On Reorder Density and its Application to Characterization of Packet Reordering*

Nischal M. Piratla[1,2] Anura P. Jayasumana[1] Tarun Banka[1]
[1]*Computer Networking Research Laboratory, Colorado State University, Fort Collins, CO 80523, USA*
[2]*Deutsche Telekom Laboratories, Ernst-Reuter-Platz 7, D-10587 Berlin, Germany*

Abstract

A formal approach for characterizing, evaluating and modeling packet reordering is presented. Reordering is, a phenomenon that is likely to become increasingly common on Internet, and it can have an adverse impact on end-to-end performance and network resource utilization. Reorder Density (RD) is a comprehensive metric for reordering. Several properties of RD are presented that provide insight into the nature of reordering present in a sequence. Based on measurements of packet reordering, certain basic patterns of reordering that are prevalent on the Internet are identified. By focusing on these basic patterns, a model is developed for reordering in networks involving two parallel paths, using a simple load-balancing scenario as an example. The model is verified using an emulation testbed.

1. Introduction

Out of order arrival of packets is an increasingly common phenomenon over the Internet. The causes for out-of-sequence delivery of packets include the following:

- Packet striping at layer 2 and 3 links [1,2]
- Retransmissions on wireless links [3] and in TCP
- DiffServ scheduling: If a flow exceeds negotiated constraints, the non-conformant packets are either dropped or given a lower priority. In the latter case, the packets will be placed in different queues resulting in out-of-order delivery [4][5]
- Ad hoc routing [6]
- Route fluttering [7]

Increased parallelism in routers and switches required to handle high link speeds and large routing tables, wireless ad hoc routing, and enhancement features such as QoS and overlay routing, are some of the factors that point to increased presence of reordering in future.

Packet reordering, irrespective of its cause, impacts applications based on both TCP and UDP significantly. In the case of TCP, when packets in forward path arrive out of order, the receiver may perceive packets as lost, resulting in a reduced congestion window, and increased retransmissions [1,8,9] that further degrade the performance. Reverse-path reordering, i.e., reordering of acknowledgements, results in the loss of TCP's self-clocking property, leading to bursty transmissions and possibility of increased congestion [1]. Approaches for mitigating the impact of out-of-order packet delivery on TCP performance include adjusting 'tcp_reordering' parameter in Linux, i.e., the number of duplicate ACKs to be allowed before classifying a following non-ACKed packet as lost. However, the non-robust nature of TCP to reordering, i.e., adjustment of tcp_reordering, makes the traffic vulnerable to security attacks [10]. In VoIP for example, a delay sensitive applications based on UDP, an out-of-order packet that arrives after the elapse of playback time is treated as lost thereby decreasing the perceived quality of voice.

Traditionally, recovery from effects of reordering has been left to the application (in case of UDP) and the transport layer (in case of TCP). To recover from reordering, the out-of-sequence packets are buffered until packets can be played back in sequence to the application. Thus an increase in out-of-order delivery by the network consumes more resources at the end-host, and also affects the end-to-end performance of these applications. Consequently, there is an effort to address this issue at intermediate nodes, at IP level. Many existing routers attempt to eliminate the reordering caused by the scheduling schemes within the router itself. This involves either a) identifying the individual streams and forwarding the packets of same stream to the same queue thus preventing reordering, or b) output re-sequencing, i.e., buffering packets at the output of the router to ensure that the packets belonging to the same stream are released in sequence [11]. For example, the network processors from vendors such as IBM, Motorola, Vitesse, TI and Intel, have built-in hardware to track flows [12]. While these approaches reduce the reordering that occurs inside a router, they cannot eliminate reordering due to multiple paths. Furthermore, the complexity of these approaches will increase significantly as the number of parallel flows in a pipe increases (due to the need to keep information on

* This research was supported in part by NSF ITR Grant No. 0121546, Ixia University Partners Program and Agilent Technologies. Nischal Piratla was with Colorado State University during this research; Email: {Nischal.Piratla, Anura.Jayasumana, Tarun.Banka}@colostate.edu

a large number of parallel flows), and as the ratio of packet time to routing latency decreases.

It is shown in [13] that the delay in sequencing the packets back in order, as in output buffering, is proportional to log (C_s/U) where C_s is the capacity of the link and U the packet size. The buffer size required to put the packets back in order also increases dramatically with link speed. Moreover, the number of table entries in routers is growing exponentially [14]. The net result is a higher end-to-end delay to packet time ratio, leading to more load splits and higher packet reordering [15]. Increase in latency required at the intermediate switches and routers to re-sequence the packets add to the end-to-end latency and the round-trip delay, thereby affecting the overall performance as well.

Next generations of networks therefore have to proactively deal with the problem of out-of-sequence packets. However, scant attention has been paid so far towards understanding the nature of reordering that occurs in networks. Information on the nature of reordering that occurs in a given end-to-end path may be used to enhance the ways that the protocols deal with this problem. Measurement and characterization of reordering in packet sequences and models that provide insight into this problem can lead to development of scalable techniques to deal with reordering and perhaps even to develop tools to diagnose network problems.

