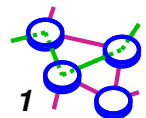


CS551

Reliable Multicast

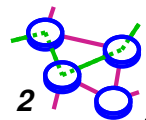
Bill Cheng

<http://merlot.usc.edu/cs551-f12>



Sender Reliable Error Detection

- ➡ Wait for ACKs from all receivers, re-send on timeout or selective ACK
 - ➡ (+) easy resource management
 - ➡ (-) wait for ACK
 - ➡ (-) receiver state in sender, not scalable
 - ➡ (-) ACK implosion

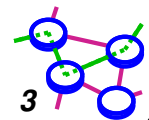


Receiver Reliable Error Detection

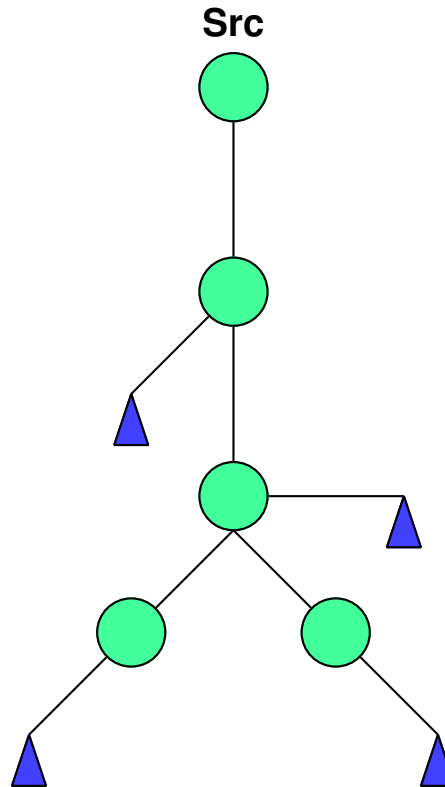


Receivers NACKs lost packets

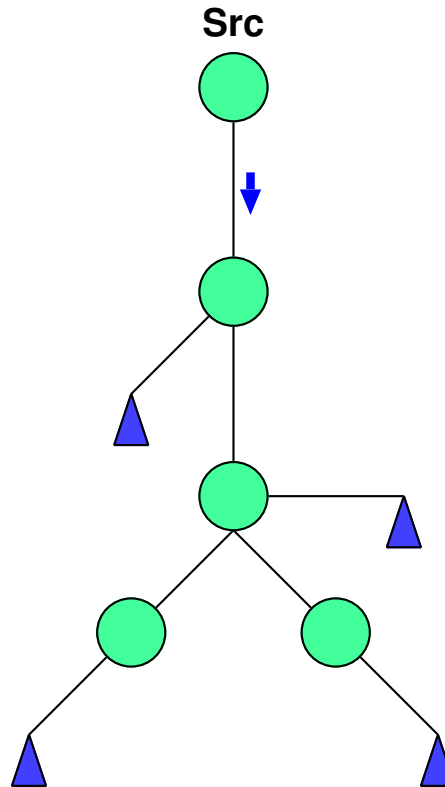
- (+) **no state at sender - good for multicast**
- (-) **does not provide 100% reliability**
- (-) **NACK implosion**



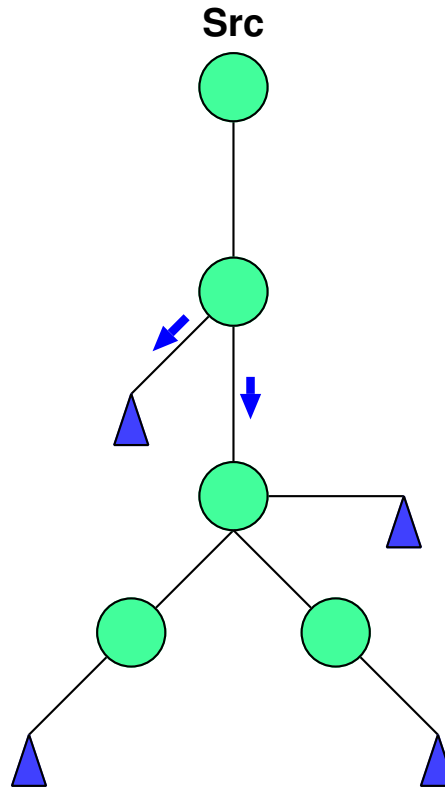
Implosion



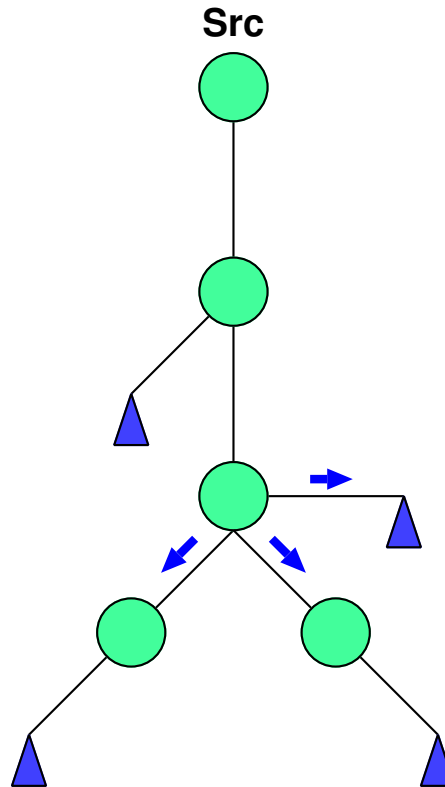
Implosion (Cont...)



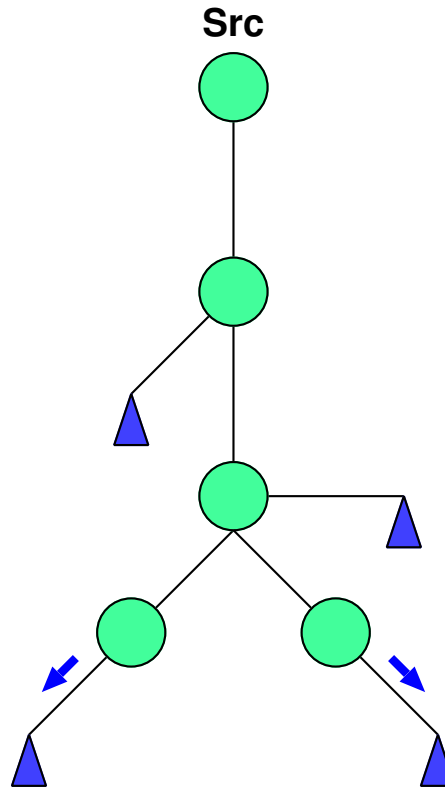
Implosion (Cont...)



