Internet Engineering Task Force
Internet Draft
Expires in November, 2003
Category: Best Current Practice
draft-ietf-ospf-scalability-04.txt

Gagan L. Choudhury
Vera D. Sapozhnikov
Gerald R. Ash
AT&T

Anurag S. Maunder
Erlang Technology

Vishwas Manral
Motorola Inc.

May, 2003

Prioritized Treatment of Specific OSPF Packets and Congestion Avoidance

Status of this Memo

IPR Notices

Copyright Notice

Abstract

This document recommends methods that are intended to improve the scalability and stability of large networks using OSPF (Open Shortest Path First) protocol. The methods include processing OSPF Hellos and

compared to other OSPF packets, and other congestion avoidance procedures. Simulation results in support of some of the recommendations are given in the appendix sections.

Table of Contents

1. Introduction

A large network running OSPF [Ref1] or OSPF-TE [Ref2] protocol may occasionally experience the simultaneous or near-simultaneous update of a large number of link-state-advertisements, or LSAs.
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, software, or procedural in nature.

The LSA storm causes high CPU and memory utilization at the router causing incoming packets to be delayed or dropped.
Delayed acknowledgments (beyond the retransmission timer value) result in retransmissions, and delayed Hello packets (beyond the router-dead interval) result in links being declared down.
The retransmissions and additional LSA originations result in further CPU and memory usage, essentially causing a positive feedback loop, which, in the extreme case, may drive the network to an unstable state.

The default value of retransmission timer is 5 seconds and that of the router-dead interval is 40 seconds. However, recently there has been a lot of interest in significantly reducing OSPF convergence time. As part of that plan much shorter (subsecond) Hello and router-dead intervals have been proposed [Ref3]. In such a scenario it will be more likely for Hello packets to be delayed beyond the router-dead interval during a network congestion caused by an LSA storm.

Appendix A explains in more detail LSA storm scenarios, their impact, and points out a few real-life examples of control-message storms. Appendix B presents a simulation study on this phenomenon.

In order to improve the scalability and stability of networks we recommend steps for prioritizing critical OSPF packets and avoiding congestion. The details of the recommendations are given in Section 2. We also do a simulation study on a subset of the recommendations in Appendix B and show that they indeed improve the scalability and stability of networks using OSPF protocol.

Appendix C provides some further recommendations with similar goals.

2. Recommendations

The Recommendations below are intended to improve the scalability and stability of large networks using OSPF protocol. During periods of network congestion they would reduce retransmissions, avoid an adjacency to be declared down due to Hello packets being delayed beyond the RouterDeadInterval, and take other congestion avoidance steps. The recommendations are unordered except that Recommendation 2 is to be implemented only if Recommendation 1 is not implemented.

(1) Classify all OSPF packets in two classes: a "high priority" class comprising of OSPF Hello packets and Link State Acknowledgment packets, and a "low priority" class comprising of all other packets. The classification is accomplished by examining the OSPF packet header. While receiving a packet from a neighbor and while transmitting a packet to a neighbor, try to process a "high priority" packet ahead of a "low priority" packet.

(2) If the Recommendation 1 cannot be implemented then reset the Inactivity Timer for an adjacency whenever any OSPF unicast packet (including any packet sent to 224.0.0.5 over a point-to-point link) is received over that adjacency (currently this is done only for the Hello packet). So OSPF would declare the adjacency to be down only if no OSPF unicast packets (or packets sent to 224.0.0.5) are received over that adjacency for a period equaling or exceeding the RouterDeadInterval. The reason for not recommending this proposal in conjunction with Recommendation 1 is to avoid potential undesirable side effects. One such effect is the delay in discovering the down status of an adjacency in a case where no high priority Hello packets are being received but the inactivity timer is being reset by other stale packets in the low priority queue.

(3) Use an Exponential Backoff algorithm for determining the value of the LSA retransmission interval (RxmtInterval). Let R(i)

represent the RxmtInterval value used during the i-th retransmission of an LSA. Use the following algorithm to compute R(i)

```
R(1) = Rmin
R(i+1) = Min(KR(i),Rmax)  for i>=1
```

where K, Rmin and Rmax are constants and the function Min(.,.) represents the minimum value of its two arguments. Example values for K, Rmin and Rmax may be 2, 5 seconds and 40 seconds respectively. Note that the example value for Rmin, the initial retransmission interval, is the same as the sample value of RxmtInterval in [Ref1].

