Internet Engineering Task Force
Internet Draft
Expires in June, 2005
Category: Best Current Practice
draft-ietf-ospf-scalability-09.txt

Gagan L. Choudhury, Editor
AT&T

December, 2004

Prioritized Treatment of Specific OSPF Version 2 Packets and Congestion Avoidance

Status of this Memo

Abstract

This document recommends methods that are intended to improve the scalability and stability of large networks using OSPF (Open Shortest Path First) Version 2 protocol. The methods include processing OSPF Hellos and LSA (Link State Advertisement) Acknowledgments at a higher priority compared to other OSPF packets, and other congestion avoidance procedures.

Table of Contents

1. Introduction

In this document as we refer to OSPF we mean OSPFv2 [Ref1].
The scalability and stability improvement techniques described here may also apply to OSPFv3 [Ref2] but that will require further study and operational experience.

A large network running OSPF protocol may occasionally experience the simultaneous or near-simultaneous update of a large number of link-state-advertisements, or LSAs. This is particularly true if OSPF traffic engineering extension [Ref3] 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, 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 neighbor adjacencies 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 (sub-second) Hello and router-dead intervals have been proposed [Ref4]. In such a scenario it will be more likely for Hello packets to be delayed beyond the router-dead interval during network congestion caused by an LSA storm.

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. A simulation study is reported in [Ref14] that quantifies the congestion phenomenon and its impact. It also studies several of the recommendations and shows that they indeed improve the scalability and stability of networks using OSPF protocol. [Ref14] is available on request by contacting the editor or one of the authors.

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 provides a list of variables used in the recommendations and their example values. Appendix C provides some further recommendations and suggestions 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.

The prioritized processing while transmitting may cause OSPF packets from a neighbor to be received out of sequence. If Cryptographic Authentication (AuType = 2) is used (as specified in [Ref1]) then successive received valid OSPF packets from a neighbor need to have a non-decreasing "Cryptographic sequence number". To comply with this requirement we recommend that in case Cryptographic Authentication (AuType = 2) is used [Ref1], prioritized processing be not done at the transmitter. This will avoid packets arriving at the receiver out of sequence. However, after security processing at the receiver (including sequence number checking) is complete, the OSPF packets may be kept in a "high-priority" queue or a "low-priority" queue based on their class and processed accordingly. The benefit of prioritized processing is clearly higher in the absence of Cryptographic Authentication since in that case prioritization can be implemented both at the transmitter and at the receiver.

However, even with Cryptographic Authentication it will be beneficial to have prioritization only at the receiver (following

security processing).

(2) If the Recommendation 1 cannot be implemented then reset the inactivity timer for an adjacency whenever any OSPF unicast packet or any OSPF packet sent to AllSPFRouters over a point-to-point link is received over that adjacency instead of resetting the inactivity timer only on receipt of the Hello packet. So OSPF would declare the adjacency to be down only if no OSPF unicast packets or no OSPF packets sent to AllSPFRouters over a point-to-point link 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)

$$R(1) = Rmin$$
$$R(i+1) = Min(KR(i),Rmax) \quad for\ i>=1$$

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 progressively using an exponential backoff mechanism but not below a certain minimum rate. At a future time, if the number of unacknowledged LSAs to this router falls below a certain "low water mark" then the rate of sending LSAs to this router should be increased progressively, again using an exponential backoff mechanism but not above a certain maximum rate. The whole algorithm is given below. It is to be noted that this algorithm is to be applied independently to each neighbor and only for unicast LSAs sent to a neighbor or LSAs sent to AllSPFRouters over a point-to-point link.

Let,
U(t) = Number of unacknowledged LSAs to neighbor at time t.
H = A high water mark (in units of number of unacknowledged LSAs)
L = A low water mark (in units of number of unacknowledged LSAs)
G(t) = Gap between sending successive LSAs to neighbor at time t.
F = The factor by which the above gap is to be increased during congestion and decreased after coming out of congestion.
T = Minimum time that has to elapse before the existing gap is considered for change.
Gmin = Minimum allowed value of gap.
Gmax = Maximum allowed value of gap.