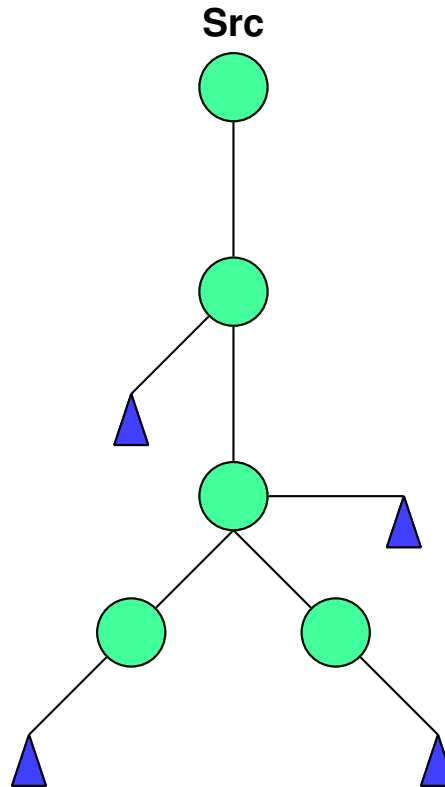
Implosion (Cont...)



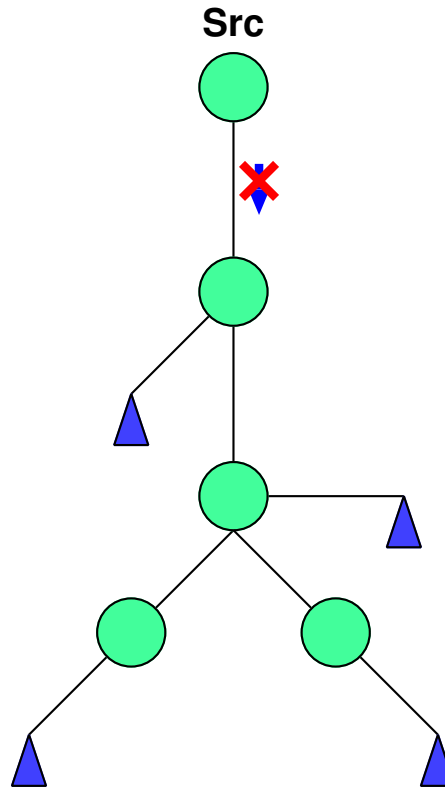
Implosion (Cont...)



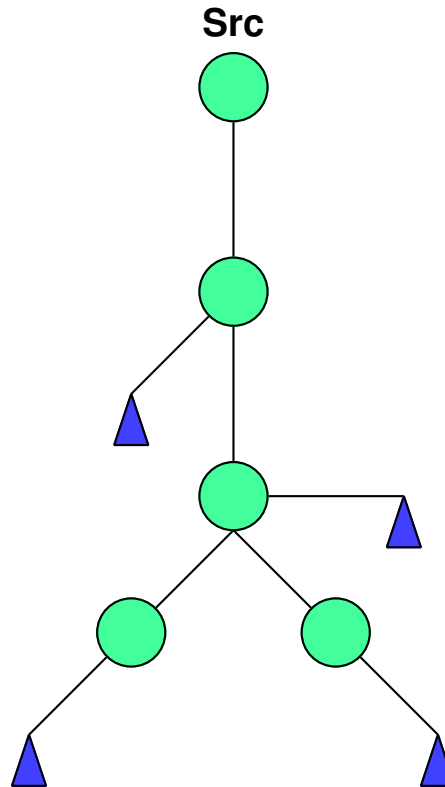
Implosion (Cont...)



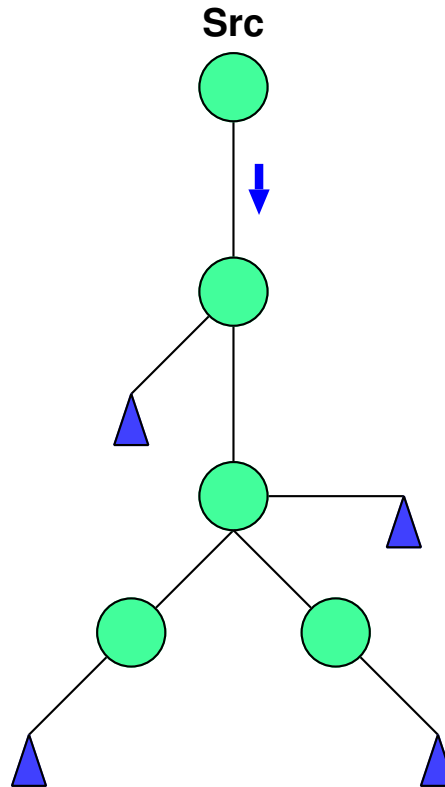
Implosion (Cont...)



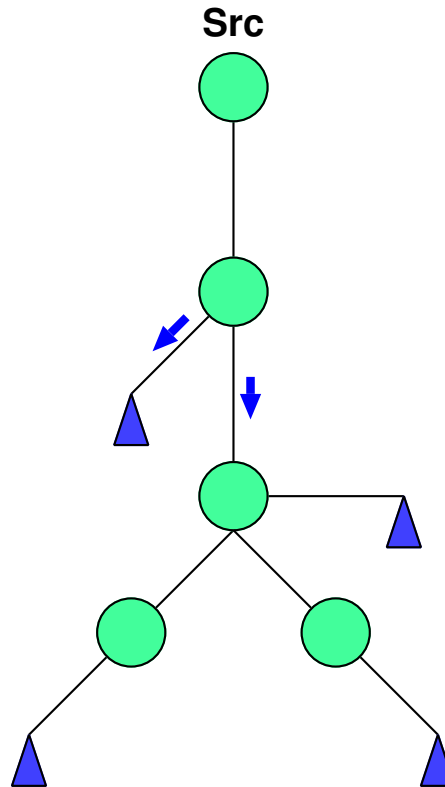
Implosion (Cont...)



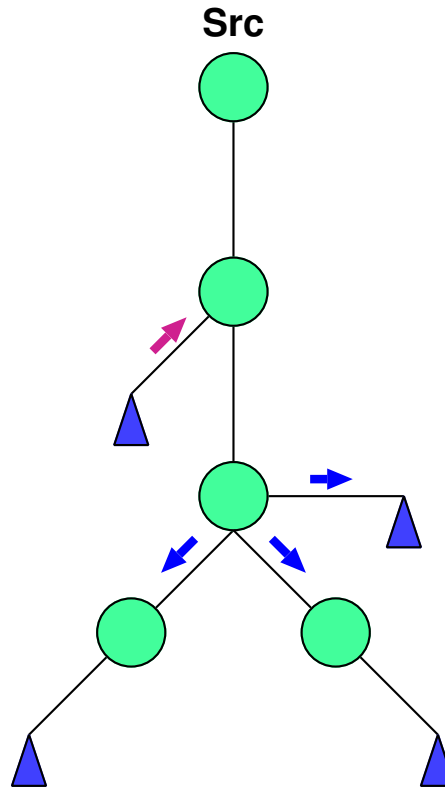
Implosion (Cont...)



