# Increasing Maximum Window Size of TCP

Yoshifumi Nishida

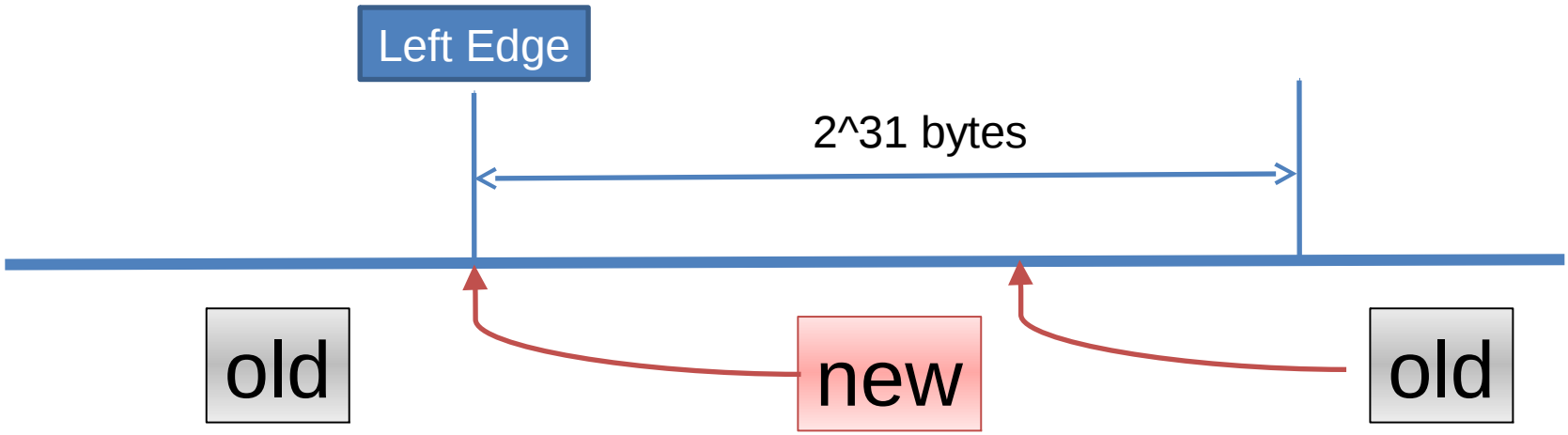
Hirochika Asai

# Background

- Current maximum window size
  - 2^30 - 2^14  (1,073,725,440) Bytes
    - Defined by RFC7323
- Can we increase this?
  - TCP’s seqnum space is 2^32
    - Nearly 4 times bigger than the maximum window size
  - Why we cannot use 2^31 – 2^15 bytes?
    - Use shiftcount =15 for window scale option

