Multicast Congestion Control: how did we solve the crying baby problem?

ICN Congestion Control

ACN Course

Email: *arumaithurai@cs.uni-goettingen.de*

Acknowledgement

SAID: A Control Protocol for Scalable and Adaptive Information Dissemination in ICN

Jiachen Chen[*,†], **Mayutan Arumaithurai**[†], Xiaoming Fu[†], and K.K. Ramakrishnan[‡]

[*] WINLAB, Rutgers University, NJ, U.S.A.
[†] Institute of Computer Science, University of Göttingen, Germany
[‡] University of California, Riverside, CA, U.S.A.

Email: *arumaithurai@cs.uni-goettingen.de*

RUTGERS

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

UC RIVERSIDE UNIVERSITY OF CALIFORNIA

Congestion

Congestion


ZIP
The urge to merge
When traffic is heavy
travel in two lanes
to the merge point
then take turns merging.
MERGE
HERE

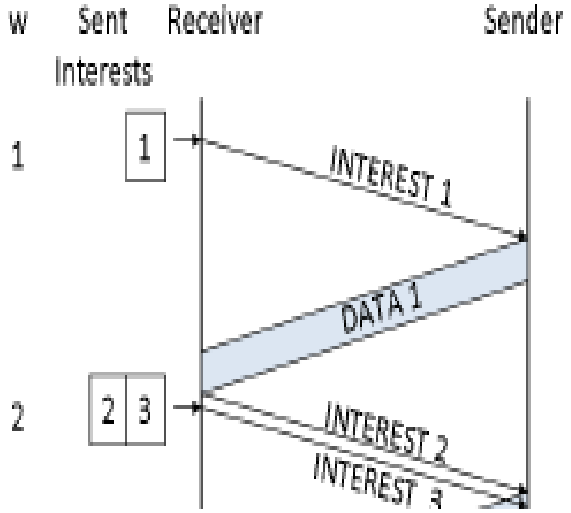
ICP [1]

- Interest control protocol
- Receiver Driven
- AIMD- similar to TCP
- Retransmission timer
- Congestion is detected when timer expires
- Reliable
- Fair
- Efficient usage of bandwidth

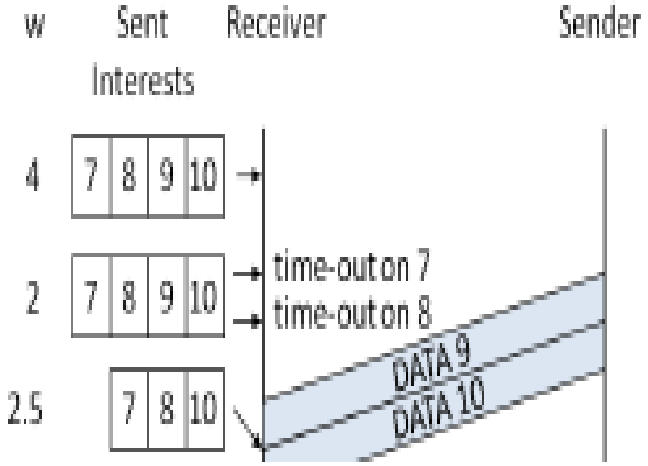
[1] G. Caroglio, M. Gallo, and L. Muscariello. Icp: Design and evaluation of an interest control protocol for content-centric networking. Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, 2012

Working

Additive Increase

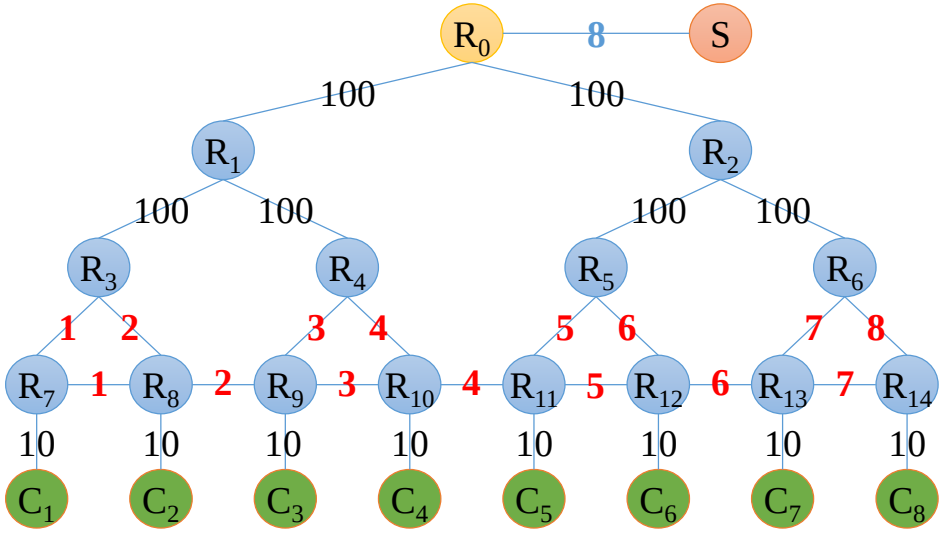


Multiplicative Decrease



Multicast Congestion Control

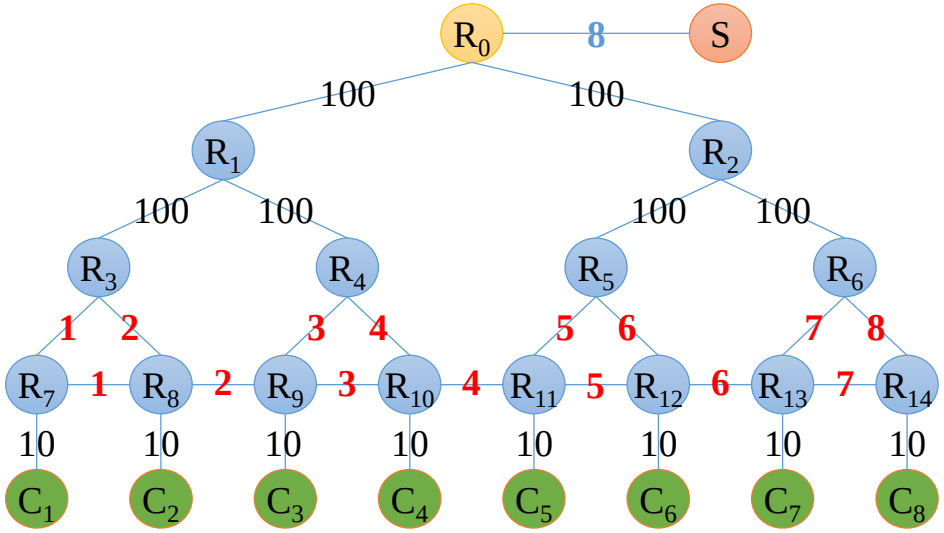
Efficient In-network Multicast – Desired, yet Difficult