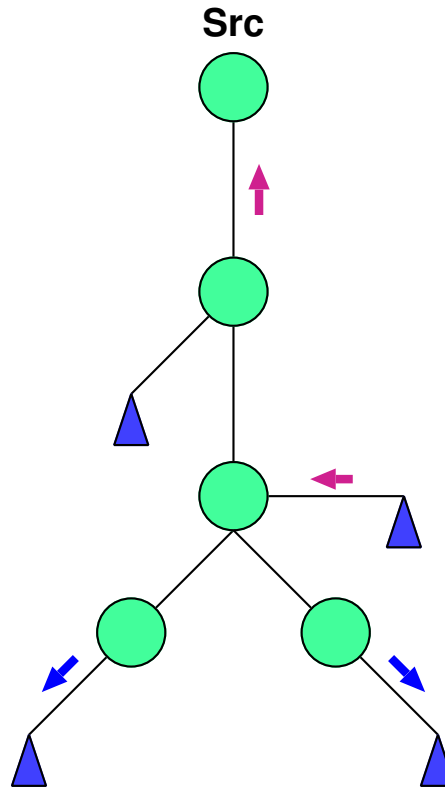
Implosion (Cont...)



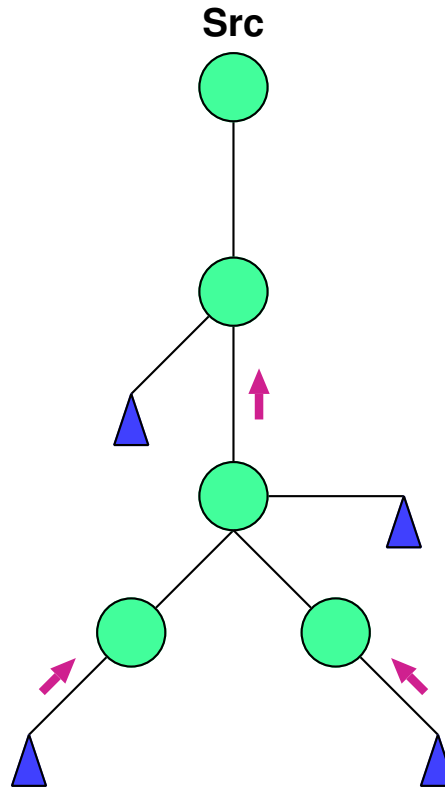
Implosion (Cont...)



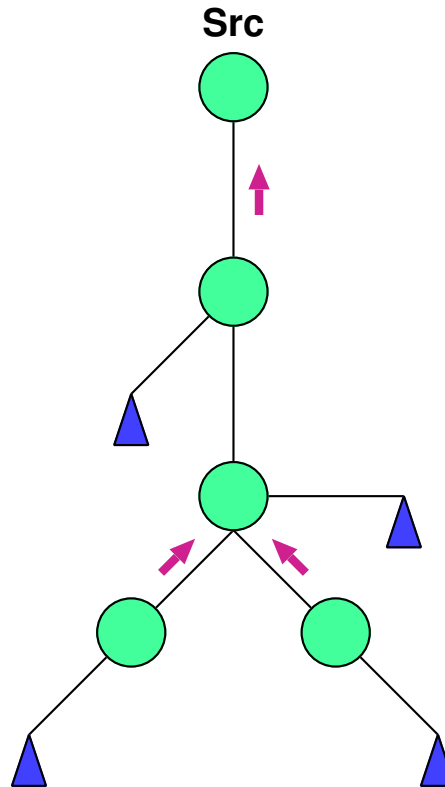
Implosion (Cont...)



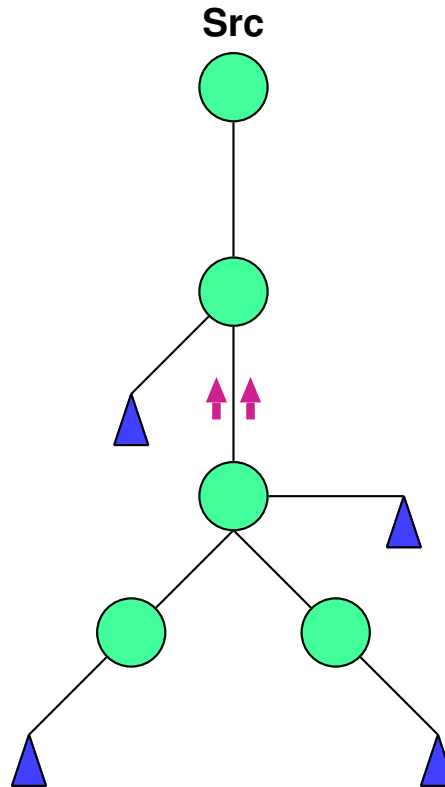
Implosion (Cont...)



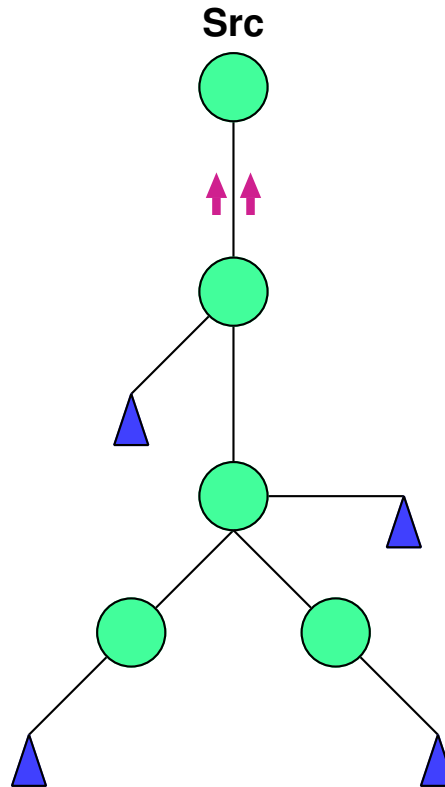
Implosion (Cont...)



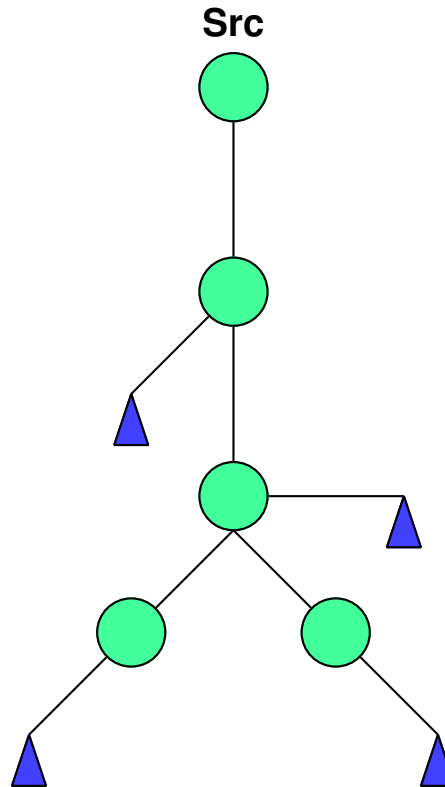
Implosion (Cont...)



