Reorder Density (RD) and Reorder Buffer-occupancy Density (RBD) : Metrics for packet reordering

Anura P. Jayasumana

Nischal M Piratla, Abhijit Bare, Tarun Banka,

Rick Whitner* & Jerry McCollom*

Computer Networking Research Laboratory

Colorado State University

*Agilent Labs

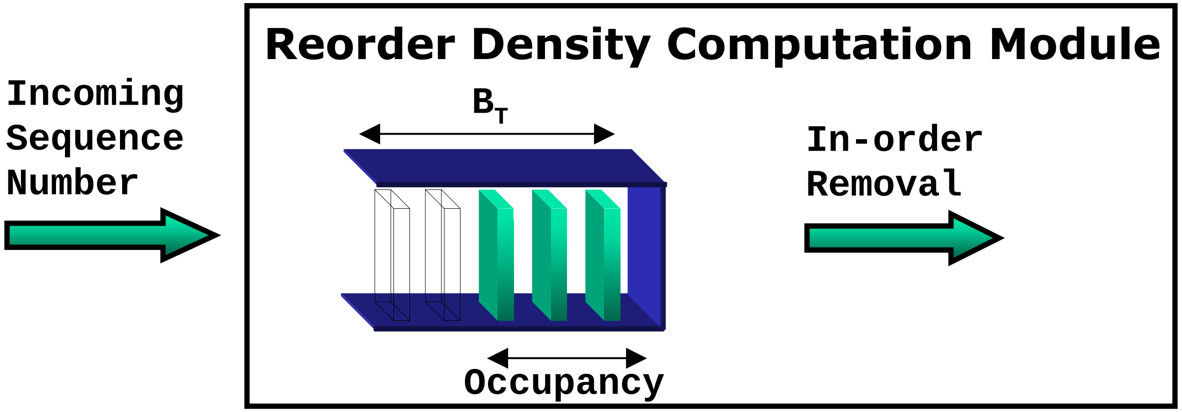
draft-jayasumana-reorder-density-03.txt

IETF Meeting, San Diego, CA August 2, 2003

Revision History

- Reorder Buffer-occupancy Density (RBD)
  - The “Reorder Density” function defined in the previous draft is renamed RBD
  - New version of RBD is orthogonal to loss and duplication
  - First version of RBD useful from application point of view for recovery from reordering
- Reorder Density (RD) – A comprehensive metric for reordering
  - Contains all information in ED (Early Density) and LD (Late Density)
  - Several useful properties including the ability to combine reorder information from subnets
  - Built in recovery features – burst losses, very early packets etc.

Concept of RBD



- If a packet with a sequence number higher than the currently expected packet arrives, it is buffered.
- Packets are removed from the buffer, when they become in-order or when the buffer is full.
- Occupancy of the buffer is recorded after each arrival is processed.
- Size of the buffer (B_T) determines when a packet is considered lost or useless.

