Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-ospf-omp-01

Curtis Villamizar
ANS

October 9, 1998

OSPF Optimized Multipath (OSPF-OMP)

Status of this Memo

Abstract

OSPF may form multiple equal cost paths between points. This is true of any link state protocol. In the absense of any explicit support to take advantage of this, a path may be chosen arbitrarily. Techniques have been utilized to divide traffic somewhat evenly among the available paths. These techniques have been referred to as Equal Cost Multipath (ECMP). An unequal division of traffic among the available paths is generally preferable. Routers generally have no knowledge of traffic loading on distant links and therefore have no basis to optimize the allocation of traffic.

Optimized Mulitpath is a compatible extension to OSPF, utilizing the Opaque LSA to distribute loading information, proposing a means to adjust forwarding, and providing an algorithm to make the adjustments gradually enough to insure stability yet provide reasonably fast adjustment when needed.

1 Overview

Networks running OSPF are often heavily loaded. Topologies often evolve to include multiple paths. Multiple paths may be initially designed to provide redundancy but also result from incremental addition of circuits to accomodate traffic growth. The redundant paths provide a potential to distribute traffic loading and reduce congestion. Optimized Mulitpath (OMP) provides a means for OSPF to make better use of this potential to distribute loading.

1.1 Past Attempts

Early attempts to provide load sensitive routing involved changing link costs according to loading. These attempts were doomed to failure because the adjustment increment was grossly course and oscillation was inevitable [2]. This early experience is largely responsible for the common belief that any form of load sensitive routing will fail due to severe oscillations resulting from instability.

Attempts to use a metric composed of weighted components of delay, traffic, and fixed costs have also been met with very limited success. The problem again is the granularity of adjustment. As the composition of weighted components switches favored paths large amounts of traffic are suddenly moved, making the technique prone to oscillations [3]. The oscillation is damped to some extent by providing a range of composite metric differences in which composite metrics are considered equal and equal cost multipath techniques are used. Even then the technique still suffers oscillations due to the course adjustments made at equal/unequal metric boundaries.

1.2 Equal Cost Multipath

A widely utilized technique to improve loading is known as Equal Cost Multipath (ECMP). If the topology is such that equal cost paths exist, then an attempt is made to divide traffic equally among the paths. Three methods of splitting traffic have been used.

1. Per packet round robin forwarding.
2. Dividing destination prefixes among available next hops in the forwarding entries.
3. Dividing traffic according to a hash function applied to the source and desination pair.

The “per packet round robin forwarding” technique is only applicable if the delays on the paths are almost equal. The delay difference must be small relative to packet serialization time. Delay differences greater than three times the packet serialization time can cause terrible TCP performance. for example, packet 2, 4, and 6 may arrive before packet 1, triggering TCP fast retransmit. The result will be limiting TCP to a very small window and very poor performance over long delay paths.

The delay differences must be quite small. A 532 byte packet is serialized onto a DS1 link in under 2.8 msec. At DS3 speed, serialization is accomplished in under 100 usec. At OC12 it is under 7 usec. For this reason “per packet round robin forwarding” is not applicable to a high speed WAN.

Dividing destination prefixes among available next hops provides a very course and unpredictable load split. Long prefixes are problematic. In reaching an end node, the majority of traffic is often destined to a single prefix. This technique is applicable to a high speed WAN but with the drawbacks just mentioned better techniques are needed.

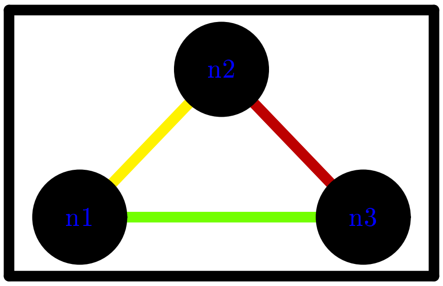

Figure 1: A very simple application of ECMP

The "source/destination hash" based technique was used as far back as the T3-NSFNET in the NSS routers. A hash function, such as CRC-16, is applied over the source address and destination address. The hash space is then split evenly among the available paths by either setting threshholds or performing a modulo operation. Traffic between any given source and destination remain on the same path. Because the technique is based on host addresses, and uses both the source and destination address, it does not suffer the course granularity problem of the prefix based technique, even when forwarding to a single prefix. Source/destination hash is the best technique available for a high speed WAN.

