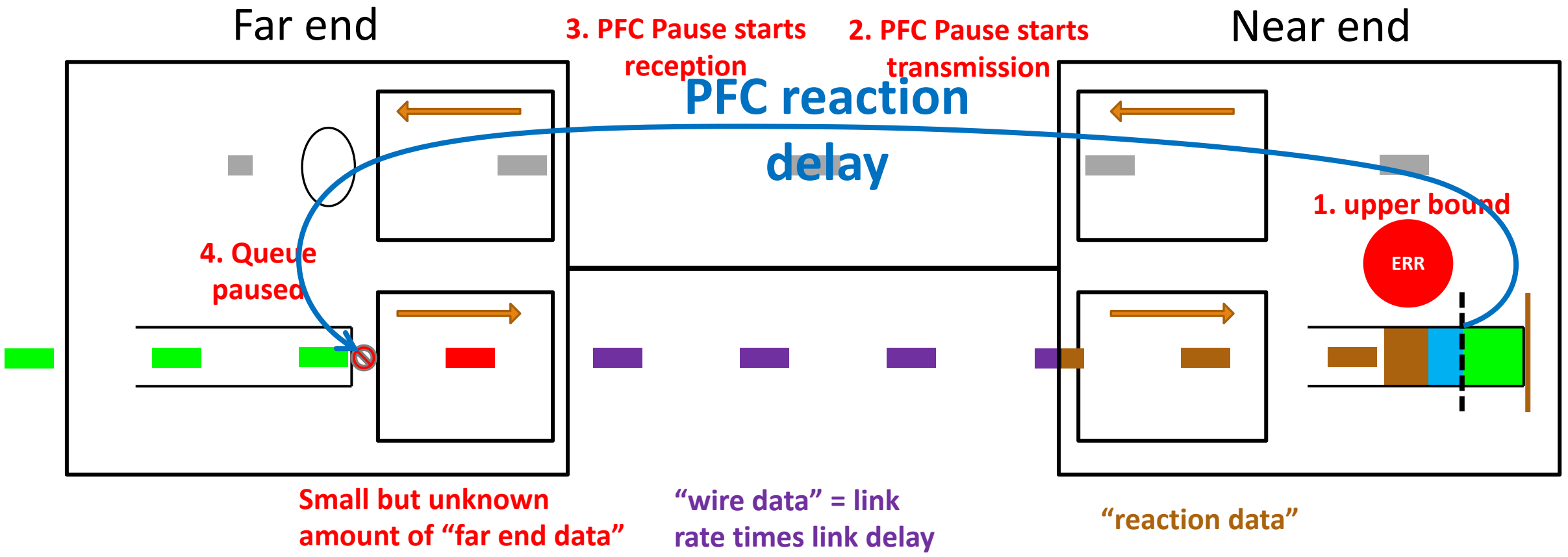
Priority Flow Control model step 7



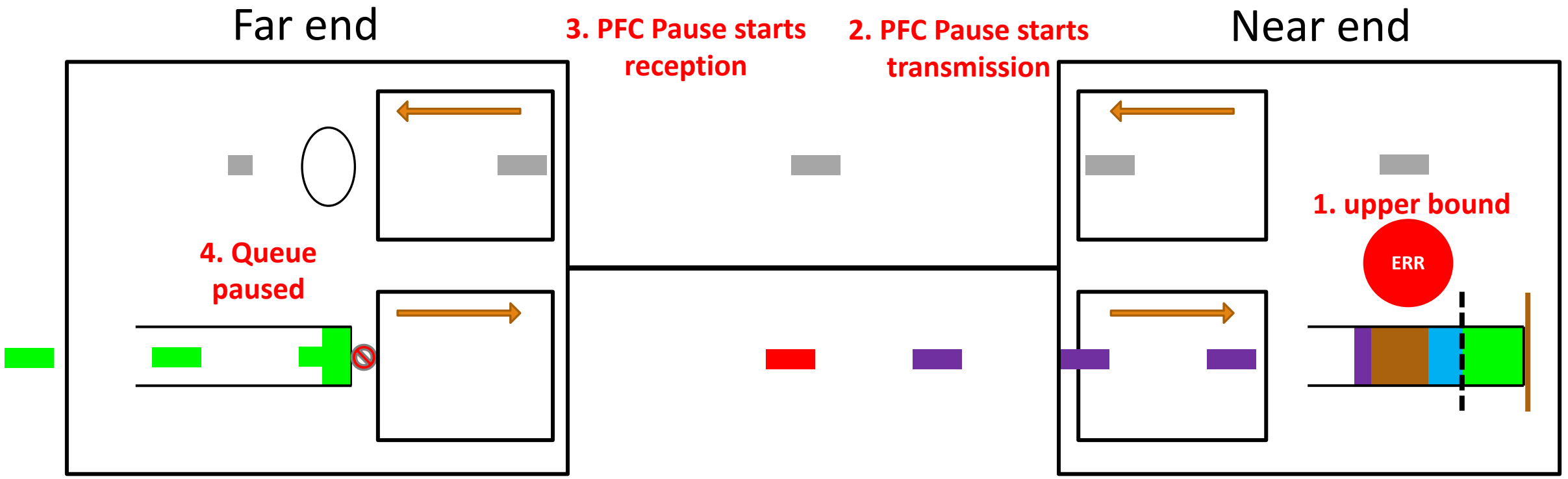