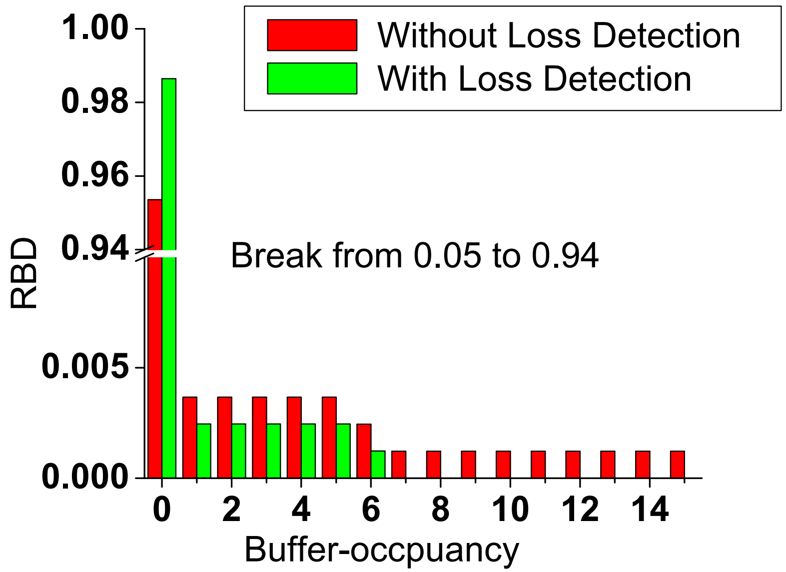
Concept of RBD

- Concern expressed at prior meeting: Dependency of metric on loss
- New version: Modify buffer-occupancies to ignore losses
  - if a packet does not arrive even after the buffer is full (go-back BT), or
  - lag by BT delay detecting losses (stay-back BT).

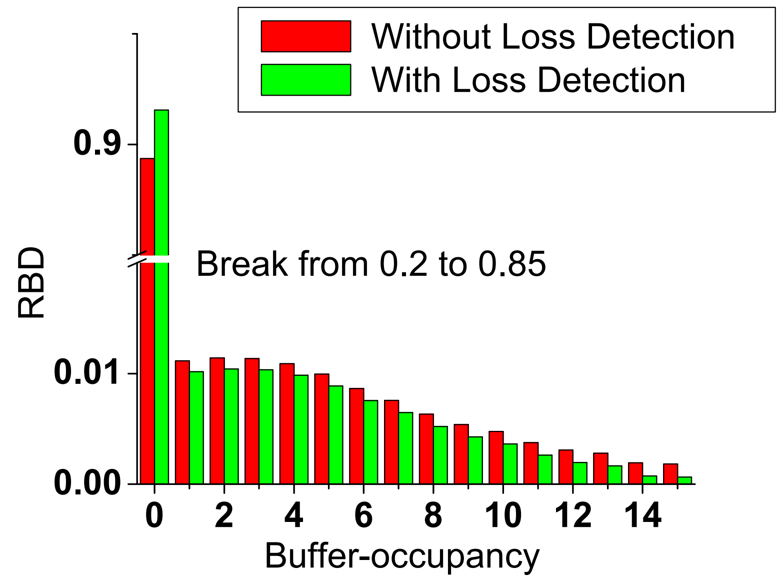
Salient Features of RBD

- New version: Orthogonal to loss and duplication
- Earlier version: Useful for application from the point of recovery at the destination, e.g., estimation of buffer allocation requirements
- Complexity:
  - Space complexity O(BT)
  - Computation complexity O(N)

RBD Measurements: With and Without Loss Detection



From 210.212.x.x to 129.82.x.x



From 202.54.x.x to 129.82.x.x

Packet Reordering

A: 1,2,8,3,4,5,6,7,9,10,11,...

B: 1,3,4,5,6,7,2,8,9,10,11,...

C: 1,2,4,3,5,6,7,8,9,10,11,...

Definition and Concept of RD

Packet Reordering:

m → $(m + d_m)$

Arrival	1	2	4	5	3	7	6
Receive_index	1	2	3	4	5	6	7
Displacement (d_m)	0	0	-1	-1	2	-1	1

Reorder Event $r(m, d_m)$: If the receive_index assigned to packet m is $(m+d_m)$, with $d_m \neq 0$ then a reordered event $r(m, d_m)$ has occurred

Earliness/Lateness: A packet is late if $d_m > 0$, and early if $d_m < 0$

Packet Reordering: Packet reordering is completely represented by the union of reorder events,

$$R = \bigcup_m \{r(m, d_m) | d_m \neq 0\}$$

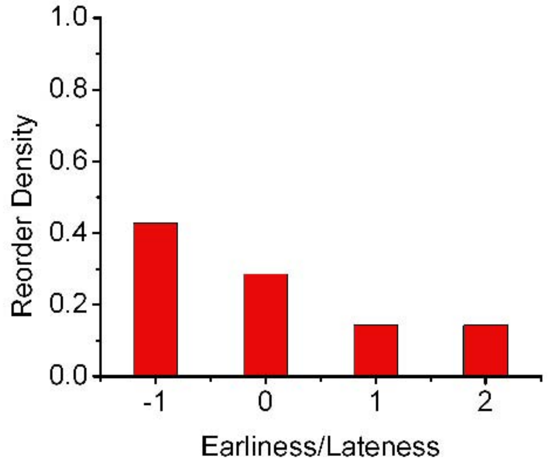
For the above sequence R = {(3, 2), (4, -1), (5, -1), (6, 1), (7, -1)}