- Efficient multicast is always a desired feature of the network
  - Applications: live streaming (e.g., livecoding.tv), file delivery (e.g., system updates), gaming, *etc.*

Key advantage: optimal in-network replication of packets

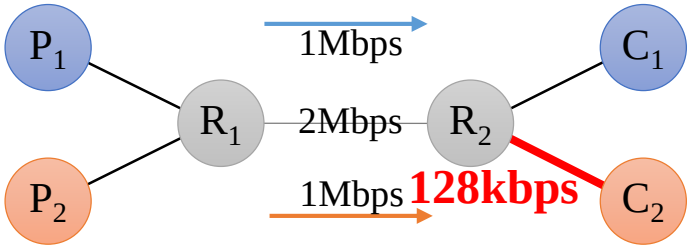
Efficient In-network Multicast – Desired, yet Difficult



Difficulty: Requirement of Reliability, Fairness

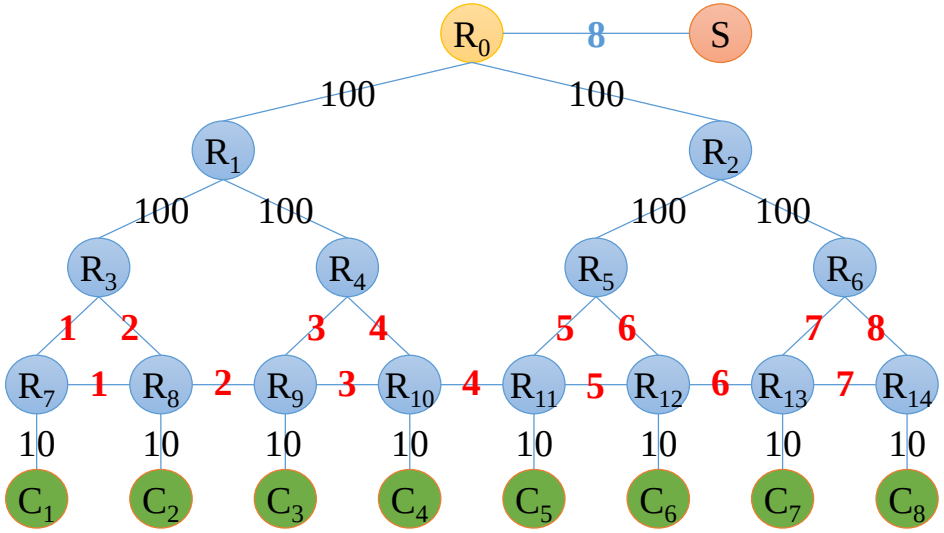
- Heterogeneous consumers
  - Bottleneck: location and fair-rate varies over time

Efficient In-network Multicast – Desired, yet Difficult



- Hop-by-hop reliability & flow-based fair queueing is not a good solution either
  - Hop-by-hop reliability is inefficient in forwarding since each router has to check the content
  - Fair queueing can cause inefficiency (compared to max-min)

Efficient In-network Multicast – Desired, yet Difficult

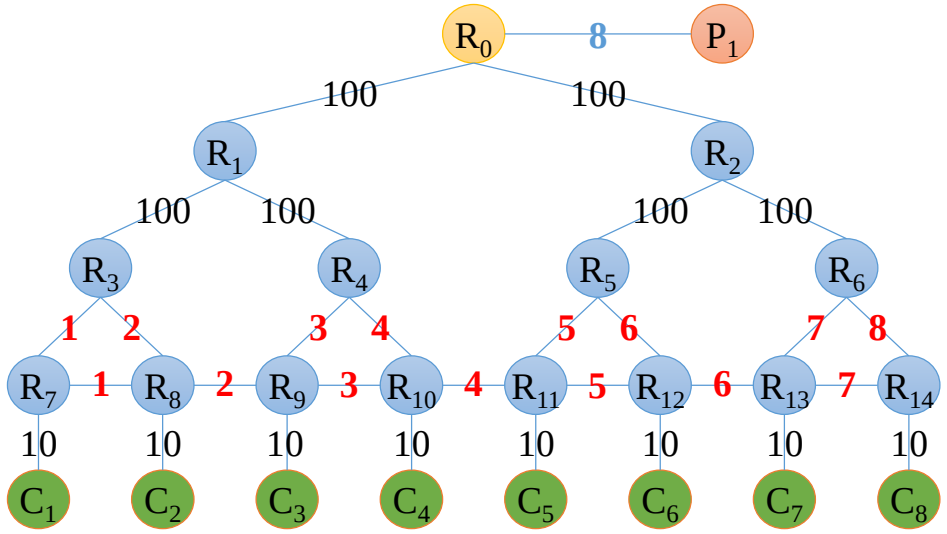


- *pgmcc* – a working solution
  - Aligns the sending rate to the slowest consumer
  - Ensures (and couples) reliability and fairness