Implosion (Cont...)



Implosion (Cont...)



Retransmission



Re-transmitter

- sender
- receiver



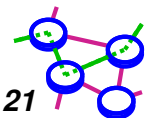
How to retransmit?

- unicast, multicast, scoped multicast, retransmission group, etc.

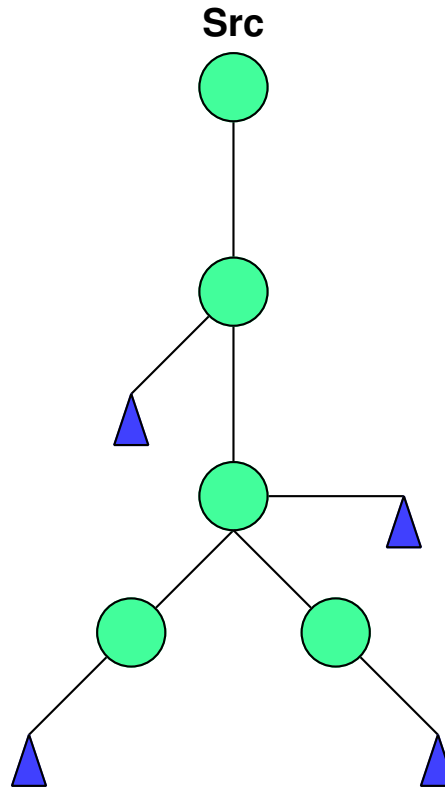


Problem with sender retransmissions

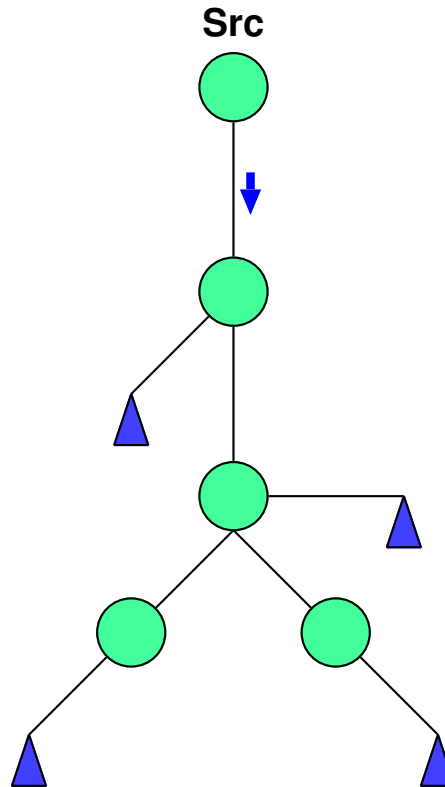
- exposure



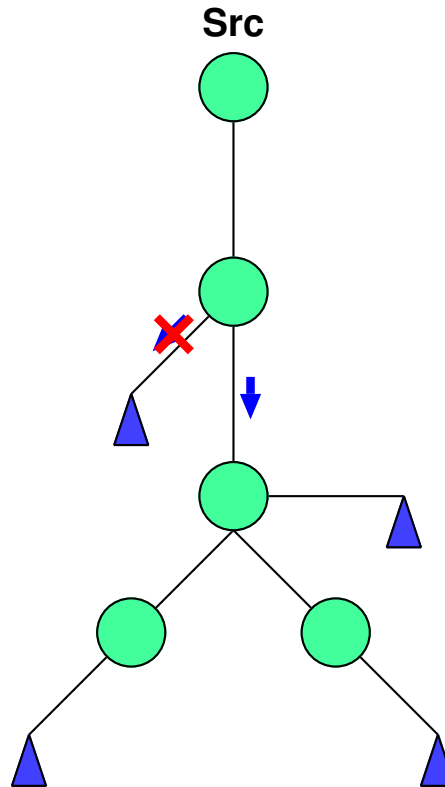
Exposure



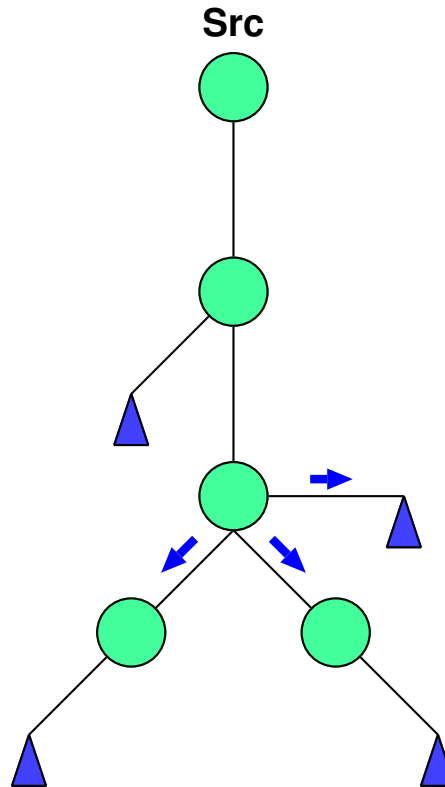
Exposure (Cont...)



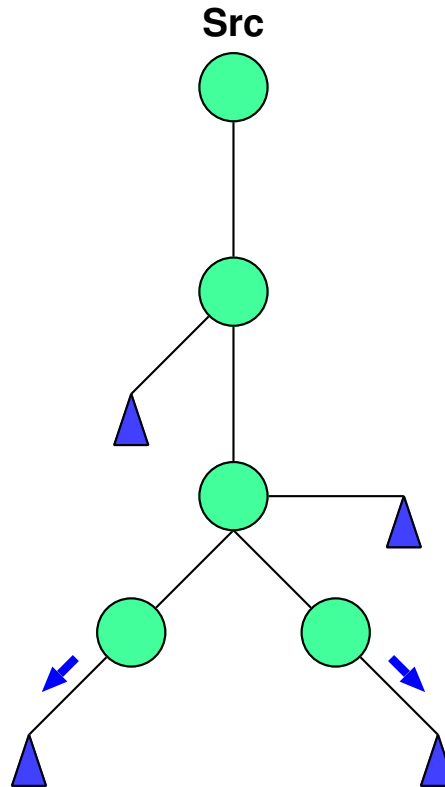
Exposure (Cont...)



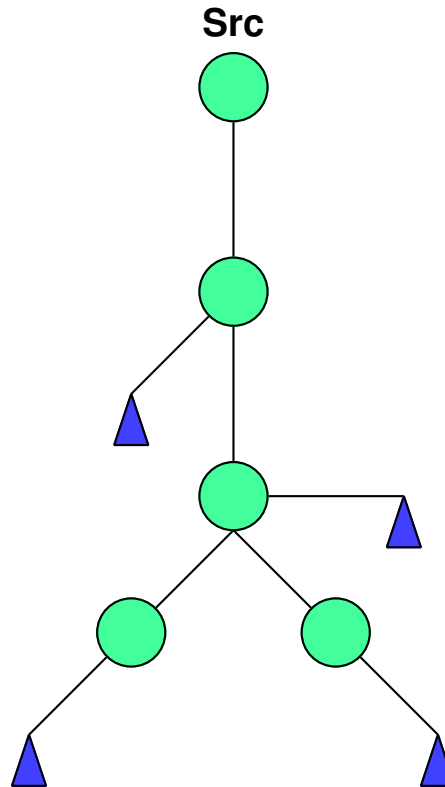
Exposure (Cont...)