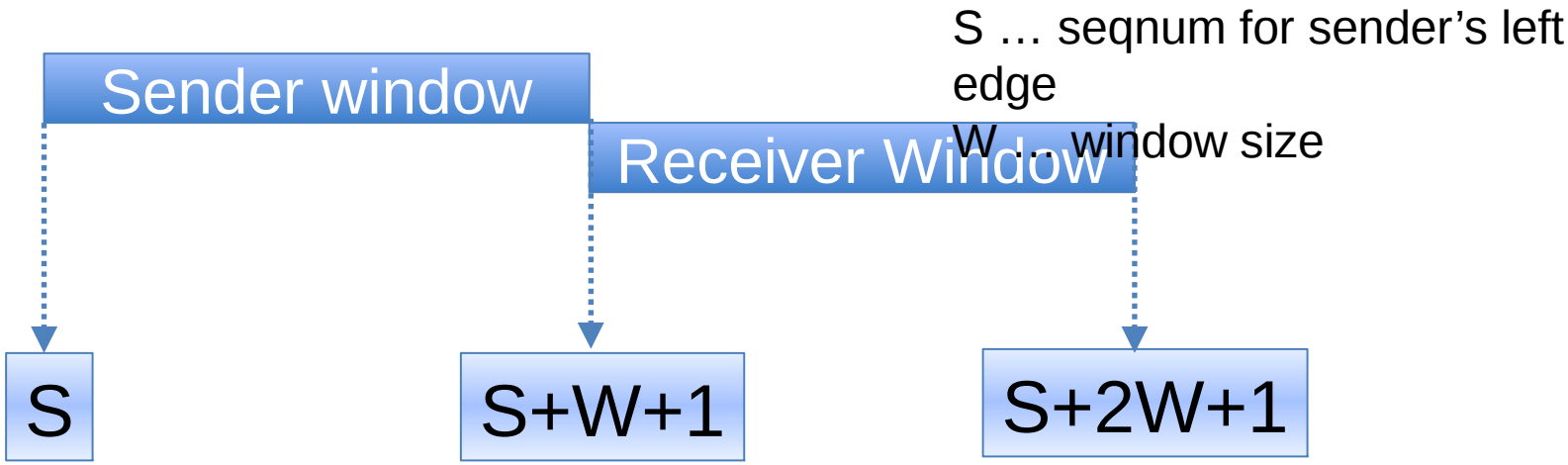
# Descriptions in RFC7323

TCP determines if a data segment is "old" or "new" by testing whether its sequence number is within 2^31 bytes of the left edge of the window, and if it is not, discarding the data as "old".



# Why max window must be < 2^30 ?

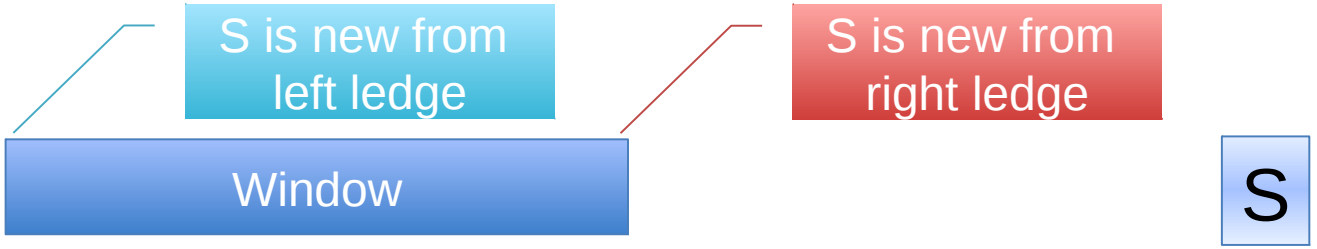
- In worst case, receiver may receive left edge + 2 * window size seqnum from sender



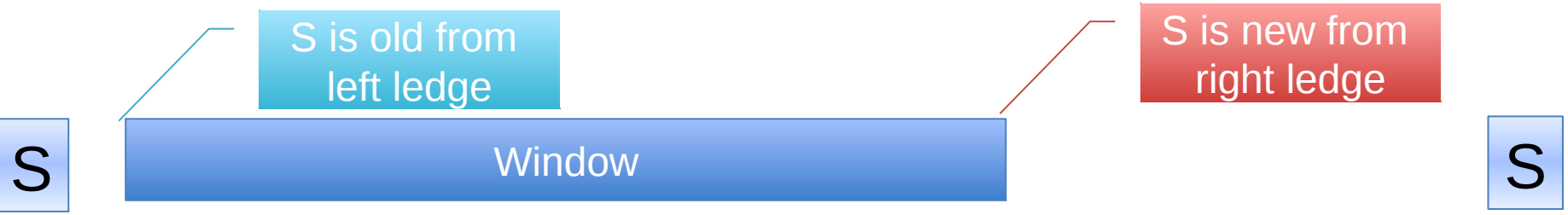

- S should be considered as "old" from S+2W+1
  - 2W +1 should be less than 2^31
  - So, W must be < 2^30
- Looks reasonable! But, is this really true?

# Do we need to know old or new?

- Our answer: Probably not. Because important point is whether seqnum S is inside of window or not!
  - If maxwin is 2^30 – 2^14




  - If maxwin is 2^31 – 2^15




    - We cannot tell S is old or new.
    - But, we can tell S is outside of the window!

# Proposal

- Increase max shift count is window scale option
  - Use 15 as max shift count
  - New maximum window size will be 2^31-2^15 bytes

# Signaling

- Possible approach
  - Use new TCP option
    - Notify peer using new max shift count
  - No signaling. Just use shiftcount=15 in WS option
    - RFC7323 allows to receive shiftcount=15
      - Parse as shiftcount=14
    - As long as a node can advertise new maximum window size (or close value), there is almost no harm
      - Conventional peer will parse it as current max window size (or close value)

# Thank You!

Please read the draft for more detailed info