A Study of Packet Delivery Performance during Routing Convergence

*

Dan Pei, Lan Wang
UCLA CSD
{*peidan, lanw*}*@cs.ucla.edu*

Daniel Massey
USC/ISI
masseyd@isi.edu

S. Felix Wu
UC Davis CSD
wu@cs.ucdavis.edu

Lixia Zhang
UCLA CSD
lixia@cs.ucla.edu

Technical Report TR-030051
UCLA Computer Science Department
November 13th, 2003

Abstract

Internet measurements have shown that network failures happen frequently, and that existing routing protocols can take multiple seconds, or even minutes, to converge after a failure. During these routing convergence periods, some packets may already be en-route to their destinations and new packets may be sent. These in-flight packets can encounter routing loops, delays, and losses. However, little is known about how many packets are delivered (or not delivered) during routing convergence periods.

In this paper, we study the impact of topological connectivity and routing protocol designs on the packet delivery during routing convergence. We examine three distributed routing protocols: RIP, Distributed Bellman Ford and BGP through protocol analysis and simulation experiments. Our study shows that the packet delivery ratio improves as the network connectivity becomes richer. However differences in routing protocol designs impact their ability to fully utilize the topological redundancy in face of component failures. Two factors in routing protocol design, keeping alternate path information at each router and quickly propagating new reachability information, appear to have the most impact on the packet delivery behavior during convergence.

1 Introduction

Internet technology advances have benefited our society and increased our productivity, but at the same time these advances have also made us critically depend on the reliability of Internet services. At a very fundamental level, all applications depend on the Internet routing infrastructure for packet delivery service. In today's Internet, routers forward data packets hop-by-hop towards their destinations according to forwarding tables built by dynamic routing protocols such as BGP[23], OSPF[18], and RIP [15]. In [2] Paul Baran suggested that adequate redundancy in network connectivity combined with a dynamic routing protocol should enable data delivery even in the face of severe component failures.

In theory the Internet has the potential to meet Baran's ideal of reliable packet delivery. In practice, however, the Internet is a large-scale, complex, loosely-coupled distributed system made of many imperfect components. Measurement results show that faults of various scale and severity occur frequently at various locations in the Internet [10, 12, 14, 28]. Although routing protocols can adapt to these failures, it takes time both to detect a failure and to propagate the necessary update messages throughout the network. [11, 10] have shown that existing routing protocols may take multiple seconds, or even minutes, to converge after a failure. During this convergence period, some packets are already en route to their destinations and new packets continue to enter the network. These "in-flight" packets may encounter looping, delays or losses; however, there has been no systematic study of packet delivery performance during routing convergence periods.

Two recent technology advances further underscore the need to understand packet delivery during routing convergence. First, a rapid decrease in bandwidth cost over the last few years has resulted in a richer Internet connectivity [9]. This richer connectivity increases the potential to deliver packets over alternate paths after a failure, but many alternate paths could lead to increased convergence time for BGP when route flap damping is deployed [4, 16]. Second, rapid increases in link bandwidth have resulted in more "in-flight" packets at any given time, including the time during routing convergence. We are not aware of any systematic studies of richer connectivity's impact on packet delivery during a convergence period.

* This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No DABT63-00-C-1027 and by National Science Fundation(NSF) under Contract No ANI-0221453. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA or NSF. A shorter version of this paper has been published in IEEE DSN 2003[21].

This paper presents a systematic study of packet delivery performance during routing convergence periods. We define a *routing convergence period* as the time period between a fault detection and restoration of new path information at *all* the routers. We use simulations to examine the performance of three routing protocols: RIP [15], a Distributed Bellman-Ford algorithm (DBF) [3], and BGP [23]. Our primary concern is packet delivery rate during the convergence; all the other factors, such as delay or jitter, are only meaningful when packets are delivered.

Our study shows that the network must have adequate physical redundancy in order to assure reliable packet delivery in face of component failures, and that the packet delivery ratio for all three protocols improves as the network connectivity becomes richer. However different protocol designs can lead to significant differences in exploiting rich topological connectivity even within our chosen set of similar routing protocols. For example, with the same topology and same packet generation rate, RIP dropped 250 packets while BGP'(a specially parameterized version of BGP) dropped fewer than 5 packets. We identified two factors in routing protocol design that appear to have the most impact on packet delivery performance during routing convergence. First, in addition to the best path, a router should keep information of some alternate path to each destination, so that when the best path fails it can switch to an alternate path instantly for packet forwarding; the packet delivery rate can be substantially improved even if the alternate path may not be the new best path. Second, once a change of connectivity is detected, the routing protocol should propagate the new information as fast as possible. These results provide insights towards improving packet delivery during routing convergence.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 gives a brief introduction to the three routing protocols studied in the paper. In Section 4 we first identify three factors that we believe have an important impact on packet delivery during routing convergence, and then analyze how well the existing routing protocols match these three factors. Section 5 presents the simulation results. Section 6 concludes the paper and discusses our future work.

2 Related Work

It is generally believed that a shorter routing convergence time reduces packet losses. Previous efforts on routing protocol design have largely focused on speeding up routing convergence and preventing routing loops. These approaches tend to achieve loop-free routing through delaying routing update propagation. The ExDBF algorithm described in [5] avoids long-lived routing loops by computing the complete path to a destination using the predecessor information. If a neighbor appears in a node's computed path to a destination D, this node will not send update message regarding D to this neighbor since this will lead to loops. [6] describes DUAL algorithm which avoids routing loops by running a "diffusion" process before switching to a longer path. The routing table is "frozen" and the affected destinations are unreachable until the diffusion process completes. Our study differs from previous work by its focus on packet delivery performance $during$ routing convergence. Our results show that a shortest convergence time does not necessarily lead to a maximal packet delivery rate. We believe that an ideal routing protocol should achieve a good balance between the routing convergence overhead and convergence time, and most importantly should maximize the packet delivery rate during convergence.

[30] simulated the convergence behaviors of several routing protocols. The authors measured the convergence time, number of routing messages, and the routing loops after node or link failures, but did not measure packet delivery during routing convergence. [20] studied the end-to-end traceroute measurements collected in 1994 and 1995. The author detected a few transient loops and conjectured that these transient loops were caused by link failures. [8] used off-line analysis of traces containing the header of every packet traversing a link on a backbone ISP to detect loops. They observed that forwarding loops are rare, and that the delay of packets which do escape a routing loop is increased by 25 to 1300 msec. They also observed that 30% of loops on a subset of the links lasted longer than 10 seconds. The paper stated that the causes of observed forwarding loops are yet to be identified in future study. Our study examines how link failures affect routing and packet forwarding by studying the forwarding and routing trace files, thus we can identify the causes of routing loops in each circumstance.

[26] simulated the loop-free MS distance vector algorithm from [17], the ExDBF algorithm from [5], and a link state protocol (SPF) using the NSFNET backbone topology. The workload used is an FTP application which does not use TCP but has a simple flow control with a maximal window size and retransmission after timeout. They measure the packet throughput, packet delay and routing load(bandwidth consumption). The authors observed that, although the SPF and ExDBF algorithms are known to have transient loops, their packet delivery performance is better than that of the loop-free MS algorithm. While this work measured the end-to-end data delivery performance under different routing protocols, our study examines the packet delivery performance with topologies of different connectivity levels. In addition, we examine in detail packet-level dynamics such as number of packet drops, number of TTL expirations, number of transient forwarding paths, forwarding path convergence delay, to understand exactly how transient routing protocol behavior affects packet delivery and hence the performance of data flows.