This recommendation is motivated by the observation that during a network congestion event caused by control messages, a major source for sustaining the congestion is the repeated retransmission of LSAs. The use of an Exponential Backoff algorithm for the LSA retransmission interval reduces the rate of LSA retransmissions while the network experiences congestion (during which it is more likely that multiple retransmissions of the same LSA would happen). This in turn helps the network get out of the congested state.

(4) Implicit Congestion Detection and Action Based on That:
If there is control message congestion at a router, its neighbors do not know about that explicitly. However, they can implicitly detect it based on the number of unacknowledged LSAs to this router. If this number exceeds a certain "high water mark" then the rate at which LSAs are sent to this router should be reduced. At a future time, if the number of unacknowledged LSAs to this router falls below a certain "low water mark" then the normal rate of sending LSAs to this router should be resumed. An example value for the "high water mark" may be 20 unacknowledged LSAs and that for the "low water mark" may be 10 unacknowledged LSAs. An example value for the rate on exceeding the "high water mark" may be 50% the normal rate. This recommendation is to be implemented only for unicast LSAs sent to a neighbor (including LSAs sent to 224.0.0.5 over point-to-point links).

Recommendations 3 and 4 both slow down LSAs to congested neighbors based on implicitly detecting the congestion but they have important differences. Recommendation 3 progressively slows down successive retransmissions of the same LSA whereas Recommendation 3 slows down all LSAs (new or retransmission) to a congested neighbor.

(5) Throttling Adjacencies to be Brought Up Simultaneously:
If a router tries to bring up a large number of adjacencies to

its neighbors simultaneously then that may cause severe congestion due to database synchronization and LSA flooding activities. It is recommended that during such a situation no more than "n" adjacencies should be brought up simultaneously. Once a subset of adjacencies have been brought up successfully, newer adjacencies may be brought up as long as the number of simultaneous adjacencies being brought up does not exceed "n". The appropriate value of "n" would depend on the router processing power, link bandwidth and propagation delay. An example value may be 4.

In the presence of throttling, an important issue is the order in which adjacencies are to be formed. We recommend a First Come First Served (FCFS) policy based on the order in which the request for adjacency formation arrives. Requests may either be from neighbors or self-generated. Among the self-generated requests a priority list may be used to decide the order in which the requests are to be made. However, once an adjacency formation process starts it is not to be preempted except for unusual circumstances such as errors or time-outs.

3. Security Considerations

This memo does not create any new security issues for the OSPF protocol. Security considerations for the base OSPF protocol are covered in [Ref1].

4. Acknowledgments

We would like to acknowledge the support of OSPF WG chairs Rohit Dube, Acee Lindem, and John Moy. We acknowledge Mitchell Erblich, Mike Fox, Tony Przygienda, and Krishna Rao for comments on previous versions of the draft. We also acknowledge Margaret Chiosi, Elie Francis, Jeff Han, Beth Munson, Roshan Rao, Moshe Segal, Mike Wardlow, and Pat Wirth for collaboration and encouragement in our scalability improvement efforts for Link-State-Protocol based networks.

5. Normative References

[Ref1] J. Moy, "OSPF Version 2", RFC 2328, April, 1998.

6. Informative References

[Ref2] D. Katz, D. Yeung, K. Kompella, "Traffic Engineering Extension to OSPF Version 2," Work in Progress.

[Ref3] C. Alaettinoglu, V. Jacobson and H. Yu, "Towards Milli-second IGP Convergence," Work in Progress.

[Ref4] Pappalardo, D., "AT&T, customers grapple with ATM net outage," Network World, February 26, 2001.

[Ref5] "AT&T announces cause of frame-relay network outage," AT&T Press Release, April 22, 1998.

[Ref6] Cholewka, K., "MCI Outage Has Domino Effect," Inter@ctive Week, August 20, 1999.