The forwarding decision for the "source/destination hash" based technique is quite simple. When a packet arrives, look up the forwarding entry in the radix tree. The next hop entry can be an array index into a set of structures, each containing one or more actual next hops. If more than one next hop is present, compute a CRC16 value based on the source and destination addresses. The CRC16 can be implemented in hardware and computed in parallel to the radix tree lookup in high speed implementations, and discarded if not needed. Each next hop entry in the structure must contain a boundary value and the next hop itself. An integer "less than" comparison is made against the boundary value determining whether to use this next hop or move to the next a comparison. In hardware the full set of comparisons can be made simultaneously for up to some number of next hops. This yields the next hop to use.

1.3 Optimized Multipath differs from ECMP

For ECMP, the boundary values are set by first dividing one more than the maximum value that the hash computation can return (65536 for CRC16) by the number of available next hops and then setting the Nth boundary to N times that number (with the Nth value fixed at one more than the maximum value regardless of underflow caused by trucating during division, 65536 for CRC16).

An equal load split is not always optimal. Consider the example in Figure 1 with the offered traffic in Table 1. If all of the link costs are set equally, then the link N1—N3 is significantly overloaded (135.75%) while the path N1—N2—N3 is lightly loaded (45.25% and 22.62%). If the cost on the N1—N3 link is equal to the cost of the N1—N2—N3 path, then N1 will try to split the load destined toward N3 across the two paths.

Given the offered traffic in Table 1 the loading on N1—N3 is reduced to 67.87% but the link loading on the path N2—N3 becomes 113.12%. Ideally node N1 should put 1/3 of the traffic toward N3 on the path N1—N2—N3 and 2/3 on the path N1—N3. To know to do this N1 must know the loading on N2–N3.

Nodes	Node Names	No Split	ECMP Traffic	OMP Traffic
n3-n1	Node 3 -> Node 1	60	30	40
n1-n3	Node 1 -> Node 3	60	30	40
n3-n2	Node 3 -> Node 2	20	50	40
n2-n3	Node 2 -> Node 3	20	50	40
n2-n1	Node 2 -> Node 1	10	40	30
n1-n2	Node 1 -> Node 2	10	40	30

Table 1: Traffic loading for the example in Figure 1

This is where Optimized Multipath (OMP) provides additional benefit over ECMP. Ignoring for the moment how node N1 knows to put 1/3 of the traffic toward N3 on the path N1—N2—N3, the way this is accomplished from a forwarding standpoint is to move the boundary in the forwarding structure from the default value of 1/2 of 65536 to about 1/3 of 65536. If there are a very large set of source and destination host addresses pairs, then the traffic will be split among the 65536 possible hash values. This provides the means for a very fine granularity of adjustment.

Having explained how a fine granularity of forwarding adjustment can be accomplished, what remains is to define how nodes in a large topology can know what the loading levels are elsewhere in the topology and defining an algorithm which can allow autonomous unsyncronized decisions on the parts of many routers in a topology to quickly converge on a near optimal loading without the risk of oscillation.

2 Flooding Loading Information

Loading information is flooded within an OSPF area using Opaque LSAs [1]. Area local scope (link-state type 10) link state attributes are flooded containing an "Opaque Type" of LSA_OMP_LINK_LOAD or LSA_OMP_PATH_LOAD. The type LSA_OMP_LINK_LOAD Opaque LSA is used to flood link loading information within an area. The type LSA_OMP_PATH_LOAD Opaque LSA is used to flood loading information for use with inter-area routes. Loading information obtained from an exterior routing protocol may also be considered if available. The means of passing loading information in an exterior routing protocol is beyond the scope of this document.

2.1 Link Loading Information

Within an area link loading is flooded using the type LSA_OMP_LINK_LOAD Opaque LSA. The format of this LSA is described in Appendix A.

The "Opaque Information" in the type LSA_OMP_LINK_LOAD Opaque LSA contains the following.

1. a measure of link loading in each direction as a fraction of link capacity,

2. a measure of packets dropped due to queue overflow in each direction (if known) expressed as a fraction,
3. the link capacity in kilobits per second (or unity if less than 1000 bytes per second).

Generally the number of ouput packets dropped will be known. In designs where drops occur on the input (generally a bad design practice), the rate of input queue drops should be recorded. These measures of loading and drop are computed using the interface counters generally maintained for SNMP purposes, plus a running count of output queue drops if available. The counters are sampled every 15 seconds.

The previous value of each of the counters is substracted from the current value. The counters required are 1) bytes out, 2) bytes in, 3) packets out, 4) packets in, 5) output queue drops, and 6) input queue drops. These counters should already exist to satisfy SNMP requirements.

A value of instantaneous load in each direction is based on byte count and link capacity. An instantaneous output queue drop rate is based on queue drops and packet count. Some of the values are filtered as described in Appendix B.1.

The last time that a type LSA_OMP_LINK_LOAD Opaque LSA with the same Opaque ID was sent is recorded and the values sent are recorded. For the purpose of determining when to reflood, an equivalent loading figure is used. The computation of equivalent loading is described in Section 2.3.

