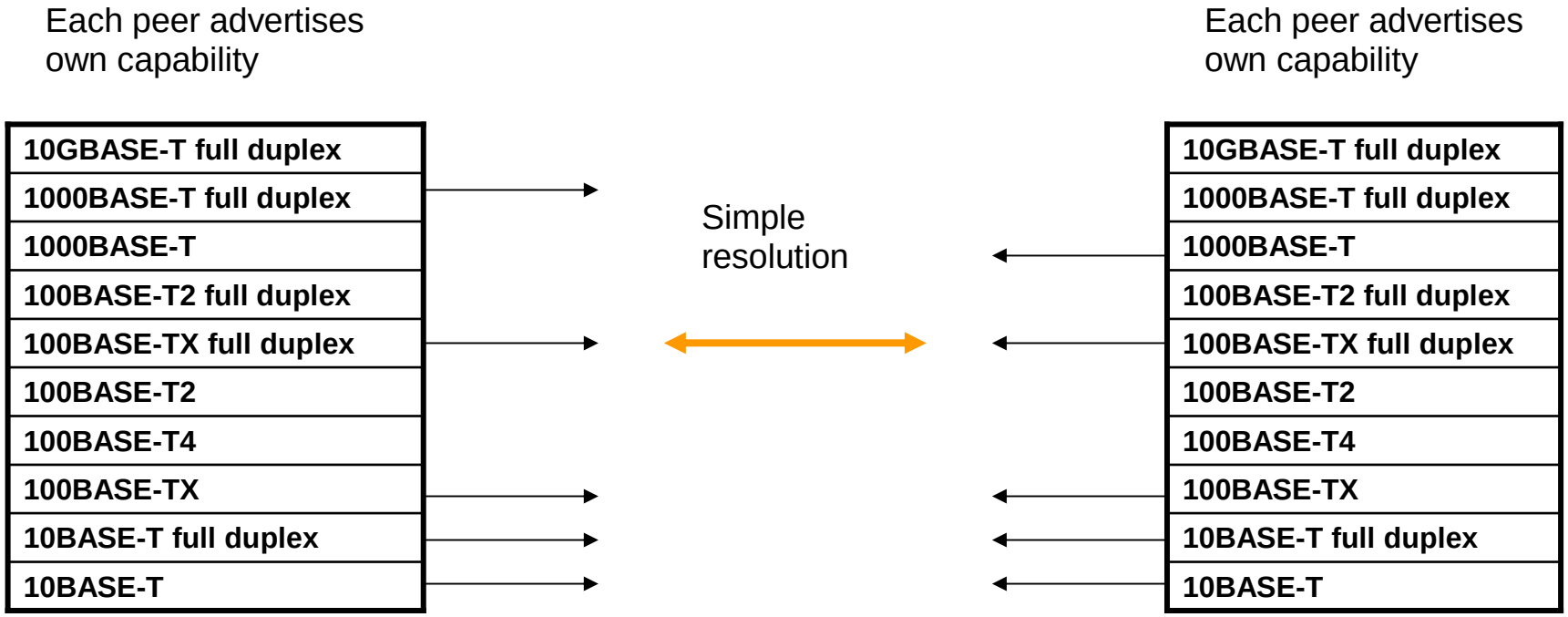
EEE Exchange of Management Information

Hugh Barrass (Cisco)

Background – capability exchange

"autonegotiation" is misnamed – it is really a peer-to-peer exchange of capabilities with a highest common mode resolution.



No state is exchanged, no "negotiation" takes place

System wake time exchange

Each peer advertises two times:

Transmit Tw

Maximum time that transmit path can holdoff sending data after deassertion of LPI

Receive Tw

Time that receiver would like the transmitter to holdoff to allow time for the receiver to wake from sleep

Simple resolution:

Each end holds off transmit data for (resolved system Tw)

= min (local Tx Tw, remote Rx Tw)

Problem exists if either end wishes to change

Changing parameters

To allow for changes, each peer sends an echo to reflect what it has received from LP

Therefore, 4 numbers are sent:

- Local transmit Tw (defined as before)
- Local receive Tw (defined as before)
- Echo transmit Tw (send back what received from LP)
- Echo receive Tw (send back what received from LP)

Consider situation where received echo parameter does not match local parameter – i.e. a change is in progress…

During a change…

When received echo does not match local parameter, you cannot know exactly when the LP will update with the new value

