Internet Engineering Task Force
Internet Draft
Expires in August, 2001
draft-ietf-ospf-scalability-00.txt

A. S. Maunder
Cisco Systems

G. Choudhury
AT&T Labs
March, 2001

Explicit Marking and Prioritized Treatment of Specific IGP Packets for Faster IGP Convergence and Improved Network Scalability and Stability

<draft-ietf-ospf-scalability-00.txt>

Status of this Memo

Copyright Notice

Abstract

There has been a lot of interest in the networking community to allow for fast failure detection followed by the fast restoration and recovery. It may be possible to provide fast recovery using special mechanisms; however, there is a strong interest in addressing this issue at a more fundamental level i.e. at IGP convergence because it addresses the problem at a much broader scale. Faster IGP convergence inevitably requires faster detection by using smaller hello interval timers (unless one relies on link level detection which is not always possible), fast flooding and more frequent SPF calculations. However, we provide analytic and simulation results[1] to show that this compromises the scalability

[1] The study was done when Anurag S. Maunder was a Sr. Member of Techical Staff at AT&T.

and stability of the network, mainly because Hello packets received at a router are indistinguishable from other packets and may experience long queueing delays during a sudden burst of many LSA updates. In this draft we suggest a need for Hello and potentially some other IGP packets to be marked explicitly so that efficient implementations can detect and act upon these messages in a priority fashion thus allowing significant reduction in convergence time for IGP while maintaining network stability.

The figures and graphs are missing from the ASCII version of the draft. The pdf versions of this draft can be found in the Internet-Drafts repository.

1 Motivation

The motivation of this draft is to address two key issues:
(1) Fast restoration under failure conditions
(2) Increased network scalability and stability

The motivation for allowing fast restoration under failure conditions is similar to the one provided in [1]draft-alaettinoglu-isis-convergence-00.txt. The theoretical limit for link-state routing protocols to re-route is in link propagation time scales, i.e., in tens of milliseconds. However, in practice it takes seconds to tens of seconds to detect the link failure and disseminate this information to the network followed by the convergence on the new set of paths. This is an inordinately long period of transient time for mission critical traffic destined to the non-reachable nodes of the network. One component of the long re-route time is the link failure detection time of between 20 and 30 seconds through three missed Hello packets with the typical hello interval of 10 seconds (between 30 and 40 seconds if missed hello threshold is 4). This component would be much shorter in the presence of link level detection, but as pointed out in [1]draft-alaettinoglu-isis-convergence-00.txt it does not work in some cases. For example, a device driver may detect the link level failure but fail to notify it to the IGP level. Also, if a router fails behind a switch in a switched environment then even though the switch gets the link level notification it cannot communicate that to other routers. Therefore for faster reliable detection at the IGP level, one has to reduce the hello interval. Reference [1]draft-alaettinoglu-isis-convergence-00.txt suggests that this be reduced to below a second, perhaps even to tens of milliseconds. A second component of the long re-route time is delayed SPF (shortest-path-first) computation. The typical delay value is 5 seconds but needs to be reduced significantly to have sub-second rerouting.