Other related work that aims at maximizing packet delivery during routing convergence include having alternate path always ready either at the routing table [29] or at line-card [1], and "non-stop forwarding" in which a router keeps forwarding packets while rebooting its routing pro-

tocol daemon [19, 25, 24].

A router cannot forward packets when a destination becomes unreachable until the reachability is restored. In an attempt to improve packet delivery after a failure, Alaettinoglu *et al* proposed that the router pre-computes a backup next hop in the line-card of the router and uses the backup next hop when the primary next hop is removed [1]. This meets the " having a backup readily available" goal that we identify later in Section 4 ,but in this case the backup is present in the line card as well as being present in the routing process.

For most of the routers, restarting the routing protocol daemon does not affect the forwarding functionality of the router if the router itself does not perform a hardware reboot. In this case, it is unnecessary for the neighbors of the restarting node R to remove R and subsequently add it back again, a process which may involve a large number of update message exchanges and route recomputation, since R's routing state before and after the restart is the same and packet forwarding is not impacted by the restart. Therefore, there are proposals of "non-stop forwarding" of these restarting routers for OSPF[19] and BGP[24]. However because a restarting router cannot participate the routing messages exchanges since its $routing$ process is inactive, it is possible that routing loops may occur if there are topological changes during the restarting period. Neighbors of the restarting router will decide whether the new topological changes they observe cause loops. If so, they will take the rebooting router as a normal failed node, and send corresponding routing messages, i.e, terminate the "non-stop forwarding" [19, 25].

3 A Brief Introduction to RIP, DBF and BGP

In this paper we consider three distributed routing protocols: RIP [15], DBF(Distributed Bellman Ford) [3], and BGP [23][1]. In addition to being widely used in network literature and practical networks, all of the three are variants of classic distance vector routing protocols. We selected three routing protocols in the same algorithm family so that we can better correlate the difference in the routing protocol design with the difference in the observed packet delivery dynamics.

RIP [15] is one of the best known routing protocols. In RIP, each router periodically advertises its shortest distance to each destination. Based on the distances learned from all its neighbors, a router selects the neighbor that leads to the shortest distance path to a given destination as the next hop, and discards the reachability information for the same destination from all other neighbor routers. Routing updates are sent every 30 seconds, and a routing entry is removed if it does not have an update within 180 seconds. Whenever a route change is detected, the router sends a "triggered updates" immediately instead of waiting for the next update interval. A damping timer is applied to space out consecutive update messages; the timer's value is randomly chosen between 1 and 5 seconds. In our simulation RIP is also enhanced with "split horizon with poison reverse" two-hop loop prevention scheme: If a node A uses B as the next hop to reach destination D, A will send B an "infinity" distance(16) to D.

The DBF algorithm is defined in [3]. In our simulation implementation the only difference between RIP and DBF protocol is that a DBF router keeps a cache of the latest routing update learned from each of its neighbors. Whenever a router notices that it cannot reach a destination through the current next hop, the router can immediately select an alternate next hop. As with RIP, the DBF protocol adopts the "split horizon with poison reverse" loop prevention mechanism and sends triggered updates upon routing changes. Note that each RIP(or DBF) update *message* may contain up to 25 destination *entries* according to RIP standard[15].

BGP [23] is a path vector protocol and each node announces to its neighbors the best *path* (a sequence of nodes) to each destination. A router keeps a copy of the latest best path received from each of its neighbors. Because BGP uses TCP for reliable delivery between neighbor nodes, routes to all destinations are advertised once only. A router sends an update only upon route changes. It sends an explicit withdrawal message to its neighbors when it cannot reach a previously reachable destination. Similar to RIP and DBF, BGP uses a timer to space out consecutive updates for the same destination by Minimum Route Advertisement Internal (MRAI)(This timer is called MRAI timer in the rest of the paper). BGP specification recommends an average MRAI value of 30 seconds with a jitter interval of 5 seconds. In our simulation, we implemented both this recommended MRAI value and a modified average MRAI value of 3 seconds with a jitter interval of 2 seconds. This smaller MRAI value makes BGP's damping delay for triggered updates comparable with that of RIP and DBF. We name this specially parameterized version of BGP *BGP'*. In most BGP implementations and in our simulation, the MRAI timer is set on a per neighbor node basis rather than the per (neighbor, destination) basis.

The BGP path information is used to prevent routing loops. When a node A receives a path from neighbor B which contains A as one of the nodes in the path, an indication of routing loop, the node should discard this new path. Our implementation treats such a path as a withdrawal message and thus is similar to the "split horizon with poison reverse" loop prevention scheme in RIP and DBF. Note that although the "counting-to-infinity" behavior cannot occur in BGP, other routing convergence problems may still occur [11].

[1] We limit our examination of BGP to shortest-path routing policy only, while in reality BGP is used to support more complex routing policies. Furthermore in our simulation each Autonomous System(AS) consists of just one single BGP router while in reality an AS consists of multiple BGP routers.

4 Routing Protocols During Convergence

In any large scale network, there will be periods when the route to a particular destination has not converged, yet hop-by-hop routing protocols (such as those used in the Internet) continue to forward packets regardless of whether the route has converged. IP packets carry a Time To Live (TTL) field which specifies the maximum number of hops the packets may travel. As long as a packet's TTL value is greater than zero and the router knows some next hop to reach the destination, the packet is forwarded to the next hop and the TTL value is decremented by 1. Although the sequence of next hops traversed by a packet during a routing convergence period (called "transient forwarding path") may be sub-optimal or even contain transient loops, the packet may still have a good chance to reach its destination.

Figure 1 shows an example of how packets can be delivered during routing convergence. In this figure, each link has a unit cost and the packet forwarding path between $R1$ and $R4$ is shown in dashed lines. Initially, as shown in Figure 1(a), $R1$ is sending packets to $R4$ along the shortest path. In Figure 1(b), the link $(R5\ R4)$ goes down. $R1$ continues to forward packets to $R5$ and $R5$ transmits the packets over the failed link. In Figure 1(c), $R5$ detects the link failure and switches to forwarding packets to $R6$. Unaware of the connectivity changes $R1$ continues to forward packets to $R5$. During this time the packets are forwarded through a non-shortest path. Finally, in Figure 1(d), $R1$ converges to the new shortest path, and forwards packets to $R2$. Note that in this example packets are only dropped between the instance the link fails and the time $R5$ switches to forwarding packets to R6 (Figure 1(b)), and that during the subsequent convergence period (Figure 1(c)) packets successfully reach the destination by going through a non-shortest path route. Also note that the time it takes to move from Figure 1(c) to Figure 1(d) counts as part of the routing convergence delay, even though packet flow has been restored. This example shows that, after a failure, a longer routing convergence period does not necessarily imply higher packet losses.

Understanding the relation between routing convergence and packet delivery raises new and interesting challenges for routing protocol design. In the remainder of this section, we identify factors that have an important impact on packet delivery during routing convergence.

4.1 Path Switch-over Period

We say a *path switch-over period* starts when a router discovers its current next hop can no longer reach a given destination and ends when the router finds a new next hop for the same destination. Because the router cannot forward any packets for that destination during the path switch-over period, an ideal network routing protocol should have a minimal path switch-over period. Forwarding packets to an alternate next hop offers a chance that the packets may eventually reach their destinations, even when the next hop is not necessarily on the new shortest path after a failure.