Priority Flow Control model step 8



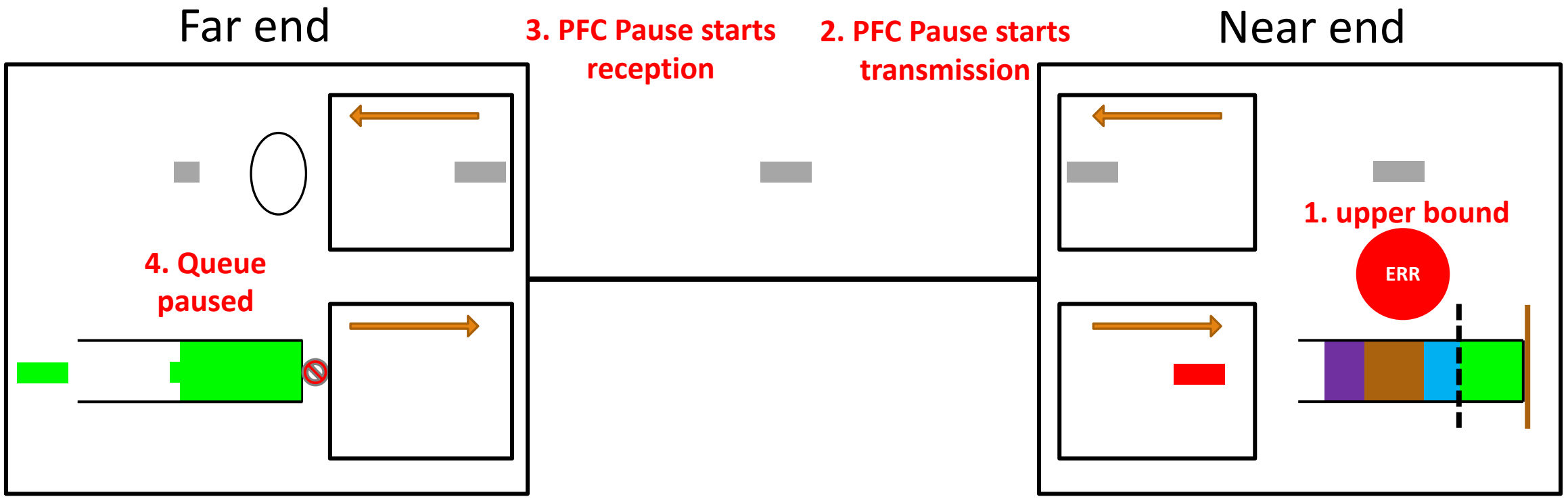
Priority Flow Control model step 8 relabeled