However, one slow receiver can affect the performance of the whole group

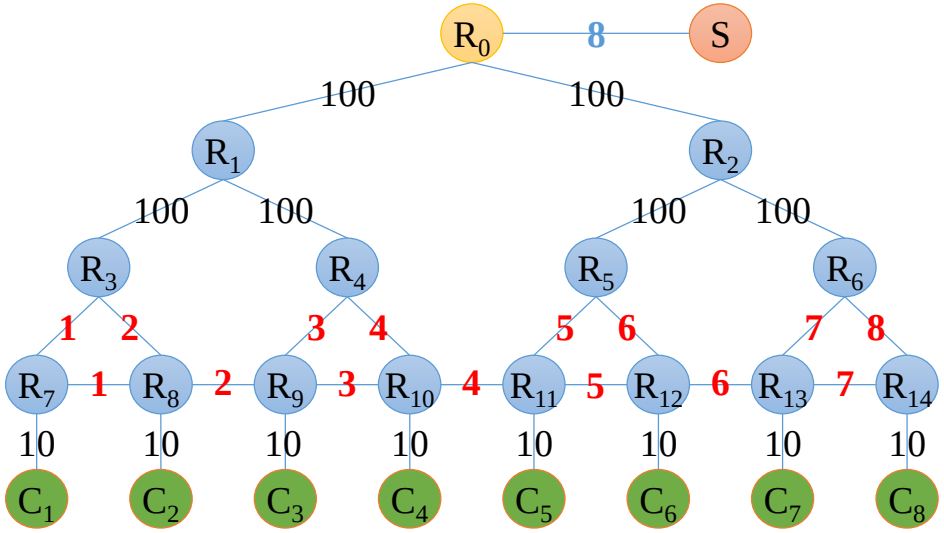
This is also known as the crying baby problem

NDN – An Architecture Born with Multicast Capability



- Solution for <u>fairness</u> and <u>reliability</u> in NDN
  - Flow balance – <u>*at maximum one*</u> Data per Interest
  - Constant packet size – 4KB/packet
  - Flow Control, Fairness: Consumers can indicate how much data they can receive by adjusting the # of Interests in flight
  - Reliability: Sequence number in each packet allows the consumers request for missing packets

NDN – An Architecture Born with Multicast Capability

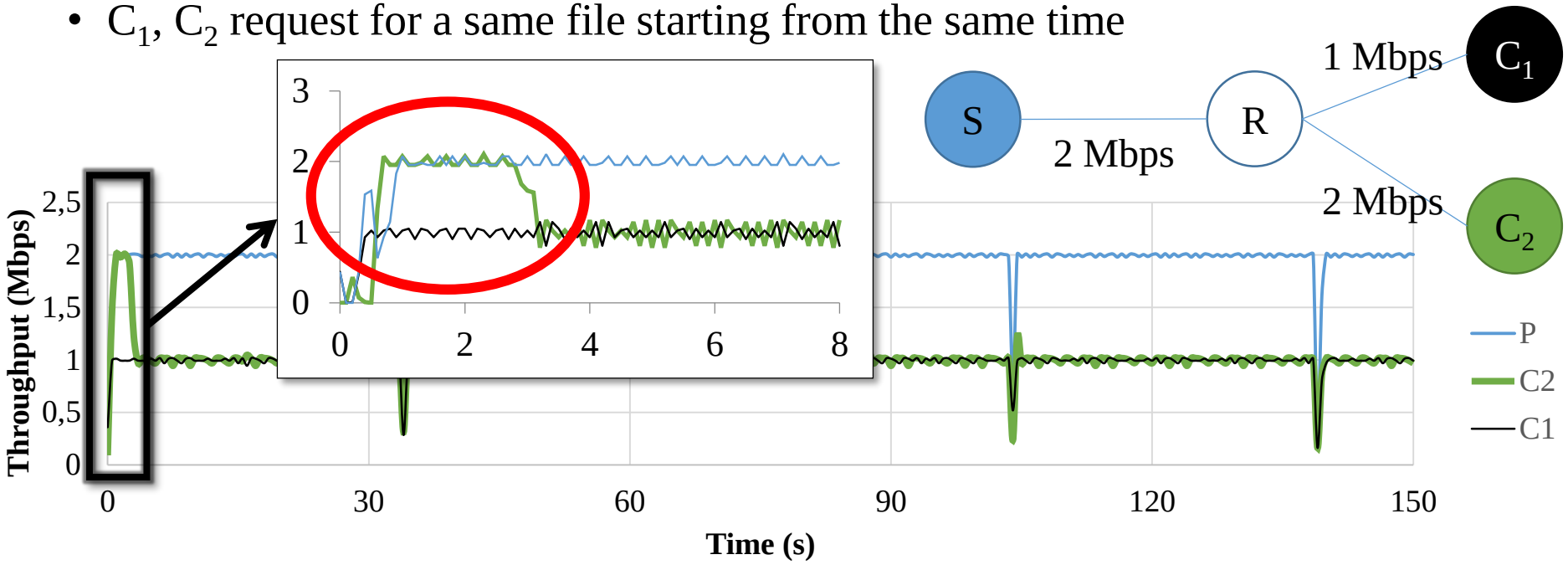


- NDN desires multicast: uses a stateful data plane
  - Pending Interest Table (PIT) – unsatisfied requests + corresponding incoming faces
  - Requests with temporal locality can be aggregated in the PIT
  - Data packets will be replicated in the network