Definition and Concept of RD

Ex. 1: No losses/ duplicates

Arrival	1	2	4	5	3	7	6
Receive_index	1	2	3	4	5	6	7
Displacement (d_m)	0	0	-1	-1	2	-1	1

Displacement (D)	-1	0	1	2
Frequency FLE[D]	3	2	1	1
Normalized Frequency RD[D]	3/7	2/7	1/7	1/7



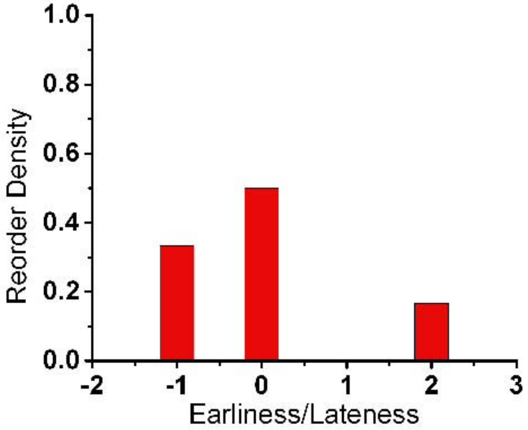
Concept of RD

Ex. 2: With packet duplication

1 is duplicated

Arrival	1	2	4	5	3	1	6
Receive_index	1	2	3	4	5	-	6
Displacement	0	0	-1	-1	2	-	0

Displacement (D)	-1	0	1	2
Frequency FLE[D]	2	3	0	1
Normalized Frequency RD[D]	2/6	3/6	0	1/6



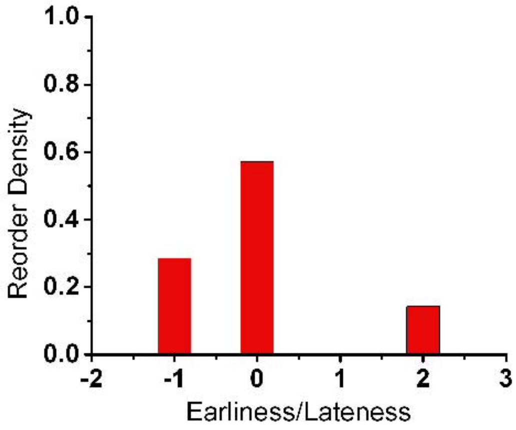
Concept of RD

Ex. 3: With packet loss, DT=2

5 is lost

Arrival	1	3	4	2	6	7	8
Receive_index	1	2	3	4	6	7	8
Displacement	0	-1	-1	2	0	0	0

Displacement (D)	-1	0	1	2
Frequency FLE[D]	2	4	0	1
Normalized Frequency RD[D]	2/7	4/7	0	1/7



RD Definition

Let S[k] be a subset of R s.t

$$S[k] = \{r(m, d_m) | d_m = k\}$$

Then,

$$RD[k] = |S[k]| / N' \text{ for } k \neq 0$$

Where N ‵ is the total non-duplicate packets received and |S[k]| is cardinality of set S[k].

$$RD[0] = 1 - \sum_{k \neq 0} |S[k]|/N'$$

Methodology

- Assignment of receive_index values
  - Do not assign receive_index values for lost packets
  - Discard duplicate arrivals, i.e., no receive_index values for such arrivals

