Models for IP/MPLS Routing Performance: Convergence, Fast Reroute, and QoS Impact

Gagan L. Choudhury
AT&T Labs, Middletown, New Jersey, USA

Abstract

We show how to model the black-holing and looping of traffic during an Interior Gateway Protocol (IGP) convergence event at an IP network and how to significantly improve both the convergence time and packet loss duration through IGP parameter tuning and algorithmic improvement. We also explore some congestion avoidance and congestion control algorithms that can significantly improve stability of networks in the face of occasional massive control message storms. Specifically we show the positive impacts of prioritizing Hello and Acknowledgement packets and slowing down LSA generation and retransmission generation on detecting congestion in the network. For some types of video, voice signaling and circuit emulation applications it is necessary to reduce traffic loss durations following a convergence event to below 100 ms and we explore that using Fast Reroute algorithms based on Multiprotocol Label Switching Traffic Engineering (MPLS-TE) that effectively bypasses IGP convergence. We explore the scalability of primary and backup MPLS-TE tunnels where MPLS-TE domain is in the backbone-only or edge-to-edge. We also show how much extra backbone resource is needed to support Fast Reroute and how can that be reduced by taking advantage of Constrained Shortest Path (CSPF) routing of MPLS-TE and by reserving less than 100% of primary tunnel bandwidth during Fast Reroute.

Keywords: IGP Tuning, Micro-Forwarding Loops, Control Message Storm, Network Stability, MPLS-TE Fast Reroute

1. Introduction

IP networks are at the heart of today's Information Superhighway. Traditionally they were designed to carry mainly high volume best effort Internet traffic. In that environment the Interior Gateway Protocols (IGP) such as Open Shortest Path First (OSPF) [1] and Intermediate System to Intermediate System (IS-IS) [2] and Exterior gateway Protocols (EGP) such as the Border Gateway Protocol (BGP) [3] were designed with a slow responsiveness to changes so as to avoid any potential for network instabilities. The slow IGP and BGP made the networks stable but that also ensured that whenever there would be a failure or link cost change in the network, that would require up to tens of seconds of IGP convergence time and up to several minutes of BGP convergence time. During the period of convergence there would be traffic losses due to black-holing over a failed link or due to micro forwarding loops due to inconsistency of routing table information at various routers. The traffic loss period would usually last during a subset of the convergence period, but in the worst case may last throughout the convergence period. The high convergence time was usually tolerable for traditional Web applications, which did not require tight real time response. However, the new trend is to carry all traffic over a common IP network. In that environment many applications require mach faster recovery from traffic loss. Specifically, many interactive gaming applications cannot tolerate a loss duration greater than about 3-5 seconds. The voice Real Time Protocol (RTP) stream usually cannot tolerate a loss duration greater than about 2 seconds in order to avoid significant number of hang-ups. There are also other applications such as some types of video, voice signaling and circuit emulation where the traffic loss duration has to be 100 ms or smaller.

This paper has three main sections. Section 2 deals with improving IGP convergence times and traffic loss durations for applications that can tolerate up to a few seconds of traffic loss. We show through a simple example how black-holing at failed interfaces and micro-forwarding loops due to inconsistency in routing tables can result in traffic losses during the convergence period. We also use a model to analyze the impact of IGP parameter tunings in order to substantially improve the convergence time and traffic loss duration. Section 3 explores some congestion avoidance and congestion control algorithms that would significantly improve stability of networks in the face of occasional massive control message storms. Specifically it shows how prioritizing Hello and Acknowledgement packets and slowing down LSA generation and retransmission generation on detecting congestion in the network can improve network scalability and stability. Section 4 deals with Fast Reroute algorithms based on Multiprotocol

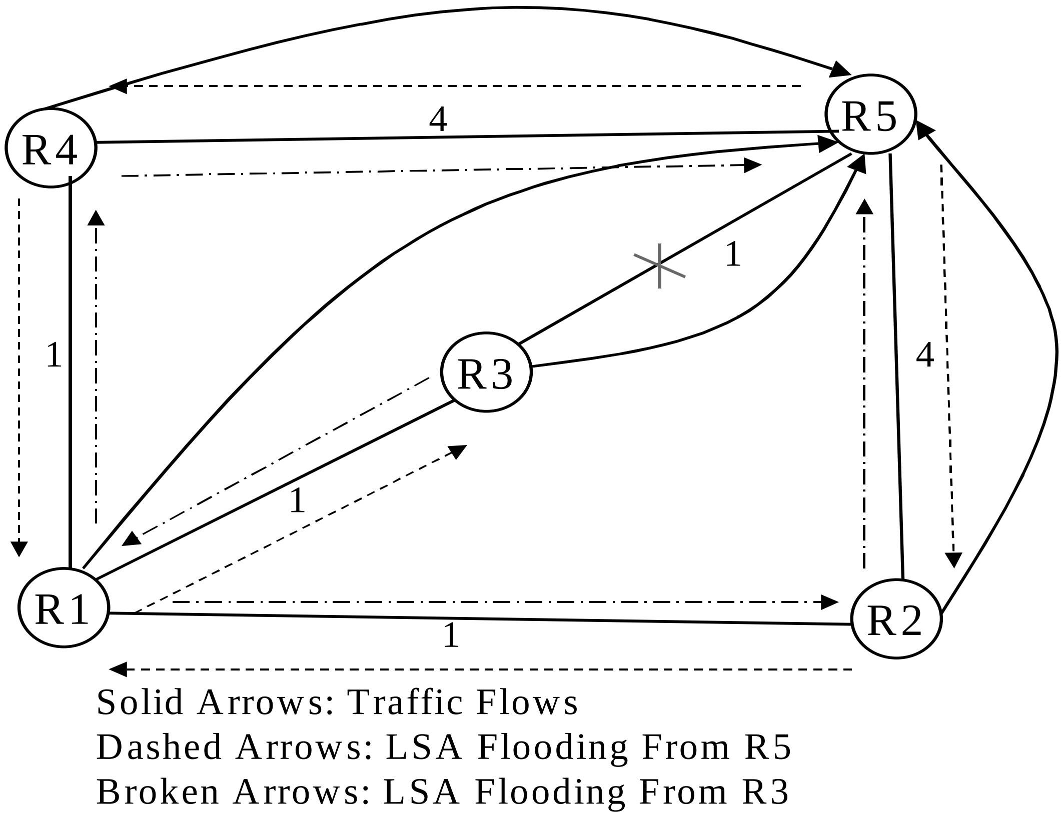
Performance, Quality of Service, and Control of Next-Generation Communication
Networks II, edited by Frank Huebner, Robert D. van der Mei, Proc. of SPIE Vol. 5598
(SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.580701

