Reordering of Packets due to Multipath Forwarding – An Analysis*

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

Abstract – Increased parallelism in routers necessary to handle high link speeds and large routing tables, wireless ad hoc routing, QoS provisioning, and overlay routing, are some of the factors that lead to an increase in reordering on the Internet. Packet reordering due to packet forwarding over multiple paths is investigated. An analytical model is derived for load splitting scenarios and verified using emulated topologies. The resulting reordering is profiled using reorder density, and analyzed with respect to path delays, path probabilities and number of paths. The variation of packet displacement with delay variation and forwarding probabilities is quantified. The special case corresponding to two paths is evaluated in detail. For any load splitting, the increase in the difference in the delay between paths leads to increased reordering, making the paths with closer delay values more preferable. The model can also be applied to a single-path case where reordering is caused by wide delay variation among packets, by deriving an equivalent set of probabilities corresponding to path splitting scenario.

Keywords: *Packet reordering, Internet measurements*

I. INTRODUCTION

Packet reordering is an increasingly common phenomenon over the Internet [1, 2, 3]. A major cause of packet reordering is the parallelism within nodes, attributed to packet stripping in layer 2 and layer 3 switches [1, 4], QoS scheduling [5, 6], etc. Here, a later arriving packet may, for example, be placed in a shorter queue leading to its departure prior to an earlier packet that was placed in a longer queue. A second cause for reordering is the parallelism among links or paths taken by a stream of packets, attributed to load-splitting [7], route fluttering [8], ad hoc routing [9], etc. An earlier packet may take a path with longer delay resulting in its arrival after a later packet that has suffered a shorter delay. Packet reordering may also be caused by retransmissions, for example, layer 2 wireless [10] or even TCP.

Irrespective of its cause, packet reordering is a significant impediment to the performance of both TCP and UDP based applications [1, 11, 12]. 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-acknowledged packet as lost. However, the non-robust nature of TCP to reordering, i.e., adjustment of tcp_reordering parameter, makes the traffic vulnerable to security attacks [13]. In addition, there are ongoing efforts to improve the robustness of TCP to such non-congestion events, reordering being an example [14].

Although router designs can be modified to reduce the amount of reordering due to the internal parallelism, the proposed solutions are not scalable with increase of link speeds. The techniques used in routers to reduce reordering due to parallelism within routers include (a) input sorting, i.e., 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 [15]. While these approaches reduce the reordering that occurs inside a router, they cannot eliminate reordering due to multiple paths or due to causes such as retransmissions. Furthermore, the complexity of these approaches increase significantly as the number of parallel flows in a pipe increases (due to the need to keep information on a large number of parallel flows), and as the ratio of packet time to routing latency decreases. In [16], it is shown that the delay in re-sequencing the packets back in order, with output re-sequencing, 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 is also shown to increase dramatically with link speed. Moreover, the number of table entries in routers has been growing rapidly over time [17], a trend that will continue into the future. The net result of all these trends is a higher end-to-end delay to packet time ratio, leading to more load splits and higher packet reordering [18]. Increase in latency required at the intermediate switches and routers to re-sequence packets add to the end-to-end latency and the round-trip delay, thereby affecting performance as well.

Scant attention has been paid so far towards understanding the nature of reordering, and its cause and effect relationships. Recent work of the IP Performance Metrics Group (IPPM) of IETF has been aimed at the development of metrics to capture the nature and extent of packet reordering, and consistent measurement techniques. These metrics are reviewed and compared in [19]. A simple metric such as percentage reordering is vague and uninformative, while Reorder Density [19, 20, 21], can capture the nature and extent of reordering in a sequence in a comprehensive and a useful manner. RD is used in [22] to characterize reordering when no more than two packet displacements intertwine at a time. In this paper, we investigate the reordering introduced to a stream of packets as a result of packets being forwarded along different paths. To the best of our knowledge, this is the first paper that models analytically the relationship between one of the causes of reordering and the nature of resulting packet reordering. The

---

* This research was supported in part by NSF ITR Grant No. 0121546, Ixia University Partners Program, and Agilent Technologies.

model is verified using emulated network configurations taking two-path load splitting and five-path load splitting as examples. Reordering is further analyzed with respect to the average delays in parallel paths and probabilities of forwarding associated with different links during load splitting. Although the analysis considers load splitting among links as the cause for reordering, results are also directly applicable to internal parallelism within routers, in which queuing disciplines replace load-balancing schemes.