The higher of the current equivalent loading computation and the previous is used when determining whether to send the type LSA_OMP_LINK_LOAD Opaque LSA. The type LSA_OMP_LINK_LOAD Opaque LSA is flooded according to elapsed time since last flooded, the current equivalent load, and the difference between the equivalent load of the most loaded and least loaded path. The reflooding decision is described in detail in Appendix B.1

The point of this graduated scale is to reduce the amount of flooding that is occurring unless links are in trouble or undergoing a significant traffic shift. Change may occur in a quiescent network due to failure external to the network that causes traffic to take alternate paths. In this case, the more frequent flooding will trigger a faster convergence. Traffic shift may also occur due to shedding of loading by the OMP algortihm itself as the algorithm converges in response to an external change.

2.2 Path Loading Information

Path loading information regarding an adjacent area is flooded by an Area Border Router (ABR) using the type LSA_OMP_PATH_LOAD Opaque LSA. The format of this LSA is described in Appendix A.

The “Opaque Information” in the type LSA_OMP_PATH_LOAD Opaque LSA contains the following.

1. the highest loading in the direction toward the destination as a fraction of link capacity,
2. a measure of total packet drop due to queue overflow in the direction toward the destination expressed as a fraction,
3. the smallest link capacity on the path to the destination.

These values are taken from the link on the path from the ABR to the destination of the summary LSA. The link with the highest loading may not be the link with the lowest capacity. The queue drop value is one minus the product of fraction of packets that are not dropped at each measurement point on the path (input and output in the direction of the path). The following computation is used.

$$Loss_{path} \;=\; 1 - \prod_{links} (1 - Loss_{link})$$

The path loading and path loss rate are filtered according to the same algorithm defined in the prior section. Rather than polling a set of counters the current value of the path loading and path loss rate is used. An equivalent load is calculated for each path to a summary LSA destination as described in Section 2.3. A type LSA_OMP_PATH_LOAD Opaque LSA is flooded according to the same rate schedule as described in the prior section.

An ABR may be configured to not send type LSA_OMP_PATH_LOAD Opaque LSA into any given area.

2.3 Computing equivalent loading

The equivalent load is the actual percent loading multiplied by a factor that provides an estimate of the extent to which TCP would be slowing down to avoid congestion. This estimate is based on the link bandwidth and loss rate, knowledge of TCP dynamics, and some assumption about the characteristics of the TCP flows being passed through the link. Some of the assumptions must be configured.

If loss is low, the equivalent load will be equal to the actual percent loading. If loss is high and loading is at or near 100%, then the equivalent load calculation provides a means of deciding which links are more heavily overloaded. The equivalent load figure is not intended to be an accurate pridiction of offerred load, simply a metric for use in deciding which link to offload.

Mathis et al provide the following estimate of loss given TCP window size and round trip time [4].

$$p \;<\; \left(\frac{MSS}{BW\ RTT}\right)^2$$

The basis for the estimate is that TCP slows down roughly in proportion to the inverse of the square root of loss. There is no way to know how fast TCP would be going if no loss were present if there are other bottlenecks. A somewhat arbitrary assumption is made that TCP would go no faster than if loss were at 0.5%. If loss is greater than 0.5% then TCP performance would be reduced. The equivalent loading is estimated using the following computation.

$$Load_{equiv} \;=\; Load_{actual}\ K\ \sqrt{Loss}$$

The inverse of the square root of 0.1% is 10 so 10 may be used for the value of "K".

The conversion of loss to estimated loading is not at all accurate. The non-linearity does affect the time to converge though convergence still occurs as long as loss is positively correlated to loading. This is discussed further in Section D.1.

3 Next hop structures

For intra-area routes, a separate next hop structure must exist for each destination router or network. For inter-area routes (summary routes), one next hop structure is needed for each set of equidistant ABRs. For external routes (AS-external or routes from an exterior routing protocol, such as BGP) one next hop structure is needed for each set of equidistant ASBRs.

The set of intra-area next hop structures is initialized after the OSPF SPF calculation is completed. An additional set of next hops is then added by relaxing the best path criteria.

3.1 Relaxing the Best Path Criteria

The exercise of setting link costs to produce the most beneficial set of equal costs paths is tedious and very difficult for large topologies. OSPF as defined in RFC–2328 requires that only the best path be considered. For the purpose of Optimized Multipath, this criteria can be relaxed to allow a greater number of multipaths but not to the point of creating routing loops. Any next hop which is closer in terms of costs than the current hop can be considered a viable next hop for multipath routing. If next hops were used where the cost at the next hop is equal or greater, routing loops would form.