The second issue we address is the ability of a network to withstand the simultaneous or near-simultaneous update of a large number of link-state- advertisement messages, or LSAs. We call this event, an LSA storm. An LSA storm may be generated due to many reasons. Here are some examples: (a) one or more link failures due to fiber cuts, (b) one or more node failures for some reason, e.g., failed power supply in an office, (c) requirement of taking down and later bringing back many nodes during a software/hardware upgrade, (d) near-synchronization of the once-in-30-minutes refresh instants of some types of LSAs, (e) refresh of all LSAs in the system during a change in software version. The LSA storm tends to drive the node CPU utilization to 100% for a period of time and the duration of this period increases with the size of the storm and the node adjacency, i.e., the number of trunks connected to it. During this period the Hello packets received at the node would see high delays and if this delay exceeds typically three or four hello intervals (typically 30 or 40 seconds) then the associated trunk would be declared down. Depending on the implementation, there may be other impacts of a long CPU busy period as well. For example, in a reliable node architecture with an active and a standby processor, a processor-switch may result during an extended CPU-busy period which may mean that all the adjacencies would be lost and need to be re-established. Both of the above events would cause more database synchronization with neighbors and network-wide LSA flooding which in turn might cause extended CPU-busy periods at other nodes. This may cause unstable behavior in the network for an extended period of time and potentially a meltdown in the extreme case. Due to world-wide increased traffic demand, data networks are ever increasing in size. As the network size grows, a bigger LSA storm and a higher adjacency at certain nodes would be more likely and so would increase the probability of unstable behavior. One way to address the scalability issue is to divide the network hierarchically into different areas so that flooding of LSAs remains localized within areas. However, this approach increases the network management and design complexity and less optimal routing between areas. Also area 0 may see the flooding of a large number of summary LSAs and some of the new protocols may not work well under the hierarchical system. Thus it is important to allow the network to grow towards as large a size as possible under a single area. The undesirable impact of large LSA storms is understood in the networking community and it is well known that large scale flooding of control messages (either naturally or due to a bug) has been responsible for several network events in the past causing a meltdown or a near-meltdown. Recently, proposals have been submitted to avoid synchronization of LSA refreshes [2]draft-ietf-ospf-refresh-guide-01.txt and reduce flooding overhead in case more than one interface goes to the same neighbor [3] draft-ietf-ospf-isis-flood-opt-00.txt, and [4]draft-ietf-ospf-ppp-flood-00.txt.

In this proposal we would like to make the point that reducing hello intervals and more frequent SPF computation would in fact reduce network scalability and stability. We will use a simple and approximate but easy-to-understand analytic model for this purpose. We will also use a more involved simulation model. Next, we would like to make the point that many of the underlying causes of network scalability could be avoided if certain IGP messages could be specially marked and provided prioritized treatment.

2 Analytic Model for Delay seen By a Received Hello Packet During a LSA Storm

For every trunk interface, a node has to send and receive a Hello packet once every hello interval. Sending of a Hello packet can be triggered by a timer and it is possible to give higher priority to timer-driven jobs and thereby ensure that it is not excessively delayed even during extended CPU-busy periods. However, a received Hello packet cannot be easily distinguished from other IGP or IP packets and therefore is typically served in a first-come-first-served fashion. We do a simple and approximate analysis of the delay experienced by this packet during an LSA storm at a node with highest adjacency. Let's assume:

- S = Size of LSA storm (i.e., number of LSAs in it). Also, it is assumed that each LSA is carried in one LSU packet.
- L = Link adjacency of the node under consideration. This is assumed to be the maximum in the network.
- t1 = Time to send or receive one IGP packet over an interface (the same time is assumed for Hello, LSA, duplicate LSA and LSA acknowledgement even though in general there may be some differences. However, this would be a good approximation if majority of the time is in the act of receiving or sending and a relatively small part for packet-type-specific work. In the numerical examples we assume t1 = 1 ms.
- t2 = Time to do one SPF calculation. For large network, this time is usually in hundreds of ms and in the numerical examples we assume t2 = 200 ms.
- Hi = Hello interval.
- Si = minimum interval between successive SPF calculations.
- ro = Rate at which non-IGP work comes to the node (e.g., forwarding of data packets). For the numerical examples we assume ro = 0.2.
- T = Total work brought in to the node during the LSA storm. For each LSA update generated elsewhere, the node will receive one new LSA packet over one interface, send an acknowledgement packet over that interface, and send copies of the LSA packet over the remaining L-1 interfaces. Also, assuming that the *implicit acknowledgement* mechanism is in use, the node will subsequently receive either an acknowledgement or a duplicate LSA over the remaining L-1 interfaces. So over each interface one packet is sent and one is received. It can be seen that the same would be true for self-generated LSAs. So the total work per LSA update is 2*L*t1. Since there are S LSAs in the storm, we get

  $$T = 2*S*L*t1 \qquad (1)$$

In Equation (1) we ignore retransmissions of LSAs in case acknowledgements are not received or processed within 5 seconds. This impact and other details are taken into account in the simulation model to be presented later.