[Ref7] Jander, M., "In Qwest Outage, ATM Takes Some Heat," Light Reading, April 6, 2001.

[Ref8] A. Zinin and M. Shand, "Flooding Optimizations in Link-State Routing Protocols," Work in Progress.

[Ref9] P. Pillay-Esnault, "OSPF Refresh and flooding reduction in stable topologies," Work in progress.

[Ref10] G. Ash, G. Choudhury, V. Sapozhnikova, M. Sherif, A. Maunder, V. Manral, "Congestion Avoidance & Control for OSPF Networks", Work in Progress.

[Ref11] B. M. Waxman, "Routing of Multipoint Connections," IEEE Journal on Selected Areas in Communications, 6(9):1617-1622, 1988.

7. Authors' Addresses

Gagan L. Choudhury
AT&T
Room D5-3C21
200 Laurel Avenue
Middletown, NJ, 07748
USA
Phone: (732)420-3721
email: gchoudhury@att.com

Vera D. Sapozhnikova
AT&T
Room C5-2C29
200 Laurel Avenue
Middletown, NJ, 07748
USA
Phone: (732)420-2653
email: sapozhnikova@att.com

Gerald R. Ash
AT&T
Room D5-2A01
200 Laurel Avenue
Middletown, NJ, 07748
USA
Phone: (732)420-4578

email: gash@att.com

Anurag S. Maunder
Erlang Technology
2880 Scott Boulevard
Santa Clara, CA 95052
Phone: (408)420-7617
email: anuragm@erlangtech.com

Vishwas Manral
Motorola Inc.
189, Prashasan Nagar,
Road Number 72
Jubilee Hills, Hyderabad
India
email: vishwas@motorola.com

Appendix A. LSA Storm: Causes and Impact

An LSA storm may be initiated due to many reasons. Here are some examples:

(a) one or more link failures due to fiber cuts,

(b) one or more router failures for some reason, e.g., software crash or some type of disaster (including power outage) in an office complex hosting many routers,

(c) Link/router flapping,

(d) requirement of taking down and later bringing back many routers during a software/hardware upgrade,

(e) near-synchronization of the periodic 1800 second LSA refreshes of a subset of LSAs,

(f) refresh of all LSAs in the system during a change in software version,

(g) injecting a large number of external routes to OSPF due to a procedural error,

(h) Router ID changes causing a large number of LSA re-originations (possibly LSA purges as well depending on the implementation).

In addition to the LSAs originated as a direct result of link/router failures, there may be other indirect LSAs as well. One example in MPLS networks is traffic engineering LSAs originated at other links as a result of significant change in reserved bandwidth resulting from rerouting of Label Switched Paths (LSPs) that went down during the link/router failure.

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 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 network routers, 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. See for example [Ref4-Ref7]. 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.

In Appendix B, we use a simulation model to show that there is a certain LSA storm size threshold above which the network may show unstable behavior caused by large number of retransmissions, link failures due to missed Hello packets and subsequent link recoveries. We also show that the LSA storm size causing instability may be substantially increased by providing prioritized treatment to Hello and LSA Acknowledgment packets and by using an exponential backoff algorithm for determining the LSA retransmission interval. Furthermore, if we prioritize Hello packets then even when the network operates somewhat above the stability threshold, links are not declared down due to missed Hellos. This implies that even though there is control plane congestion due to many retransmissions, the data plane stays up and no new LSAs are originated (besides the ones in the original storm and the refreshes). These observations are the basis of the first three recommendations in Section 2.

One might argue that the scalability issue of large networks should be solved solely by dividing the network hierarchically into multiple areas so that flooding of LSAs remains localized within areas. However, this approach increases the network management and design complexity and may result in less optimal routing between areas. Also, ASE LSAs are flooded throughout the AS and it may be a problem if there are large numbers of them. Furthermore, a large number of summary LSAs may need to be flooded across Areas and their numbers would increase significantly if multiple Area Border Routers are employed for the purpose of reliability. Thus it is important to allow the network to grow towards as large a size as possible under a single area.

Our recommendations are synergistic with a broader set of scalability and stability improvement proposals. [Ref8] proposes flooding overhead reduction in case more than one interface goes to the same neighbor. [Ref9] proposes a mechanism for greatly reducing LSA refreshes in stable topologies.