  Result: Orthogonal to loss and duplication

- Detection of lost packets
- Detection of duplicate packets

Methodology - Detecting lost packets

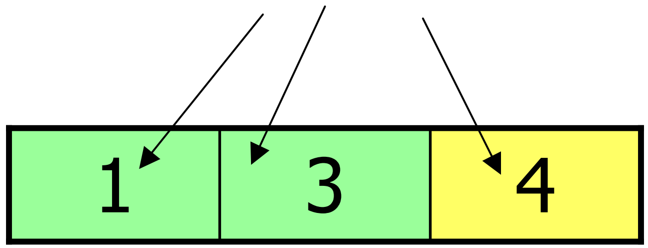
- A packet is lost, if it is not within next DT arrivals, w.r.t where it was in the ordered sequence
- Two detection methods:
  - Go-back DT:
    - RD is computed in real-time
    - For previous DT arrivals, receive_index values are reassigned and RD is computed again, only *if a packet is lost*
  - Stay-back DT:
    - RD is computed with a lag DT arrivals
    - Next DT arrivals are checked for any lost packets before assigning the current receive_index value

Methodology – Detecting duplicates

- A packet is duplicate, if it is among early arrivals
- Due to DT threshold on loss, the number of early arrivals to be recorded is DT for the worst case