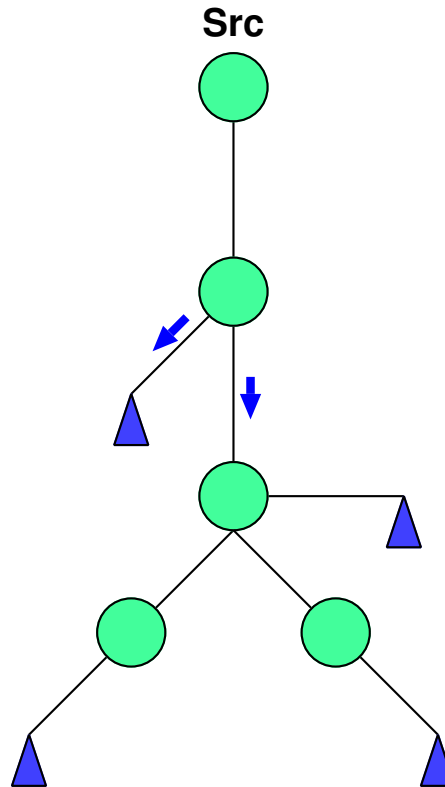
Exposure (Cont...)



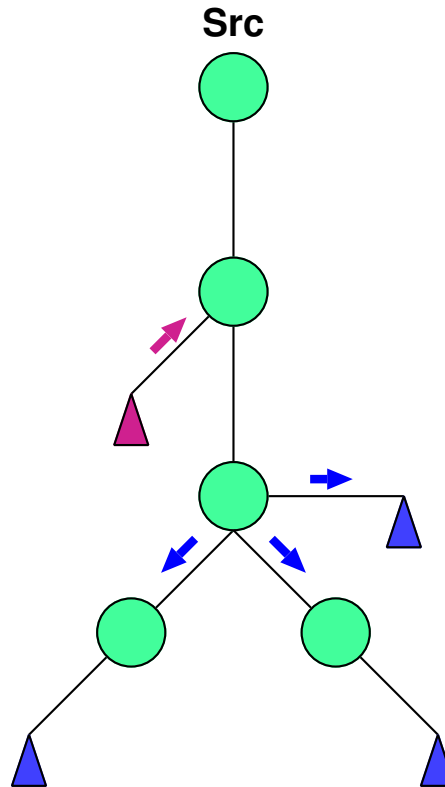
Exposure (Cont...)



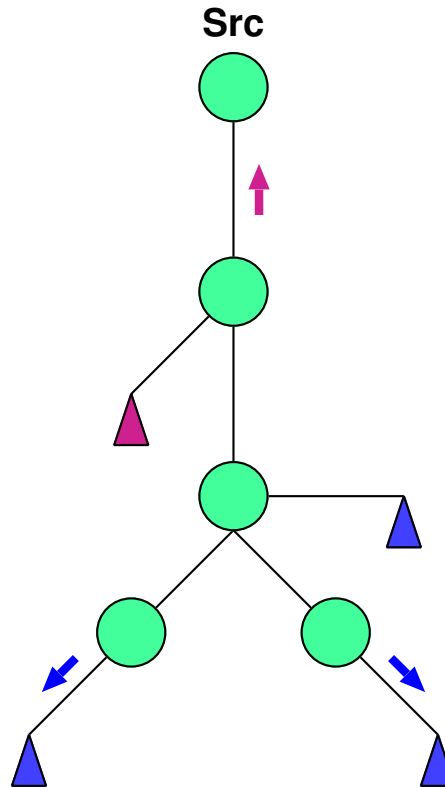
Exposure (Cont...)



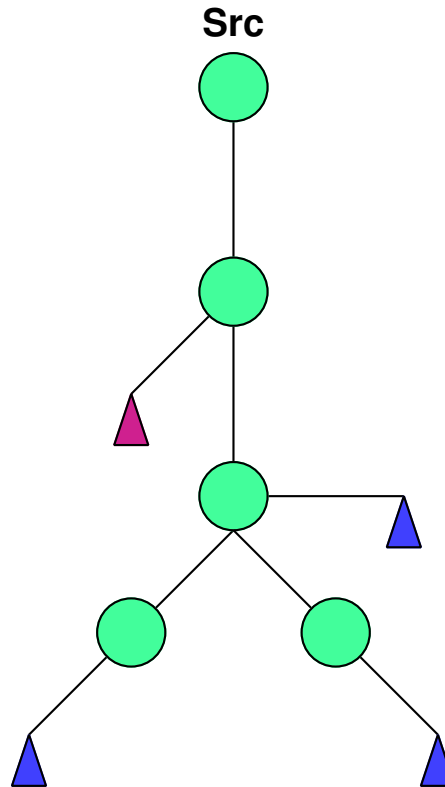
Exposure (Cont...)



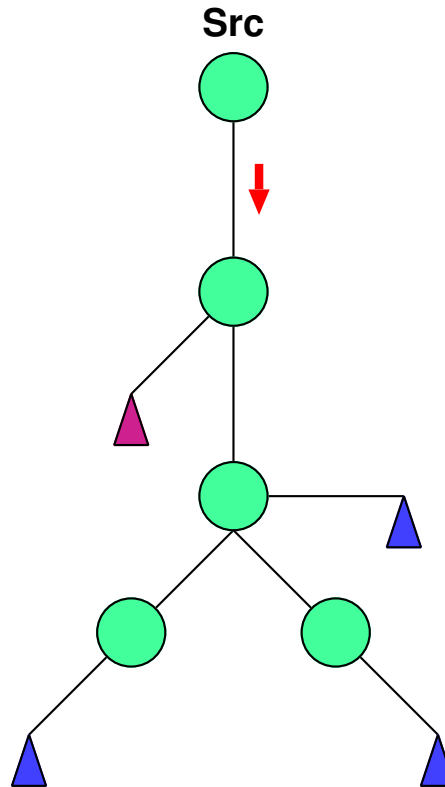
Exposure (Cont...)



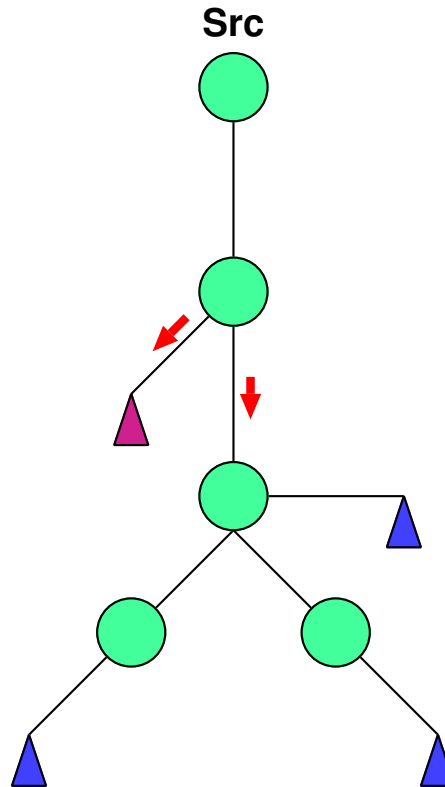
Exposure (Cont...)