Label Switching Traffic Engineering (MPLS-TE) [10] that effectively bypass IGP convergence and allows traffic loss durations of 100 ms or smaller as required by some applications. We explore the scalability of primary and backup MPLS-TE tunnels where MPLS-TE domain is in the backbone-only or edge-to-edge. We also show how much extra backbone resource is needed to support Fast Reroute and how can that be reduced by taking advantage of Constrained Shortest Path (CSPF) routing of MPLS-TE and by reserving less than 100% of primary tunnel bandwidth during Fast Reroute.

2. IGP Convergence Time Improvement

We explain the IGP convergence time following a link failure event using a simple network example as shown in Figure 1.

Figure 1: IGP Convergence following a Link failure


R4
R5
4
1
R3
1
4
1
R1
R2
1
Solid Arrows: Traffic Flows
Dashed Arrows: LSA Flooding From R5
Broken Arrows: LSA Flooding From R3

The network has 5 Routers and 6 links and has the following four traffic flows: R1-to-R5, R2-to-R5, R3-to-R5, and R4-to-R5, i.e., all flows have R5 as destination. The numbers next to each link represent IGP link cost and all traffic is routed on shortest path using these costs additively. So all traffic flows use the R3-to-R5 link as the final link on their paths. Suppose at time T=0 the R3-to-R5 link fails. We assume that all links are POS (Packet over Sonet) and at time T=50 ms the two end-point Routers R3 and R5 detect the failure by receiving the physical layer Loss of Signal (LOS) message. If the physical layer signal is not received or IGP cannot act on it then loss detection may take up to the Router Dead Interval (up to 30 or 40 seconds) but we do not consider that case. Typically the routers will wait for a Carrier Delay of 2 seconds (to avoid false alarms or damp link flapping) before letting the IGP act on it. Each of the routers R3 and R5 will flood a Link-State-Advertisement message (LSA), usually following an LSA_Origination delay, to indicate the failure to the other routers (LSA is an OSPF term and we will stick to that). The other routers, R1, R2 and R4, will detect the link failure as soon as they get the LSA either from R3 or from R5. Following a failure detection (either directly or through LSA from neighbor) a router would start Shortest Path First (SPF) calculation after waiting a period of SPF_Delay since the failure detection by

IGP and also a period of SPF_Hold since the previous SPF calculation, whichever wait is longer. Following the SPF calculation, each router will install new routes to its forwarding tables bypassing the failed link. The route installation time has a fixed part and a variable part proportional to the number of prefixes affected. If indirect addressing is used with many prefixes combined into route aggregates then the variable time will be proportional to the number of route aggregates affected. For the purpose of our simple example let's assume that for all routers the physical layer detection time is 0.1 seconds, Carrier delay is 2 seconds, LSA origination delay is 0.5 seconds, SPF_Delay is 0 and SPF_Hold is 5 seconds (the time from failure detection by IGP to the start of SPF computation would be anywhere between 0 and 5 seconds), SPF calculation time is 0.2 seconds, and Time to install new routes to forwarding table is 0.5 seconds. Next we estimate the convergence time at the various routers.

Convergence at Router R3: 1) Failure detected at physical layer at T=0.1 seconds, 2) Failure detected by IGP at T=2.1 seconds, 3) LSA flooding to other routers starts at T=2.6 seconds, 4) SPF calculation starts at T=6.1 seconds (random wait due to SPF_Hold following IGP failure detection assumed to be 4 seconds). 5) SPF calculation ends at T=6.3 seconds, 6) Routes installed at forwarding table, i.e., convergence completed at T=6.8 seconds.

Convergence at Router R5: 1) Failure detected at physical layer at T=0.1 seconds, 2) Failure detected by IGP at T=2.1 seconds, 3) LSA flooding to other routers starts at T=2.6 seconds, 4) SPF calculation starts at T=5.1 seconds (random wait due to SPF_Hold following IGP failure detection assumed to be 3 seconds). 5) SPF calculation ends at T=5.3 seconds, 6) Routes installed at forwarding table, i.e., convergence completed at T=5.8 seconds.

Convergence at Router R1: 1) Failure detected by IGP at T=2.7 seconds based on receiving LSA originated from R3 and processing it (a second LSA originated from R5 would be received later but that would be redundant), 2) LSA re-flooding to other routers started immediately at T=2.7 seconds, 3) SPF calculation starts at T=7.2 seconds (random wait due to SPF_Hold following IGP failure detection assumed to be 4.5 seconds). 4) SPF calculation ends at T=7.4 seconds, 5) Routes installed at forwarding table, i.e., convergence completed at T=7.9 seconds.