Name	Faces
/ICN/16/papers/SAID.pdf/_s0	***R3, R4***
…	…

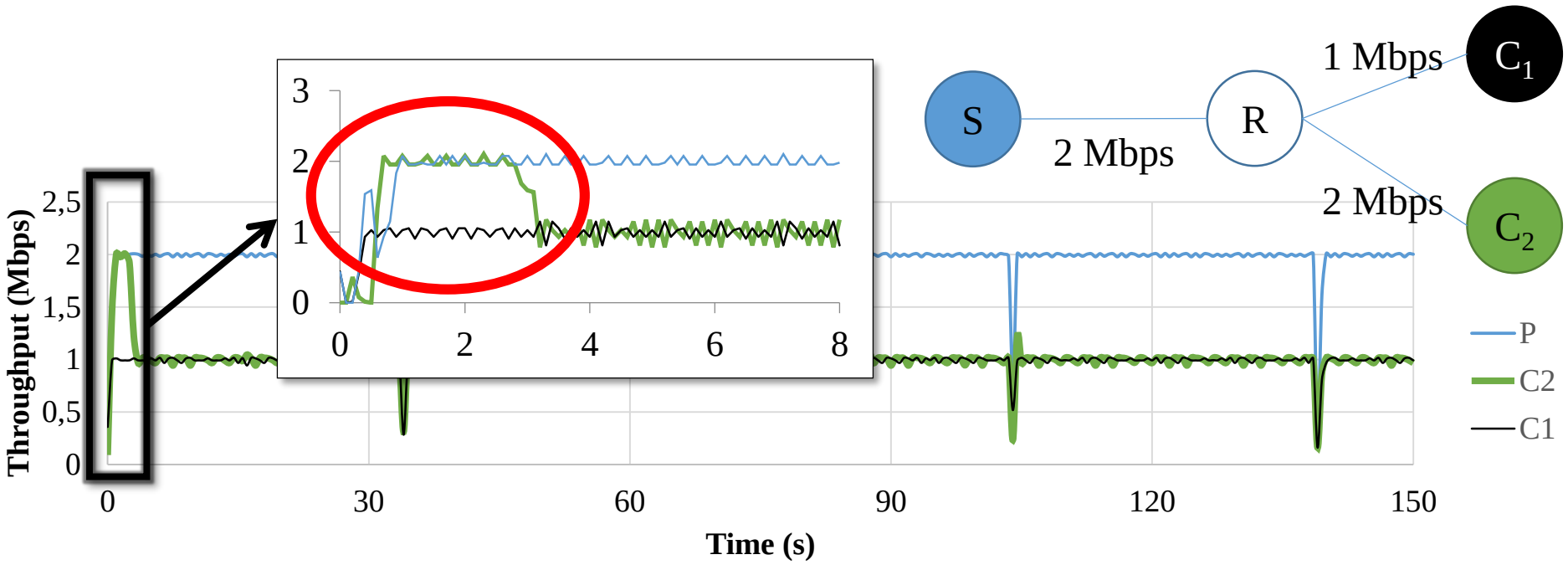
Problem with NDN Multicast – Consumers Getting Out-of-synch

- Testbed Evaluation with Simple Topology
  - C_1, C_2 request for a same file starting from the same time

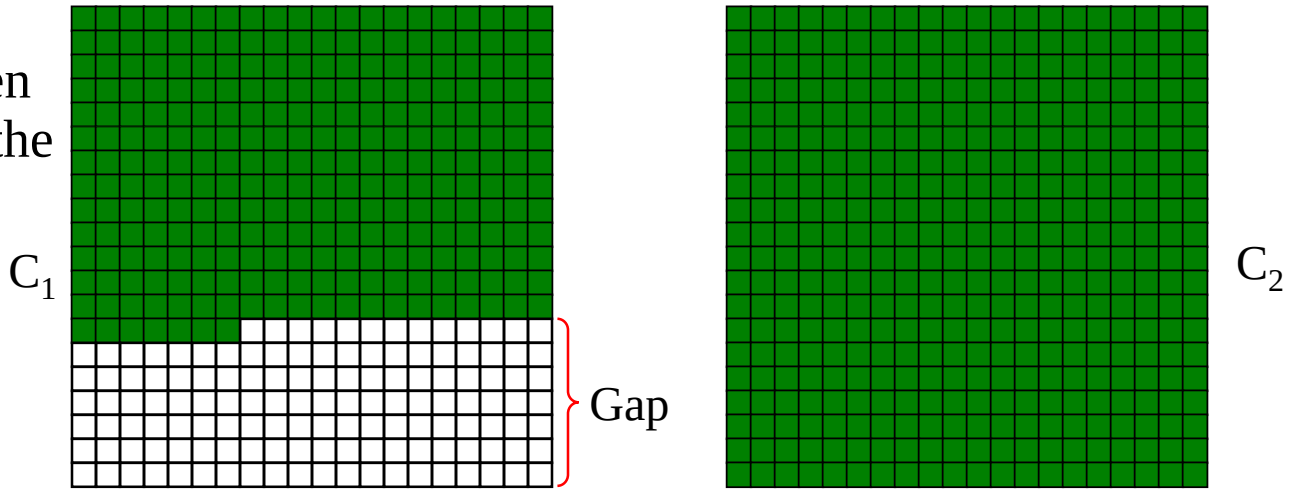


Why?

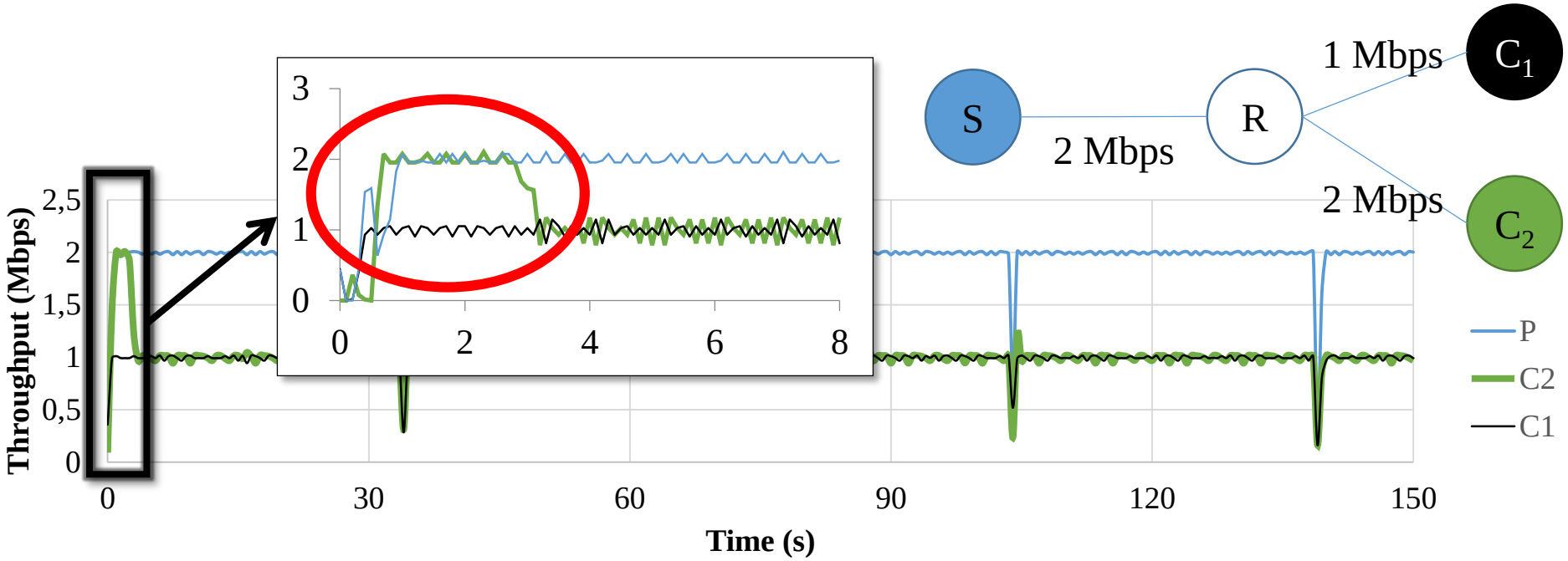
Problem with NDN Multicast – Consumers Getting Out-of-synch