The simplest and safest approach is to use the most pessimistic assumption until you know that the LP is updated

Therefore consider both the local and the received echo

- For transmitter, use longer value – i.e. longer holdoff
- For receiver, use shorter value – i.e. shallower sleep

Pessimistic approach only applies for short time during change

- Also, require one change at a time – don't change a parameter unless echo matches current local

Required Function

Data holdoff time:

Resolved Tx Tw = max(local Tx Tw, received echo Tx Tw)

Data holdoff time = resolved system Tw

= min (resolved Tx Tw, remote Rx Tw)

Note that these functions are always evaluated in the same manner – no state or timing dependence

The receiver function does not necessarily need to be specified

Deepest sleep (receiver):

Resolved Rx Tw = min(local Rx Tw, received echo Rx Tw)

Deepest sleep = pessimistic LP resolved system Tw

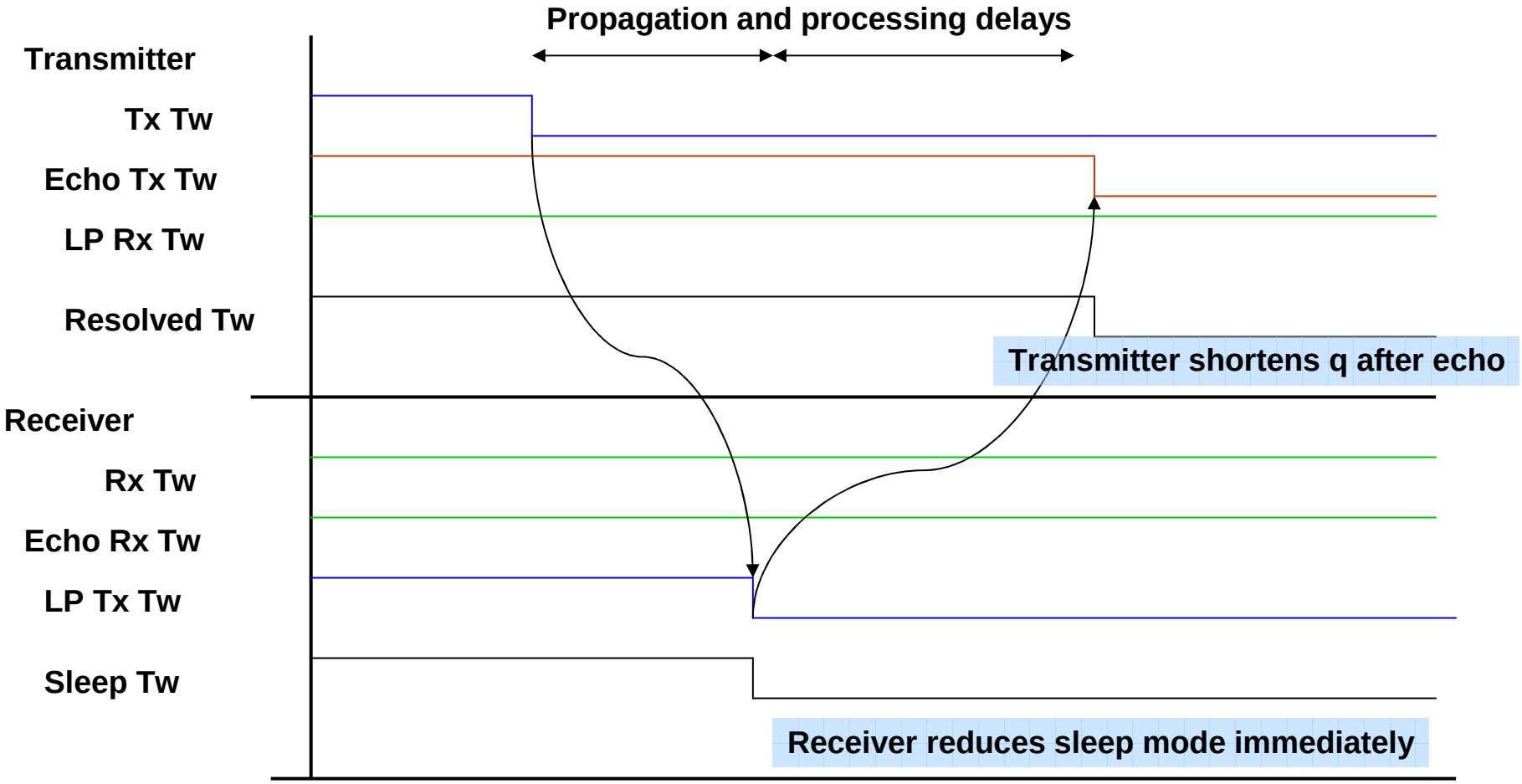
= min (resolved Rx Tw, remote Tx Tw)