[Ref10] proposes a wide range of congestion control and failure recovery mechanisms (some of those ideas are covered in this draft but [Ref10] has other ideas not covered here).

Appendix B. Simulation Study

The main motivation of this study is to show the network congestion and instability caused by large LSA storms and the improvement in stability and scalability that can be achieved by following the recommendations in this memo.

Appendix B.1. The Network Under Simulation

We generate a random network over a rectangular grid using a modified version of Waxman's algorithm [Ref11] that ensures that the network is connected and has a pre-specified number of routers, links, maximum number of neighbors per router, and maximum number of adjacencies per router. The rectangular grid resembles the continental U.S.A. with maximum one-way propagation delay of 30 ms in the East-West direction and maximum one-way propagation delay of 15 ms in the North-South direction. We consider two different network sizes as explained in Section B.2.

The network has a flat, single-area topology.

Each link is a point-to-point link connecting two routers.

We assume that router CPU and memory (not the link bandwidth) is the main bottleneck in the LSA flooding process. This will typically be true for high speed links (e.g., OC3 or above) and/or links where OSPF traffic gets an adequate Quality of Service (QoS) compared to other traffic.

Different Timers:

  LSA refresh interval = 1800 seconds,
  Hello refresh interval = 10 Seconds,
  Router-Dead interval = 40 seconds,
  LSA retransmission interval: two values are considered, 10 seconds and 5 Seconds (note that a retransmission is disabled on the receipt of either an explicit acknowledgment or a duplicate LSA over the same interface that acts as an implicit acknowledgment)
  Minimum time between successive origination of the same LSA = 5 seconds,
  Minimum time between successive Dijkstra SPF calculations is 1 second.

Packing of LSAs: It is assumed that for any given router, the LSAs originated over a 1-second period are packed together to form an LSU but no more than 3 LSAs are packed in one LSU.

LSU/Ack/Hello Processing Times: All processing times are expressed in terms of the parameter T. Two values of T are considered, 1 ms and 0.5 ms.

In the case of a dedicated processor for processing OSPF packets the processing time reported represents the true processing time. If the processor does other work and only a fraction of its capacity can be dedicated to OSPF processing then we have to inflate the processing time appropriately to get the effective processing time and in that case it is assumed that the inflation factor is already taken into account as part of the reported processing time.

The fixed time to send or receive any LSU, Ack or Hello packet is T. In addition, a variable processing time is used for LSU and Ack depending on the number and types of LSAs packed. No variable processing time is used for Hello.
Variable processing time per Router LSA is (0.5 + 0.17L)T where L is the number of adjacencies advertised by the Router LSA. For other LSA types (e.g., ASE LSA or a "Link" LSA carrying traffic engineering information about a link), the variable processing time per LSA is 0.5T.

Variable processing time for an Ack is 25% that of the corresponding LSA.

It is to be noted that if multiple LSAs are packed in a single LSU packet then the fixed processing time is needed only once but the variable processing time is needed for every component of the packet.

The processing time values we use are roughly in the same range of what has been observed in an operational network.

LSU/Ack/Hello Priority: Two non-preemptive priority levels and three priority scenarios are considered. Within each priority level processing is FIFO with new packets of lower priority being dropped when the lower priority queue is full. The higher priority packets are never dropped.

In Priority scenario 1, all LSUs/Acks/Hellos received at a router are queued at the lower priority.

In Priority scenario 2, Hellos received at a router are queued at the higher priority but LSUs/Acks are queued at lower priority.

In Priority scenario 3, Hellos and Acks received at a router are queued at the higher priority but LSUs are queued at lower priority.

All packets generated internally to a router (usually triggered by a timer) are processed at the higher priority. This includes the initial LSA storm, LSA refresh, Hello refresh, LSA retransmission and new LSA origination after detection of a failure or recovery.

Buffer Size for Incoming LSUs/Acks/Hellos (lower priority): Buffer size is assumed to be 2000 packets where a packet is either an Ack, LSU, or Hello.