- Why?
  - Out-of-synch – when the gap is larger than the cache size
  - Cache cycling therefore the two consumers are seen as different flows



Problem with NDN Multicast – Consumers Getting Out-of-synch



The "two flows" compete the bandwidth between P and R

The "when" can vary according to the scenario, but it will eventually happen.

How to Prevent Out-of-Synch?

- Why out-of-synch is happening in NDN (Congestion Control Solutions)?
  - Couple *in-sequence reliability* with *fairness*

- We argue that:
  - Not every application requires reliability (e.g., games), but every application requires fairness
  - Even fewer applications require in-sequence reliability
    - E.g. file transfer, it is ok to eventually get the whole file
    - E.g. Live streaming, it is sufficient for the playout buffer to be full

Therefore, reliability can be decoupled from fairness in multicast dissemination.

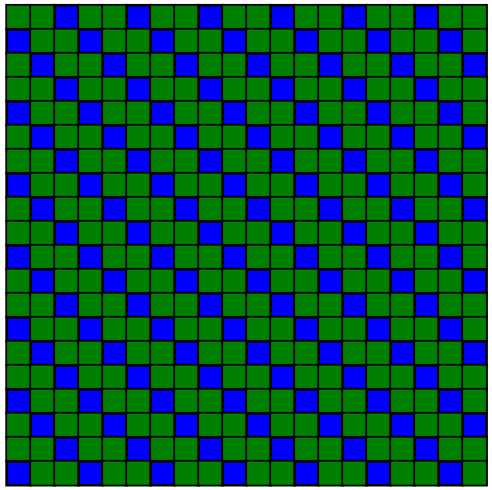
Intuition of our solution (SAID)

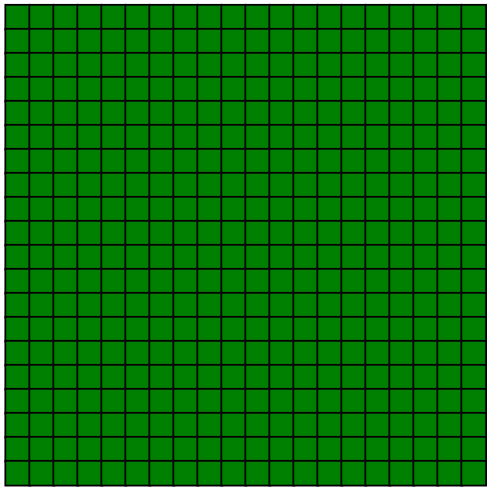
- The consumers control *how much* data they are going to receive, but *NOT which exact packets*
  - Control rate – adjust the # of in-flight requests (receiver window based flow balance)
- A new request type – Any Next Packet (ANP) request
  - Let the network decide which packets to send, but fresh packets

Intuition of our solution (SAID)

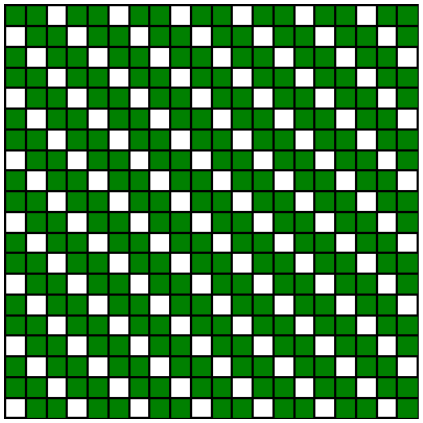
- The consumers control *how much* data they are going to receive, but *NOT which exact packets*
  - Control rate – adjust the # of in-flight requests (receiver window based flow balance)
- A new request type – Any Next Packet (ANP) request
  - Let the network decide which packets to send, but fresh packets
- ***Maximize the utility* of the packets that are sent *at the first time***
  - We can get 2/3 of the packets used twice regardless of the file size and no cache
- Seek to repair these holes later