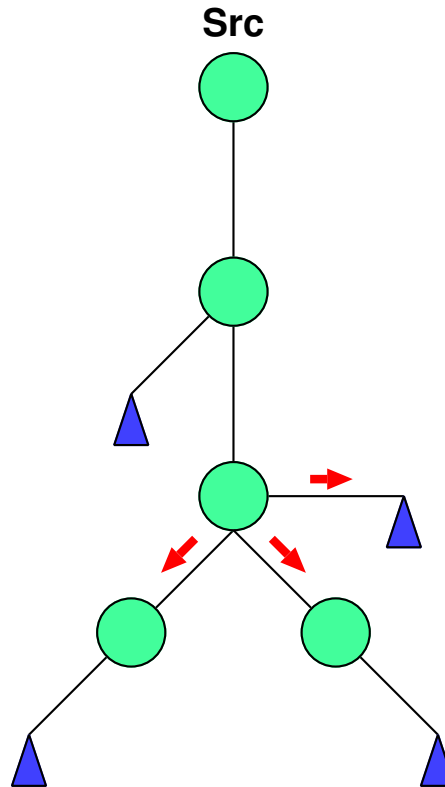
Exposure (Cont...)



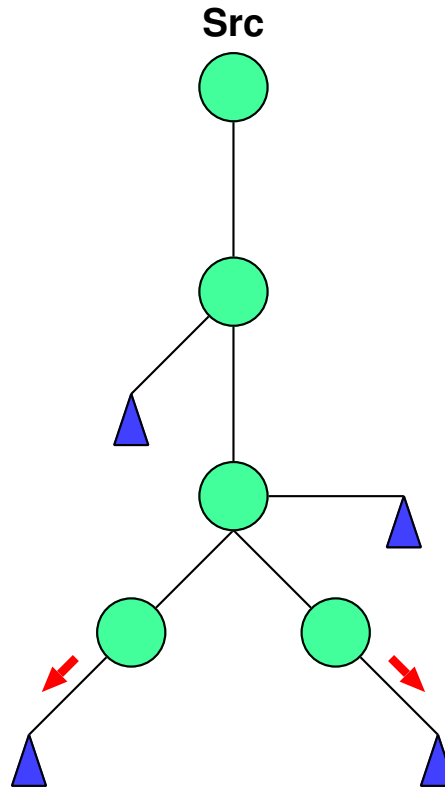
Exposure (Cont...)



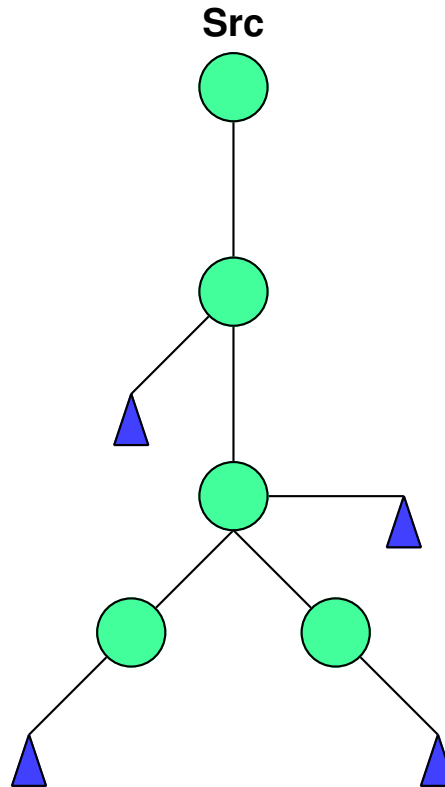
Exposure (Cont...)



Exposure (Cont...)



Exposure (Cont...)

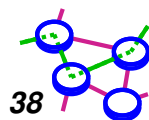


Aside - Using the Routers



Routers do transport level processing

- buffer packets**
- fuse ACKs**
- send retransmissions**
- this solves implosion and exposure problems, but:**
 - not scalable**
 - violate end-to-end argument**



CS551

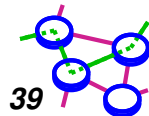
Scalable Reliable

Multicast

[Floyd97c]

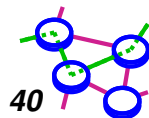
Bill Cheng

<http://merlot.usc.edu/cs551-f12>



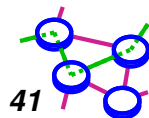
SRM (Scalable Reliable Multicast)

- ➔ Originally designed for *wb*
- ➔ Receiver reliable
 - ▬ NACK-based
- ➔ Every member may multicast NACK or retransmission
- ➔ No assistance from routers

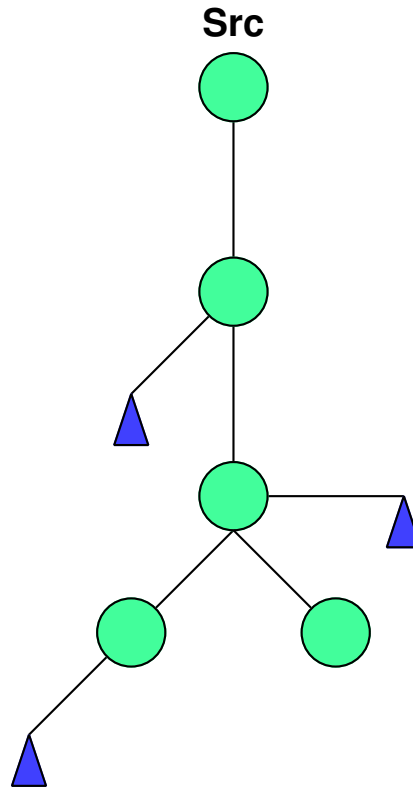


SRM Retransmission

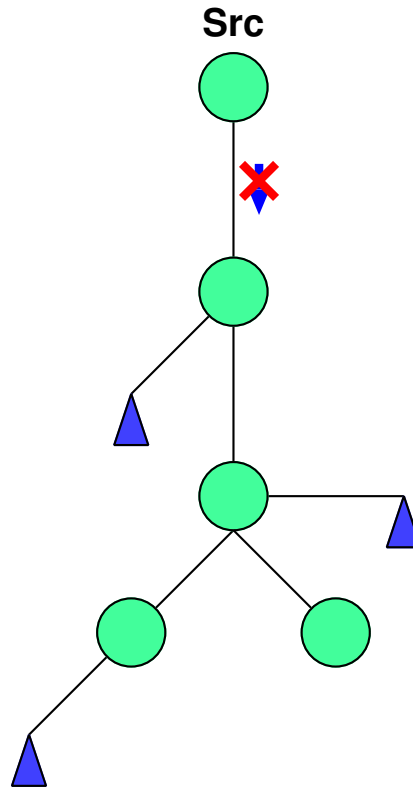
- ➔ All transmissions go to everyone
- ➔ Receivers notice losses
 - ▬ how?
 - if there is a missing sequence number from subsequence data
 - use session messages to deal with loss of last packet
- ➔ Losses result in repair requests (to everyone)
- ➔ Repair requests produce repairs (to everyone)
- ➔ Requests and repairs are spaced and avoided
 - ▬ via randomization and suppression