Background

Some examples…

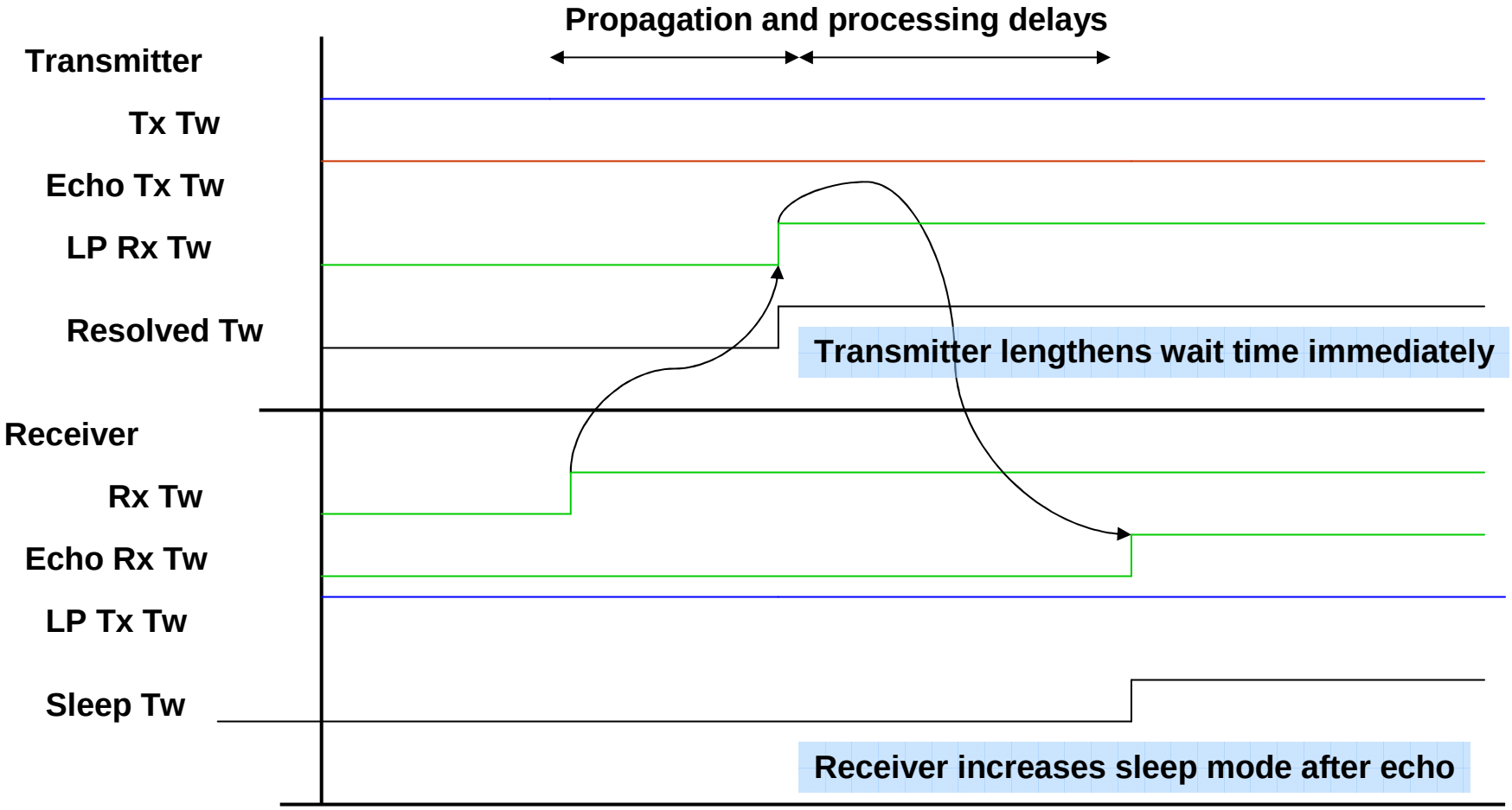
How it works

Transmitter reduces Tx Tw (new value is < LP Rx Tw)