While it is not difficult to define a measure that varies in some form with packet reordering in a sequence, in order for a metric to be useful, it needs to capture information on the nature of reordering and be useful for measurement, analysis and modeling. Consider the sequence (1, 4, 5, 2, 3, 6, 7), n-reordering lists only packet 2 as reordered, reorder extent shows packet 2 and 3 as reordered [16], and percentage reordering may treat two to four packet as reordered packets depending on the definition. Reorder Buffer-occupancy Density metric [17,18] measures reordering in terms of the occupancy of a buffer required to recover from reordering at the receiver. Reorder Density metric [18,19] captures both lateness and earliness in a sequence and has additional attributes such as low spatial and computational complexity, robustness, orthogonality to loss and duplication, etc. Metric RD is unique in that it measures over individual subnets can be combined to obtain end-to-end reordering in a network [20]. Due to these features and broader applications, RD is chosen to study and characterize reordering in the networks.

This paper provides a mathematical representation for generic packet reordering. Moreover, a set of useful properties for RD are listed that will help in understanding reordering in the networks, and also the probable causes. The generalized expressions for RD for the basic reordering patterns are useful in characterization and modeling of packet reordering in the networks. For example, in load-balancing scenarios, we can statistically estimate the number of each possible basic pattern, and then apply the generalized RD expressions to them. A reorder model for simple load-balancing scenario is developed using this methodology and is verified against an emulated topology.

Section 2 discusses the representation of packet reordering in detail. Internet measurements with observation of basic patterns are presented in Section 3. Section 4 derives the RD for basic patterns. In Section 5, a reorder model for simple load-balancing scenario is developed and is verified against emulated network topology. Conclusions are presented in Section 6.

2. Representation of Packet Reordering

Without loss of generality, consider a sequence of packets (1,2…N) transmitted over a network. A receive_index (1,2…) is assigned to each packet as it arrives at the destination, according to the order in which it is received. Duplicate packets are not assigned a receive_index, and a receive_index values equal to the sequence numbers of lost packets are never used. A packet is treated as lost, if it does not arrive within threshold D_T positions from its expected position, in the absence of reordering. A detail discussion on detection of lost and duplicate packets and selection of D_T is presented in [19].

If the receive_index assigned to packet m is (m+d_m), with $d_m \neq 0$, we say that a reorder event has occurred, and it is denoted by r(m, d_m). In the absence of reordering, the sequence number of the packet and the receive_index value are the same, i.e., $d_m = 0$ for each packet. A packet is late if $d_m > 0$, and early if $d_m < 0$. Thus, packet reordering of a sequence of packets is completely represented by the union of reorder events, R, referred to as the reorder set:

$$R = \bigcup_m \{r(m, d_m) \mid d_m \neq 0\} \quad (1)$$

Table 1. Example of the representation of packet reordering for arrived sequence (1,2,6,4,3,7,3)

Arrived Sequence	1	2	6	4	3	7	3
Receive_index	1	2	3	4	6	7	-
Displacement	0	0	-3	0	3	0	-

If there is no reordering in a packet sequence then R = {ϕ}. Table 1 illustrates the assignment of receive_index values, and the computation of displacements for an arrived sequence at the measurement point. In this example, the packet with sequence number 5 is lost.

Therefore, the receive_index assignment skips this value. Packet with sequence number 3 is duplicated and there is no receive_index value assigned to the second copy of it. The reorder set for this case, R = {(3, 3), (6, -3)}.

Consider a sequence (1, 3, 4, 2, 5, 8, 6, 7), where packet 2 is late by two positions and 8 is early by 2 positions. Due to lateness of 2, both 3 and 4 are early by one position, and similarly due to earliness of 8, both 6 and 7 are late by one position. Thus, for $d_m > 0$ (< 0), the next d_m (previous d_m) packets are displaced by –1 (+1) positions respectively. In this case, the packet with sequence number 'm' is said to be associated with a primary event or p-event, while the affected d_m packets have undergone secondary events (s-events). The p-event r (m, d_m) results in following set S' of s-event(s):

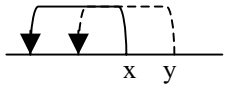
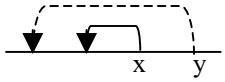
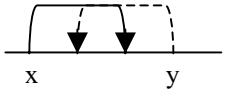
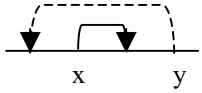
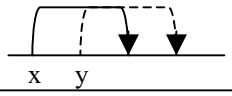
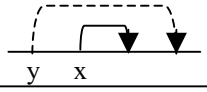
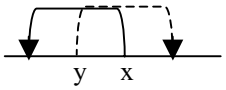
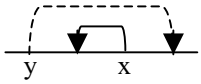
$$S = \{r(m+j, -1), 1 \le j \le d_m\} \quad \text{for } d_m > 0 \quad (2)$$

$$S = \{r(m+j, 1), d_m \le j \le -1\} \quad \text{for } d_m < 0 \quad (3)$$

In general the effects of reordering of packets in a sequence are localized. Therefore, one can envisage partitioning the sequence of packets arriving at the receiver into reordered-segments (RS) such that the reordering is localized within reordered-segments. By definition, reordering event(s) due to packets within RS are not propagated beyond it. If a p-event belongs to a particular RS, then all its associated s-events must belong to the same RS. The definition does not preclude in-order packets before or after the event(s) from belonging to RS. For example, in the sequence (1, 3, 4, 5, 2, 6, 7, 9, 10, 8, 11, 12, 13), the possible reorder-segments are (1, 3, 4, 5, 2), and (6, 7, 9, 10, 8, 11, 12, 13), where p-events associated with packets 2 and 10 contain their s-events in their respective reorder-segments respectively. Also, the sequence itself is a standalone RS. A reordered-segment that cannot be partitioned further into RS' is defined as a minimal-reordered-segment (MRS).