The remainder of the paper is organized as follows: Section II provides a brief introduction to RD. The proposed packet-reordering model for multi-path load splitting is presented in Section III. The model is verified using emulations and the nature of reordering is further analyzed using two-path and five-path load splitting as examples in Section IV. Section V concludes the paper.

II. REORDER DENSITY

This section reviews Reorder Density (RD), a metric for representing reordering in a sequence of packets. Comprehensive discussions of this metric, and its unique features, including its usage in cascade of networks can be found in [20, 21, 22].

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. Lost and duplicate packets are not assigned any receive_index values. If the receive_index assigned to packet 'm' is $(\mathrm{m}+d_m)$, then d_m is the displacement of the packet 'm'. In the absence of reordering, the sequence number of packets and the receive_index values are equal, i.e., $d_m = 0$. A packet is late if $d_m > 0$, and early if $d_m < 0$. Reorder density RD is an array with RD[k] equal to the fraction of packets with displacement k, where k takes both negative and positive values, with $k < 0$ corresponding to early packets, $k > 0$ corresponding to late packets and $k = 0$ corresponding to packets that arrive at the correct position. A example of RD computation is illustrated in Table I.

TABLE I
ASSIGNMENT OF RECEIVE_INDEX VALUES AND COMPUTATION OF DISPLACEMENTS FOR ARRIVAL SEQUENCE (2, 1, 3, 7, 5, 6, 8, 2)

Arrived Sequence	2	1	3	7	5	6	8	2
Receive_index	1	2	3	5	6	7	8	-
Displacement	-1	1	0	-2	1	1	0	-

III. REORDER MODEL FOR LOAD-SPLITTING

Packet reordering frequently occurs due to parallelism within the nodes or among the links, as explained in Section I. Distributing communication activity across multiple outgoing links when there is a large amount of traffic is a commonly available router configuration option. A similar distribution of the packets also occurs within routers, where packets are sent to different queues/processors for router table lookup and forwarding. The former causes the packets of same stream to take different outgoing links and follow different paths in the network to reach the destination. These paths could differ in latencies, thus the packets may arrive out-of-order at the destination. In this section, we derive a model for packet reordering due to such multi-path forwarding, also referred to as load splitting among multiple links.

To explain the terminology used in modeling, consider the two-path load splitting example illustrated in Fig. 1. The packets of stream generated at source enter Router 1, and go through either path X or path Y to reach Router 2, which is linked to the destination. In the example, the average delay of path Y is 80 ms and path X is 28 ms. Therefore, if one of the packets from the stream takes path Y, then it is delayed on average by additional 52 ms compared to all other packets that take path X. If the inter-packet gap in the packet stream at source is equal to 10 ms, then the packet going through path Y arrives on average after five of its successive packets in source stream, thus causing packet reordering. Therefore, for each path we can associate a displacement Δ that a packet suffers going through it. For primary preferred path (namely, X), Δ is set to zero, and in the above example Δ = 5 for alternate path.

Now, consider the case with 'n' alternate paths in addition to the preferred primary path from Router 1 to the destination. Let $P_n, P_{n\text{-}1},\ldots, P_2, P_1$ correspond to the path probabilities with which packets leaving Router 1 choose the respective paths. These paths have associated Δ values of $n, n\text{-}1, ..., 2, 1$ respectively. Note that some of the path probabilities may be zero. Thus P_0, the probability with which a packet takes the preferred primary path, is given by $P_0 = 1 - P_1 - P_2 - \ldots - P_n$. The effective displacement d_m of each packet at the destination depends on the combination and interference of displacements of its neighboring packets.

Due to its simplicity, again consider the example of two-path load splitting to understand the impact of neighboring packets on the effective displacement of packets at the destination. Fig. 2 shows the displacement of an arbitrary packet 'i' that takes the alternate path (as path Y in Fig. 1). In this example, a non-zero Δ value (5 in the example) is associated with alternate path. Thus, packet 'i' is Δ places away from the original position.