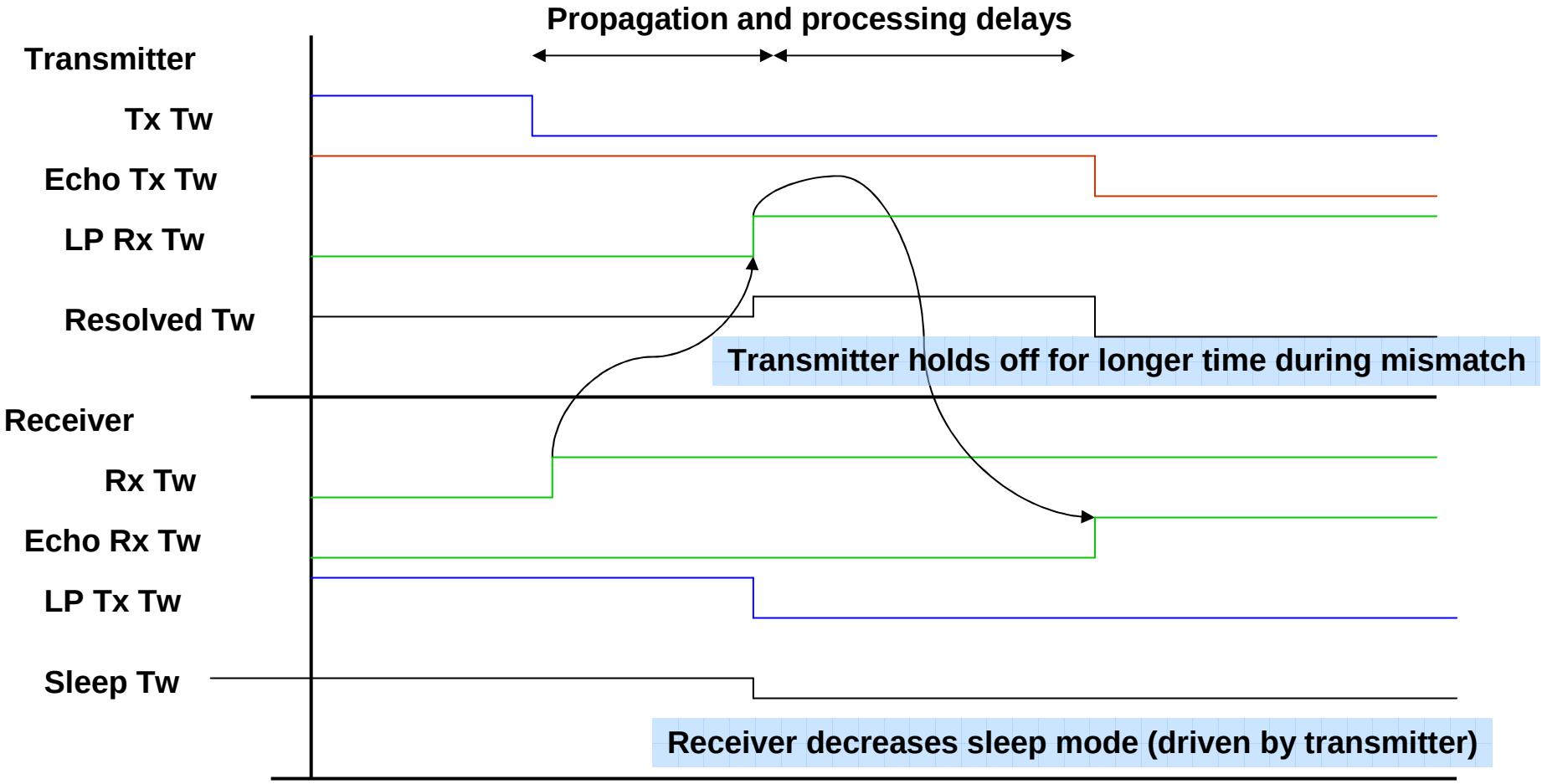
How it works

Receiver increases Rx Tw (new value is < LP Tx Tw)



How it works

Both change Tx decreases & Rx increases simultaneously



Some notes

At all times the Tx holdoff is >= the Rx sleep mode

No corner cases exist

Transmitter never reduces until after receiver has already reduced

Receiver never increases until after transmitter has already increased

When changes have settled times become equal

Mechanism uses the absolute minimum number of frames

Only ever two frames to make a change

- Allows infinite granularity or number of modes

For both transmit queue depth & receive sleep depth