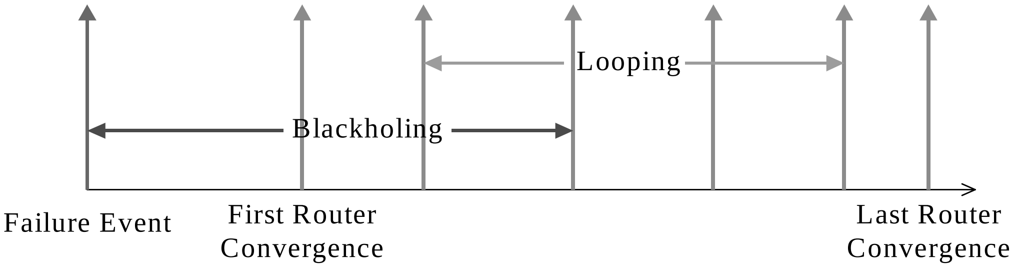
Convergence at Router R2: 1) Failure detected by IGP at T=2.7 seconds based on receiving LSA originated from R5 and processing it (a second LSA originated from R3 would be received later but that would be redundant), 2) LSA re-flooding to other routers started immediately at T=2.7 seconds, 3) SPF calculation starts at T=4.7 seconds (random wait due to SPF_Hold following IGP failure detection assumed to be 2 seconds). 4) SPF calculation ends at T=4.9 seconds, 5) Routes installed at forwarding table, i.e., convergence completed at T=5.4 seconds.

Convergence at Router R4: 1) Failure detected by IGP at T=2.8 seconds based on receiving LSA originated from R3 (flooded later by R1) and processing it (a second LSA originated from R5 would be received later but that would be redundant), 2) LSA re-flooding to other routers started immediately at T=2.8 seconds, 3) SPF calculation starts at T=7.8 seconds (random wait due to SPF_Hold following IGP failure detection assumed to be 5 seconds). 4) SPF calculation ends at T=8.0 seconds, 5) Routes installed at forwarding table, i.e., convergence completed at T=8.5 seconds.

Chronology of Events: At time T=0 the link fails and all traffic flows starts black-holing at the R3-to-R5 link. At time T=5.4 Seconds, router R2 converges and uses the R2-to-R5 route and traffic loss stops for the R2-to-R5 flow.
At time T=5.8 Seconds, router R5 converges but this event has no impact on the status of various flows since R5 is the destination of all flows, not the source or transit point. At time T=6.8 Seconds, router R3 converges and black-holing of all traffic ends. However, R1 has not converged yet and so a micro-forwarding loop starts between R1 and R3 for the three flows starting at R1, R3 and R4. At time T=7.9 Seconds, router R1 converges. From R1 to R5, there are two equal cost multi-paths and so half the traffic would take the R1-to-R2-to-R5 route and the other half would take the R1-to-R4-to-R5 route based on a hash function of source and destination addresses. The first route would have no traffic loss since R2 has converged as well. However, the second route would cause a micro-forwarding loop between R1 and R4 since R4 has not converged yet. So at this time half the traffic of the remaining three flows would go through and the remaining half would see loss due to looping. At time T=8.5 seconds the router R4 converges. At this time the routing tables at all routers would be consistent and all traffic losses would end assuming that the remaining network still has enough capacity to carry all the flows.
In general there would be a period of black-holing followed by a period of looping after a failure event (the looping may or may not happen). The looping happens only during the period when a subset of routers has converged and

another subset has not converged and so there is inconsistency in routing information available at various routers. Besides the link failures shown earlier, there may be router failure or shared risk link group (SRLG) failures which implies that failure of a single transport layer facility may take down multiple logical links at layer 3 which all ride over the single shared facility. Also, in many cases a link or all links out of a router are costed-out (IGP cost made very high) to divert traffic away in anticipation of a maintenance event. In such a situation no black-holing happens but there is still a period when a subset of routers have raised the IGP cost and another subset have not and so looping can still happen. At a later point of time the link is costed-in, i.e., the IGP cost set back to its original value. Here again black-holing would not happen but looping may happen.

Figure 2: Black-Holing and Looping Following a Failure Event



We have developed a generic model for an IP network that can simulate any combination of failure or cost-out/cost-in events. In addition to the simple model described above it also simulates other details of IGP processing. Also, at each router it allows for processing of tasks other than IGP processing. At first it does the failure detection, LSA propagation, SPF computation and route installation at all the routers in parallel (for each of the failure events) and based on that determines the partial convergence instants and full convergence instants at all routers. Next it computes routing table at each router and whenever a router has a state change (a partial or full convergence happens) that is reflected in its routing table. Next it looks at all network events starting from the first failure or cost-out/cost-in event until convergence happens at each router. Initially it routes all the flows in the network. Next, whenever an event happens it again routes all the flows and determines how much traffic, if any, is black-holed or loops. For each flow it determines all loss durations, either due to black-holing or looping. The loss durations may or may not be contiguous. It reports the maximum loss duration and total loss duration for each flow.

We applied the model to an IP backbone network with 36 routers and 49 links. We show the convergence time and traffic loss duration results under two conditions: Untuned IGP, and Tuned IGP.

Untuned IGP: Carrier delay = 2 seconds. SPF_Delay = 5 seconds. SPF_Hold = 10 seconds. LSA_Origination Delay = 500 ms. MinLSInterval (Minimum Time between successive origination of the same LSA) = 5 seconds. MinLSArrival = 1 second. Maximum time a process can hold on to Router CPU = 200 ms. Time between a Router failure and link failure detection at all neighbors is random between 0 and 1 second. Time to install routes to forwarding table following a failure is random between 0.5 and 15.5 seconds (upper end with a change in next hop for all prefixes and lower end with a change in next hop for just one prefix).