The packets that affect the displacement of packet 'i' are '$i+1$', '$i+2$'… till '$i+\Delta$'. If one of these packets also takes the alternate path then the effective displacement of 'i' is 'Δ-1', and if two of them take path Y then the displacement is 'Δ-2' and so on. The packets subsequent to '$i+\Delta$' have no effect, as

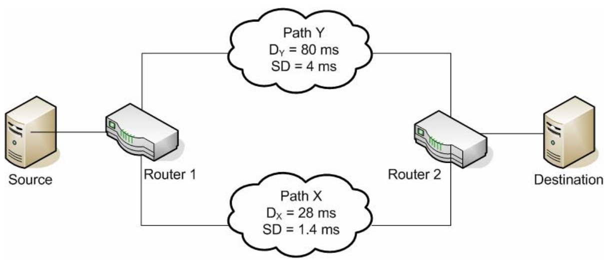


Fig. 1. Example topology for two-path load splitting

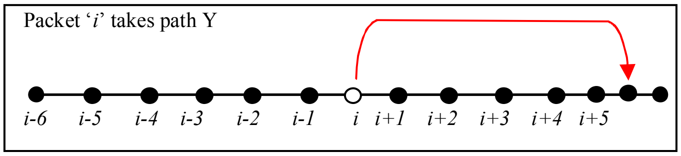


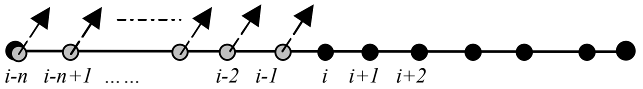
Fig. 2. Displacement of packet ‘*i*’ due to additional delay in path Y traversed by it

their displacements do not change the displacement of ‘*i*’. Similarly, packets before ‘*i*’ do not affect the displacement of ‘*i*’ either. On the other hand, if packet ‘*i*’ is one that has taken the primary path, the packets that affect the displacement of packet ‘*i*’ are ‘*i-1*’, ‘*i-2*’… till ‘i-Δ’ Thus, the displacement of any packet ‘*i*’ is affected by its neighboring packets in range ‘i-Δ’ to ‘$i+\Delta$’. The distribution of displacements of this arbitrary packet ‘*i*’ gives RD of the stream at the destination.

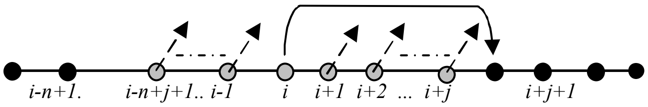
With multi-path load splitting, each alternate path is associated with a different Δ value. Let us use the notation described for multi-path and consider the displacement of an arbitrary packet ‘*i*’. If this packet ‘*i*’ is one that has taken the primary path, then packets before ‘i- n’ will not affect its displacement, as n is the maximum displacement a packet suffers due to alternate paths. Similarly, packets after ‘*i*’ will not affect the displacement of ‘*i*’ as the least displacement that they can have is $\Delta = 0$. It is illustrated in Fig. 3(a).

If packet ‘*i*’ gets displaced by ’*j*’, i.e., it takes the path associated with path probability P_j, then the packets with sequence numbers less than ‘*i-n+j+1*’ and greater than ‘*i+j*’ will not affect its effective displacement as depicted in Fig. 3(b). Packets with sequence numbers less than ‘*i-n+j+1*’, i.e., ‘*i-n+j*’*,* ‘*i-n+j-1*’, may get displaced by a maximum of n, but fail to enclose packet ‘*i*’s new position. Similarly, packets with sequence numbers greater than ‘*i+j*’, i.e., ‘*i+j*’, ‘*i+j+2*’, etc., may get displaced by a minimum of zero units, yet cannot enclose the new position of packet ‘*i*’. However, there could be cases where another packet along with packet ‘*i*’ could land in the gap between ‘*i+j*’ and ‘*i+j+1*’; the probability of such cases occurring is assumed negligible for these modeling scenarios.

To model the packet reordering for the multi-path scenario, we first derive the probability associated with the effective displacement ‘*L*’ of this arbitrary packet ‘*i*’. Initially, we consider the case where ‘*i*’ takes the shortest path, followed by the case in which ‘*i*’ takes an alternate path. Recall, we use the term ‘effective displacement’ to refer to the difference between the packet sequence number and its receive_index. An interference probability is associated with the packets in an arrival sequence. It is the probability that an arriving packet affects the displacement of the packet under consideration. If a packet ‘*i-m*’ does not affect the effective displacement of packet under consideration ‘*i*’, then the interference probability, $P_{mi} = 0$. Note the first subscript corresponds to the earliness of the interfering packet with respect to the one under consideration.