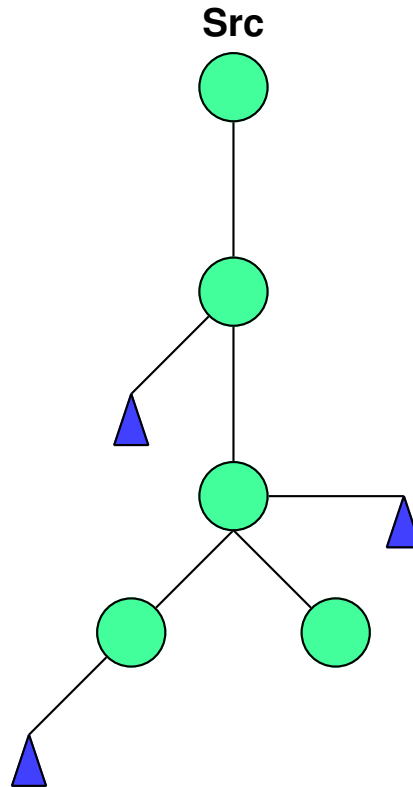
SRM Request Suppression



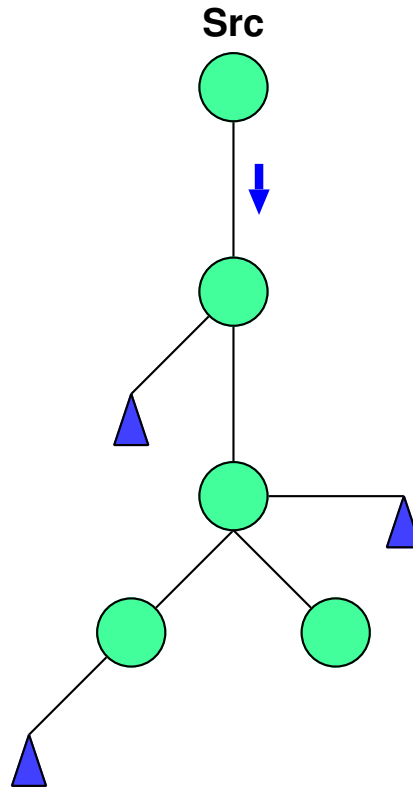
SRM Request Suppression (Cont...)



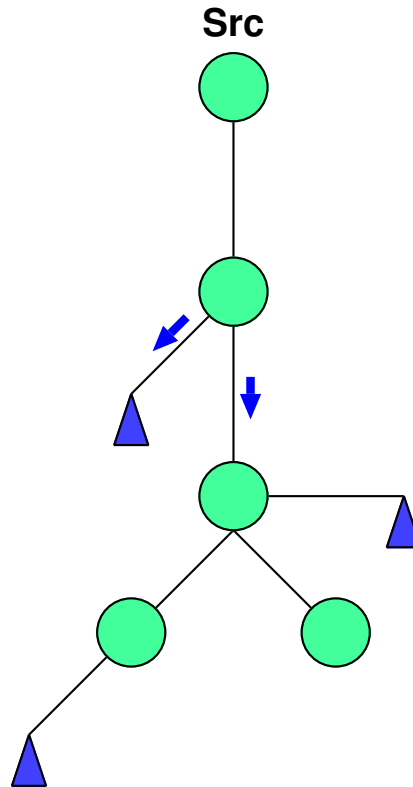
SRM Request Suppression (Cont...)



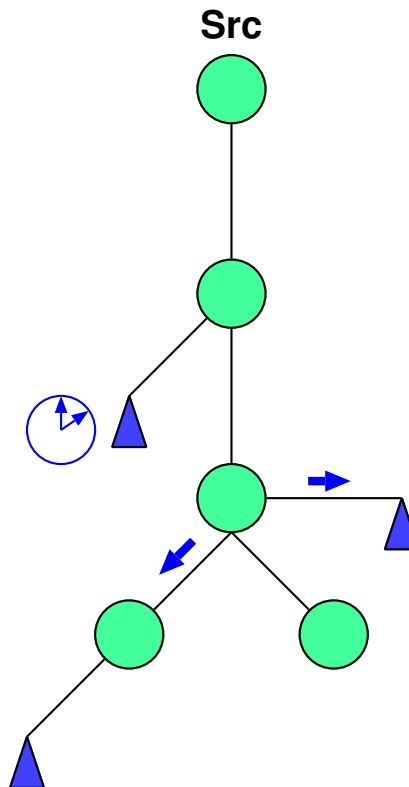
SRM Request Suppression (Cont...)



SRM Request Suppression (Cont...)

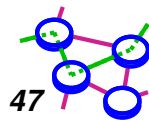


SRM Request Suppression (Cont...)

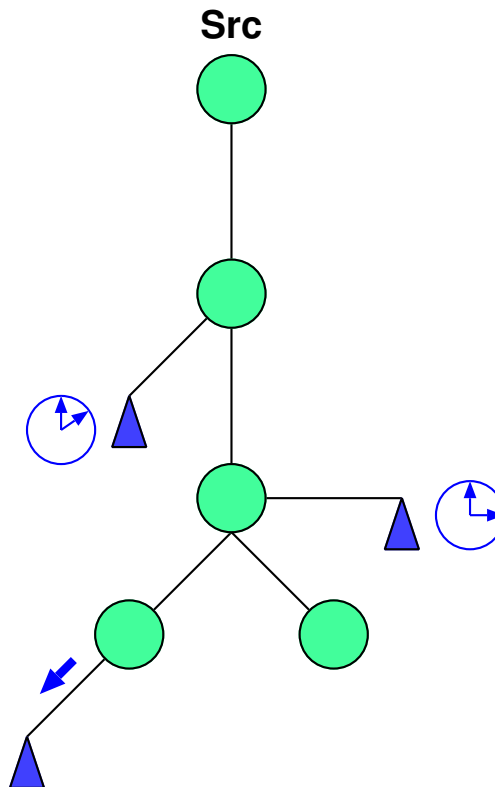


When noticing skipped sequence number:

➡ start a timer whose timeout is proportional to distance from Src

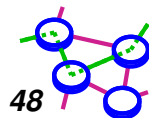


SRM Request Suppression (Cont...)