Primary events can inter-twine in different forms. If there is only a single p-event in a MRS then it is called an independent reordering (IR) event. Measurements over the Internet, provided in Section 3, show that most of the MRS' associated with packet streams over Internet contain at most two p-events; an MRS' with more than two p-events is a relatively rare occurrence. In the case of two p-events within an MRS, one event may be embedded in the other, or the two events may overlap each other. We call the former an embedded reordering (ER), and the latter an overlapped reordering (OR). IR, OR and ER together will be referred to as the basic patterns of reordering. In an IR, the p-event is associated with either earliness or lateness. For ER and OR patterns, the p-events that combine may have different signs of displacements. Consider two p-events given by r(x, d_x) and r(y, d_y) in a MRS. There are four possible combinations of OR and ER patterns each, depending on the direction of displacements d_x and d_y. The possible combinations of OR and ER patterns are listed in Table 2. Conventionally, for ER, r(y, d_y) indicates the enclosing p-event, whereas for OR it is the second p-event in MRS. To differentiate between events, a dashed line is used for r(y, d_y).

Table 2. Different combinations of p-events r(x, d_x) and r(y, d_y) in basic overlapped and embedded patterns

Overlapped	Embedded
Earliness overlaps with earliness: $d_x < 0,\ d_y < 0$ 	Earliness encloses earliness: $d_x < 0,\ d_y < 0$ 
Earliness overlaps with lateness: $d_x > 0,\ d_y < 0$ 	Earliness encloses lateness: $d_x > 0,\ d_y < 0$ 
Lateness overlaps with lateness: $d_x > 0,\ d_y > 0$ 	Lateness encloses lateness: $d_x > 0,\ d_y > 0$ 
Lateness overlaps with earliness: $d_x < 0,\ d_y > 0$ 	Lateness encloses earliness: $d_x < 0,\ d_y > 0$ 

RD is defined as the discrete density of the frequency of packets with respect to their displacements, i.e., the lateness and earliness from the original position. Let $S[k]$ denote the set of reorder events in R with displacement equal to k, i.e.,

$$S[k] = \{r(m, d_m) \in R \mid d_m = k\} \quad (4)$$

Thus, RD[k] is defined as $|S[k]|$ normalized with respect to the total number of received packets (N'). Note that N' does not include duplicates or lost packets. RD[0] corresponds to the packets for which receive_index is same as the sequence number. Thus,

$$RD[k] = |S[k]|/N' \quad \text{for } k \neq 0 \quad (5)$$

$$RD[0] = 1 - \sum_{k \neq 0} |S[k]|/N' \quad (6)$$

RD corresponding to a sequence of in-order packets is a unit impulse (at $k = 0$). Consider sending this sequence of packets through a network. RD of the sequence observed

at the output corresponds to the reordering introduced by the network, and hence we call it the reorder response ($\boldsymbol{J}[k]$) of the network. In fact, the reorder response of a subnet is a function of many factors including the background or cross traffic in the network, and statistical characteristics of the inter-packet gap. We hypothesize however that a reorder response exists for a network operating under stationary conditions and a given inter-packet gap distribution, provided the reordering introduced by the network is not dependent on the sequence number of the packet. We expect that a theory developed for reordering in networks under such stationary conditions will lead the way towards a more general solution in future. The reorder response $\mathbf{J}[k]$ of a network formed by cascading two subnets, with reorder responses $\mathbf{J_1}[k]$ and $\mathbf{J_2}[k]$ respectively, is given by the convolution of $\mathbf{J_1}[k]$ and $\mathbf{J_2}[k]$, i.e., $\mathbf{J}[k] = \mathbf{J_1}[k] * \mathbf{J_2}[k]$. This property is proved and extensively discussed in [19]. Other important properties of packet reordering and RD include but not limited to:

1. When lost and duplicate packets are detected, and receive_index is assigned as specified, $\sum d_i = 0$.
2. Reorder Density (RD) satisfies $\sum_k (k * \mathrm{RD}[k]) = 0$. This follows directly from property 1.
3. If packet reordering introduced by a network consists only of IR associated with early arrival of the packets, in the absence of losses, then RD satisfies the following conditions:
$$\mathrm{RD}[k] = 0 \text{ for } k > 1 \quad (7)$$
4. If packet reordering introduced by a network consists only of IR associated with late arrival of the packets, in the absence of losses, then RD satisfies the following conditions:
$$\mathrm{RD}[k] = 0 \text{ for } k < -1 \quad (8)$$
5. If packet reordering introduced by a network consists only of IR with late and/or early arrival of the packets, in the absence of losses, then RD satisfies the following condition:
$$\mathrm{RD}[-1] - \sum_{k>0} (k*\mathrm{RD}[k]) = \mathrm{RD}[1] - \sum_{k<0} (|k|*\mathrm{RD}[k]) \geq 0 \quad (9)$$
6. Consider a sequence of 'N' packets partitioned into 's' RS', S_1, S_2,..S_s with lengths n_1, n_2, ..n_s respectively. RD of a sequence is the weighted average of RDs of individual RSs', i.e.,
$$\mathrm{RD}[k] = \sum_{i=1}^{s} \left(\frac{n_i}{N} * \mathrm{RD}_i[k]\right) \quad (10)$$
where $\mathrm{RD}_i[k]$ is the RD for i^{th} RS. Note, sum of n_i = N. The proofs of properties are available in [19, 21].

3. Internet Measurements

Reordering observed on packet streams transferred over the Internet are presented and analyzed next. The script used for these measurements and other tools for reorder measurement can be found at [21]. Multiple files were downloaded to a host on subnet 129.82.x.x (in Colorado, USA) from different sites around the world listed in Table 3. To ensure that the connections were not short-lived, only sites with files larger than 2MB were chosen. IP2Location software, and ping provided the geographical location, the one-way delay value (D), and approximate standard deviation of the delay (SD).

Table 3. Servers for reorder measurements, with mean one-way delay (D) and standard deviation of delay (SD) in milliseconds (ms)

Net	*Subnet*	*Location*	*D*	*SD*
1	192.35.x.x	Armenia	75.5	1.5
2	209.211.x.x	MA, USA	45.5	0.81
3	62.94.x.x	Lazio, Italy	81.9	0.19
4	130.195.x.x	Wellington, NZ	95.4	0.17
5	202.54.x.x	India	145	3

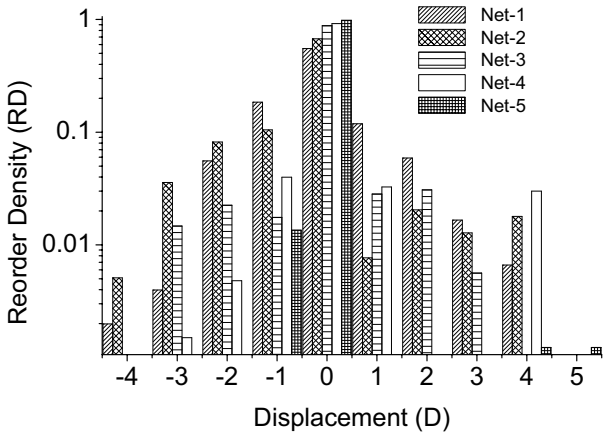


Figure 1. RD based on measurements for Net-1, 2, 3, 4 and 5. (With D_T = 10)

Out of multiple sets of measurements [20], only five are presented here, corresponding to a range of end-to-end delays. TCP based applications HTTP and FTP were used for servers in Armenia, India, Italy and USA. From host on subnet 130.195.x.x, a media file was played that used RTP/RTCP. Using 'tcpdump' to collect the information of

received data, the TCP and RTP sequence numbers were mapped to (1, 2,..). Fig. 1 shows the observed RD. The characteristics that can be derived from RDs include the following:

1) Net-5 has minimum reordering as more than 90% of the packets had zero displacement.
2) Net-1 and Net-2 have approximately one in every three packets with non-zero displacement.
3) Net-1 server is farthest (geographically) and Net-2 is the nearest from the receiver. However, standard deviation of delay (SD) of Net-1 is less than that of Net-2. A higher SD value, rather than higher mean delay, appears to cause more reordering in the network.
4) RD for Net-5 has only four components, i.e., RD[0], RD[-1] and RD[4] and RD[5]. In this case, recovery from reordering is feasible with a buffer size of five packets, at the receiver. Also, the number of retransmissions triggered by 3 duplicate ACKs in TCP is equal to sum of RD[4] and RD[5] components [20].

Table 4 lists the percentage of all MRS patterns corresponding to the basic patterns, IR, OR and ER defined in Section 2. The events that do not fit these categories are lumped under 'Other' category. These results indicate that the basic reorder patterns account for 88% to 100% of the MRS'. Thus, RD for basic pattern MRS' can be used to characterize most of the reordering over the Internet. It was further observed that a majority of p-events are lateness-based events [20].

Table 4. Percentage of basic reorder patterns (two p-events) over the Internet

Net	*IR%*	*OR%*	*ER%*	*Other*
1	64	24	0	12
2	45	38	17	0
3	86	12	1	1
4	84	2	4	10
5	100	0	0	0

4. Reorder Density for Basic Patterns

In this section, the expressions of RD for basic patterns are derived. As the steps for different overlapping and embedded types are very similar, we will derive RD for IR, a case each for OR and ER. Complete derivations for all basic patterns are present in [20].