(a) Packet ‘*i*’ takes the shortest path. The packets that may affect the effective displacement of ‘*i*’ are shown using arrows



(b) Packet ‘*i*’ takes j^{th} path. The packets that may affect the effective displacement of ‘*i*’ are shown using arrows

Fig.3. Effective displacement of packet ‘*i*’, given (a) ‘*i*’ takes preferred path, (b) ‘*i*’ takes j^{th} path

Consider the case in which packet ‘*i*’ takes the preferred path. The probability of ‘*i-m*’ interfering with the effective displacement of ‘*i*’, i.e., it moving by more than ‘m’, is:

$$P_{mi} = (P_m + P_{m+1} + ... + P_n) = \left\{ \begin{array}{ll} \sum_{k=m}^{n} P_k & \text{for } 0 < m \le n \\ 0 \text{ otherwise} & \end{array} \right\} \qquad (1)$$

Therefore, the probability that none of the packets interfere with effective displacement of ‘*i*’, given ‘*i*’ is not displaced:

$$P\{L = 0 \,|\, i \text{ takes preferred path}\} = \prod_{l=1}^{n} (1 - P_{li}) = C_0 \qquad (2)$$

Consider the case where only packet ‘*i-m*’ belonging to [*i-n, i-1*] interferes with the displacement of packet ‘*i*’, then

$$\begin{aligned} &P\{L = -1 \,|\, (i \text{ on shortest path, only pkt.'} i - m \text{'interferes})\} \\ &= P_{mi} \prod_{l=1, l \ne m}^{n} (1 - P_{li}) = \left(\prod_{l=1}^{n} (1 - P_{li}) \right) \frac{P_{mi}}{(1 - P_{mi})} = C_0 \frac{P_{mi}}{(1 - P_{mi})} \end{aligned} \qquad (3)$$

Thus,

$$P\{L = -1 \,|\, i \text{ takes preferred path}\} = C_0 \sum_{m=1}^{n} \frac{P_{mi}}{(1 - P_{mi})} \qquad (4)$$

Similarly, let packets *i-m₁* and *i-m₂* be the only two packets belonging to the range [*i-n, i-1*] interfering with packet ‘*i*’. These could be first and second packets preceding ‘*i*’, first and third packets, first and fourth so on, or second and third preceding ‘*i*’, second and fourth, etc. Considering all the possible combinations of m_1 and m_2 values, with $m_1 < m_2$, and transforming the index variables, following expression is obtained for the probability that packet ‘*i*’ gets displaced by ‘–2’ units:

$$\begin{aligned} &P\{L = -2 \,|\, i \text{ takes preferred path}\} \\ &= C_0 \sum_{m_1=1}^{n-1} \sum_{m_2=m_1+1}^{n} \frac{P_{m_1 i} P_{m_2 i}}{(1 - P_{m_1 i})(1 - P_{m_2 i})} \end{aligned} \qquad (5)$$

Generalizing this expression for displacement of packet ‘*i*’ equal to ‘*–k*’*, i.e.,* ‘*k > 0*’ packets interfering, we have:

$$\begin{aligned} &P\{L = -k \,|\, i \text{ takes preferred path}\} \\ &= C_0 \sum_{m_1=1}^{n-k+1} \sum_{m_2=m_1+1}^{n-k+2} \cdots \sum_{m_k=m_{k-1}+1}^{n} \frac{P_{m_1 i} P_{m_2 i} \ldots P_{m_k i}}{(1 - P_{m_1 i})(1 - P_{m_2 i}) \ldots .(1 - P_{m_k i})} \end{aligned} \qquad (6)$$

Now, let's consider the case where packet '*i*' takes the j^{th} alternate path (0< *j* =<n). For this case, as illustrated in Fig. 3(b), packets from '*i-n+j+1*' to '*i+j*' may affect the effective displacement of packet '*i*'. The interference probabilities for these packets are given as:

$$P_{mi} = (P_{m+j+1} + P_{m+j+2} + \ldots + P_n)$$
$$= \left\{ \begin{array}{l} \sum\limits_{k=j+m+1}^{n} P_k \quad \text{for } -j \le m \le n-j, m \ne 0 \\ \quad 0 \text{ otherwise} \end{array} \right\} \qquad (7)$$