- T2 = Time period over which the work comes. Due to differences in propagation times and congestion at other nodes, it is possible for the work arrival time to be spread out over a long interval. However, since we are considering the node with highest adjacency, i.e., one with highest congestion, (this is assuming that all nodes have the same processing power and about the same non-IGP workload) most of the work will come in one chunk. We verified this to be usually true using simulations. One part of T2 will be of the order of link propagation delay and we assume that there is a second part which is proportional to T. Therefore we get,

T2 = A + B*T (2)

Where A and B are constants. For the numerical examples we assume
A = 10 ms and B = 0.1.

- D = Maximum delay experienced by a Hello packet during the LSA storm. We assume first-come-first-served service and hence the delay seen by the Hello packet would be the total outstanding work at the node at the arrival instant plus its own processing time. We assume that outstanding work steadily increases over the interval T2 and so the maximum delay is seen by a Hello packet that comes near the end of this interval. We write down an approximate expression for D and then explain the various terms on the right hand side:

D = T – T2 + max(1,2*T2/Hi)*t1 + max(1,T2/Si)*t2 + ro*T2 (3)

The first term is the total work brought in due to the LSA storm. The second term is the work the node was able to finish since we are assuming that it was continuously busy during the period T2. The third term is the total work due to the sending and receiving of Hello packets during the period T2. Note that it is assumed that at least one Hello packet is processed, i.e., itself. The fourth term is due to SPF processing during the period T2 and we assume that at least one SPF processing is done. The last term is the total non-IGP work coming to the node over the interval T2

- Dmax = maximum allowed value of D, i.e., if D exceeds this value then the associated link would be declared down. In the numerical examples below we assume

Dmax = 3*Hi (4)

If we assume that the previous Hello packet was minimally delayed then exceeding Dmax really means four missed hellos since the Hello packet under study itself came after a period Hi. In the numerical examples below, both D and Dmax change with choice of system parameters and we are mainly interested in identifying if D exceeds Dmax. For this purpose we define the following ratio variable

Delay Ratio = D/Dmax (5)

and identify if Delay Ratio exceeds 1.

In Figures 1-3 we plot the Delay Ratio as a function of LSA Storm size with node adjacencies 10, 20 and 50 respectively. All parameters except for the ones noted explicitly on the figures are as stated earlier. Figure 1 assumes Hello packets every 10 seconds and SPF calculation every 5 seconds which are typical default values today. With a node adjacency of 10, the Delay Ratio is below 1 even with an LSA storm of size 1000. However, with a node adjacency of

20, the Delay Ratio exceeds 1 at around a storm of size 800 and with a node adjacency of 50, the Delay Ratio exceeds 1 at around a storm of size 325.