In RIP, a router R only keeps the information for the next hop along the shortest path. R loses the reachability to a destination whenever the router detects the failure of the link to the next hop or the next hop reports the destination is unreachable. Although R's other neighbors may not be affected by the failure, these neighbors will not inform R their reachability to the destination until the next periodic update is due. Therefore, after a failure, a RIP router may take up to 30 seconds before it learns an alternate path. As a result, RIP suffers from a potentially long path switch-over period. In contrast to RIP, a router running DBF or BGP keeps a cache of the reachability information learned from all its neighbors. When it can no longer reach a destination through the current next hop, the router can immediately select an alternate next hop for the destination, achieving a zero time path switch-over. However note that there is no guarantee that the selected alternate next hop leads to the shortest route to the destination, nor that the next hop can even reach the destination. This leads us to consider the next important factor, probability of choosing valid paths.

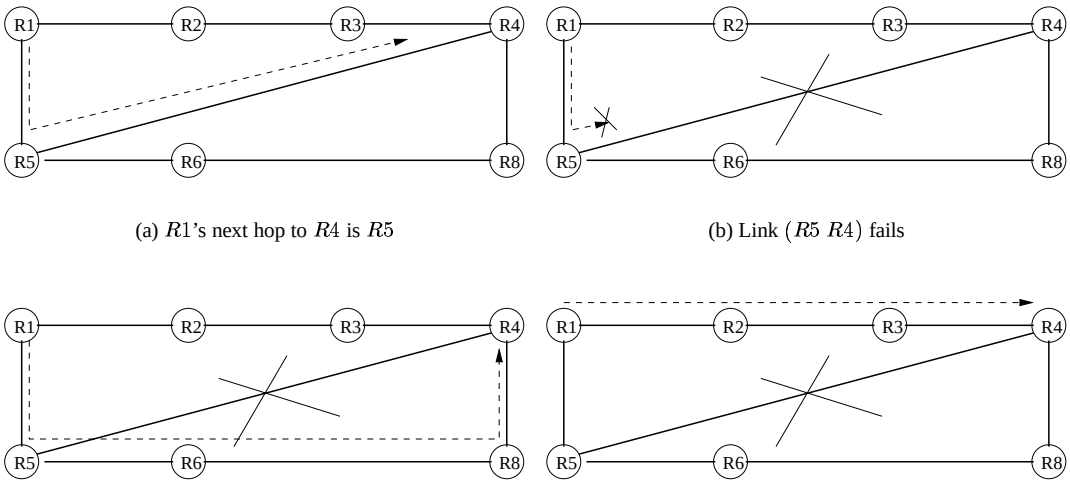
4.2 Probability of Choosing Valid Paths

Ideally, when the existing path fails, the router should switch to a new path which does not use the failed link. We call any alternate path that avoids the failed link a *valid path*. The second factor of an ideal routing protocol is the high probability of choosing a new next hop with a valid path if there exist multiple alternate paths. A valid path can be sub-optimal, as long as packets can reach the destination while the routing protocol is converging.

"Split horizon with poison reverse" avoids two-hop loops, thus helps increase the probability of valid alternate paths after a failure. In all the three protocols we studied, BGP is the only one that allows a node to check whether a chosen alternate path contains a failed link in some restricted cases. For example, [22] utilized this feature of BGP to substantially reduce the routing convergence time. However due to the existence of routing policies and other constraints, after a route failure BGP may still alternate among a number of new routes before converging to a stable route. Such transient route instability can happen in all the three studied protocols, and is caused by inconsistent connectivity information perceived by different routers while the latest update is being propagated through the entire network. This leads us to consider the third important factor, propagation time of Failure information.

4.3 Propagation Time of Failure Information

Upon detection of a physical failure, an ideal routing protocol should propagate the failure information through the network as quickly as possible, so that all the routers



(a) $R1$'s next hop to $R4$ is $R5$

(b) Link $(R5\ R4)$ fails

(c) $R5$'s next hop becomes $R6$

(d) $R1$'s next hop becomes $R2$

Figure 1. Packet Could still Be Delivered during Convergence

can recompute the shortest paths to the affected destinations. However this propagation takes time, and due to the distributed nature of distance-vector and path-vector algorithms. a router which has received the failure notification may not find the new best path, or even a valid path, at this time because the reachability information it received from its neighbors earlier may have been invalidated by the failure. Nevertheless when a router changes its paths, it will inform its neighbors about the new paths. If the new path is also invalid, packets following that path are likely to be lost. However if the new path is valid but sub-optimal, packets sent along that path have a good chance to reach their destinations.

In RIP or DBF, upon detecting a failure a router sends a triggered update instead of waiting for the next 30-second update interval. BGP sends only triggered routing updates upon route changes. In this sense, all the three studied protocols attempt to achieve the goal of quick propagation of failed paths. However, because damping timers for triggered update (3-second average for RIP, DBF and BGP', and 30-second average for BGP) are used to space out consecutive updates, a node delays the sending of all update messages except the first one. One exception is made in BGP which does not apply the timer to withdrawal messages in order to propagate the unreachability information quickly. But A link failure may cause route changes to multiple destinations, and updates regarding these destinations may not be received by a BGP router at the same time. After a BGP router has processed all the changed path and sent out corresponding updates, it turns on the MRAI timer. After the timer is on, any newly changed paths to destinations not in the previous update messages are delayed by the per neighbor MRAI timer, but not by per(neighbor, destination) MRAI timer.

4.4 The Impact of Network Topology

In addition to the routing protocol design choices, our claim is that the ideal factors of a minimal path switchover period, a high probability of picking a valid path, and fast propagation of updated connectivity information, all benefit from a richly connected network topology where routers have a high connectivity degree. Intuitively, as a network becomes more interconnected, the number of alternate paths increases, and the probability that an alternate path goes through the same failed link decreases. Furthermore, the rich connectivity also reduces the average path length between any two points in the network. This reduced path length helps reduce the propagation time of the failure information, which is related to not only the link propagation delay but also the triggered update (or MRAI) timer. For example, [13] has shown that the BGP convergence time is proportional to the product of MRAI timer value and the length of the longest backup paths.

5 Simulation Results and Analysis

We simulated RIP, DBF, BGP and BGP' using IRLSim simulator [27]. In the simulated networks, each link has a unit cost, a propagation delay of 1 ms, and a transmission rate of 10 Mbps. A link failure is detected by the two nodes attached to it within 5ms after the failure happens. Each node has a packet queue sizes of 200 packets and zero CPU processing delay. Note that because this paper is a comparative study of different routing protocols, the exact values of these parameters should have little impact on the results.