If none of the packets interfere with '*i*', then packet '*i*' will have a displacement of '*j*' units:

$$P\{L = j \mid i \text{ takes } j^{th} \text{ path}\} = \prod_{l=-j}^{n-j-1} (1 - P_{li}) = C_j \qquad (8)$$

Note $P_{0i} = 0$, by definition. As in (3), we can show that

$$P\{L = j-1 \mid (i \text{ takes } j^{th} \text{ path}, \text{pkt.}'m'\text{interferes})\} = C_j \frac{P_{mi}}{(1-P_{mi})}$$

$$P\{L = j-1 \mid i \text{ takes } j^{th} \text{ path}\} = C_j \sum_{m=-j}^{n-j-1} \frac{P_{mi}}{(1-P_{mi})} \qquad (9)$$

$$P\{L = j-2 \mid i \text{ takes } j^{th} \text{ path}\}$$
$$= C_j \sum_{m_1=-j}^{n-j-2} \sum_{m_2=m_1+1}^{n-j-1} \frac{P_{m_1 i} P_{m_2 i}}{(1-P_{m_1 i})(1-P_{m_2 i})} \qquad (10)$$

Extending it for only '*k > 0*' packets interfering with packet *i'*,

$$P\{L = j-k \mid i \text{ takes } j^{th} \text{ path}\}$$
$$= C_j \sum_{m_1=1}^{n-j-k} \sum_{m_2=m_1+1}^{n-j-k+1} \cdots \sum_{m_k=m_{k-1}+1}^{n} \frac{P_{m_1 i} P_{m_2 i} \ldots P_{m_k i}}{(1-P_{m_1 i})(1-P_{m_2 i})\ldots.(1-P_{m_k i})} \qquad (11)$$

Note, the same effective displacement of '*i*' could result with '*i*' taking different paths. For example, when packet '*i*' takes the preferred path then the interference of one of its preceding packets from the range [*i-n, i-1*] leads to displacement of '-1' for packet '*i*'. But if '*i*' takes the alternate path with $\Delta = 2$, and three of the packets from range [*i-n+3, i+2*] interfere with its displacement, then packet '*i*' can still have an effective displacement of '-1'. Accounting for all such occurrences, and using (2) to (11), RD[*k*] for any '*k*' can be computed as follows:

$$\text{RD}[k] = \left\{ \begin{array}{l} (P\{L = k \mid i \text{ takes preferred path}\}) * P_0 \\ +P\{L = k \mid i \text{ takes } 1^{\text{st}} \text{ alternate path}\} * P_1 \\ \qquad \vdots \\ +P\{L = k \mid i \text{ takes } n^{\text{th}} \text{ alternate path}\} * P_n \end{array} \right\}$$

resulting in,

$$\text{RD}[k] = \sum_{j=0}^{j=n} \left\{ (P\{L = k / i \text{ takes } j^{\text{th}} \text{ path}\}) * P_j \right\} \qquad (12)$$

The case where the packets are routed through two paths, depicted in Fig. 1 is of significant interest. Therefore, we simplify the model for this case as follows.

Let *P* be the probability of a packet taking the alternate path, and let the offset of packets taking this path be Δ. In this case, $P_j = 0$ for '*j*' not equal to Δ or 0, $P_\Delta = P$, and $P_0 = 1-P$. Thus, Eqn. (12) reduces to:

$$\text{RD}[k] = \left\{ \begin{array}{ll} (1-P)\binom{\Delta}{|k|} P^{|k|} (1-P)^{\Delta-|k|} & \mathit{for}\ -\Delta \le k < 0 \\ P^{\Delta+1} + (1-P)^{\Delta+1} & \mathit{for}\ k = 0 \\ P\binom{\Delta}{k} P^{\Delta-k} (1-P)^{k} & \mathit{for}\ 0 < k \le \Delta_k \end{array} \right\} \qquad (13)$$

It can be observed that RD for both positive and negative displacements is binomial distribution. Thus, these components are symmetrical around *P* = 0.5, and also RDs for *P* = x and *P* = 1-x, where 0<x<1, are mirror images about origin for two-path load splitting case.