Tuned IGP: Carrier delay = 100 ms. Instead of using static high values for SPF_Delay/Hold use an adaptive binary exponential backoff mechanism with low starting value for delaying SPF computation. For the first SPF computation use SPF_Delay = 50 ms and each time a new SPF_Delay computation is needed multiply this value by 2 until it reaches 5 seconds. If no SPF computation is needed for 5 seconds then go back to the starting value of 50 ms. LSA_Origination Delay = 10 ms. MinLSInterval = 1 second to start with and then use a binary exponential backoff mechanism until it reaches at least 5 seconds. MinLSArrival = 200 ms. Reduce the maximum time a process can hold on to Router CPU to 50 ms. As soon as a router fails, immediately send a message to all line cards so that the time between a Router failure and link failure detection at all neighbors is no longer than 50 ms. Use address indirection so that the time to install routes to forwarding table following a failure does not exceed 1.5 seconds even when all prefixes change the next hop.

Table 1: IGP Convergence Time and Traffic Loss Duration

Event Type	Convergence Time Range (Seconds)		Traffic Loss Duration Range (Seconds)	
	Untuned IGP	Tuned IGP	Untuned IGP	Tuned IGP
Single Link Failure	3.5 - 14.2	0.8 - 2.0	3.6 - 8.3	1.2 - 2.0
Two Parallel Link Failures	3.5 - 24	0.8 - 3.0	4.9 - 21.6	1.8 - 3.0
Single Router Failure	3.5 - 29.2	0.8 - 4.1	4.2 - 22.5	0.8 - 3.6
Link Cost-Out	1.4 - 8.9	0.8 - 4.0	0.03 - 3.6	0 - 0.7
Link Cost-In	1.4 - 8.9	0.7 - 4.1	0.02 - 4.0	0 - 0.38

Table 1 above shows that the IGP convergence time and traffic loss duration is significantly improved with IGP tuning but in most cases it is still well over 1 second. To reduce traffic loss duration below 1 second we have to use a fast reroute technique to be described in Section 4.

3. Congestion Avoidance and Congestion Control Algorithms

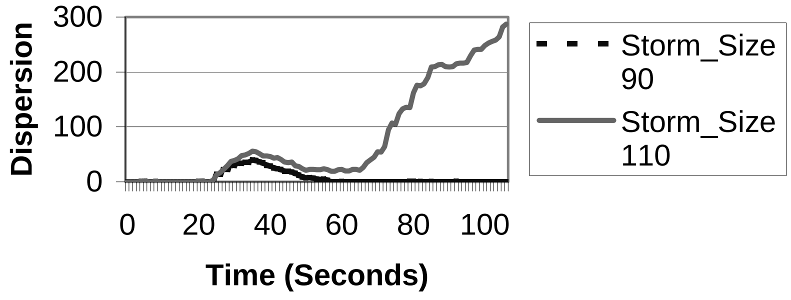
A large network (running OSPF or IS-IS, although we will use OSPF terminology) may occasionally experience the simultaneous or near-simultaneous update of a large number of LSAs. This is particularly true if OSPF traffic engineering extension [4] is used which may significantly increase the number of LSAs in the network. We call this event an LSA storm and it may be initiated by an unscheduled failure or a scheduled maintenance event. The failure may be hardware (Link/Router), software (e.g., a new software release requiring a refresh of the entire LSA database), or procedural (e.g., erroneous import of a large number of external routes to IGP) in nature. The LSA storm causes high CPU and memory utilization at the router processor causing incoming packets to be delayed or dropped. Delayed acknowledgments (beyond the retransmission timer value) results in retransmissions, and delayed Hello packets (beyond the Router-Dead interval) results in links being declared down. A trunk-down event causes Router LSA origination by its end-point routers. If traffic engineering LSAs are used for each link then that type of LSAs would also be originated by the end-point routers and potentially elsewhere as well due to significant changes in reserved bandwidths at other links caused by the failure and reroute of Label Switched Paths (LSPs) originally using the failed trunk. Eventually, when the link recovers that would also trigger additional Router LSAs and traffic engineering LSAs. The retransmissions and additional LSA originations result in further CPU and memory usage, essentially causing a positive feedback loop. We define the LSA storm size as the number of LSAs in the original storm and not counting any additional LSAs resulting from the feedback loop described above. If the LSA storm is too large then the positive feedback loop mentioned above may be large enough to indefinitely sustain a large CPU and memory utilization at many routers in the network, thereby driving the network to an unstable state. In the past, network outage events have been reported in IP and ATM networks using link-state protocols such as OSPF, IS-IS, PNNI or some proprietary variants. In many of these examples, large scale flooding of LSAs or other similar control messages (either naturally or triggered by some bug or inappropriate procedure) have been partly or fully responsible for network instability and outage.

We note that the original reason for the slow responsiveness of the IGP and BGP protocols was to avoid message storm and network instability. However, we noted in Section 2 that in order to meet the fast convergence demands of many applications it is important to speed up the responsiveness of the protocols. Two such examples mentioned in Section 2 are faster LSA generation (reduce MinLSInterval) and faster SPF computation (reduce SPF_Delay/Hold). If we do this naively then we would actually increase the risk of network stability. To avoid that in both cases we used an exponential backoff algorithm so that on the detection of the first failure the IGP would respond very fast, but it will gradually slow down with the detection of each subsequent failure event until it gets as slow as the original untuned IGP design. This technique helps, but in this Section we quantify the network stability issue and consider a few other techniques for improving the stability. The work here builds on ongoing work at the IETF [5] and ATM Forum [6,7] as well as a previous technical report on this subject [8].