In order to study the protocol behavior at different topological connectivity levels, we choose to use a family of regular network topologies. A random topology presents

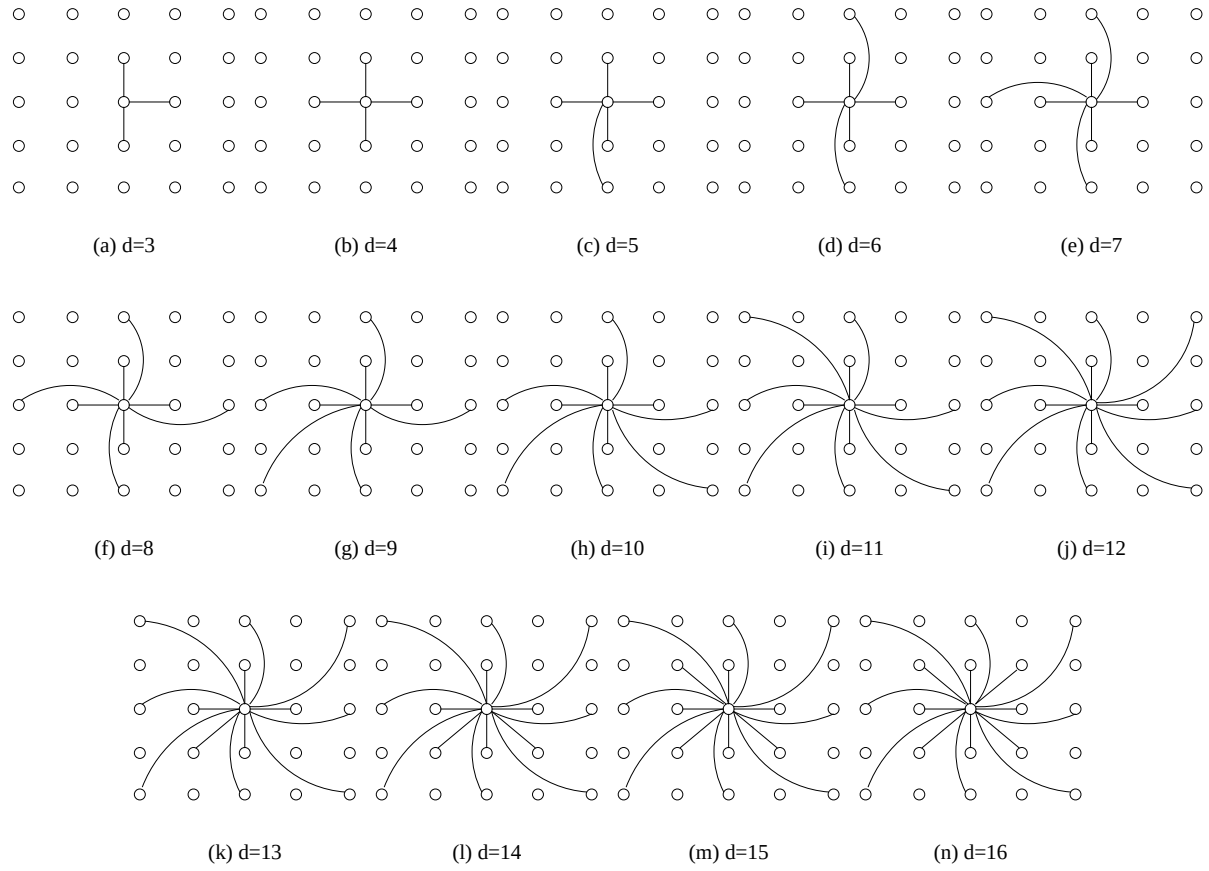


Figure 2. Constructing links for a node with degree(d) from 3 to 16

a random factor in each simulation run. Using regular topologies removes this undesirable random factor and allows us to clearly identify the impact of connectivity level on the protocol performance. The simulated network topology is a mesh of N rows by N columns and each node in the network (except those on the border) has the same node degree d. There are various ways to construct such topologies, we use a deterministic method similar to the one used by Baran in [2]. Figure 2 shows for a node with d from 3 to 16, how to connect it to the rest of the network. As an example Figure 3 shows three example topologies for $N = 5$ and $d = 4, 5, 6$.

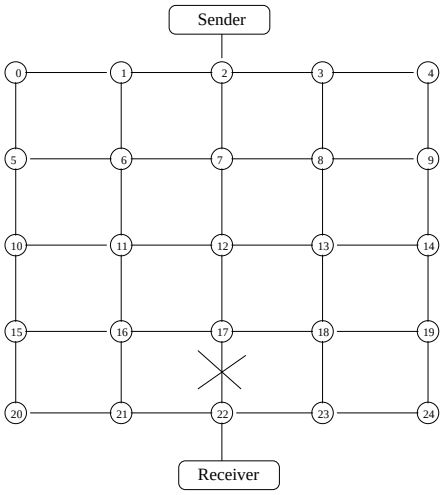
We run each simulation experiment for 800 seconds. There is a warm-up period after each simulation starts, during which time period the network nodes exchange routing update messages and the routing table at each node stabilizes. At time $t = 390$ seconds, a single sender starts sending IP packets with TTL=127 to a single receiver at a constant transmission rate of 20 packets/second. The sender and the receiver are connected to a randomly chosen router on the first row and last row of the regular topology, respectively. At time $t = 400$ seconds, one of the links along the shortest path between the sender and receiver is randomly chosen to fail. We simulated topologies with $N = 7$ and d ranging from 3 to 16. For each topology with a different node degree, we conducted 100 simulation runs to collect statistically valid performance measurement. Due to the space limitation, in the paper we use the topologies with $N = 5$ shown Figure 3 to help explain our observations.

5.1 Packets Drops due to No Reachability

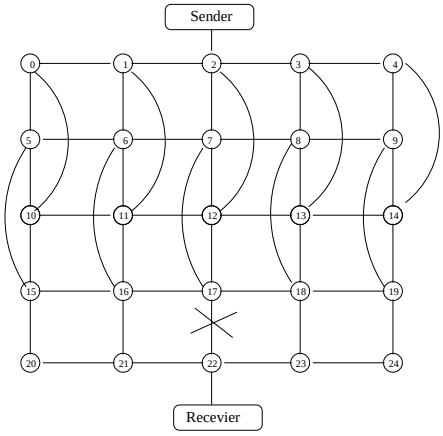
When a packet arrives at a router which is in the switch-over period after a failure, the packet is dropped because the the router does not know the next top to reach the destination. Figure 4 shows the average number of packet drops due to lack of reachability over 100 simulation runs.

Observation 1 *For all the three examined routing protocols, the number of packet drops decreases as the node degree increases until it reaches 6.*
When the node degree is 6 or more, there are virtually no packet drops with DBF, BGP, and BGP'. But in RIP, packet drops improve only slightly with the increase in node degree.

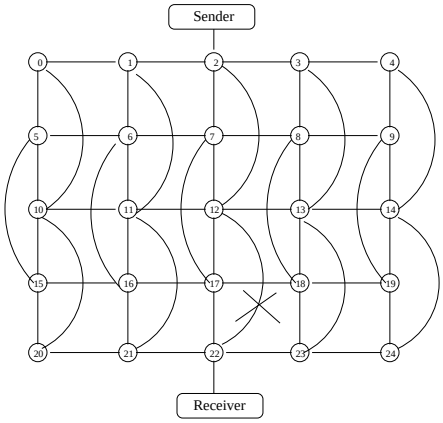
Our explanation is the following. Consider the behavior of a node A that lies on the shortest path from the sender to the receiver. In a sparse network, it is often the case that A is chosen by its neighbors as the next hop to the destination. Thus when A learns that its current next hop



(a) degree 4



(b) degree 5



(c) degree 6

Figure 3. Link Failures in Networks with node degree 4,5 and 6

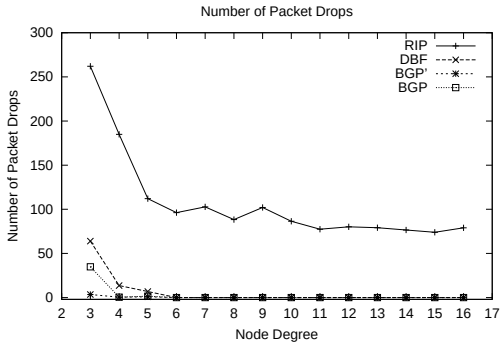


Figure 4. Number of Packet Drops due to no route Vs. node-degree

to the destination is no longer valid, its neighbors cannot immediately offer an alternate path to the destination. For example, consider the behavior of node 17 in Figure 3(a) (a sparse topology with degree 4). Node 17 is chosen by neighbors $12, 16, 18$ as the next hop to the destination and nodes $12, 16, 18$ use Poison-Reverse loop prevention to inform node 17 that their distance to the receiver is infinity. When link $(17\ 22)$ fails, node 17 finds no alternate path until next periodic update cycle when the new reachability through nodes [16, 21, 22] or [18, 23, 22] is discovered. Similar situations can happen to the other nodes along the shortest path before the failure, i.e. nodes 2, 7, and 12.