LSA Refresh: Each LSA is refreshed once in 1800 seconds and the refresh instants of various LSAs in the LSDB are assumed to be uniformly distributed over the 1800 seconds period, i.e., they are completely unsynchronized. If however, an LSA is originated as part of the initial LSA storm then it goes on a new refresh schedule of once in 1800 seconds starting from its origination time.

LSA Storm Origination: As defined earlier, "LSA storm" is the simultaneous or near simultaneous origination of a large number of LSAs. In the case of only Router and ASE LSAs we normally assume that the number of ASE LSAs in the storm is about 4 times that of the Router LSAs, but the ratio is allowed to change if either the Router or the ASE LSAs have reached their maximum possible value.
In the case of only Router and Link LSAs (carrying traffic engineering information) we normally assume that the number of Link LSAs in the storm is about 4 times that of the Router LSAs, but the ratio is allowed to change if either the Router or the Link LSAs have reached their maximum possible value. For any given LSA storm we keep originating LSAs starting from router index 1 and moving upwards and stop until the correct number of LSAs of each type have been originated. The LSAs originated at any given router is assumed to start at an instant uniformly distributed between 20 and 30 seconds from the start of the simulation. Successive LSA originations at a router are assumed to be spaced apart by 400 ms.
It is to be noted that during the period of observation there are

other LSAs originated besides the ones in the storm. These include refresh of LSAs that are not part of the storm and LSAs originated due to possible link failures and subsequent possible link recoveries.

Failure/Recovery of Links: If no Hello is received over a link (due to CPU/memory congestion) for longer than Router-Dead Interval then the link is declared down. At a later time, if Hellos are received then the link would be declared up. Whenever a link is declared up or down, one Router LSA is originated by each router on the two sides of the point-to-point link. If "Link LSAs" carrying traffic engineering information is used then it is assumed that each router would also originate a Link LSA. In this case it is also assumed that due to rerouting of LSPs, three other links in the network (selected randomly in the simulation) would have significant change in reserved bandwidth which would result in one Link LSA being originated by the routers on the two ends of each such link.

Appendix B.2. Simulation Results

In this section we study the relative performance of the three priority scenarios defined earlier (no priority to Hello or Ack, priority to Hello only, and priority to both Hello and Ack) with a range of Network sizes, LSA retransmission timer values, LSA types, processing time values and Hello/Router-Dead-Interval values:

Network size: Two networks are considered. Network 1 has 100 routers, 1200 links, maximum number of neighbors per router is 30 and maximum number of adjacencies per router is 50 (same neighbor may have more than one adjacencies). Network 2 has 50 routers, 600 links, maximum number of neighbors per router is 25 and maximum number of adjacencies per router is 48. Dijkstra SPF calculation time for Network 1 is assumed to be 100 ms and that for Network 2 is assumed to be 70 ms.

LSA Type: Each router has 1 Router LSA (Total of 100 for Network 1 and 50 for Network 2). There are no Network LSAs since all links are point-to-point links and no Summary LSAs since the network has only one area. Regarding other LSA types we consider two situations. In Situation 1 we assume that there are no ASE LSAs and each link has one "Link" LSA carrying traffic engineering information (Total of 2400 for Network 1 and 1200 for Network 2). In Situation 2 we assume that there are no "Link" LSAs and half of the routers are ASA-Border routers and each border router has 10 ASE LSAs (Total of 500 for Network 1 and 250 for Network 2). We identify Situation 1 as "Link LSAs" and Situation 2 as "ASE LSAs".

LSA retransmission timer value: Two values are considered, 10 seconds and 5 seconds (default value).

Processing time values: Processing times for LSUs, Acks and Hello packets have been previously expressed in terms of a common parameter T. Two values are considered for T, which are 1 ms and 0.5 ms respectively.

Hello/Router-Dead-Interval: It is assumed that Router-Dead interval is four times the Hello interval. In one case it is assumed that Hello interval is 10 seconds and Router-Dead-Interval is 40 seconds (default values), and in the other case it is assumed that Hello interval is 2 seconds and Router-Dead-Interval is 8 seconds.