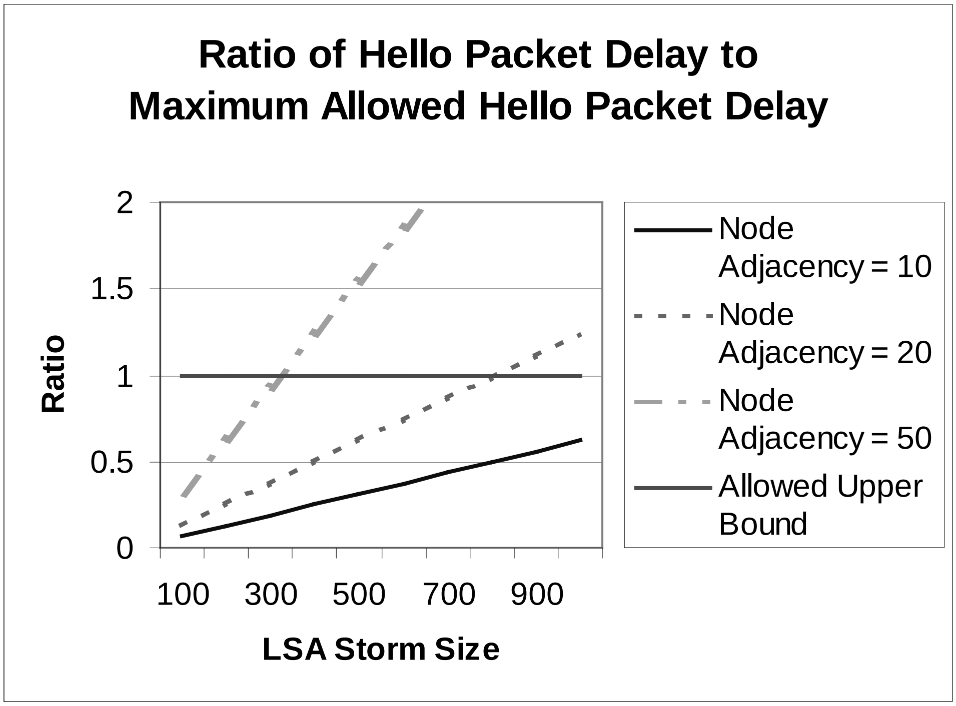

Figure 1: Delay Ratio with Hello Every 10 Seconds, SPF Every 5 Seconds, Dmax = 30 seconds

In a large network it is not unusual to have LSA storms of size several hundreds since the LSA database size may be several thousands. This is particularly true if there are many type 5 LSAs and there are special LSAs for carrying information about available bandwidth at trunks as is common in ATM networks and might be used in MPLS-based networks as well.

Figure 2 decreases the hello interval to 2 seconds and SPF calculation is done once a second. LSA storm thresholds are significantly reduced. Specifically, with a node adjacency of 10, the Delay Ratio exceeds 1 at around a storm of size 310; with a node adjacency of 20, the Delay Ratio exceeds 1 at around a storm of size 160; and with a node adjacency of 50, the Delay Ratio exceeds 1 at around a storm of size only 65.

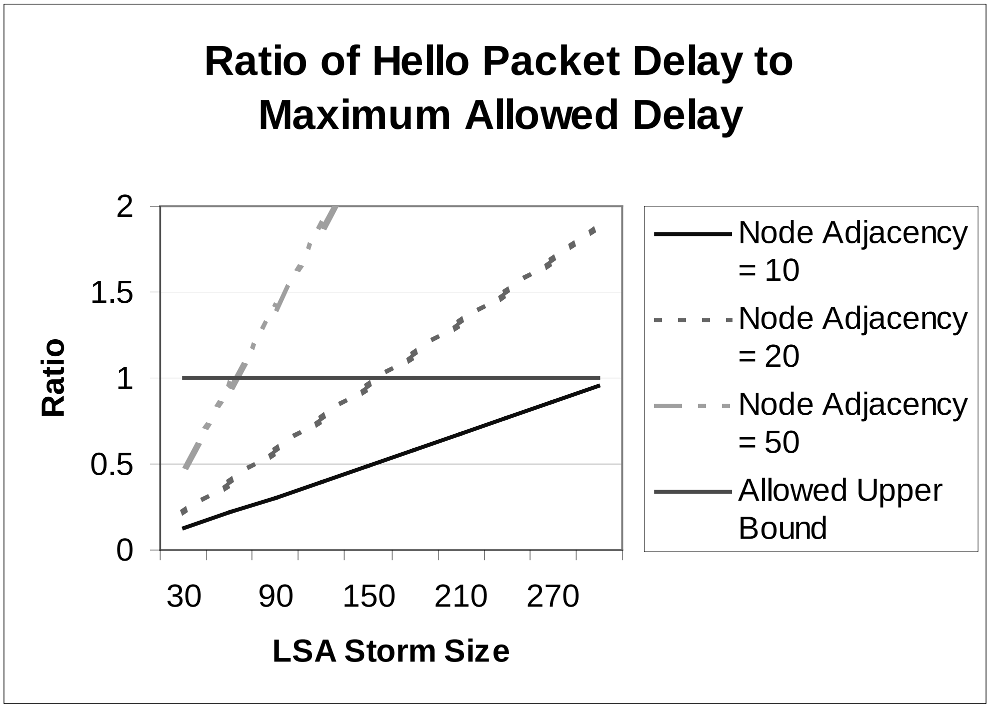

Figure 2: Delay Ratio with Hello Every 2 Seconds, SPF Every 1 Second, Dmax = 6 seconds

Figure 3 decreases the hello interval even further to 300 ms and SPF calculation is done once every 500 ms. LSA storm thresholds are really small now. Specifically, with a node adjacency of 10, the Delay Ratio exceeds 1 at around a storm of size 40, with a node adjacency of 20, the Delay Ratio exceeds 1 at around a storm of size 20, and with a node adjacency of 50, the Delay Ratio is already over 1 even with a storm of size 10.