On the other hand, a densely connected network makes it likely that a node A has one or more neighbors whose shortest path to the destination does not go through A. For example, consider node 12 in Figure 3(c) (a topology with degree 6). Before the failure node 12's next hop to the destination is node 22 and under DBF, BGP and BGP' node 12 also keeps the information from its neighbor node 17 about its reachability to the destination. When link $(12\ 22)$ fails, node 12 immediately finds the alternate path [12, 17, 22] and starts forwarding packets along this new path. Comparison of Figure 3(a) and 3(c) shows that increasing the node degree to 6 essentially guarantees that all the nodes on the initial shortest path can find a valid alternate path after any link along the shortest path fails. But a network running RIP largely depends on the periodic updates to propagate information about alternative paths after a failure and node 12 does not learn of the alternate path via node 17 until the next update. A higher node degree only slightly reduces the propagation delay of the periodic update messages and the number of packet drops in RIP decreases only slightly.

5.2 Number of TTL Expirations

Figure 5 shows the number of packets dropped due to TTL expirations. Given the large TTL value (127) and the small size of the simulated topology, all the TTL expirations are caused by routing loops during convergence.

Observation 2 *For topologies with a node degree below*

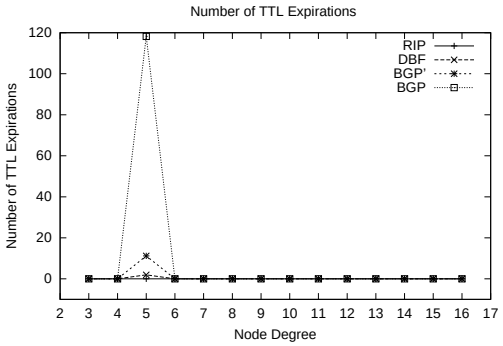


Figure 5. Number of TTL Expirations During Convergence

6, BGP has the largest number of TTL expirations while RIP has no TTL expirations. When the node degree is 6 or higher, there are no TTL expirations.

The reason no TTL expirations occur under RIP is the following. Whenever a link failure happens, a triggered update is sent quickly. and with our 1ms link delay, the failure information can propagate along the path in a few milliseconds. Furthermore, packets enter the network at a relatively slower rate of 20 packets/second and because a RIP node keeps no alternative path information, the node next to the sender will drop all the incoming packets when the current shortest path fails. As Figure 4 shows, RIP avoids looping by simply dropping all the incoming packets till new reachability is established.

Analysis of the routing and forwarding traces shows that BGP's slow convergence problem [11] combined with the MRAI timer are mainly responsible for the forwarding loops with the topology of node degree 5. For example, nodes 2,7, and 12 in Figure 3(b), can easily form a routing loop. At one moment after failure of link (17 22) during the simulation, node 2's path is [2 12 7 17 22] while node 12's path is [12 2 7 17 22]. However, MRAI timers of both node 2 and node 12 have been turned on by some previously exchanged updates and no new updates can be sent before one of the timers expires, thus a forwarding loop is formed. This example shows that different nodes based on inconsistent information might form a transient loop, and the looping period is lengthened by the MRAI timer. The number of TTL expirations in BGP is about 10 times of that in BGP', and this is consistent with the fact that average MRAI timer value in BGP(30 seconds) is about 10 times of that in BGP'(3 seconds).

Although the MRAI timer value for BGP' is about the same as the damping timer for DBF, the two routing protocols show noticeable difference in the number of TTL expirations. This difference is due to some specific details in how the damping timer is applied. A single DBF update message can contain as many destination entries as the message size allows (25 destinations). Thus given the size of simulated topology (49 nodes total), a single update is likely to contain all the affected destinations by the link failure. On the other hand, BGP is a path-vector protocol and a single BGP update can only contain those destinations that share the same path. Because our simulation implemented the MRAI timer on a per neighbor basis (as in most vendor BGP implementations), after a link failure only the first BGP update message can propagate quickly. Any further update messages will be regulated by the MRAI timer, resulting in a longer time when different nodes have inconsistent routing information. Thus BGP' suffered more from transient loops which lead to a higher number of TTL expirations. Note that the results could have been different had the MRAI timer been implemented on a per (neighbor, destination) basis.

We see that in our simulations, RIP has no loops and DBF has fewer loops than BGP (and BGP'). Conversely, BGP(and BGP') has more routing information than DBF which has more than RIP. Although BGP assures that a node picks a path that does not contain itself, this does not always prevent transient looping. Furthermore a common implementation simplification in the BGP's MRAI sets the time on a per neighbor rather than per (neighbor, destination) basis and lengthens BGP's convergence time.

5.3 Instantaneous Throughput

Figure 6 shows the instantaneous throughput (at each second) measured in packet/second at the receiver with node degrees 3, 4, 5, and 6, respectively. The results for node degree 7 and higher are similar to that of node degree 6. Note that due to TTL expirations in BGP and BGP' at degree 5, their throughput performance are worse than those at degree 3, 4 and 6. For clarity, we normalize time by subtracting the 390 second warm-up period, thus the failure is injected at $t = 10$ seconds in Figure 6. Note that the results shown in the figure are the average throughput of 100 simulation runs.

Observation 3 *In a sparse network, a link failure on the existing path between the sender and receiver tends to cause an instant throughput drop with all the protocols under study. For BGP, BGP', and DBF, the throughput then increases gradually and resumes full throughput around the triggered update timer values. RIP does not resume full throughput until the 30 second periodic update value.*
In a dense network, increased node degree reduces the throughput drop for DBF, BGP, and BGP' to negligible amount. RIP does only slightly better as the network becomes more dense.