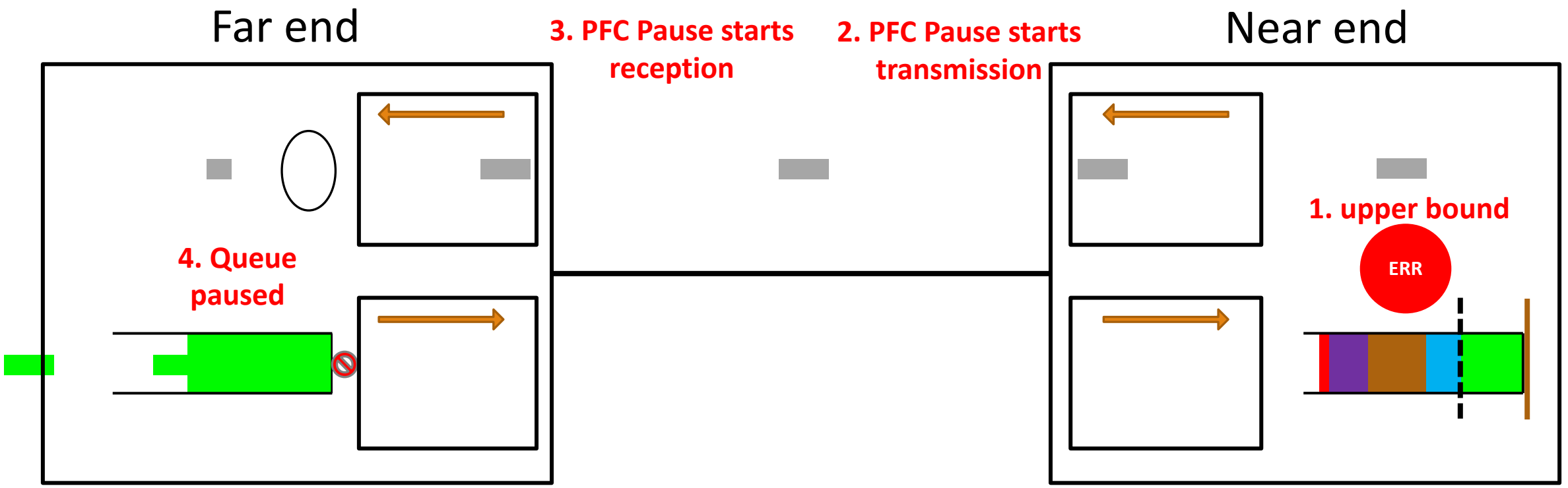
Priority Flow Control model step 9



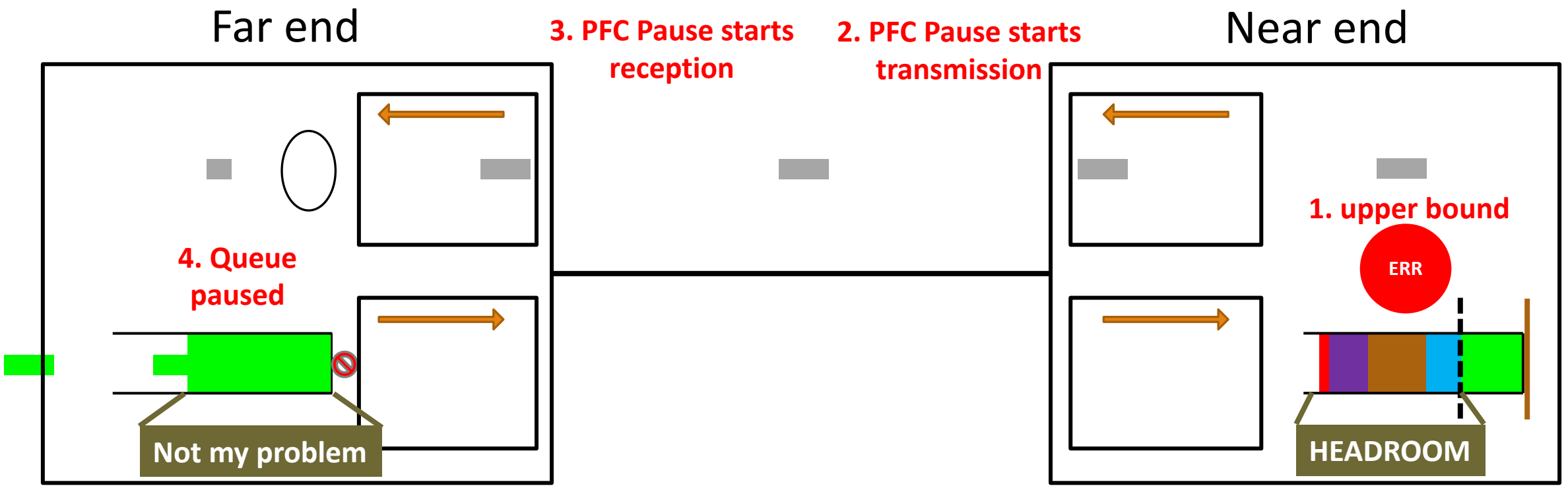
Priority Flow Control model step 10



Priority Flow Control model step 11



Priority Flow Control model step 11 relabeled



Reaction time vs. data storage

- The reaction time measures the time from near-end buffer set-point detection to far-end queue shut-off. This total time, multiplied by the data rate, gives a number of bits that will have to be accommodated in the near-end input buffer, assuming that the transmitter is running continuously. This is called the reaction data.
- Typically, there is a per-frame overhead in the buffer, so we must assume minimum-length frames, to get the worst case.
- There is additional frame data that must be included in the headroom calculation, stored (potentially) in the far end, the medium, and the near end, that must be also included in the headroom calculation.

PFC reaction delay

1. Near end sees that the buffer upper bound has been reached.
   - PFC Generation time
2. The PFC starts transmission (a PTP point).
   - PFC wire time
3. The first bit of the PFC is received at the far end (a PTP point).
   - PFC action time
4. The priority queue specified in the PFC is stopped.

PFC generation time: Buffer full to PFC transmission starts

A. It takes some time to generate the PFC.
B. It takes some time for the PFC to progress down the MAC stack to the transmission point defined by 802.3.
C. There may be whole or partial frames that have been committed to the MAC and must be output before the PFC can be output.
D. There may be delays caused by MACsec.

- A and B are likely constant, though their values may be vary with link speed.
- The worst-case for C depends on link speed, but also on configuration variables such as max PDU size and preemption. The times will vary widely from one implementation to another.
- D affects both the PFC itself, and the committed frames in C.

PFC wire time:

- This is what PTP does for a living.

PFC action time:
PFC receipt to transmission stopped

A. It takes some time for the PFC to progress up the MAC stack to the PFC handler.
B. It takes some time to pause the transmission queue.
C. There may be delays to the received PFC caused by MACsec.

- A and B are likely constant, though their values may be vary with link speed.
- C is likely constant for a given size of PFC, but pipelining of the decryption process can have odd effects.
- Given the port configuration (e.g. link speed), and information from the manufacturer, the worst-case for all of these values can be determined sufficiently accurately for the headroom calculation.