We consider a 100-Router, 1200 link flat network with maximum adjacency at a node of 50. The network was generated randomly over a rectangular grid, resembling the continental USA, using a modified version of Waxman's algorithm [9] that ensures that every node in the network is at least doubly connected. The various protocol timers are assumed to be as follows: the LSA refresh interval is 1800 seconds, the Hello refresh interval is 10 seconds, the Router-Dead Interval is 40 seconds, the LSA retransmission interval is 10 seconds, the minimum time between successive origination of the same LSA is 5 seconds, and the minimum time between successive shortest-path-first (SPF) calculations is 1 second. LSA and Hello processing times are assumed to be about 1 ms and the processing time for an LSA Acknowledgement is assumed to be about 0.25 ms. In general multiple LSAs may be packed in a single packet but we do not assume any such packing. The total queue size to store LSAs and Acknowledgements is assumed to be 2000 messages and new messages are dropped when the queue is full. An LSA storm is generated about 20 seconds after the start of the simulation by failing a number of links. The failed links are so chosen that they do not disconnect the network. The simulation is repeated with different sizes of LSA storms and by employing different congestion avoidance and control mechanisms. A key measure of system stability is the quantity, "dispersion", which is the number of LSAs that have been generated but not processed in at least one node at a given point in time. The dispersion count shoots up following the LSA storm. In a stable system, it should come down after a period of time but in an unstable system it may stay high indefinitely.

In Figure 3, where no priority is given to Hello and Acknowledgements, the dispersion versus time charts show that the system is stable with an LSA storm of size 90 but tends to show unstable behavior at a storm size of 110. Figure 4 also confirms this behavior, where the CPU utilization at the busiest router is plotted over time. The CPU utilization hits 100% soon after the LSA storm. With an LSA storm of size 90 the CPU utilization eventually comes down, but it stays at 100% indefinitely with an LSA storm of size 110. The unstable behavior is sustained due to many retransmissions and links being declared down due to the expiry of the Router-Dead Interval. Many new LSAs are generated as a result of the link going down and eventually coming back up as well as due to changes in bandwidth at many links resulting from the rerouting of traffic in the network.

Figure 3. Dispersion Over Time: No Priority to Hello or Ack



In Figure 5, the Hello packets are given higher priority compared to LSAs and Acknowledgements. As a result, the Router-Dead Interval never expires and links are not declared down even if a node is under severe congestion. The system stays stable beyond the storm size of 110. However, at a storm of size 140 the dispersion stays indefinitely high due to too many retransmissions. But, no links are declared down and the dispersion does not reach the very high level as observed in Figure 3.

Figure 4. CPU Utilization at Busiest Router: No Priority to Hello/Ack

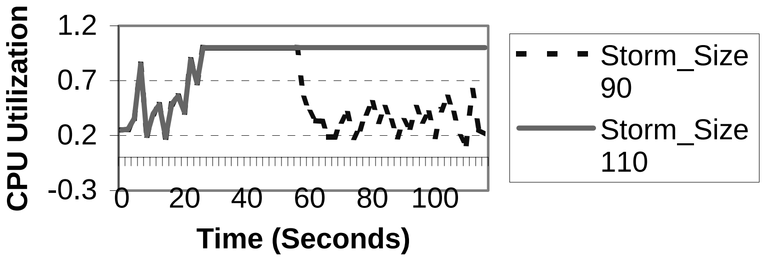
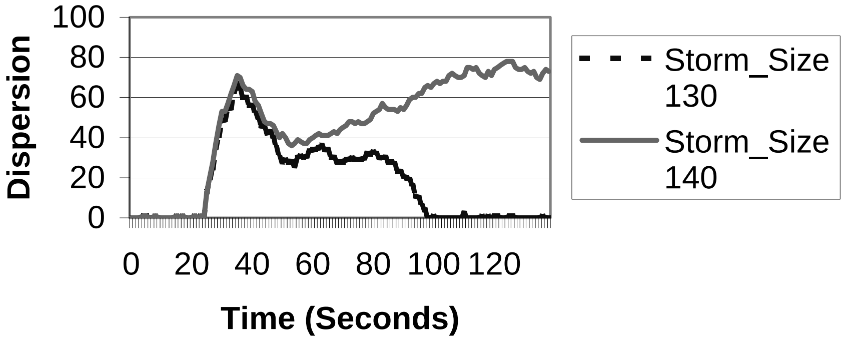


Figure 5. Dispersion Over Time: Priority to Hello only



In Figure 6, both the Hello and Acknowledgement packets are treated at a higher priority compared to LSAs. As a result, in addition to the Router-Dead Interval not expiring, the retransmission timers also expire less frequently resulting in fewer retransmissions. We see that in this case the system stays stable at a much higher storm size compared to those in Figure 3 and Figure 5. However, there is no slow-down of LSA generation and retransmission generation and eventually at a storm size of around 260 the system again shows a sustained congestion behavior.