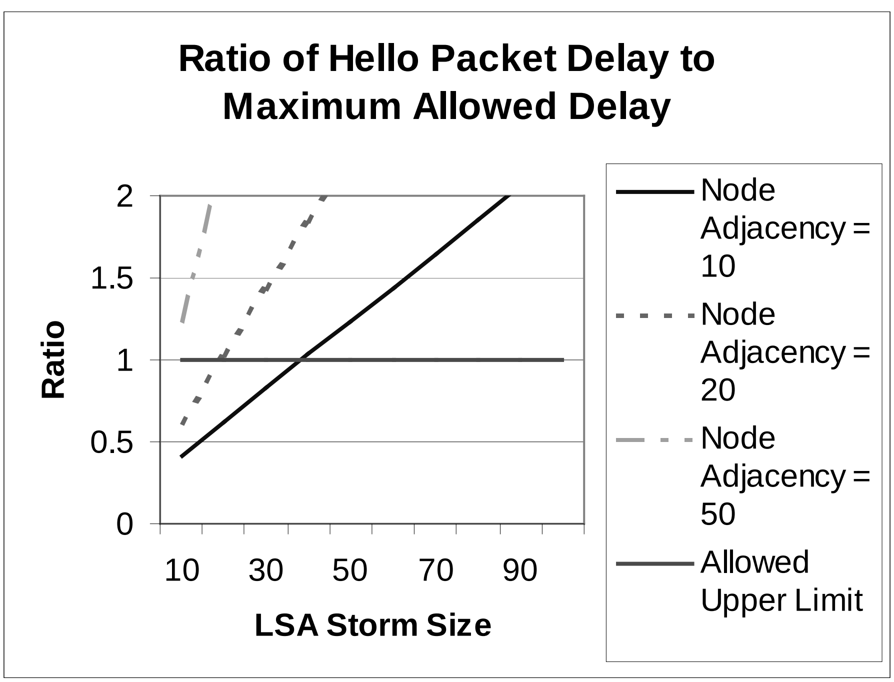

Figure 3: Delay Ratio with Hello Every 300 ms, SPF Every 500 ms, Dmax = 900 ms

Whenever Delay Ratio exceeds 1, the associated link is declared down even if it is actually up and eventually other undesirable events start (e.g., trunk flapping and cascading of extended CPU overload periods to other nodes). Therefore, the LSA storm threshold at which the Delay Ratio exceeds 1 may also roughly be considered as the network stability threshold. Figures 1-3 show that the stability threshold rapidly decreases as the hello interval and SPF computation interval decreases. One reason for this is the increased CPU work due to more frequent hello and SPF computations, but the dominant reason is that Dmax itself decreases and so a smaller CPU busy interval is needed to exceed it. Specifically, Dmax is 30 seconds in Figure 1, 6 Seconds in Figure 2 and only 900 ms in Figure 3. It is clear from the above examples that in order to maintain network stability as the hello interval decreases, it is necessary to provide faster prioritized treatment to received Hello packets which can of course be only done if those packets can be distinguished from other IGP or IP packets.