Based on Network size, LSA type and processing time values we develop 6 Test cases as follows:

Case 1: Network 1, Link LSAs, retransmission timer = 10 sec.,
T = 1 ms, Hello/Router-Dead-Interval = 10/40 sec.

Case 2: Network 1, ASE LSAs, retransmission timer = 10 sec.,
T = 1 ms, Hello/Router-Dead-Interval = 10/40 sec.
Case 3: Network 1, Link LSAs, retransmission timer = 5 sec.,
T = 1 ms, Hello/Router-Dead-Interval = 10/40 sec.

Case 4: Network 1, Link LSAs, retransmission timer = 10 sec.,
T = 0.5 ms, Hello/Router-Dead-Interval = 10/40 sec.

Case 5: Network 1, Link LSAs, retransmission timer = 10 sec.,
T = 1 ms, Hello/Router-Dead-Interval = 2/8 sec.

Case 6: Network 2, Link LSAs, retransmission timer = 10 sec.,
T = 1 ms, Hello/Router-Dead-Interval = 10/40 sec.

For each case and for each Priority scenario we study the network stability as a function of the size of the LSA storm. The stability is determined by looking at the number of non-converged LSUs as a function of time. An example is shown in Table 1 for Case 1 and Priority scenario 1 (No priority to Hellos or Acks).

LSA STORM SIZE	Number of Non-Converged LSUs in the Network at Time(in sec)								
	10s	20s	30s	35s	40s	50s	60s	80s	100s
100 (Stable)	0	0	24	29	24	1	0	1	1
140 (Stable)	0	0	35	48	46	27	14	1	1
160 (Unstable)	0	0	38	57	55	40	26	65	203

Table 1: Network Stability Vs. LSA Storm
(Case 1, No priority to Hello/Ack)

The LSA storm starts a little after 20 seconds and so for some period of time after that the number of non-converged LSUs should stay high and then come down for a stable network.
This happens for LSA storms of sizes 100 and 140. With an LSA storm of size 160, the number of non-converged LSUs stay high indefinitely due to repeated retransmissions, link failures due to missed Hellos for more than the Router-Dead interval which originates additional LSAs and also due to subsequent link recoveries which again originate additional LSAs. We define network stability threshold as the maximum allowable LSA storm size for which the number of non-converged LSUs come down to a low level after some time. It turns out that for this example the stability threshold is 150.

The network behavior as a function of the LSA storm size can be categorized as follows:

(1) If the LSA storm is well below the stability threshold then the CPU/memory congestion lasts only for a short period and during this period there are very few retransmissions, very few dropped OSPF packets and no link failures due to missed Hellos. This type of LSA storm is observed routinely in operational networks and networks recover from them easily.

(2) If the LSA storm is just below the stability threshold then the CPU/memory congestion lasts for a longer period and during this period there may be considerable amount of retransmissions and dropped OSPF packets. If Hello packets are not given

priority then there may also be some link failures due to missed Hellos. However, the network does go back to a stable state eventually. This type of LSA storm may happen rarely in operational networks and they recover from it with some difficulty.

(3) If the LSA storm is above the stability threshold then the CPU/memory congestion may last indefinitely unless some special procedure for relieving congestion is followed. During this period there are considerable amount of retransmissions and dropped OSPF packets. If Hello packets are not given priority then there would also be link failures due to missed Hellos. This type of LSA storm may happen very rarely in operational networks and usually some manual procedure such as taking down adjacencies in heavily congested routers is needed.

(4) If Hello packets are given priority then the network stability threshold increases, i.e., the network can withstand a larger LSA storm. Furthermore, even if the network operates at or somewhat above this higher stability threshold, Hellos are still not missed and so there are no link failures. So even if there is congestion in the control plane due to increased retransmissions requiring some special procedures for congestion reduction, the data plane remains unaffected.

(5) If both Hello and Acknowledgement packets are given priority then the stability threshold increases even further.

In Table 2 we show the network stability threshold for the five different cases and for the three different priority scenarios defined earlier.