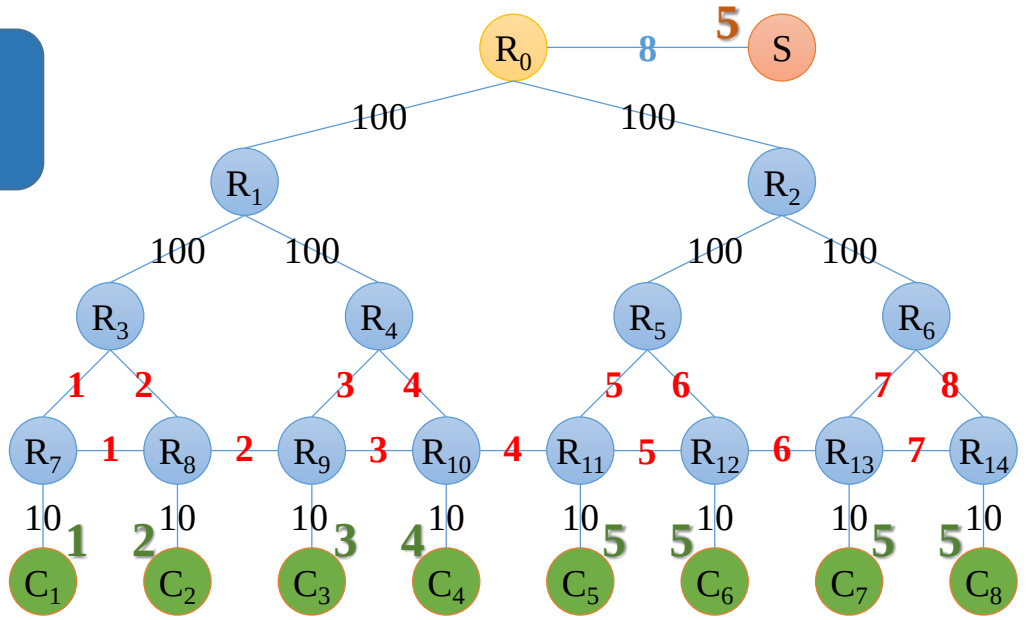
This occurs even if C1 or C2 start later





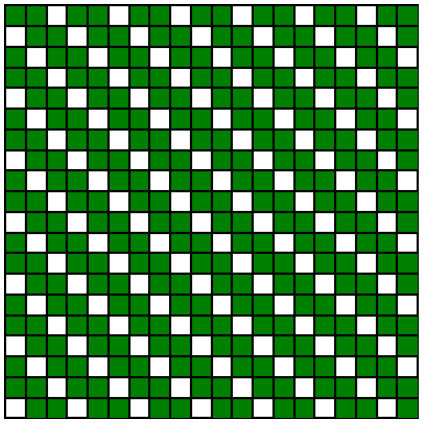
Fairness: Efficient Multicast Module (using ANP request)

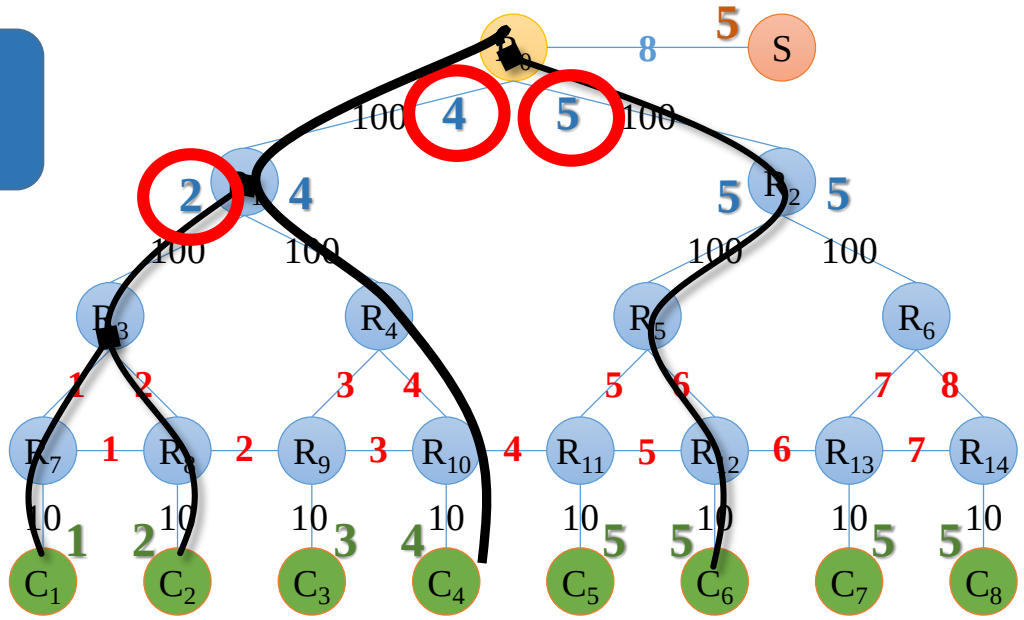




- With fairness, but no reliability guarantee
- Each consumer should receive (whichever is smaller):
  - Fair share on the bottleneck between sender and the consumer (no matter where the bottleneck is)
  - Sending rate of the sender

Fairness: Efficient Multicast Module (using ANP request)