RD with Stay-back Procedure

- Ex. 1 (1, 3, 4, 2, 6, 7, 8) with DT = 2

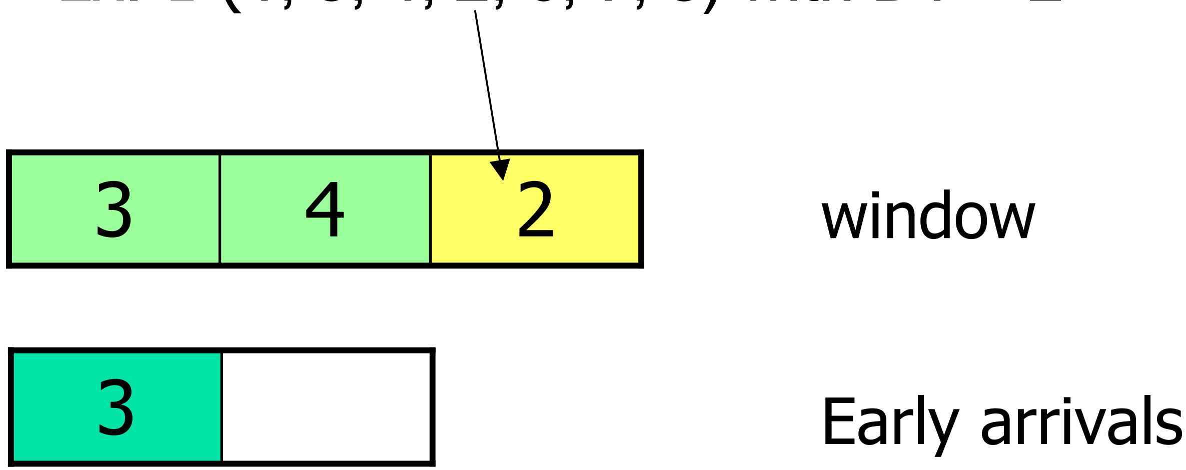


window

Arrival	1	3	4				
Receive_index	1						
Displacement	0						

RD with Stay-back Procedure

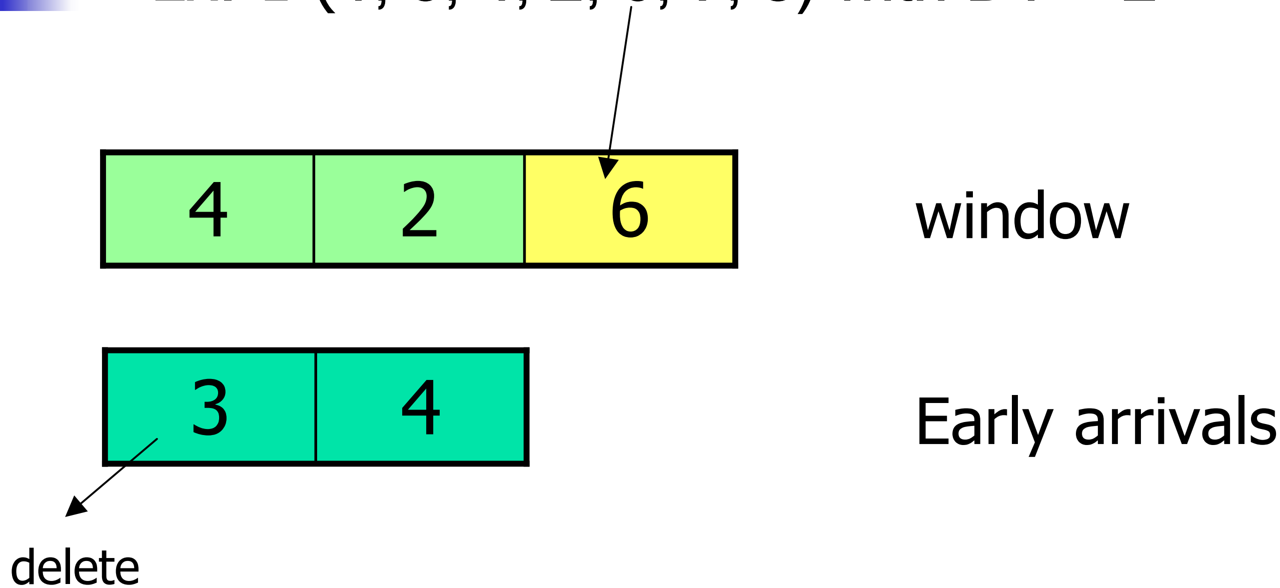
- Ex. 1 (1, 3, 4, 2, 6, 7, 8) with DT = 2



Arrival	1	3	4	2			
Receive_index	1	2					
Displacement	0	-1					

RD with Stay-back Procedure

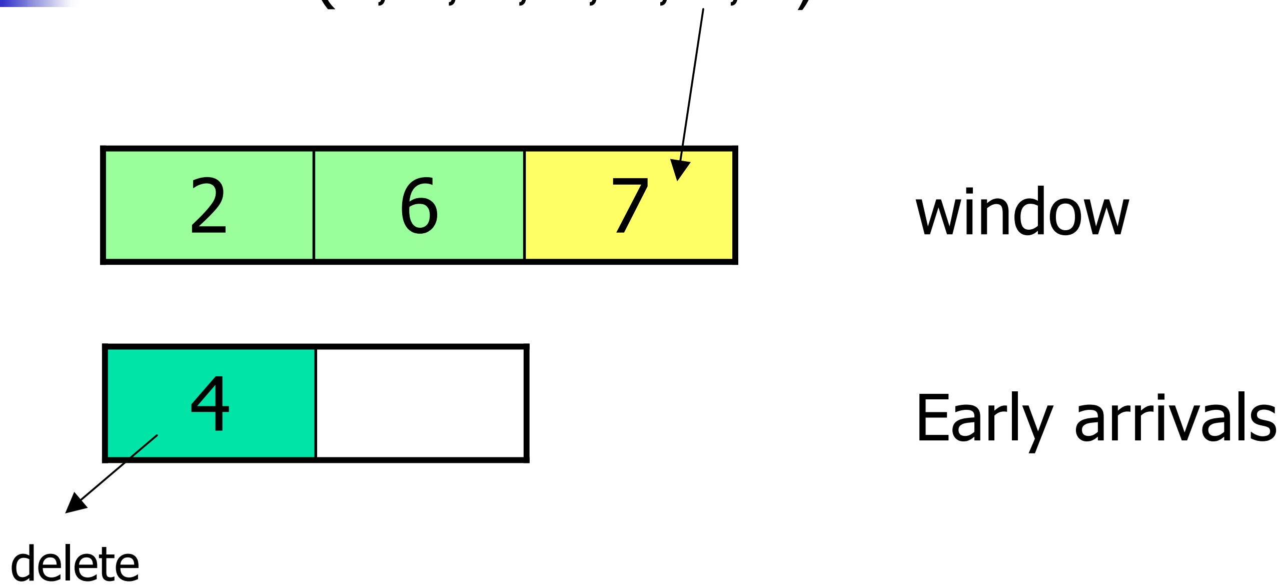
- Ex. 1 (1, 3, 4, 2, 6, 7, 8) with DT = 2