Measuring and/or calculating the reaction time

- **PFC generation time:** The worst-case is required; the typical case is not interesting. The worst-case is going to be essentially constant, or at worst, somewhat dependent on link speed and port configuration. It can be calculated by the implementor, or at worst, obtained through some combination of one-time measurement and calculation. No standard is required. The far end does not care about this number.
- **PFC wire time:** PTP measures the link delay. Both ends know the answer.
- **PFC action time:** Again, the worst-case is required, and it depends on link speed and port configuration. It can be calculated by the implementor, or at worst, obtained through some combination of one-time measurement and calculation. No standard is required. **This value must be somehow conveyed to the near-end.**

Total headroom calculation: Four pieces

- **Near-end data:** Depending on the implementation, there may be a certain amount of data in the pipeline between the PTP/802.3 reception measurement point and the point at which the data is added to the input buffer to cause a PFC pause situation. This must be added to the headroom requirements.
- **Reaction data:** This is the data transmittable during the PFC reaction delay (reaction delay multiplied by link speed).
- **Wire data:** After the transmission stops, a certain amount of data is stored in the medium (link delay multiplied by link speed).
- **Far-end data:** There can be a some amount of data from the stopped queue already committed by the far end to be transmitted at the moment the queue is stopped.

Obtaining the headroom data

- **Near-end data:** The worst case is a constant, perhaps a constant that varies with link speed. It can be calculated by non-standard means without regard to the far-end system. It **must be calculated**, but does not have to be shared with the far-end system.
- **Reaction data:** Trivial calculation (PFC reaction delay multiplied by link speed). But, one component of the reaction delay (the PFC action time) **must be conveyed from the far end to the near end**.
- **Wire data:** PTP link delay multiplied by link speed.
- **Far-end data:** The far end can calculate the worst case by non-standard means without regard to the neat-end system. **This quantity must be conveyed to the near end**.

The complete headroom solution

To obtain the headroom, we (the near end) must add:

1. The near end data, calculated but not shared.
2. The reaction data, link speed times reaction delay, which delay consists of:
   a) The PFC generation delay, which we can calculate.
   b) The link delay, which we measure from PTP.
   c) The PFC reception delay, which we calculate from link speed and PFC size.
   d) The PFC action delay, which must be conveyed to us by a protocol.
3. The wire data, link delay times link speed.
4. The far-end data, which must be conveyed to us by a protocol.

Actually, only one far-end parameter need be conveyed (perhaps per-priority)

We can note that the far-end PFC action time and the far-end data (red in the illustrations) can be combined into a single parameter passed to the near-end. One could pass either:

P = (PFC action time) + ((far-end data) / (link speed))

or:

P’ = ((PFC action time)*(link speed)) + (far-end data)

Conveying the far-end delay/data parameter to the near end.

- **IEEE Std 802.1AB LLDP.**
- The value conveyed varies with link speed, and perhaps port configuration, and these change slowly enough that LLDP is well-suited to carry them.
- With the new multi-frame LLDP, there is no problem even if this parameter is different for each of the 8 priority levels.

This method vs. a timestamp protocol

- We already have widely-implemented hardware and software for the PTP link delay measurement.
- Experience with ITU Y.1731 has shown that achieving widespread availability of a timestamp protocol is slow and expensive.
- The method presented, here, can be implemented by any vendor. The values measured/calculated are independent of any other vendor.
- This method probably can be included in IEEE P802.1Qcz.
- If the calculation is wrong, the headroom is wrong.

A timestamp protocol vs. this method

- Running a timestamp protocol at random (not regular!) intervals, and taking the maximum as the worst case, will give a headroom value that is probably sufficient.
- Actual measurement protects against erroneous calculations.
- A timestamp protocol probably needs a new PAR.
- The near-end PFC generation time and far-end PFC action time can vary significantly between the best case and the worst case. The worst case may happen only rarely.
  - Of course, *it may be precisely that worst that triggers the need for PFC*. In that situation, the need for PFC would occur just *before* the buffer headroom has been correctly measured.

What to do?

- Ultimately, this is an engineering judgement decision.
- Which is easier to do?
  1. At product design time, use some combination of analysis and measurement to determine the parameters that enable a system to calculate its near-end data, and far-end delay plus far-end data values. Operationally, run the PTP common mean link delay service, and pass the far-end calculated value in LLDP.
  2. Standardize and implement a timestamp protocol that gives a running measure of the headroom delay, and use it.

DISCUSSION

Thank you