Figure 6. Dispersion Over Time: Priority to Hello and Ack

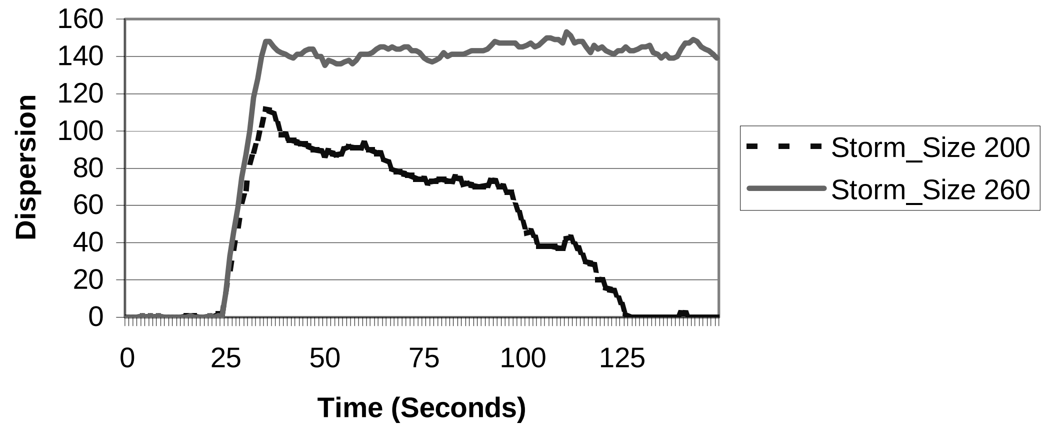


Figure 7. Dispersion Over Time: Priority to Hello/Ack + Congestion Control

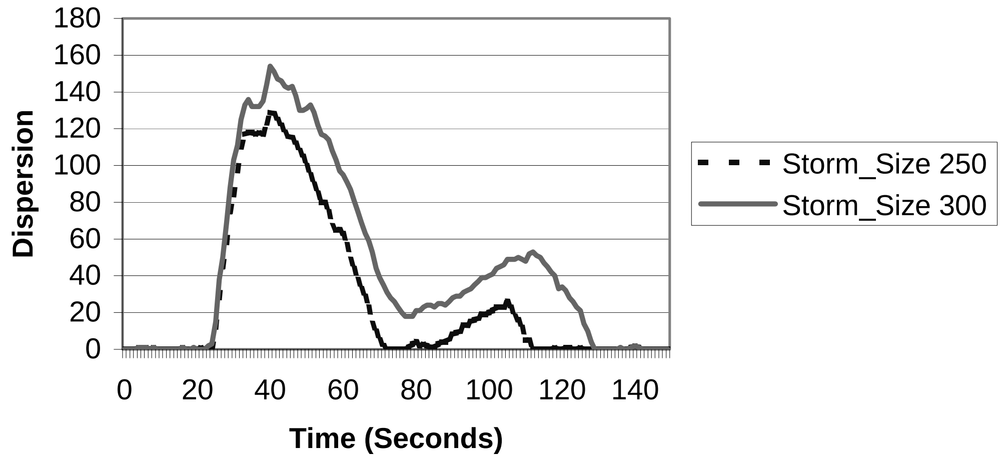


Figure 8. Congestion State at Busiest Router Over Time:
(1 = no cong, 2 = low cong, 3 = high cong.)

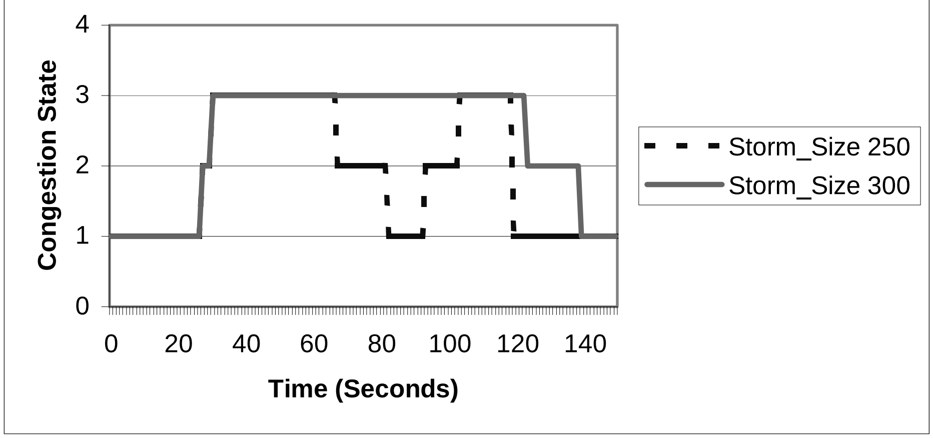


Figure 7 employs a congestion detection and control mechanism in addition to prioritizing Hello and LSA Acknowledgement packets. Three congestion levels are used, "no congestion", "low congestion" and "high congestion". Each router determines a "local" congestion state based on its outstanding work queue size. If the queue size is below one-third the total queue size (2000 messages in the current example) then the local state is "no congestion", if it is at or above one-third but below two-thirds the total queue size then local state is "low congestion", and if it is at or above two-thirds the total queue size then local state is "high congestion." The local congestion state is sent to all neighbors using two bits in the Hello messages. Each router determines its "overall" congestion level as the maximum of its "local" congestion state and those received from all its neighbors based on received Hello messages. The congestion state at a router is determined based on its overall congestion level but also using some restriction on the frequency of state transition. Specifically, a router can enter a less congested state only after it has stayed in the previous state for at least 15 seconds, but there is no such waiting period for transitioning to a more congested state. This allows congestion control action to take place immediately after detecting congestion but also prevents frequent oscillation among states. In a "no congestion" state, a router uses default values of all protocol parameters as described previously. In the "low congestion" state both the retransmission timer and the timer controlling origination of successive LSAs are doubled. In "high congestion" state the above two timers are doubled again.