Arrival	1	3	4	2	6		
Receive_index	1	2	3				
Displacement	0	-1	-1				

RD with Stay-back Procedure

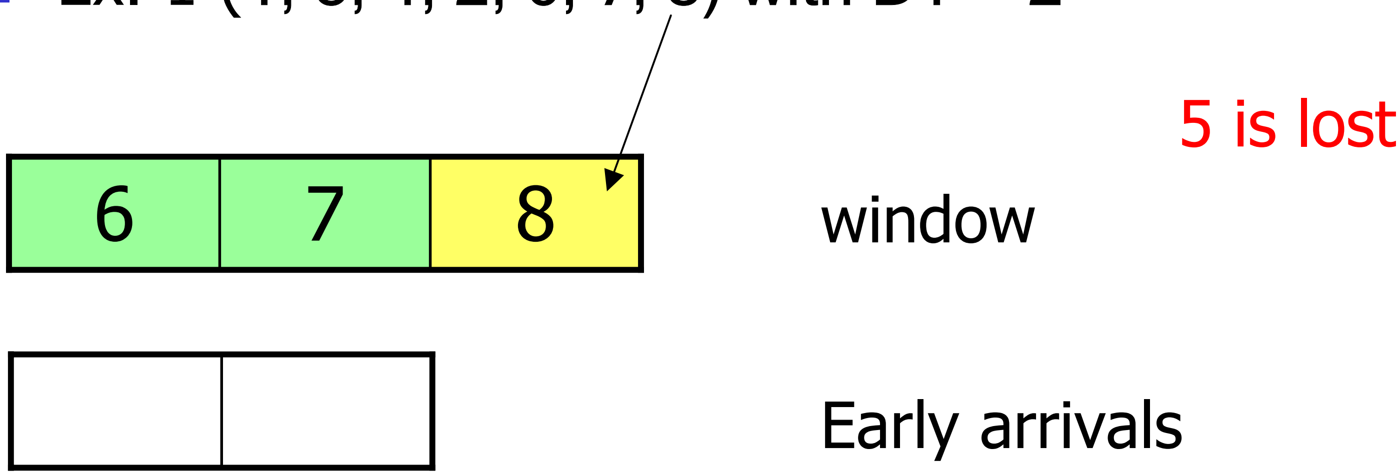
- Ex. 1 (1, 3, 4, 2, 6, 7, 8) with DT = 2