Because RIP routers do not keep alternate path information, after the failure they must wait for other routers' periodic announcements (or triggered update if they notice path changes) to learn an alternate path. Consequently RIP's throughput right after the failure is almost zero. For

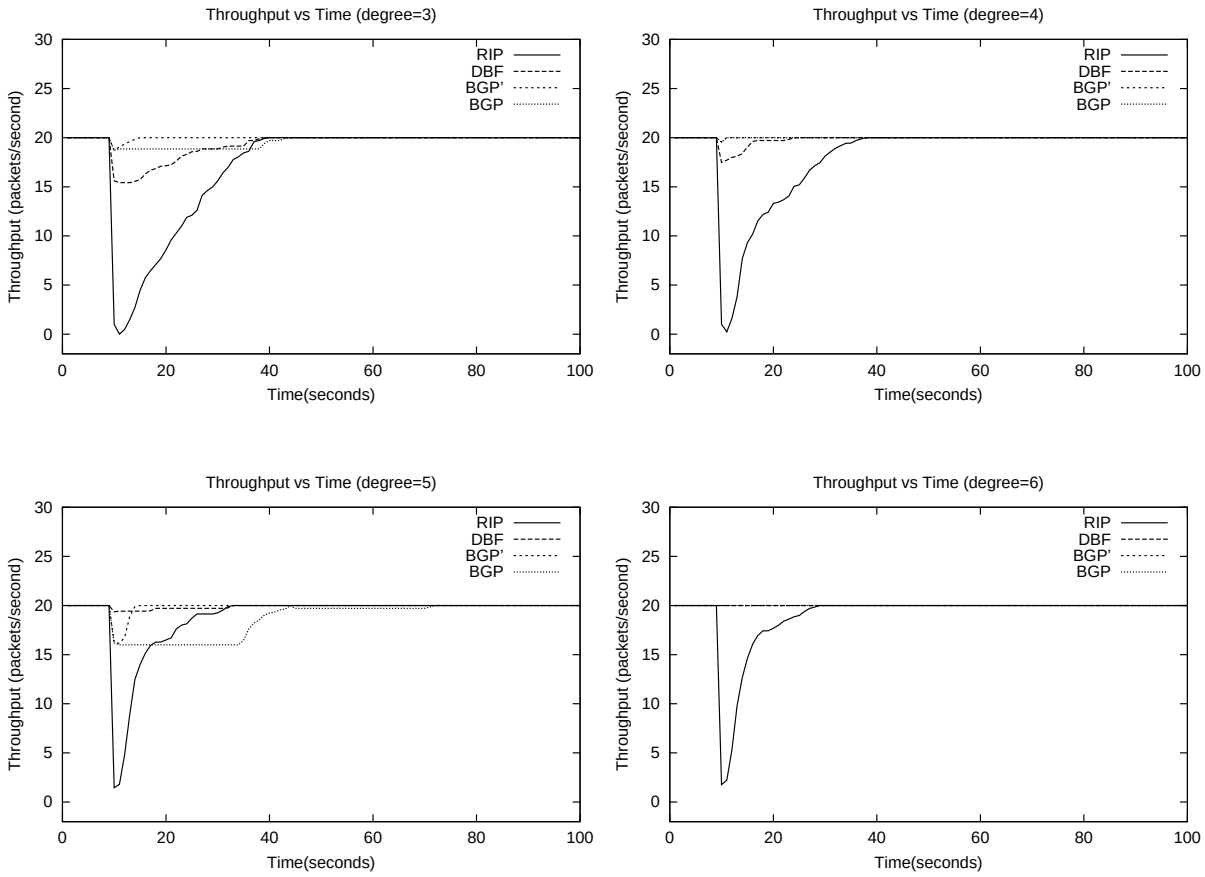


Figure 6. Instantaneous Throughput

node degree 3, RIP's throughput climbs back to the original throughput at about 30 seconds later after failure, which matches the periodical update interval. The time it takes RIP to climb back to the original throughput decreases slightly as the node degree increases: more neighbors mean it's more likely to receive a periodic announcement earlier.

With BGP, BGP', and DBF, the throughput does not drop to zero since it is possible for the routers to have alternate paths available. For node degree 3, BGP's gradual throughput increase begins at about 25 seconds after the failure and ends at about 35 seconds after the failure, which match the value of the MRAI timer. Similarly, the sharp throughput increase of BGP' begins at about 1 second after the failure and ends at about 5 seconds after the failure. The results show that increased node degree can reduce the throughput a lot with BGP, BGP' and DBF. The time it takes DBF to climb back to the original throughput decreases quickly as the node degree increases: it's about 30 seconds at degree 3, and 20 seconds at degree 4, and almost zero seconds at degree 6.

5.4 Forwarding Path Convergence Delay

The *forwarding path convergence delay* starts when the failure is detected by a router and ends when the forwarding path between the sender and receiver stabilizes over the final shortest path after the failure. After that, all the packets are delivered along the converged path. Note the forwarding path convergence delay is different from the routing convergence time. The forwarding path convergence delay ends when each node on the path from our sender to receiver has converged to the final next hop, but other remote nodes in the simulation may still be experiencing path changes. Note that the routing convergence time of DBF and RIP are measured by looking at the last triggered update messages, while the real convergence is achieved through a broadcast message. Therefore the numbers shown in the figure might be up to five seconds shorter than the actual convergence time, but it doesn't affect our observation

Figure 7(a) shows the average forwarding convergence delay, and Figure 7(b) shows the average time for the routing protocol to converge. Figure 4 has shown that there are basically no packet drops with DBF, BGP', and BGP in networks with node degree 6 and higher. Note however,

that Figure 7(a) shows that path convergence delay, especially those of BGP and BGP', are above zero even at high degrees.

Observation 4 *BGP' has a much shorter path convergence delay than BGP, although the difference of packet drops between these two versions of BGP is much less noticeable.*

The work in [7] shows that BGP's MRAI timer can be adjusted to minimize the network convergence delay, and our results in 7(b) shows that in the topologies simulated a smaller MRAI value reduces both the network routing convergence delay and the forwarding convergence delay. Furthermore, we also observed that the number of packet drops during routing convergence is not directly proportional to the forwarding convergence delay or the network convergence delay. For example, at node degree 6, the network convergence delay difference is about 60 seconds, and the forwarding convergence delay difference between BGP and BGP' is about 11 seconds, but the packet drops difference shown in Figure 4 is negligible. That is, with a degree 6 or higher, tuning the MRAI value might minimize the network convergence time (potentially with more message overhead), but it does not necessarily help packet delivery as much.

Figure 8 shows that the average number of transient paths is always higher than 0 even in a very high degree. However, Figure 7(a) has shown that the $average$ forwarding path convergence times are very close to zero in DBF and BGP' at high node degrees. This shows that some routers might have just quickly tried some transient paths before the convergence. Futhermore, Figure 8 also shows that the average number of transient forwarding paths first increases and then decreases as the node degree increases. An increase of node degree at the low degrees increases the chance that there $exist$ some alternate paths for the nodes. As long as the local node has not learned the new best path, it will choose the best one from the available alternate paths. As a result, the number of transient forwarding paths increases. On the other hand, when the node degree increases to be even higher, it could almost guarantee that there $exist$ some alternate paths after a failure. Therefore, when degree increases at high degrees, it is more likely that the local node has a copy of the path sent by the the eventual next hop at the time of the failure. As a result, the average number of transient paths decreases as node degree increases at high degrees. This can also help explain the trend of the forwarding path convergence curve.

5.5 Instantaneous Packet Delay

Those packets delivered during the convergence might traverse more hops than the new best path and result in longer end-to-end packet delay. Figure 9 shows the average instantaneous delay of those packets delivered at time t for networks with node degree 3, 4, 5, and 6. The results

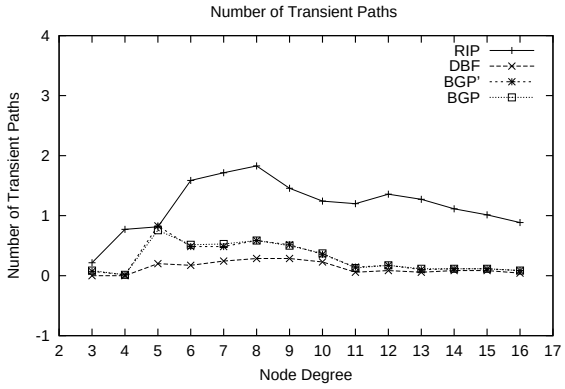


Figure 8. Extra Forwarding Paths Traversed by Packets

for node degree 7 and higher are similar to that of node degree 6. Note that the average path lengths of networks with different degrees are different, therefore it is unfair to compare directly the absolute value of delay between networks with different degrees.