If the single p-event of IR pattern is $r(m, d_m)$ and the length of a RS with this single p-event is n_i, then using Eqs. (2) and (3), RD for IR is given as:

$$\left\{\begin{array}{l} RD[d_m] = 1/n_i, RD[-sgn(d_m)] = |d_m|/n_i \\ \text{where } sgn(d_m) = +1 \text{ if } d_m > 0 \text{ and } = -1 \text{ if } d_m < 0 \end{array}\right\} \quad (11)$$

Any value of RD[k], $k \neq 0$, that is not explicitly specified is zero. RD[0] is always given by expression in Eq.(6).

For ER and OR, the p-events can combine in different ways depending on the sign of displacements as discussed earlier. In case of OR, we define a related quantity called overlap length V_{xy}. Overlap length indicates the number of sequence numbers shared by the s-event sets, if the two p-events were to occur independently. For example, consider the sender's side sequence (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) resulting in received sequence (1, 2, 4, 6, 3, 7, 8, 5). If only r(3, 3) occurs then the s-event set is {(4, -1), (5, -1), (6, -1)}. Similarly, if only r(5, 4) occurs then the s-event set is {(6, -1), (7, -1), (8, -1), (9, -1)}. The common sequence number in s-event sets is just '6', for this case. Therefore, $V_{xy} = 1$.

In OR with lateness overlapping lateness, displacements of overlap region packets are affected by both p-events. Since both the p-events are late, each packet in overlap region V_{xy} has a displacement of –2.. All the remaining packets associated with s-events of first and second p-events have a displacement of -1. Therefore, RD for a RS of size n_i, with only single OR of type lateness overlaps with lateness is given by:

$$\left\{\begin{array}{l} \text{Initialize RD}[k] = 0 \text{ for all } k, \text{ then} \\ RD[-2] \mathrel{+}= V_{xy}/n_i \\ RD[-1] \mathrel{+}= (d_x + d_y - 2*V_{xy})/n_i \\ RD[d_x] \mathrel{+}= 1/n_i; RD[d_y] \mathrel{+}= 1/n_i \end{array}\right\} \quad (12)$$

Note, the notation '+=' is increment syntax as in C/C++.

In ER with lateness embedded in lateness the event $r(y, d_y)$ encloses $r(x, d_x)$. Due to the displacement of packet 'y', there are d_y packets are early by one position. Since packet 'x' is one of them, for it to be effectively be displaced by d_x packets, it has to be actually displaced by $(d_x + 1)$ positions. Thus the inner event involves $(d_x + 2)$ packets. There are $(d_x + 1)$ packets effectively with minus two displacement each, and $(d_y - d_x - 2)$ packets early by one position. Therefore, RD for a RS of size n_i, with only single ER of type lateness enclosing lateness, and no other patterns is given as:

$$\left\{\begin{array}{l} \text{Initialize RD}[k] = 0 \text{ for all } k, \text{ then} \\ RD[-1] \mathrel{+}= (d_y - d_x - 2)/n_i \\ RD[-2] \mathrel{+}= (d_x + 1)/n_i \\ RD[d_x] \mathrel{+}= 1/n_i; RD[d_y] \mathrel{+}= 1/n_i \end{array}\right\} \quad (13)$$

Similarly, RD can be derived for the remaining possible basic patterns. After the initialization, i.e., RD[k] = 0 for all k, the resulting expressions for all basic patterns are listed in Table 5.

Table 5. RD expressions for all the permutations of two primary events for OR and ER

OR	ER
Earliness with earliness: d_x <0, d_y <0	
RD[1] += $(-d_x - d_y - 2V_{xy})/n_i$ RD[2] += V_{xy} / n_i RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$	RD[1] += $(d_x - d_y - 2)/ n_i$ RD[2] += $(-d_x + 1)/ n_i$ RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$
Earliness with lateness: d_x >0, d_y <0	
RD[-1] += $(d_x - 1 - V_{xy})/ n_i$ RD[1] += $(-d_y - 1 - V_{xy})/ n_i$ RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$	RD[1] += $(-d_x - d_y)/ n_i$ RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$
Lateness with lateness: d_x >0, d_y >0	
RD[-2] += V_{xy} / n_i RD[-1] += $(d_x + d_y - 2V_{xy})/n_i$ RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$	RD[-2] += $(d_x + 1)/ n_i$ RD[-1] += $(d_y - d_x - 2)/n_i$ RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$
Lateness with earliness: d_x <0, d_y >0	
RD[-1] += $(d_y - V_{xy} - 1)/ n_i$ RD[1] += $(-d_x - V_{xy} - 1)/ n_i$ RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$	RD[-1] += $(d_x + d_y)/ n_i$ RD[d_x] += $1/ n_i$ RD[d_y] += $1/ n_i$

5. Reorder Model for Load Balancing

A methodology for deriving the RD for more general scenarios, based on RD for basic patterns in Section 4, is presented next. Although, we consider a simple load-balancing scenario as an example, the goal is to illustrate that the properties derived in Section 4 can be used to analyze reordering due to different network scenarios.