Case Number	Maximum Allowable LSA Storm Size For		
	No Priority to Hello or Ack	Priority to Hello Only	Priority to Hello and Ack
Case 1	150	190	250
Case 2	185	215	285
Case 3	115	127	170
Case 4	320	375	580
Case 5	120	175	225
Case 6	185	224	285

Table 2: Maximum Allowable LSA Storm for a Stable Network

We also considered one more scenario with priority to Hello and Ack and with a truncated binary exponential backoff of the retransmission interval with an upper limit of 40 seconds (for the same LSA, each successive retransmission interval is doubled but not to exceed 40 seconds). The maximum allowed LSA storm size for this scenario significantly exceeded the numbers given in the third column.

Appendix B.3. Observations on Simulation Results

Table 2 shows that in all cases prioritizing Hello packets increases the network stability threshold, and in addition, prioritization of LSA Acknowledgment packets increases the stability threshold even further. The reasons for the above observations are as follows. The main sources of sustained CPU/memory congestion (or positive feedback loop) following an LSA storm are (1) LSA retransmissions and (2) links being declared down due to missed Hellos which in turn causes further LSA origination and future recovery of the link causing even more LSA originations.
Prioritizing Hello packets avoids and practically eliminates the second source of congestion. Prioritizing Acknowledgments significantly reduces the first source of congestion, i.e., LSA retransmissions. It is to be noted that retransmissions can not be completely eliminated due to the following reasons. Firstly, only the explicit Acknowledgments are prioritized but duplicate LSAs carrying implicit Acknowledgments are still served at the lower priority. Secondly, LSAs may get greatly delayed or dropped at the input queue of receivers and therefore Acknowledgments may not even get generated in which case prioritizing Acks would not help. Another factor to keep in mind is that since Hellos and Acks are prioritized, the LSAs see bigger delay and potential for dropping. However, the simulation results show that on the whole prioritizing Hello and LSA Acks are always beneficial and significantly improve the network stability threshold.

As stated in Section B.2, exponenetial backoff of LSA retransmission interval further increases the network stability threshold.

Our simulation study also showed that in each of the cases, instead of prioritizing Hello packets if we treat any packet received over a link as a surrogate for a Hello packet (an implicit Hello) then we get about the same stability threshold as obtained with prioritizing Hello packets.

Appendix C. Other Recommendations

(1) Explicit Marking: In Section 2 we recommended that OSPF packets be classified to "high" and "low" priority classes based on

examining the OSPF packet header. In some cases (particularly in the receiver) this examination may be computationally costly. An alternative would be the use of different TOS/Precedence field settings for the two priority classes. [Ref1] recommends setting the TOS field to 0 and the Precedence field to 6 for all OSPF packets. We recommend this same setting for the "low" priority OSPF packets and a different setting for the "high" priority OSPF packets in order to be able to classify them separately without having to examine the OSPF packet header. Two examples are given below:

Example 1: For "low" priority packets set TOS field to 0 and Precedence field to 6, and for "high" priority packets set TOS field to 4 and Precedence field to 6.

Example 2: For "low" priority packets set TOS field to 0 and Precedence field to 6, and for "high" priority packets set TOS field to 0 and Precedence field to 7.

This recommendation is not needed to be followed if it is easy to examine the OSPF packet header and thereby separately classify "high" and "low" priority packets.

(2) Further Prioritization of OSPF Packets: Besides the packets designated as "high" priority in Recommendation 1 of Section 2 there may be a need for further priority separation among the "low" priority OSPF packets. We recommend the use of three priority classes: "high", "medium" and "low". While receiving a packet from a neighbor and while transmitting a packet to a neighbor, try to process a "high priority" packet ahead of "medium" and "low" priority packets and a "medium" priority packet ahead of "low priority" packets. The "high" priority packets are as designated in Recommendation 1 of Section 2. We provide below two candidate examples for "medium" priority packets. All OSPF packets not designated as "high" or "medium" priority are "low" priority.

One example of "medium" priority packet is the Database Description (DBD) packet from a slave (during the database synchronization process) that is used as an acknowledgment. A second example is an LSA carrying intra-area topology change information (this may trigger SPF calculation and rerouting of Label Switched paths and so fast processing of this packet may improve OSPF/LDP convergence times).

An implementor may decide not to follow this recommendation if the processing cost of identifying and separately queueing the "medium" priority packets is deemed to be high.