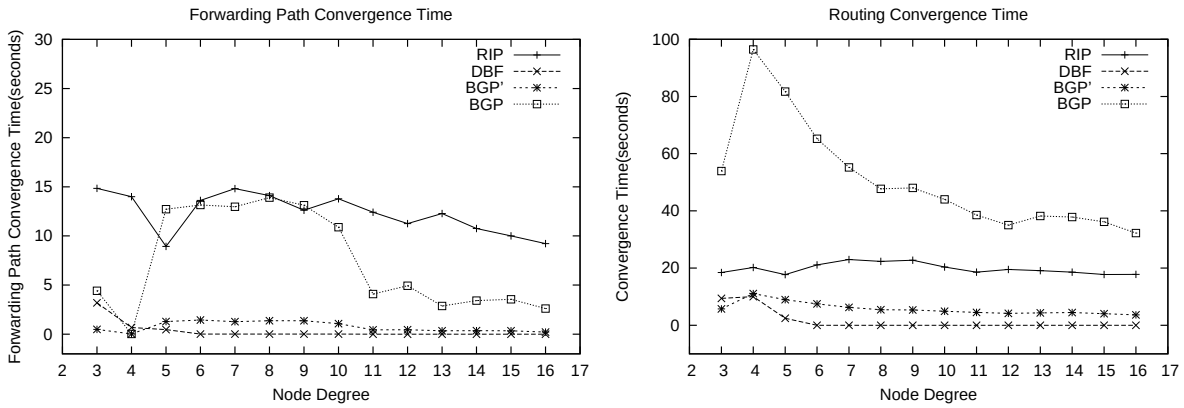
Observation 5 *When node degree increases, DBF, BGP, and BGP' might experience more extra delay than the eventual steady delay, and those packets escaping from forwarding loops have even longer delays.*

BGP's extra delay at degree 6 is larger than those at degree 4, but study of trace files shows that these extra delays are caused by those extra packets delivered due to the higher degree. Note that the delay for degree 5 oscillates at about $t = 40$ seconds. Study of the packet forwarding trace files shows that at about $t = 40$ seconds, some packets involved in loops had escaped from the loops, and these packets have delays much larger than those simply traversed some sub-optimal alternate paths.

6 Conclusion

The Internet has grown in multiple dimensions. Growth in size increases the frequency of component failures, growth in link speed increases the potential number of packets en-route during a routing convergence period, and growth in topological connectivity offers increased number of alternate paths after any component failure. However it relies on the correct routing protocol design to best utilize the rich redundancy in connectivity to assure continuous packet delivery during routing convergence.

In contrast to most of previous efforts on routing protocol designs which focused on preventing routing loops and minimizing convergence time, in this paper we evaluated the impact of topological connectivity and routing protocol design choices on packet delivery during routing convergence periods. Our study shows that, although increased network connectivity improves the packet delivery ratio in



(a) Forwarding Path Convergence Time

(b) Network Routing Convergence Time

Figure 7. Forwarding Path Convergence Time and Routing Convergence Time

general, differences in routing protocol designs can lead to significant differences in packet delivery performance. Our observations show that

- When a router keeps information about alternate paths, it can instantly switch packet forwarding to an alternate path P when the best path to a destination fails. Even in cases when P is not the final path that will be used when the routing protocol has converged, packets forwarded to P have a good chance of reaching their destinations in a well connected network. Increases in network connectivity increase the probability of packet delivery.

- Once a failure is detected, the routing protocol should propagate the failure information as fast as possible. Not only can fast failure report minimize the convergence time, it can also improve the delivery ratio of those en-route packets at the time of failure.

- Contrary to common intuition, a path vector routing protocol does not necessarily eliminating loops. Our simulation demonstrated that transient routing loops can occur in a network using BGP as the routing protocol. Furthermore, the looping duration is lengthened by BGP's MRAI timer.

We also observed that, although using alternative paths without validity verification can potentially lead to routing loops, increases in network connectivity can quickly reduce the probability and duration of looping. In a well connected network, how to handle the *counting-into-infinity* issue associated with distance vector routing protocols deserves a re-examination. For example, in order to reduce the risk of *counting-into-infinity*, RIP design keeps no alternate path information which results in packet losses once the best path to a destination fails, and new reachability can take long to establish. Similarly, other existing loop prevention approaches such as those proposed in [6] eliminate routing loops by paying a high cost of delaying routing updates and stopping packet delivery during convergence. Our study shows that, in a network with redundant connectivity, after a path failure a distance vector routing protocol simply counts to the next-best path instead of *counting-into-infinity*. The higher the redundancy in connectivity, the sooner the *counting-into-infinity* is likely to stop. This work represents a first step towards understanding packet delivery during routing convergence. Our investigation started with the simplest case of packet delivery between a single source and a single destination connected by a regular topology, and measured the routing and packet delivery dynamics after a single isolated failure. As a next step to gain further insight, we plan to extend the simulation experiments to larger network sizes, multiple pairs of data sources and destinations, as well as multiple failures which can potentially overlay with each other in time.

Two other directions for future work include extending the routing protocol family being examined from distance-vector and path-vector based routing protocols (RIP, DBF, and BGP) to include link-state routing protocols, and comparing the results with those of distance-vector based protocols; and extending the packet delivery performance measure from IP layer to include end-to-end TCP performance during routing convergence period.

7 Acknowledgments

We would like to thank Randy Bush for encouraging us to work on the research direction in this paper. We would also like to thank Ke Zhang, Xiaoliang Zhao and Mohit Lad for their comments on an earlier version of this paper and thank the DSN 2003 anonymous reviewers for their valuable and insightful comments which helped improve the quality of this paper. We would also like to thank Peter Follett, Charlie Fritzius and Stephen Sakamoto for provid-