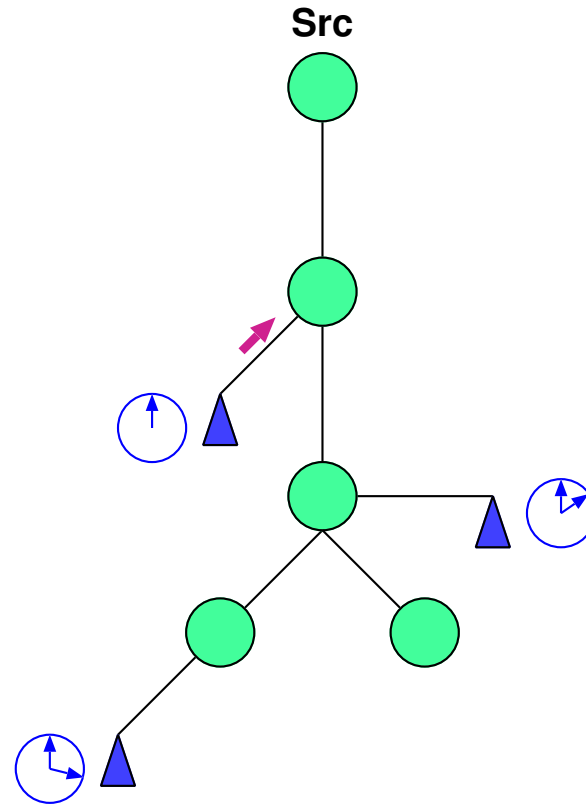


When noticing skipped sequence number:

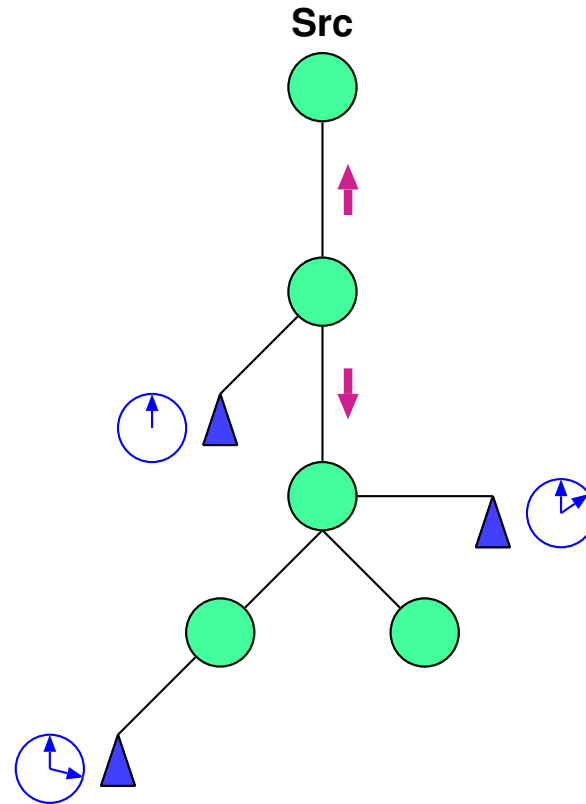
- ▢ start a timer whose timeout is proportional to distance from Src



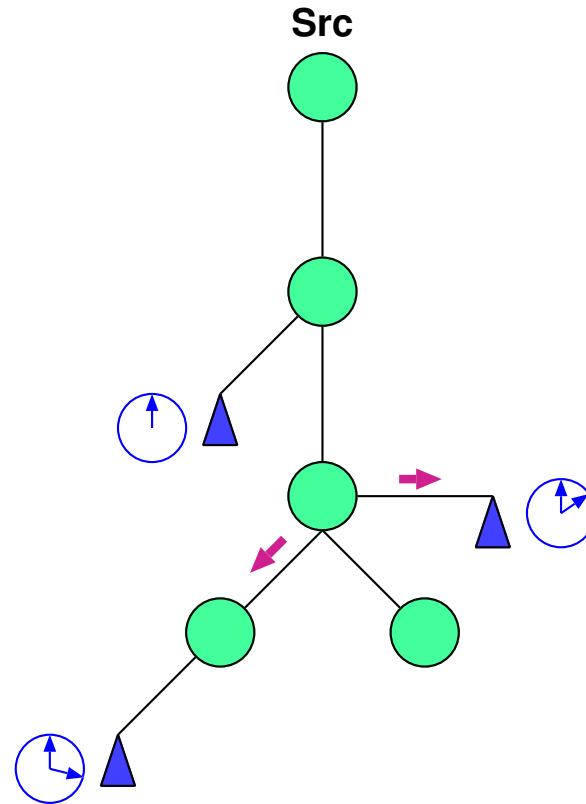
SRM Request Suppression (Cont...)



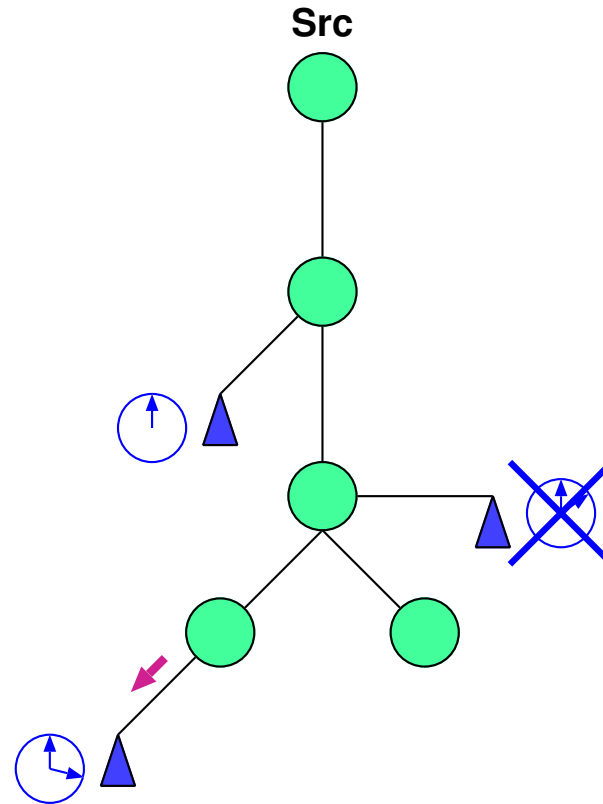
SRM Request Suppression (Cont...)



SRM Request Suppression (Cont...)



SRM Request Suppression (Cont...)

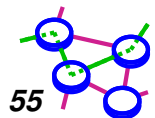


SRM Summary

- ➔ **NACK/Retransmission suppression**
 - ➔ delay before sending
 - ➔ delay based on RTT estimation
 - ➔ deterministic and stochastic components

- ➔ **Periodic session messages**
 - ➔ discover lost final packets
 - ➔ used to estimate OTT from sender to receivers

- ➔ **Adaptive algorithm to adjust constants**



What's Missing?

➡ Losses at link (A,C)
causes retransmission
to the whole group

➡ *Better: only retransmit
to those members who
lost the packet*

⇒ local recovery

○ router support

⇒ A to C

⇒ where have we seen
this before?

○ SNOOP