IV. VERIFICATION AND ANALYSIS

The model for RD developed in Section III is verified using an emulated network topology for two-path and five path load splitting scenarios. Fig. 1 shows the two-path case. The nature of packet reordering is further analyzed for two-path load splitting to derive a more intuitive understanding.

NISTNet emulator was used for probabilistically routing packets to networks X and Y, as well as for introducing path delays. A stream of 50,000 packets with a constant gap of 10 ms between the packets was sent through this emulated network and reordering was measured at the destination.

The average one-way delays of the paths are D_X and D_Y. The value of D_X was set to 28 ms, and D_Y was varied to obtain various Δ values for the alternate path. The value of Δ for path Y depends on the difference between D_X and D_Y values and the inter-packet gap value for the stream of packets. When D_Y is 80 ms, for example as shown in Fig. 1, Δ=5 for path Y as shown earlier. The standard deviations (SD) of these delay values were also varied for different sets of measurements to evaluate the sensitivity of the model to delay variation.

Fig. 4 depicts the RDs obtained from Eqn.(13) and from the emulation testbed. Loss of packets was not introduced in the emulations, as RD evaluation is insensitive to the losses. SD value for both paths was set to 0.5% of the mean delay. Fig. 4 indicates very close agreement between the measured RD and the prediction based on the model. The distributions of RDs for *P* = 0.1 and 0.9, and *P* = 0.3 and 0.7, are mirror images around displacement equal to zero. As *P* increases from 0.1 to 0.5, the number of packets taking the alternate path increases leading to more non-zero displacements. However, after *P* = 0.5, the number of packets taking the alternate path is higher, making it the preferred path. Thus, path X has a Δ = -5 and path Y has a Δ = 0.

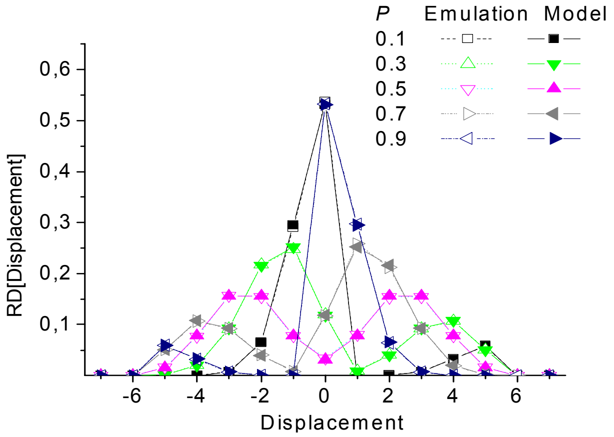


Fig. 4. RD from emulation Vs. RD from model for different *P*

In Fig. 5, we illustrate the impact of the delay of alternate path on the reorder profile by varying the delay of the alternate path, yet keeping the probability of taking the alternate path constant (P = 0.3). As the displacement that a packet suffers in an alternate path, Δ, increases the effective spread of the RD distribution also increases. There is good agreement between the model and the measured values. Increasing the delay of the alternate path leads to more packets with non-zero displacements. The values of RD[k] for k < 0 correspond to early packets. To recover from reordering, these packets need to be buffered till the arrival of the next in-sequence packet. This observation also confirms that to counter reordering, it is advisable to choose a next path that has closer latencies to that of the preferred path.

In the emulation results above, the distribution of delay for path Y had a standard deviation of 0.5% of mean delay. The two-path model presented in Section III assumes constant Δ. To evaluate the effectiveness of the model when this assumption does not hold, we vary the value of SD to 0.5, 1, 2, 5, 10 and 15% of mean delays on both paths. Recall that the average delay values were chosen such that the effective Δ was computed to be five. As SD was increased beyond 5%, i.e., 1.4 ms and 4.0 ms for D_X and D_Y respectively, RD had components beyond displacement 5. This change can be attributed to the delay deviations becoming significant with respect to the inter-packet gap, which was set to 10 ms. The RMS error between RD from model and RD from emulation shown in Fig. 6 for SD values equal to 0.5%, 1%, 2%, 5%, 10% and 15% are 0.0028, 0.0038, 0.0054, 0.0152, 0.015, and 0.015 respectively. Even though the analysis did not take the delay variations into account, these results indicate the validity of the model for such cases too with a maximum mean square error of approximately 1.5%.