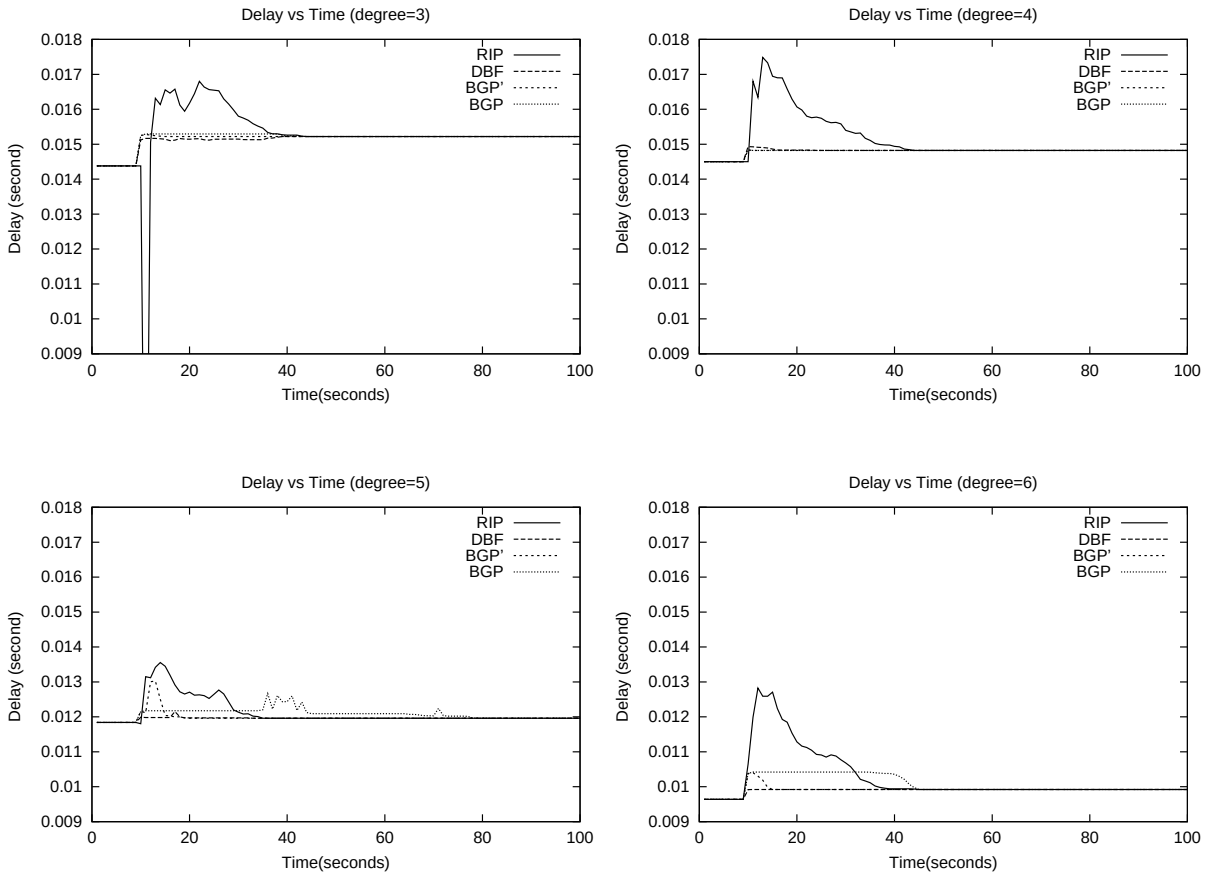


Figure 9. Instantaneous Packet Delay

ing us additional computing resources for the simulation experiments.

References

[1] C. Alaettino and A. Zinin. Igp fast reroute. Talk slides, http://www.packetdesign.com/publications, 2002.

[2] P. Baran. On Distributed Communication Networks. *IEEE Transactions on Communications*, 12(1):1–9, 1964.

[3] D. Bertsekas and R. Gallager. *Data Network*. Prentice-Hall, 1992.

[4] R. Bush, T. Griffin, and Z. M. Mao. Route Flap Damping: Harmful? http://www.ripe.net/ripe/meetings/archive/ripe-43/presentations/ripe43-routing-flap.pdf, 2002.

[5] C. Cheng, R. Riley, S. Kumar, and J. Garcia-Lunes-Aceves. A Loop-Free Extended Bellman-Ford Routing Protocol Without Bouncing Effect. In *Proceedings of ACM Sigcomm*, pages 224–236, August 1989.

[6] J. J. Garcia-Luna-Aceves. A unified approach to loop-free routing algorithm using distance vectors or link states. In *Proceedings of ACM Sigcomm*, September 1989.

[7] T. Griffin and B. Premore. An Experimental Analysis of BGP Convergence Time. In *Proceedings of ICNP*, November 2001.

[8] H. Hengartner, S. Moon, R. Mortier, and C. Diot. Dection and Analysis of Routing Loops in Packet Traces. In *Proceedings of ACM IMW 2002*, October 2002.

[9] G. Huston. The State of BGP Routing. http://www.ietf.org/proceedings/01mar/slides/plenary-2/index.html.

[10] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of Link Failures in an IP Backbone. In *Proceedings of ACM IMW 2002*, October 2002.

[11] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proceedings of ACM Sigcomm*, August 2000.

[12] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental Study of Internet Stability and Wide-Area Network Failures. In *Proceedings of FTCS99*, June 1999.

[13] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja. The Impact of Internet Policy and Topology on Delayed Routing Convergence. In *Proceedings of the IEEE INFOCOM*, April 2001.

[14] R. Mahajan, D. Wetherall, and T. Anderson. Understanding bgp misconfiguration. In *Proceedings of ACM Sigcomm*, August 2002.

[15] G. Malkin. Routing Information Protocol Version 2. RFC 2453, SRI Network Information Center, November 1998.

[16] Z. Mao, R. Govindan, G. Varghese, and R. Katz. Route Flap Damping Exacerbates Internet Routing Convergence. In *Proceedings of ACM Sigcomm*, August 2002.

[17] P. M. Merlin and A. Segall. A failsafe distributed routing protocol. *IEEE Transactions on Communications*, 27:1280–7, 1979.

[18] J. Moy. OSPF Version 2. RFC 2328, SRI Network Information Center, September 1998.

[19] J. Moy, P. Padma, and A. Lindem. Hitless OSPF Restart. http://www.ietf.org/internet-drafts/draft-ietf-ospf-hitless-restart-04.txt, October 2002.

[20] V. Paxson. End-to-End Routing Behavior in the Inthernet. *IEEE/ACM Transactions on Communications*, 5(5):610–615, 1997.

[21] D. Pei, L. Wang, D. Massey, S. F. Wu, and L. Zhang. A study of packet delivery performance during routing convergence. In *IEEE DSN*, June 2003.

[22] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, F. S. Wu, and L. Zhang. Improving BGP Convergence Through Assertions Approach. In *Proceedings of the IEEE INFOCOM*, June 2002.

[23] Y. Rekhter and T. Li. Border Gateway Protocol 4. RFC 1771, SRI Network Information Center, July 1995.

[24] S. R. Sangli, Y. Rekhter, R. Fernando, J. Scudder, and E. Chen. Graceful Restart Mechanism for BGP. http://www.ietf.org/internet-drafts/draft-ietf-idr-restart-05.txt, October 2002.

[25] A. Shaikh, D. Dube, and A. Varma. Avoiding Instability during Shutdown of OSPF. In *Proceedings of the IEEE INFOCOM*, June 2002.

[26] A. U. Shankar, C. Alaettinoglu, K. Dussa-Zieger, and I. Matta. Transient and steady-state performance of routing protocols: Distance-vector versus link-state. *Journal of Internetworking: Research and Experience*, 6:59–87, 1995.

[27] A. Terzis, K. Nikoloudakis, L. Wang, and L. Zhang. IRL-Sim: A general purpose packet level network simulator. In *Proceedings of the 33rd ACM-SIAM Symposium on Discrete Algorithms*, April 2000.

[28] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang. Observation and Analysis of BGP Behavior under Stress. In *Proceedings of the ACM IMW 2002*, October 2002.

[29] Z. Wang and J. Crowcroft. Shortest path first with emergency exits. In *Proceedings of ACM Sigcomm*, pages 166–176, 1990.

[30] W. Zaumen and J. J. G.-L. Aceves. Dynamics of Distributed Shortest-Path Routing Algorithms. In *Proceedings of ACM Sigcomm*, August 1991.