Consider the case where a certain number of packets in a flow undergo significantly higher delay due to packets being routed via a longer path Y instead of path of choice X. Such a condition could occur, e.g., as an outcome of load balancing at a router. Let the difference between the average delays of the paths 'X' and 'Y' be such that the packets traversing via path Y arrive at the destination after 'Δ' of its subsequent packets that traverse via path X. Fig. 2 illustrates the displacement of packet with sequence number 11, where Δ = 3. Packet with sequence number 11 arrives after packets with sequence numbers 12, 13 and 14. In networks, the value of 'Δ' may also vary depending on the variance of delays in paths X and Y (due to internal parallelism). But for the development of a reordering model with load balancing scenario, 'Δ' is considered to be constant.

To derive a RD for a packet stream received over the network with the above-described behavior, the number of IR, OR and/or ER patterns in the received sequence is estimated, and then the RD expressions provided in Section 4 for the basic patterns are used to compute the effective RBD.

5.1. Basic Pattern Count

Let the probability that a packet takes path Y be 'P' and path X be (1-P). Consider an arbitrary packet 'i', then

$$\Pr\{i^{th} \text{ 'pkt. displaced by } \Delta\} = P \quad (14)$$

Consider a sequence of size 'N,' where N >> Δ. For a constant 'Δ', the resultant multiple reordering types are restricted to IR and OR. Given that packet 'i' takes path Y, it is involved in an IR event if none of its previous previous 'Δ-1' packets and 'Δ' packets within the reorder event get displaced. Therefore,

$$\Pr\{\text{IR with } i^{th} \text{ pkt.}/\ i^{th} \text{ pkt. takes Y}\} = (1-P)^{2\Delta-1} \quad (15)$$

Note, even if packet 'i-Δ' is displaced, it is still an IR, due to the displacement of the i^{th} packet. The probability of such an occurrence given packet 'i' and 'i-Δ' are displaced is given by:

$$\Pr\{\text{IR with } (i-\Delta)^{th}\ pkt./\ i^{th}\ pkt. \text{ takes Y}\} = P(1-P)^{2\Delta-2} \quad (16)$$

However, these IR events have 'Δ-1' displacement associated with primary events. Thus, the number of IR events with 'Δ' displacement in the sequence using Eq. (14)and (15), is approximated as:

$$\text{IR\_Count\_with\_}\Delta \cong NP(1-P)^{2\Delta-1} \quad (17)$$

and the number of IR events with 'Δ-1' displacement in the sequence, using Eq. (14) and (16), is approximated as:

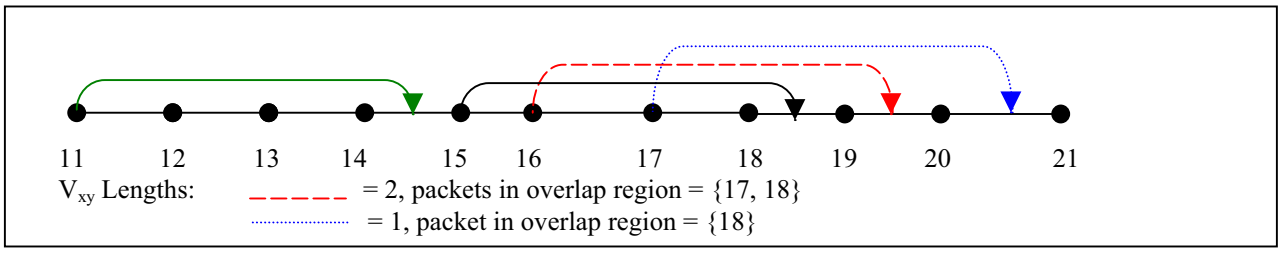


Figure 2. Displacement of packets illustrating independent reorder event with packet 11 and feasible overlapping reorder events, given the displacement of packet 15

$$\text{IR\_Count\_with\_}\Delta\text{-}1 \cong NP^2(1-P)^{2\Delta-2} \quad (18)$$

As all primary events correspond to late packets passing through Y, in the case of OR patterns, only "lateness overlapping lateness" type interferences are feasible. However for a given 'Δ', the overlap length 'V_{xy}' between two p-events has a range of [1, Δ-1]. For example, as illustrated in Fig. 2, given the sequence (11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21), if 15 is displaced by three units, then 16 is displaced by three units or 17 is displaced by three units, it results in an OR event with 'V_{xy}' equal to two or one respectively. Note, the effective displacement of packet 15 is two whereas the following primary event packet (16 or 17) is three.

Let us consider the case where only 15 and 17 are displaced forming an OR event. In this scenario, if any of the packets 13, 14 (previous Δ-1 packets) or 16, 18 (inner Δ-1 packets) or 19, 20 (Δ -V_{xy} packets) get displaced, and then a cluster of more than two overlapping events is formed. Note, packet 20 may be displaced as in the case discussed in IR events. However, the probability of such occurrences is small, and we neglect it. Therefore, for a basic OR pattern with overlap V_{xy}, there is a set of '3Δ - 2 - V_{xy}' packets that should not be associated with primary events. The probability of the occurrence of an OR event, given packet 'i' is displaced, is:

$$\Pr.\{\text{OR with } V_{xy} \text{ associated with } i^{th} \text{ pkt.} / \; i^{th} \text{ pkt. displaced}\} = (1-P)^{3\Delta-2-V_{xy}} P \quad (19)$$

Thus, using Eq.(19), the average number of OR patterns for a large received sequence of size N, is approximated as:

$$\text{OR\_Count} \cong \sum_{V_{xy}=1}^{\Delta-1} N(1-P)^{3\Delta-2-V_{xy}} P^2 \quad (20)$$

As P increases, the number of complex reorder patterns involving more than two primary events (both overlapped and embedded) increases. As observed in Table 4, the overlap patterns with more than two p-events are not common over the Internet.