Though the components of RD are very small for values beyond 'Δ', as illustrated in Fig. 6 (note Log scale for Y axis), further increase in SD value led to greater spread in the distribution. However, such a variation can be accommodated in the model by taking multiple Δ values with respect to path Y, which can be computed by sampling the distribution of delay values.

Next we evaluate the reordering caused by a node that routes packets to the destination via five different paths. The delay value of primary path is kept at 28 ms, and the delays of alternate paths were set to 50, 80, 100 and 130 ms respectively. These delays correspond to 2, 5, 7 and 10 positions of displacement for the four alternative paths. An SD value of 0.5% of its mean delay was used for each of the links. The packets take the corresponding paths with probabilities P_0= 0.5, P_2= 0.25, P_5= 0.125, P_7= 0.0625 and P_{10}= 0.0625 respectively. The decreasing probabilities were associated with increasing delay, as we feel that it is logical to choose a longer path only if a shorter path is not available during load-splitting.

Fig. 7 illustrates the comparison between the RD derived from the model and that at the destination of the emulated topology. The model fits with RD from emulation results. The RMS error between the two curves is only 0.0034. In the case of five-path load splitting, the earliness and lateness are not binomial distributions as they result from different Δ values associated with five different paths. Since the longer paths had

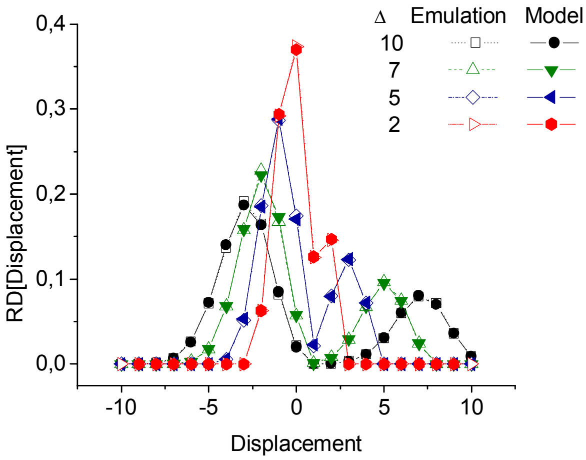


Fig. 5. Variation of RD with Δ, for fixed P = 0.3 for alternate path

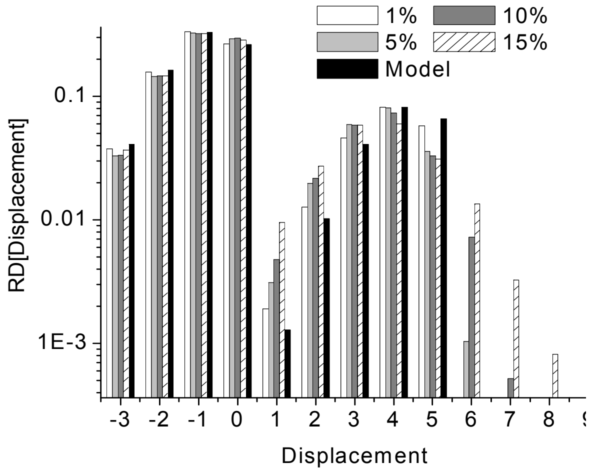


Fig. 6. Impact of Standard deviation (SD) in delay D_Y on RD

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.

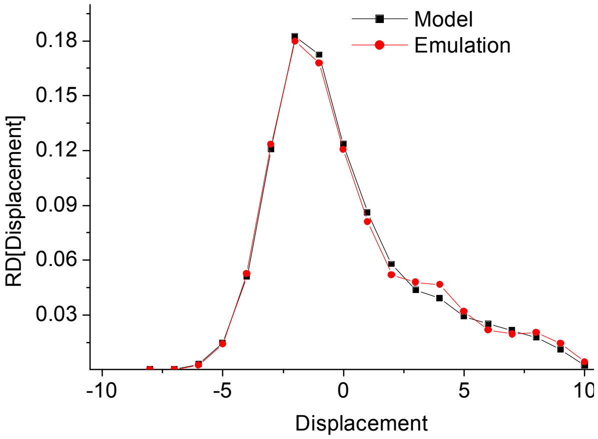


Fig.7. Comparison of RD from the Model and that obtained by emulation for five-path load splitting topology. Path probabilities are $P_0 = 0.5$, $P_1 = 0$, $P_2 = 0.25$, $P_3 = 0$, $P_4 = 0$, $P_5 = 0.125$, $P_6 = 0$, $P_7 = 0.0625$, $P_8 = 0$, $P_9 = 0$, $P_{10} = 0.0625$.

lower probabilities the RD monotonically decreases with increasing earliness and increasing lateness. On the contrary, if the longer paths had higher probabilities, more packets would have arrived far away from their actual positions resulting in highly undesirable reordering.