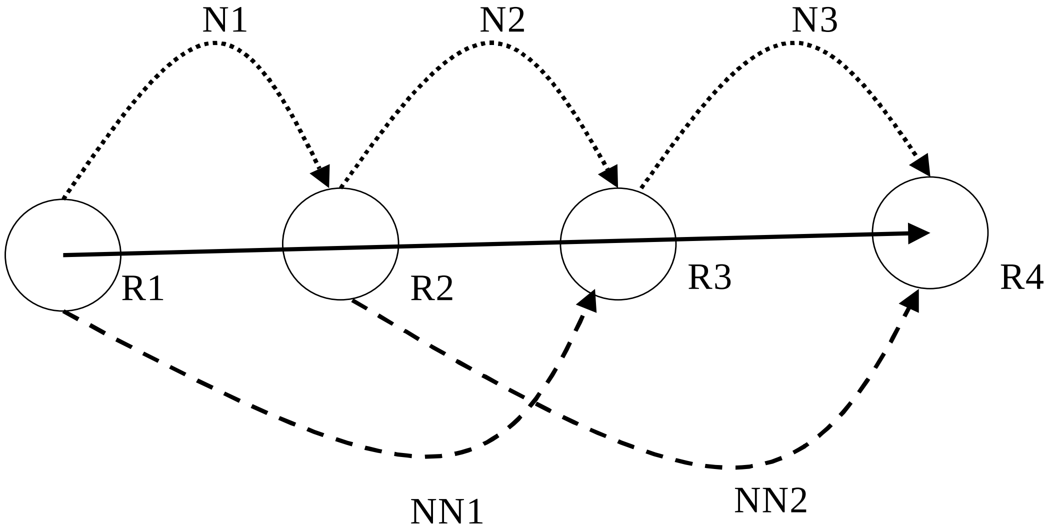
Figure 7 shows that with the addition of the congestion control mechanism, the system stays stable at a significantly higher LSA storm level. Figure 8 plots the congestion state at the busiest router as a function of time. Initially the router is in "no congestion". Soon after the storm the router (also its neighbors, although not shown in the picture) enters a "high congestion" state. This period significantly slows down retransmissions and the generation of new LSAs, which helps in relieving congestion. After a while the node (also its neighbors) comes out of the congestion state. There may be some further excursions to higher congestion states but eventually all congestion goes away and the network gets back to its normal state of "no congestion." At a higher storm size the router stays in the "high congestion" state longer but in both cases recovery from congestion happens

4. Fast Reroute to Bypass IGP Convergence

In order to reduce the traffic loss duration to around 100 ms or lower, it is necessary to pre-configure a backup path and switch over to it as soon as a failure is detected at the physical layer (typical detection time is around 50 ms and detection plus switchover to backup path is within 100 ms). After IGP convergence is complete it is necessary to switch again to a path determined by IGP (typically without any further traffic loss) and also recalculate new backup paths to be ready for further failures. This technique is known as Fast Reroute (FRR). The most common forms of FRR use "Link-Bypass" paths and "Router-Bypass" paths, the former protecting only against link failures and the latter protecting against both link and router failures. However, other FRR protections are also possible. For example if multiple Layer 3 links (between different pairs of Routers) ride over a common physical facility then they form a Shared Risk Link Group (SRLG) and it may be possible to use "SRLG-bypass" FRR paths to protect against failure of the physical facility. In this paper we only consider FRR techniques using MPLS-TE [10, 11] but it is also possible to have FRR without using MPLS-TE [12, 13].

Figure 9 shows a primary MPLS-TE tunnel and the associated backup tunnels for Fast reroute.

Figure 9. Primary and Backup MPLS-TE tunnels



The solid line shows the primary tunnel going through four routers R1, R2, R3 and R4. The three top backup tunnels N1, N2 and N3 are next-hop link bypass tunnels. The two bottom backup tunnels, NN1 and NN2 are next-next-hop router bypass tunnels. If the link from R2 to R3 fails then it can be protected either by the link-bypass tunnel N2 or by the router-bypass tunnel NN2. If Router R3 fails however, it can only be protected by NN2 and not by any of the link-bypass tunnels. For the last leg of the primary tunnel, however, only protection option available is the link-bypass tunnel N3. If either the source R1 or destination R4 of the primary tunnel fails then there is no protection. The backup tunnels shown above are one-to-one and if there are P primary tunnels and R Routers along the path of a primary tunnel on the average then there would be P*(R-1) backup tunnels. A second possibility is to have facility backup tunnels where all tunnels going over one direction of a single link can be protected by a single next-hop tunnel and/or (N-1) next-next-hop tunnels where N is the number of neighbors of the router on the other end of the link. The number of facility backup tunnels needed is usually much less than the number of one-to-one backup tunnels.

The two main issues with MPLS-TE FRR are the scalability and the amount of additional bandwidth needed in the network for the protection and we address them next. We consider an IP network with 30 backbone or Provider (P) routers and 63 Access or Provider Edge (PE) routers. Each PE is dual homed to two different P routers (total of 126 links). In addition, there are 54 links interconnecting the backbone or P routers. We consider three cases. In Case 1 MPLS-TE FRR is done in the backbone part only. So FRR protection is possible only for backbone link failures and for transit traffic for backbone router failures. In Case 2 MPLS-TE FRR is done separately in the backbone and access. So FRR protection would also be possible for access link failures. In Case 3 MPLS-TE is done edge-to-edge so that FRR protection would be possible for all failures except for the failure of the PE routers. Table 2 quantifies the number of primary and facility backup tunnels needed under various scenarios. It also looks at the most congested router and shows for how many tunnels it is an end-point and mid-point respectively. Every router on a tunnel exchanges an RSVP Path message and an RSVP Reservation Request message to its neighbors every 30 seconds (1 neighbor of an end-point router and two neighbors of a mid-point router). We show CPU utilization at the most congested router assuming per-message processing times of 1 ms and 100 microseconds respectively. As expected, the number of tunnels and the CPU utilization at the most congested router increases as we move from "link-bypass only" to "Link + router bypass" and also as we move from MPLS-TE in "backbone-only" to "separately in backbone and access" to "edge-to-edge". The number of backup tunnels and the CPU utilization at the routers would go up significantly with one-to-one backup tunnels. However, we also note that the CPU utilization may also be reduced significantly by using the refresh overhead reduction mechanism of RFC 2961 [14].