5.2. RD Model

Given a sequence of size N, the counts of reordering are given by Eq. (17), (18), and (20). Applying expressions for RD to each IR and OR patterns derived in Section 4, the resulting RD for the load-balancing scenario can be evaluated using the following assignments:

For IR of size 'Δ', using Eq. (17),

$$\left\{\begin{array}{l}\text{Initialize RD}[k] = 0 \text{ for all } k, \text{ then} \\ \text{RD}[\Delta] \mathrel{+}= NP(1-P)^{2\Delta-1} \\ \text{RD}[-1] \mathrel{+}= N\Delta P(1-P)^{2\Delta-1}\end{array}\right\} \quad (21)$$

and for IR of size 'Δ-1', using Eq.(18),

$$\left\{\begin{array}{l}\text{Initialize RD}[k] = 0 \text{ for all } k, \text{ then} \\ \text{RD}[\Delta\text{-}1] \mathrel{+}= NP^2(1-P)^{2\Delta-2} \\ \text{RD}[-1] \mathrel{+}= (\Delta-1)NP^2(1-P)^{2\Delta-2}\end{array}\right\} \quad (22)$$

In the case of OR, using Eq. (20), initialize RD[k] = 0 for all k, then do the following assignments:

$$\left\{\begin{array}{l}\text{for } V_{xy}=1 \text{ to } '\Delta-1' \{ \\ \quad \text{RD}[\Delta] \mathrel{+}= N(1-P)^{3\Delta-2-V_{xy}} P^2 \\ \quad \text{RD}[\Delta\text{-}1] \mathrel{+}= N(1-P)^{3\Delta-2-V_{xy}} P^2 \\ \quad \text{RD}[-1] \mathrel{+}= 2N(\Delta - V_{xy})(1-P)^{3\Delta-2-V_{xy}} P^2 \\ \quad \text{RD}[-2] \mathrel{+}= NV_{xy}(1-P)^{3\Delta-2-V_{xy}} P^2 \quad \}\end{array}\right\} \quad (23)$$

5.3. Emulation and Verification

To verify the model for RD derived in previous section, the network topology shown in Fig. 3, using NISTNet is emulated. The average one-way delay of paths X and Y were D_x = 28 ms and D_Y = 80 ms respectively. The inter-packet gap at the source was set to 10 ms. From these values, it is apparent that the packet in path Y may get displaced by approximately five positions with respect to those taking path X, i.e., Δ = 5 was used for the expressions in the model. The standard deviation (SD) values for delays on these paths were set to 5% of the average delays of the respective paths, i.e., 1.4 ms and 4.0 ms that are maintained less than the inter-packet gap. As uniform distributions are used for delay, there were displacements of lengths that were greater and less than Δ = 5. However, the model was able to approximate the measurements even with such deviations.

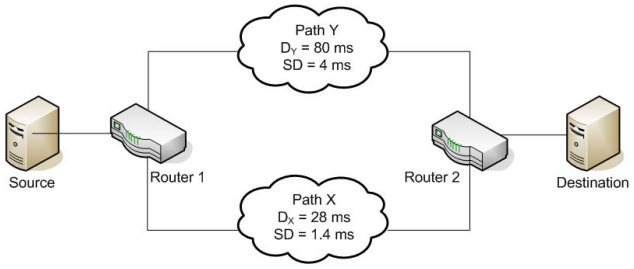


Figure 3. Emulated network topology using NISTNet to attain to alternate paths X and Y

A stream of 50,000 UDP packets was sent through the network, in which a packet takes an alternate route Y with probability of P, (= 0.01, 0.03, 0.05 and 0.07), to emulate varied scenarios.

Fig. 4 illustrates RD from the derived model using Eqs. (21)-(23) and from measurements using emulation that computed RD on-the-fly. The root mean-square error (RMSE) values for P = 0.01, 0.03, 0.05, and 0.07 between

model and emulation RDs are 8.83E-4, 0.0040, 0.0096 and 0.022 respectively. It can be observed that the RD from model follows the RD obtained from emulation for low values of *P*, validating the assumption during the derivation of the model about the magnitude of *P*. However, as *P* increases beyond 0.05, the number of multiple overlap events with more than two p-events becomes significant. By considering three p-event-overlap cases during the analysis, the methodology for the model could be extended to follow the RD from emulated measurements, even for higher values of *P*.