V. CONCLUSIONS

A model was derived for the profile of reordering, in terms of reorder density, due to multi-path packet forwarding. The model was verified using measurements from an emulated network. In the case of two-path load splitting, the RD distributions are mirror images for complementary probabilities of packets taking alternate route. In addition, as the delay of the alternate path increases the non-zero components of RD increases, resulting in increased buffer requirements for recovery from reordering. Though these models depict reordering due to load splitting, the methodology can also be used for evaluating effects due to parallelism with the nodes, i.e., parallelism due to queuing or parallelism due to processing in a router. This is currently under investigation. In addition, the inter-packet gaps are not always constant at the sending side, e.g., the gap with TCP streams depends on the congestion in the network. Another case of interest is when the path delays follow a wide distribution.

Characterization of reordering in packet sequences is critical for providing insight into the problems due to reordering. We believe that such insight will lead to the development of scalable techniques to deal with reordering, and also to the development of tools that diagnose network problems. Router vendors can evaluate their queuing or load splitting disciplines, and load balancing schemes using this model to predict the amount and extent of reordering. Similarly, ad hoc routing schemes, QoS criteria and other design and implementation decisions can make use of these models to keep reordering within acceptable limits.

REFERENCES

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

[2] 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, Fall 2005.

[3] L. Gharai, C. Perkins and T. Lehman, "Packet Reordering, High Speed Networks and Transport Protocol Performance", *Proc. IEEE ICCCN*, Oct. 2004, pp. 73-78.

[4] 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.

[5] 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.

[6] G. Iannaccone, S. Jaiswal and C. Diot, “Packet Reordering Inside the Sprint Backbone,” *TR01-ATL-062917,* Sprint ATL, Jun. 2001.

[7] D. A. Khotimsky, “A Packet Resequencing Protocol for Fault-Tolerant Multipath Transmission with Non-Uniform Traffic Splitting,” *Proc. IEEE GLOBECOM*, Dec. 1999, pp. 1283-1289.

[8] V. Paxson, “Measurements and Analysis of End-to-End Internet Dynamics,” Ph.D. Dissertation, Computer Science Department, University of California, Berkeley, 1997.

[9] 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.

[10] J. Bellardo and S. Savage, “Measuring Packet Reordering,” *Proc. IMW’02,* Nov. 2002, pp. 97-105.

[11] E. Blanton and M. Allman, “On Making TCP More Robust to Packet Reordering”, *ACM Computer Comm. Review*, Jan. 2002, pp. 20-30.

[12] 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.

[13] I. Aad, J-P. Hubaux, E. W. Knightly, “Denial of Service Resilience in Ad-hoc Networks,” *Proc. MobiCom’04*, Sept/Oct. 2004, pp. 202-215.

[14] S. Bhandarkar, A. L. N. Reddy, M. Allman and E. Blanton, “Imporving the Robustness of TCP to Non-Congestion Events,” IETF draft (work in progress).

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

[16] Y. Xia and D. Tse, “Analysis on Packet Resequencing for Reliable Network Protocols,” *Proc. IEEE INFOCOM,* San Francisco, CA, Mar. 2003, pp. 990-1000.

[17] 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.

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

[19] N. M. Piratla, A. P. Jayasumana and A. A. Bare, “A Comparative Analysis of Packet Reordering Metrics,” *Proc. 1st Intl. Conf. Communication System Software and Middleware (COMSWARE 2006),* New Delhi, India, Jan. 2006.

[20] 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/Reorder/draft-jayasumana-reorder-density-05.txt.

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

[22] N. M. Piratla, A. P. Jayasumana and T. Banka, "On Reorder Density and its Application to Characterization of Packet Reordering," *Proc. 30th IEEE Local Computer Networks Conference (LCN 2005)*, Sydney, Australia, Nov. 2005, pp. 156-163.