3 Simulation Study

We have also developed a simulation model to capture more accurately the impact of an LSA storm on all the nodes of the network. It captures the actual congestion seen at various nodes, propagation delay between nodes and retransmissions in case an LSA is not acknowledged. It also tries to approximate a real network implementation and uses processing times that are roughly in the same order of magnitude as measured in the real network (of the order of milliseconds). There are two categories of IGP messages. Category one messages are triggered by a timer and include the Hello refresh, LSA refresh and retransmission packets. Category 2 messages are not triggered by a timer and include received Hello, received LSA and received acknowledgements. Timer-triggered messages are given non-preemptive priority over the other type. A beneficial effect of this strategy is that Hello packets are sent out with little delay even under intense CPU overload. However, the received Hello packets and the received acknowledgement packets may see long queueing delays under intense CPU overload. Figures 4 and 5 below show sample results of the simulation study when applied to a network with about 300 nodes and 800 trunks. The hello interval is assumed to be 5 seconds, the minimum interval between successive SPF calculations is 1 second, and a trunk is declared down if no Hello packet is received for three successive hello intervals, i.e., 15 seconds. During the study, an LSA storm of size 300 and 600 (Figures 4 and 5 respectively) are created at instant of time 100 seconds. Three LSAs are packed within one LSU packet and it is assumed that they remain packed the same way during the flooding process. Besides the storm, there are also the normal once-in-thirty-minutes LSA refreshes and those LSAs are packed one per LSU packet. We define a quantity “dispersion” which is the number of LSU packets generated in the network but not received and processed in at least one node. Figures 4 and 5 plot dispersion as a function of time. Before the LSA storm, the dispersion due to normal LSA refreshes remains small. As expected, right after the storm the dispersion jumps and then comes down again to the pre-storm level after some period of time. In Figure 4 with an LSA storm size 300, the "heavy dispersion period" lasted about 11 seconds and no trunk losses were observed. In Figure 5 with an LSA storm of size 600, the "heavy dispersion period" lasted about 40 seconds. Some trunk losses were observed a little after 15 seconds within the "heavy dispersion period" but eventually all trunks recovered and the dispersion came down to the pre-storm level.

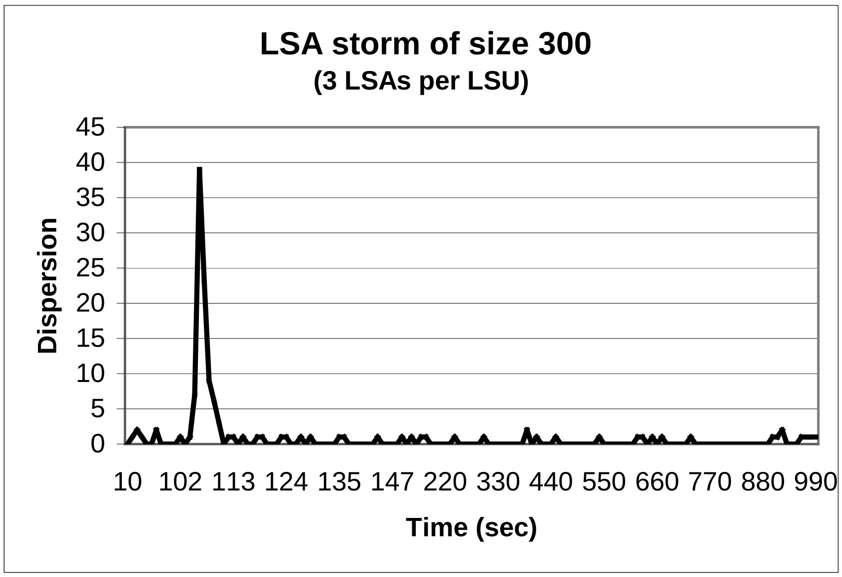

Figure 4: Dispersion Versus Time (LSA Storm Size = 300)