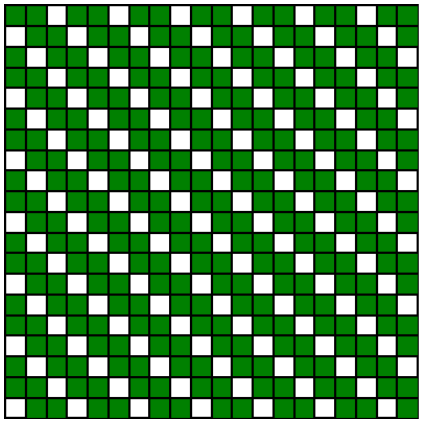


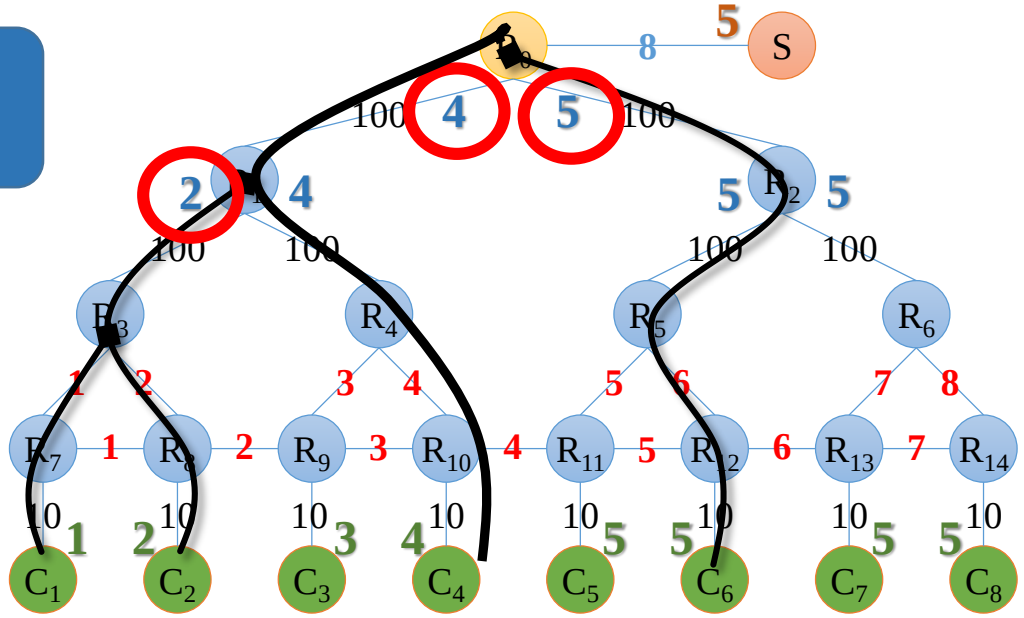
- Efficient link utility
  - The network transmits *at the receive rate on each hop* between the branching point and the consumer
  - For a consumer C, its branching point BR_C is defined as:
    - The bifurcation router nearest to C that has another consumer that can receive faster than C
    - 1st hop router of the provider when the receive rate is equal to the sending rate

Essentially Max-Min Fairness

SAID – Building Blocks

Fairness: Efficient Multicast Module (using ANP request)

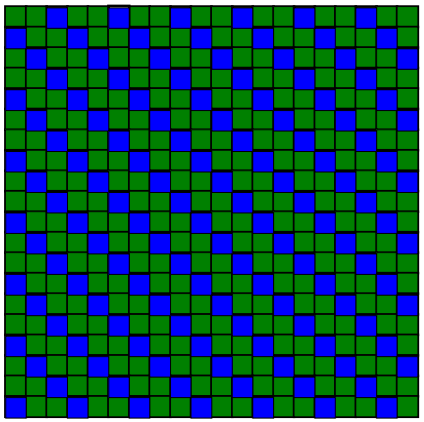


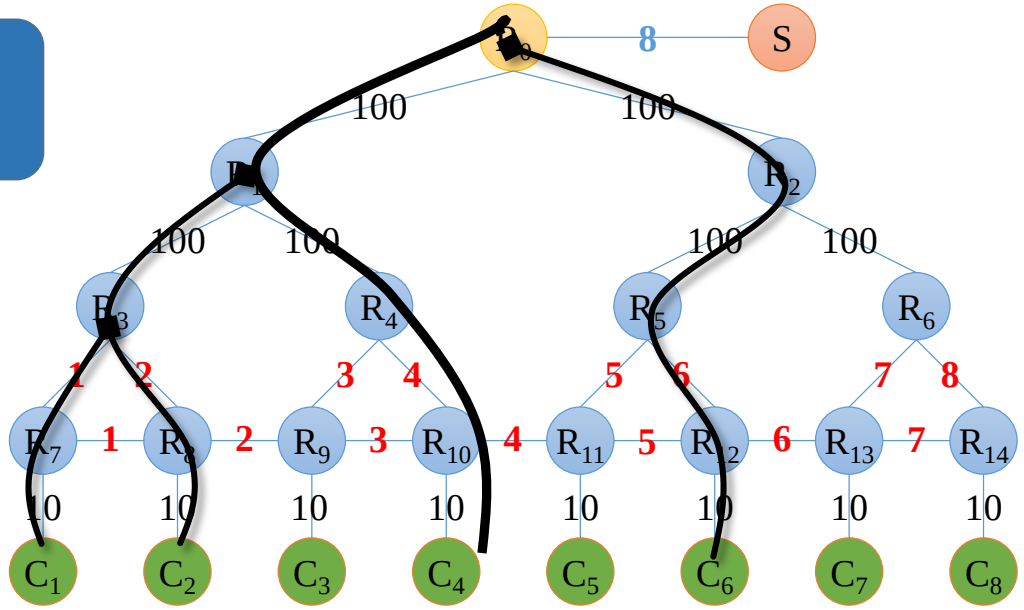


Issue: Holes are not an indication of congestion

SAID – Building Blocks

Fairness: Efficient Multicast Module (using ANP request)



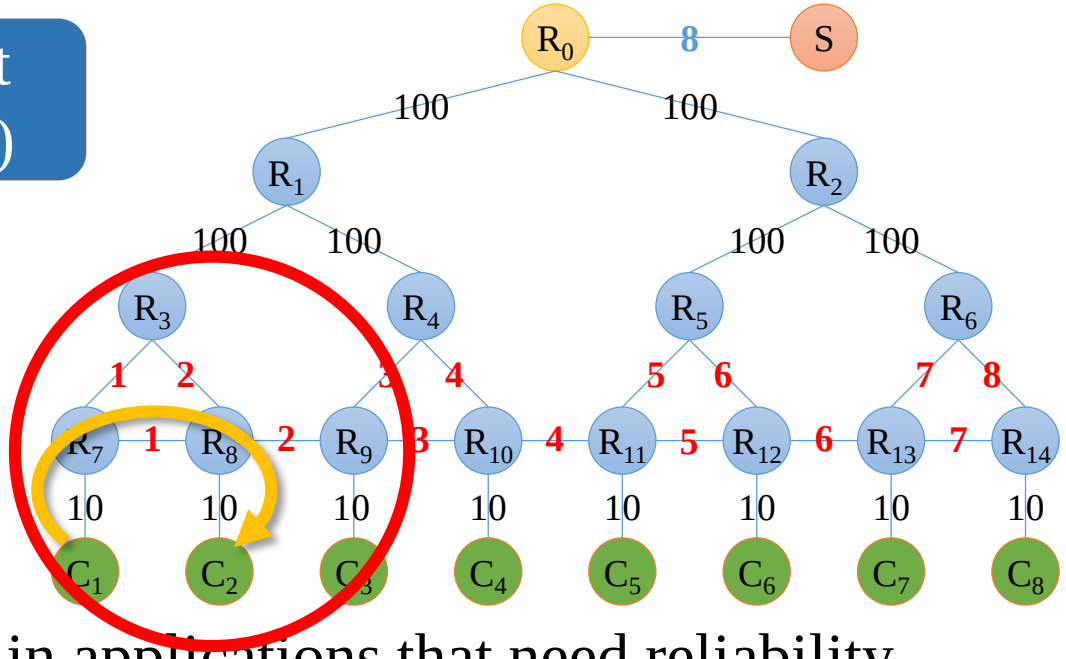


Reliability: How do we get these missing packets ?

