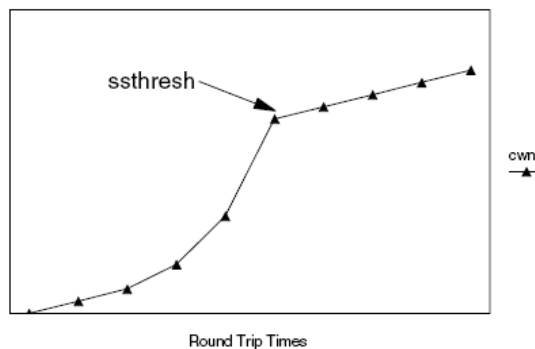


# [ TCP Congestion Algorithms ]

- v Slow Start
  - r Rate at which new packets are sent is rate at which acknowledgements are received
  - r A new window is added on the sender's side called the congestion window
  - r One packet is sent, when acknowledged, two are sent until acknowledged (congestion window of size two), then four are sent (congestion window of size four) until acknowledged, etc
  - r When packets start being lost, then this is the limit of throughput for sender
  - r Slow Start optimises throughput when communicating in scenarios of intermediate routers

# [ Congestion Avoidance ]

- v Congestion Avoidance
  - γ When a timeout occurs or duplicate ACKs are received, one assumes that there is network congestion
    - v Assumption that packet loss due to corruption is very rare
  - γ When this occurs, one saves half the size of the current congestion window in ssthresh and one sets the current congestion window back to one.
  - γ On timeout, usually one performs slow start at this stage and when the new congestion window is equal to ssthresh, one uses congestion avoidance growth
    - v Congestion Avoidance uses a linear growth pattern



(diagram taken from IBM Redbook)

# [ Fast Retransmit and Recovery ]

## v Fast Retransmit

- γ When an out-of-order packet is received, the receiver will send a duplicate ACK
- γ Sender has no way of knowing if duplicate ACK is due to out-of-order packet or a lost packet
- γ The sender will then wait for a small number of duplicate ACKs
  - v If three or more ACKs are received, the sender will immediately re-transmit the lost packets without waiting for a timeout

## v Fast Recovery

- γ After enabling fast retransmit, one sets the congestion window to the region to enable congestion avoidance (not slow start)
- γ This is done since if duplicate ACKs were received, that means that the flow in between the sender and receiver is still stable, thus one should not resort to slow start congestion window