The equation below shows how the gap is to be changed after a time T has elapsed since the last change:

```
          _
         |
         | Min(FG(t),Gmax) if U(t+T) > H
G(t+T) = | G(t) if H >= U(t+T) >= L
         | Max(G(t)/F,Gmin) if U(t+T) < L
         |_
```

Min(.,.) and Max(.,.) represent the minimum and maximum values of the two arguments respectively.
Example values for the various parameters of the algorithm are as follows: H = 20, L = 10, F = 2, T = 1 second, Gmin = 20 ms, Gmax = 1 second.

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 progressively 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, total bandwidth available for control plane traffic and propagation delay.
The value of "n" should be configurable.

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.

In some of the Recommendations above we refer to point-to-point links. Those references should also include cases where a broadcast network is to be treated as a point-to-point connection from the standpoint of IP routing [Ref5]

3. Security Considerations

This memo does not create any new security issues for the OSPF protocol.

4. Acknowledgments

We would like to acknowledge the support and helpful comments from OSPF WG chairs Rohit Dube, Acee Lindem, John Moy, Routing Area directors Alex Zinin and Bill Fenner, and IESG reviewers. We acknowledge Vivek Dube, 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 Reference

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

[Ref2] R. Coltun, D. Ferguson and J. Moy, "OSPF For IPV6", RFC 2740, December, 1999.

6. Informative References

[Ref3] D. Katz, K. Kompella, D. Yeung "Traffic Engineering (TE) Extensions to OSPF Version 2," RFC 3630, September, 2003.

[Ref4] C. Alaettinoglu, V. Jacobson and H. Yu, "Towards Milli-second IGP Convergence," Work in Progress.

[Ref5] N. Shen, A. Lindem, J. Yuan, A. Zinin, R. White and S. Previdi, "Point-to-point operation over LAN in link-state routing protocols," Work in Progress.

[Ref6] Pappalardo, D., "AT&T, customers grapple with ATM net outage," Network World, February 26, 2001.

[Ref7] "AT&T announces cause of frame-relay network outage," AT&T Press Release, April 22, 1998.

[Ref8] Cholewka, K., "MCI Outage Has Domino Effect," Inter@ctive Week, August 20, 1999.

[Ref9] Jander, M., "In Qwest Outage, ATM Takes Some Heat," Light Reading, April 6, 2001.

[Ref10] A. Zinin and M. Shand, "Flooding Optimizations in Link-State Routing Protocols," Work in Progress.

[Ref11] P. Pillay-Esnault, "OSPF Refresh and flooding reduction in stable topologies," Work in progress.

[Ref12] G. Ash, G. Choudhury, V. Sapozhnikova, M. Sherif, A. Maunder, V. Manral, "Congestion Avoidance & Control for OSPF Networks", Work in Progress.

[Ref13] B. M. Waxman, "Routing of Multipoint Connections," IEEE Journal on Selected Areas in Communications, 6(9):1617-1622, 1988.

[Ref14] G. Choudhury, G. Ash, V. Manral, A. Maunder and V. Sapozhnikova, "Prioritized Treatment of Specific OSPF Packets and Congestion Avoidance: Algorithms and Simulations," AT&T Technical Report, August, 2003.

[Ref15] K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPV4 and IPV6 Headers", RFC 2474, December, 1998.

7. Contributing Authors and their Addresses

In addition to the Editor, several people contributed to this document. The names and contact information of all authors are given below.