SAID – Building Blocks

Fairness: Efficient Multicast Module (using ANP request)

Reliability: Repair Module



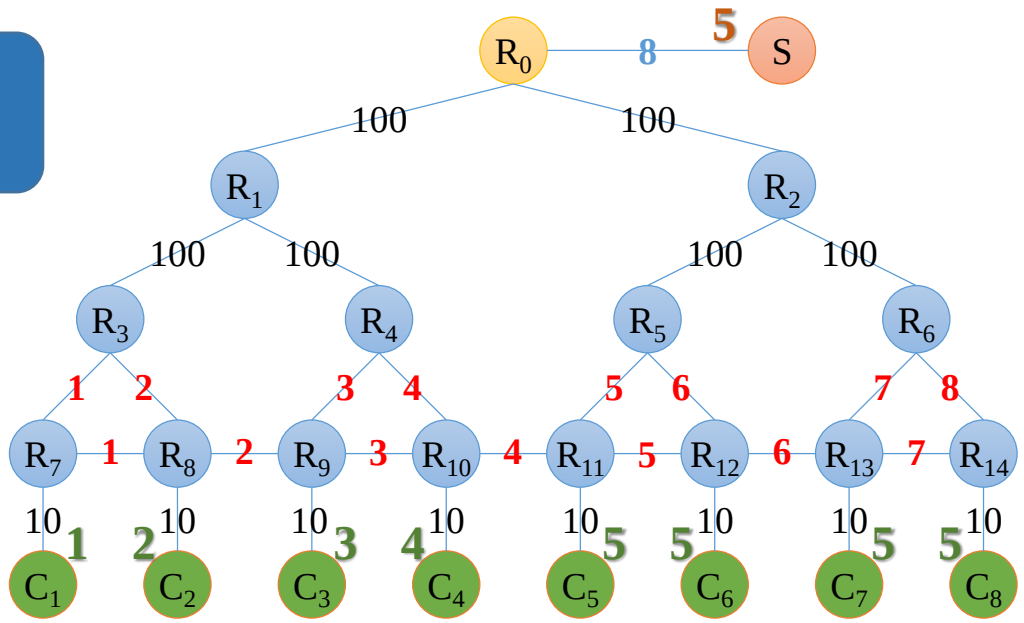
- Request for missing packets in applications that need reliability
- Each consumer can decide when to repair and which packets to repair
  - E.g. Live streaming in order to keep the playout buffer of 1s is full
- Use basic NDN specific sequence request model that can fully utilize the benefits of NDN
  - Preserves privacy and trust
  - Get repair from best sources (including other consumers) via different paths

SAID – Building Blocks

Fairness: Efficient Multicast Module (using ANP request)

Reliability: Repair Module

Sending Rate Module (Optional)



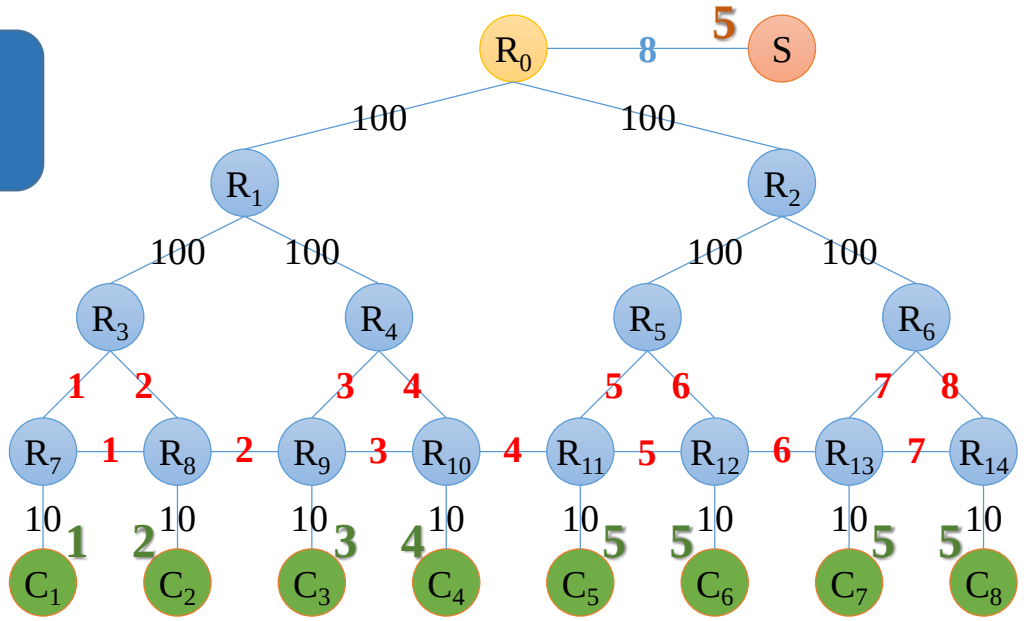
- Sender can send at any rate (upper bounded by the fastest receiver)
  - Align to slowest, fastest, intermediate
  - Constant application suitable rate
- Tradeoff between session completion time (for faster receivers), additional sources & amount of repair traffic

SAID – Building Blocks

Fairness: Efficient Multicast Module (using ANP request)

Reliability: Repair Module

Sending Rate Module



Thank you for your attention!