Table 2. Scalability of MPLS-TE Tunnels in a Network with 30 backbone routers and 63 access routers (Facility Backup)

MPLS-TE Domain	Type of FRR Tunnels	Total Number of Tunnels		View at Most Congested Router			
		Primary	Backup	# of Tunnel End Points	# of Tunnel Mid Points	CPU Utilization Assuming	
						1 ms per message	100 microsec per message
Backbone Only	Link Bypass only	870	108	72	198	3.1%	0.3%
	Link + Router Bypass		905	139	339	5.5%	0.5%
Separately in Backbone and Access	Link Bypass Only	1122	360	34	269	3.8%	0.4%
	Link + Router Bypass		688	72	325	4.8%	0.5%
Edge-to-Edge	Link Bypass Only	3906	360	24	1097	14.8%	1.5%
	Link + Router Bypass		2256	130	1463	20.4%	2.0%

Next, using the same network and also using a certain point-to-point traffic matrix we estimate how much backbone resource would be needed (in terms of the total number of equivalent OC-48 links and total OC-48 miles) under various conditions. This is shown in Table 3. MPLS-TE is used either in the backbone-only or edge-to-edge. In terms of traffic engineering we consider two cases. In one case all tunnels are assigned zero bandwidth so that traffic always follow the shortest path. However, the network design ensures that there is enough capacity on the links to cover normal routing and routing under all single link and single router failure cases (both IGP reroute and Fast Reroute). In the other case the tunnels are assigned real bandwidths and a constrained shortest path first (CSPF) algorithm is used (we may start with the best estimate of tunnel bandwidth and later adjust it based on measurements). This case is more efficient since spare capacity on a non-shortest path can be utilized. The backup tunnels are designed to protect either 100% of primary tunnel bandwidth or only 60% of primary tunnel bandwidth with the expectation that we only need to protect high priority traffic which forms less than 60% of primary tunnel bandwidth. No bandwidth reservation is made for the backup tunnels but they are routed such that they will have enough bandwidth on their path if needed. Next each possible single failure scenario is considered and enough link capacity is kept in the network design such that in all cases the backup tunnels would have at least as much bandwidth as they are protecting.

The main observations on Table 3 are as follows. At first we concentrate on the case of no traffic Engineering. We observe that the edge-to-edge case requires substantially more resource to support Fast Reroute compared to the backbone-only case. This is because the edge-to-edge case provides protection in many more cases (e.g., access link failures and complete protection for backbone router failures) as mentioned earlier. However, protecting only 60% of primary bandwidth can significantly reduce the extra resource need. Next looking at the case of no traffic engineering we see that there is substantial resource saving by using MPLS-TE, which can use non-shortest path routing following failures. Saving is more with Edge-to-Edge case since there we have many more smaller-sized tunnels compared to the fewer and fatter tunnels in the backbone-only case.

Table 3. Backbone Resource Need Estimation With MPLS-TE and FRR

MPLS-TE Domain	Backbone Resource Need	No Use of Traffic Engineering, Zero BW Tunnels, SPF Routing			Traffic Engineering Used, Tunnels With BW, CSPF Routing		
		Only IGP Reroute	IGP + Fast Reroute With FRR Protection At		Only IGP Reroute	IGP + Fast Reroute With FRR Protection At	
			100%	60%		100%	60%
Backbone Only	Total OC-48s	721	739	721	561	574	562
	Total OC-48 Miles	498,181	512,966	498,181	393,773	402,120	393,967
Edge-To-Edge	Total OC-48s	721	870	734	508	625	527
	Total OC-48 Miles	498,181	600,186	507,300	372,347	450,613	387,320

5. Acknowledgements

The author would like to acknowledge collaboration and helpful discussions with many colleagues including Jerry Ash, Rich Kwapniewski, Ragu Raghuram, and Vera Sapozhnikova.

6. References

1. J. Moy, “OSPF Version 2,” IETF RFC 2328, April, 1998.
2. R. Callon, “Use of OSI IS-IS for Routing in TCP/IP and Dual Environments,” IETF RFC 1195, December, 1990.
3. Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP 4)," IETF RFC 1771, March, 1995.
4. D. Katz, K. Kompella and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2," IETF RFC 3630, September 2003.
5. G. Choudhury, Editor, "Prioritized Treatment of Specific OSPF Packets and Congestion Avoidance," Internet Draft, work in progress.
6. af-cs-0200.000, “PNNI Routing Congestion Control, Version 1.0,” June 2004.
7. af-cs-0201.000, “PNNI Routing Resynchronization Control, Version 1.0,” June 2004.
8. Choudhury, G., Ash, G., Manral, V., Maunder, A., Sapozhnikova, V., "Prioritized Treatment of Specific OSPF Packets and Congestion Avoidance: Algorithms and Simulations," AT&T Technical Report, August 2003.
9. Waxman, B.M., "Routing of Multipoint Connections," IEEE Journal on Selected Areas in Communications, 6(9): 1617-1622, 1988.
10. J. Boyle et al, “Applicability Statement for Traffic Engineering with MPLS,” IETF RFC 3346, August, 2002.
11. P. Pan, G. Swallow, A. Atlas, “Fast Reroute Extensions to RSVP-TE for LSP Tunnels,” Work in Progress in IETF.
12. A. Atlas, R. Torvi, G. Choudhury, C. Martin, B. Imhoff, D. Fedyk, “IP/LDP Local Protection,” Work in Progress in IETF.
13. S. Bryant, C. Filsfils, S. Previdi, M. Shand, “IP fast Reroute Using Tunnels,” Work in Progress in IETF.
14. L. Berger et. al. ,"RSVP Refresh Overhead Reduction Extensions," IETF RFC 2961, April, 2001.