Arrival	1	3	4	2	6	7	
Receive_index	1	2	3	4			
Displacement	0	-1	-1	2			

RD with Stay-back Procedure

- Ex. 1 (1, 3, 4, 2, 6, 7, 8) with DT = 2

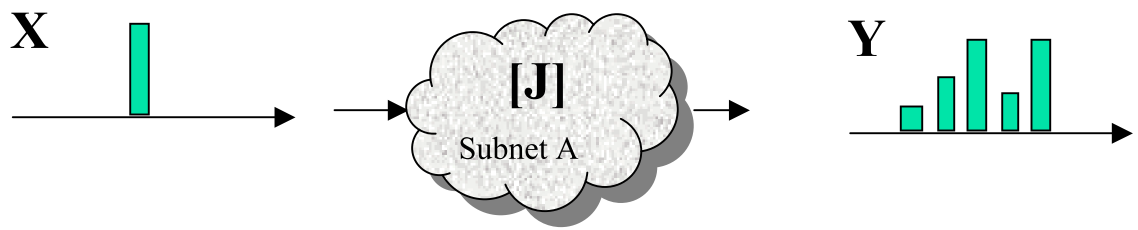


Arrival	1	3	4	2	6	7	8
Receive_index	1	2	3	4	6	7	8
Displacement	0	-1	-1	2	0	0	0

Salient Features of RD

- As length of the sequence being evaluated increases, RD approaches the PDF of the packet displacement
- Orthogonal to loss and duplication
- Robustness – Receive_index self correcting in presence of losses, burst losses, sense of proportionality
- Usefulness: Ex. Can be used for setting TCP parameter "dupthresh" dynamically
- Complexity:
  - Space complexity O(DT)
  - Computation complexity O(N)
- Broader applicability: Reorder Response

Reorder Response



Reorder response of a network [J]:

RD at the output of the subnet for a sequence of in-order packets

- A reorder response exists for a network in which the packet reordering is not based on packet sequence number.
- The reorder response depends on the distribution of inter-packet gaps and the network operating condition
- The reorder density Y at the output of the subnet for an out-of-order input packet sequence with RD X is given by

$$Y = X * J$$

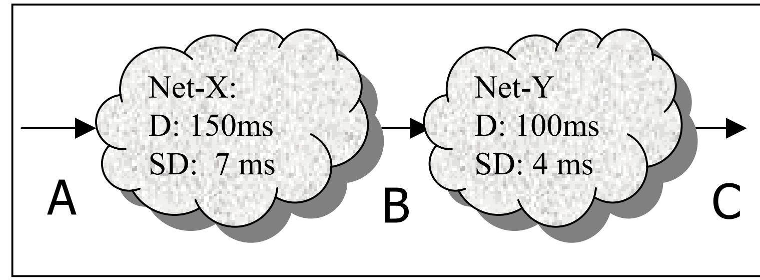
- Ref: Reorder Density – A Metric for Packet Reordering, Piratla, Bare & Jayasumana. See also http://www.cnrl.colostate.edu

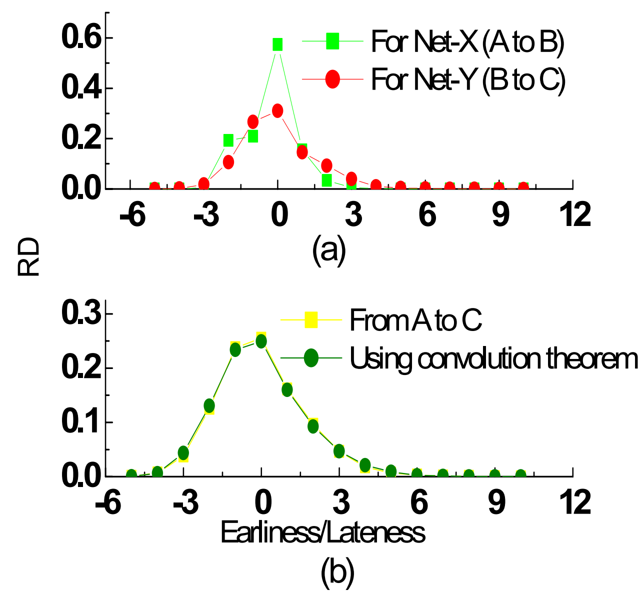
RD and Reorder Response

At steady conditions:

*The reorder response **J**[k] of a network formed by cascading two subnets, with reorder responses **J1**[k] and **J2**[k] respectively, is given by the convolution of **J1**[k] and **J2**[k], i.e.,*

$$J[k] = J1[k] * J2[k].$$





Resources

- Our site http://www.cnrl.colostate.edu
  - Algorithms
  - Perl Scripts
  - Java Applets
  - Papers on Reordering
  - Information on Work in Progress

Future

- Merge two drafts?
- A separate ippm draft ?
- An informational RFC
- Further publications outside IETF