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

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

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

Vishwas Manral
Sinett Semiconductors,
Infantry Road,
Bangalore 500 081
India
email: vishwas@sinett.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 [Ref3] 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 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. See for example [Ref6-Ref9]. 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 [Ref14] a simulation model is used 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. It is also shown 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. If it is not possible to prioritize Hello packets then resetting the inactivity timer on receiving any valid OSPF packets can also provide the same benefit. 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 support the first three recommendations in Section 2. The authors of this draft have also done simulations to verify that the other recommendations in Section 2 helps avoid congestion and allows a graceful exit from a congested state.

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.

The recommendations in the draft are synergistic with a broader set of scalability and stability improvement proposals. [Ref10] proposes flooding overhead reduction in case more than one interface goes to the same neighbor. [Ref11] proposes a mechanism for greatly reducing LSA refreshes in stable topologies.

[Ref12] proposes a wide range of congestion control and failure recovery mechanisms (some of those ideas are covered in this draft but [Ref12] has other ideas not covered here).

Appendix B. List of Variables and Values

F = The factor by which the gap between sending successive LSAs to a neighbor is to be increased during congestion and decreased

after coming out of congestion (used in Recommendation 4). Example value is 2.

G(t) = Gap between sending successive LSAs to a neighbor at time t (used in Recommendation 4).

Gmax = Maximum allowed value of gap between sending successive LSAs to a neighbor (used in Recommendation 4). Example value is 1 second.

Gmin = Minimum allowed value of gap between sending successive LSAs to a neighbor (used in Recommendation 4). Example value is 20 ms.

H = A high water mark (in units of number of unacknowledged LSAs). Exceeding this mark would trigger a potential increase in the gap between sending successive LSAs to a neighbor. (used in Recommendation 4). Example value is 20.

K = A multiplicative constant used in increasing the RxmtInterval value used during successive retransmissions of the same LSA (used in Recommendation 3). Example value is 2.

L = A low water mark (in units of number of unacknowledged LSAs) Dropping below this mark would trigger a potential decrease in the gap between sending successive LSAs to a neighbor. (used in Recommendation 4). Example value is 10.

n = Upper limit on the number of adjacencies to be brought up simultaneously (used in Recommendation 5).

R(i) = RxmtInterval value used during the i-th retransmission of an LSA (used in Recommendation 3).

Rmax = The maximum allowed value of RxmtInterval (used in Recommendation 3). Example value is 40 seconds.

Rmin = The minimum allowed value of RxmtInterval (used in Recommendation 3). Example value is 5 seconds.

T = Minimum time that has to elapse before the existing gap between sending successive LSAs to a neighbor is considered for change (used in Recommendation 4). Example value is 1 second.

U(t) = Number of unacknowledged LSAs to a neighbor at time t (used in Recommendation 4).

Appendix C. Other Recommendations and Suggestions

(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.

It is to be noted that the TOS/Precedence bits have been redefined by Diffserv (RFC 2474, [Ref15]). It is also to be noted that the different TOS/Precedence field settings suggested above only need to be agreed among the systems on the link. 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. If Cryptographic Authentication (AuType = 2) is used (as specified in [Ref1]) then prioritized treatment is to be provided only at the receiver and after security processing, but not at the transmitter since that may cause packets to arrive out of sequence and violate the requirements of "Autype = 2".

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). However, if the processing cost of identifying and separately queueing the LSA in this example is deemed to be high then the implementer may decide not to do it.

(3) Processing large number of LSA Purges: Occasionally some events in the network, such as Router ID changes, may result in a large number of LSA re-originations and LSA purges. In such a scenario one may consider processing LSAs in different order, e.g., processing LSA purges ahead of LSA originations. We, however, do not recommend out-of-order LSA processing for several reasons. Firstly, detecting the LSA type ahead of queueing may be computationally expensive. Out-of-order processing may also cause subtle bugs. We do not want to recommend a major change in the LSA processing paradigm for a relatively rare event such as Router ID change. However, a Router with a changing ID may flush the old LSAs gradually without causing a storm.

Full copyright statement

Intellectual Property Considerations

assurances of licenses to be made available, or the result of an

attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr. The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.