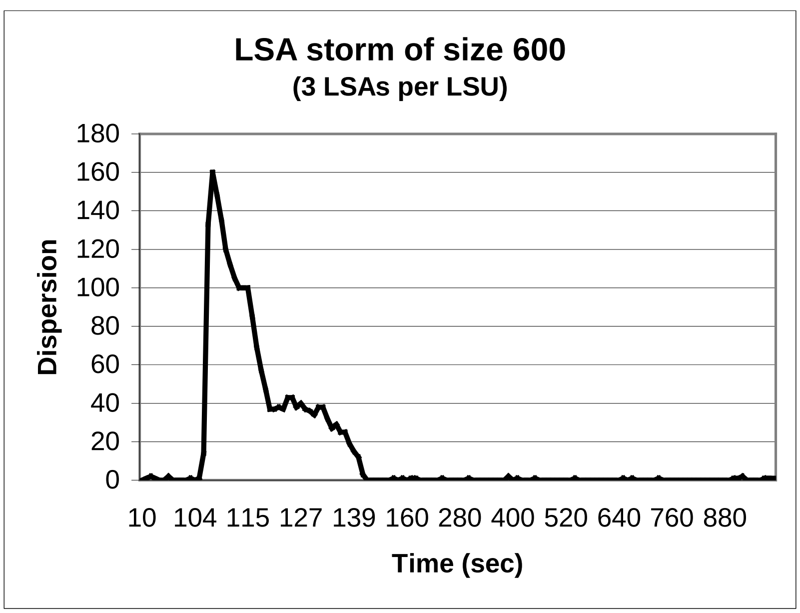

Figure 5: Dispersion Versus Time (LSA Storm Size = 600)

4 Need for Special Marking and Prioritized Treatment of Specific IGP packets

The analytic and simulation models show that a major cause for unstable behavior in networks is received Hello packets at a node getting queued behind other work brought in to the node during an LSA storm and missing the deadline of typically three or four hello intervals. This need not happen to outgoing Hello packets that are triggered by a timer since the node CPU can give it prioritized treatment. Clearly, if the received Hello packet can be specially marked to distinguish it from other IGP and IP packets then they can also be given prioritized treatment and they would not miss the deadline even during a large LSA storm. Some specific field of IP packets may be used for this purpose. Besides the Hello packets there may be other IGP packets that could also benefit from special marking and prioritized marking. We give two examples but clearly others are possible.

- One example is the LSA acknowledgement packet. This packet disables retransmission and if a large queueing delay to this packet expires the retransmission timer (typical default value is 5 seconds) then a needless retransmission will happen causing extra traffic load. Retransmission event is usually rare due to the reliable nature of transmission links, but during the 600 LSA storm simulation in Figure 5 many retransmission events were noted. Usually, retransmission events happen more with a longer CPU busy period. Clearly, a special marking and prioritization of the LSA acknowledgement packet would eliminate many needless retransmissions.
- A second example is an LSA carrying a bad news, i.e., a failure of a trunk or a node. It is preferable to transmit this information faster than other LSAs in the network that either carry good news or are just once-in-30-minutes refreshes. The explicit identification can also be used to trigger the SPF calculation after processing LSAes carrying bad information. This will obviate the need of lowering the SPF calculation interval under all circumstances and thus reducing the processing overhead.

The example in this draft focussed explicitly on the control domain. However, it can easily be seen that having an explicit identification for certain ‘chosen’ packets will help minimize their drop probability in the traffic plane also. The explicit identification allows these control packets to be easily distinguished from the data packets in the line card and hence their processing (forwarding) can be expedited even under large traffic conditions.

5 Summary

In this proposal we point out that if a large LSA storm is generated as a result of some type of failure/recovery of nodes/trunks or synchronization among refreshes then the Hello packets received at a node may see large queueing delays and miss the deadline of typically three or four hello intervals. This causes the trunk to be down and is potentially the beginning of unstable behavior in the network. This is already a concern in today’s network but would be a much bigger concern if the hello interval and minimum interval between SPF calculations are substantially reduced (below or perhaps well below a second) in order to allow faster rerouting, as proposed in [1]draft-alaettinoglu-isis-convergence-00.txt. To avoid the above, we propose the use of a special marking for Hello packets (perhaps using a special field in IP packets) so that they may be distinguished from other IGP and IP packets and provided a prioritized treatment during intense CPU overload periods caused by LSA storms. We also point out that other IGP packets could benefit from special markings as well. Two examples are LSA acknowledgement packets and LSA packets carrying bad news.

5 Acknowledgments

The authors would like to thank members of the High-Speed Packet Switching division of AT&T for their help during the study.

6 References

[1] draft-alaettinoglu-isis-convergence-00.txt   November, 2000

[2] draft-ietf-ospf-refresh-guide-01.txt   July, 2000

[3] draft-ietf-ospf-isis-flood-opt-00.txt   October, 2000

[4] draft-ietf-ospf-ppp-flood-00.txt   November, 2000

8 Authors’ Addresses

Anurag S. Maunder
Cisco Systems
email: amaunder@cisco.com

Gagan Choudhury
AT&T Labs, Middletown, NJ, USA
email: gchoudhury@att.com