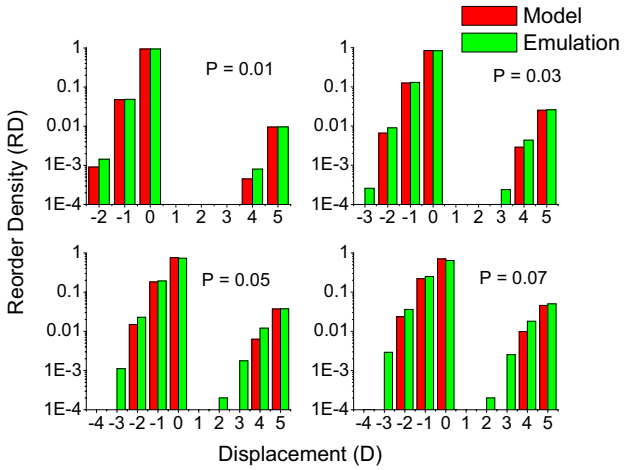


Figure 4. Comparison of RD derived from model and emulations for *P* = 0.01, 0.03, 0.05 and 0.07

6. Conclusions

A formal representation for packet reordering was presented. The representation leads to a metric, reorder density, which captures the nature and amount of reordering. RD is unique among the metrics for reordering in that it can combine reordering in subnets to obtain end-to-end reordering. This feature allows the usage of RD for complex networks. Furthermore, this paper demonstrated the feasibility of modeling reordering caused due to load balancing in terms of RD. At present, we are looking at the queuing schemes within routers and switches and their impact on reordering, and the reorder response of networks.

References

[1] J. C. R. Bennett, C. Partridge and N. Shectman, "Packet Reordering is Not Pathological Network Behavior," *Trans. Networking IEEE/ACM*, Dec. 1999, pp.789-798.

[2] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose and D. Towsley, "Measurement and Classification of Out-of-sequence Packets in Tier-1 IP Backbone," *Proc. IEEE INFOCOM*, Mar. 2003, pp. 1199- 1209.

[3] J. Bellardo and S. Savage, "Measuring Packet Reordering," *Proc. IMW'02,* Nov. 2002, pp. 97-105.

[4] S. Bohacek, J. Hespanha, J. Lee, C. Lim and K. Obraczka, "TCP-PR: TCP for Persistent Packet Reordering," *Proc. IEEE 23rd ICDCS*, May 2003, pp. 222-231.

[5] G. Iannaccone, S. Jaiswal and C. Diot, "Packet Reordering Inside the Sprint Backbone," *Tech. Report, TR01-ATL-062917,* Sprint ATL, Jun. 2001.

[6] Z. Fu, B. Greenstein, X. Meng and S. Lu, "Design and Implementation of a TCP-friendly Transport Protocol for Ad Hoc Wireless Networks," *Proc. IEEE ICNP*, Oct. 2002, pp. 216-225.

[7] V. Paxson, "Measurements and Analysis of End-to-End Internet Dynamics," *Ph.D. dissertation*, Computer Science Department, UC Berkeley, 1997.

[8] E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering", *ACM Computer Comm. Review*, 32(1), Jan. 2002, pp. 20-30.

[9] M. Laor and L. Gendel, "The Effect of Packet Reordering in a Backbone Link on Application Throughput," *IEEE Network,* Sep./Oct. 2002, pp. 28-36.

[10] I. Aad, J-P. Hubaux, E. W. Knightly, "Denial of Service resilience in ad hoc networks," *Proc. MobiCom'04*, Sep./Oct. 2004, pp. 202-215.

[11] H. Liu, "A Trace Driven Study of Packet Level Parallelism," *Proc. IEEE ICC*, NY, 2002, pp. 2191-2195.

[12] e2e forum: http://www.postel.org/pipermail/end2end-interest/2004-August/004233.html.

[13] Y. Xia and D. Tse, "Analysis on Packet Resequencing for Reliable Network Protocols," *Proc. IEEE INFOCOM,* Mar. 2003, pp. 990-1000.

[14] M. Ruiz-Sanchez, E. W. Biersack and W. Dabbous, "Survey and Taxonomy of IP Address Lookup Algorithms," *IEEE Network,* Mar./Apr. 2001, pp. 8-23.

[15] A. A. Bare, "Measurement and Analysis of Packet Reordering using Reorder Density," *MS Thesis*, Dept. of Computer Science, Colorado State University, 2004.

[16] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov and J. Perser, "Packet Reordering Metric for IPPM," IETF draft (work in progress).

[17] T. Banka, A. A. Bare and A. P. Jayasumana, "Metrics for Degree of Reordering in Packet Sequences," *Proc. IEEE LCN,* Nov. 2002, pp. 333-342.

[18] A. P. Jayasumana, N. M. Piratla, A. A. Bare, T. Banka, R. Whitner and J. McCollom, "Reorder Density and Reorder Buffer-occupancy Density - Metrics for Packet Reordering Measurements," IETF draft (work in progress). http://www.cnrl.colostate.edu/packet_reorder.html

[19] N. M. Piratla, A. P. Jayasumana and A. A. Bare, "RD: A Formal, Comprehensive Metric for Packet Reordering," *IFIP Networking Conference 2005*, LNCS 3462, pp. 78-89.

[20] N. M. Piratla, "A Theoretical Foundation, Metrics and Modeling of Packet Reordering and Methodology of Delay Modeling using Inter-Packet Gaps," *Ph.D. Dissertation*, Dept. of Electrical and Computer Engineering, Colorado State University, 2005.

[21] Algorithms, Java Applets and Perl Script for RD. http://www.cnrl.colostate.edu/packet_reorder.html.