In considering the paths beyond the next hop path, only the best paths should be considered. There is no way to determine if subsequent routers have relaxed the best path criteria. In addition, there is no need to consider the additional paths if the best path criteria is relaxed downstream. If best path criteria is relaxed downstream, the best paths must be part of the downstream next hop structure. If there are additional paths the the downstream is able to use to further distribute the load, the entire set of paths will still converge toward optimal loading.

3.2 Offloading Congestion Outside the OSPF Area

For inter-area routes or external routes, a separate next hop structure must exist for each such route is it is desireable to reduce loading outside of the area and the loading within the area is sufficiently low to safely allow this.

For intra-area routes if a type LSA_OMP_PATH_LOAD Opaque LSA exists for the summary LSA and more than one ABR is advertising the summary route and the equivalent load for the summary LSA is greater than 90% and the equivalent load within the area is sufficiently smaller than the inter-area loading, then a next hop structure can be created specifically to allow offloading of the intra-area route. For external routes, if an equivalent loading exists, and more than one ASBR is advertising the route, and the equivalent load is greater than 95% and the equivalent load within the area is sufficiently smaller than the external route loading, then a separate structure is used.

Hysterysis must be used in the algorithm for determining if an equivalent load on a summary LSA or external route is considered sufficiently larger than the intra-area equivalent load or if an external route loading is considered sufficiently larger than the inter-area equivalent load. For for the purpose of describing this algorithm one euivalent load is referred to as the more external, and the other as the more internal equivalent load.

If the more external equivalent load exceeds the more internal equivalent load by 15% and the more internal equivalent load is under 85%, then a separate next hop structure is created. If the more external equivalent load falls below 20% of the more internal equivalent load or the more internal equivalent load exceeds 98%, then an existing separate next hop structure is marked for removal and combined with the more internal next hop structure (see Section 3.3). The more external equivalent load should not fall significantly below the more internal unless either the traffic toward the more external destination increases or the loading on the more internal increases, since the more internal equivalent load will become the critical segment on the separate next hop structure if the load is sufficiently shifted but is unlikely to overshoot by 20%. These threshholds should be configurable at least per type of routes (inter-AS or external).

The degree to which Summary LSA loading and external route loading will be considered is limited. This serves two purposes. First, it prevents compensating for external congestion to the point of loading the internal network beyond a fixed threshhold. Second, it prevents triggering the removal of the next hop structure, which if allowed to occur could trigger a hysteresis loop. This mechanisms are described in Section 3.4, and Appendix C.4.

3.3 Creating and destroying next hop structures

As described in Section 3.2 separate next hop structure is needed if the loading indicated by the type LSA_OMP_PATH_LOAD Opaque LSA or exterior routing protocol is sufficiently high to require separate balancing for traffic to the summary-LSA or exterior route and the intra-AS loading is sufficiently low.

When a separate next hop structure is created, the same available paths appear in the structure, leading to the same set of ABR or ASBR. The balance on these available paths should be copied from the existing more internal next hop structure. By initializing the new next hop structure this way, a sudden change in loading is avoided if the summary route or external route sinks a great deal of traffic.

When a separate next hop structure can be destroyed, the traffic should be transitioned gradually. The next hop structure must be marked for deletion. The traffic share in this separate next hop structure should be gradually changed so that it exactly matches the traffic share in the more internal next hop structure. The gradual change should follow the adjustment rate schedule described in Section 4.1 where the move increment is increased gradually as moves continue in the same direction. Once the separate next hop structure marked for deletion matches the more internal next hop structure, the summary route or external route can be changed to point to the more internal next hop structure and the deletion can be made.

3.4 Critcally loaded segment

For every set of paths, one link or part of the path is identified as the "critcally loaded" segment. This is the part of the path with the highest equivalent load as defined in Section 2.3. For an inter-area route with a separate next hop structure, the critically loaded segment may be the critcally loaded segment for the intra-area set of paths, or it may be the summary LSA if the equivalent load on the summary LSA is greater. For an external route with a separate next hop structure, the critcally loaded segment may be the critcally loaded segment for the internal route or it may be the external route if the equivalent load on the external route is greater. In considering loading reported for summary LSA or external routes, the loading may be clamped to some configured ceiling (see Appendix C.4). If intra-area loading exceeds this ceiling, the summary LSA loads or external routes loads are in effect ignored.

Each next hop structure has exactly one "critcally loaded" segment. There may be more than one path in the next hop structure sharing this critcally loaded segment. A particular Opaque LSA may be the critcally loaded segment for no next hop structures if it is lightly loaded. Another Opaque LSA may be the critcally loaded segment for many next hop structures if it is heavily loaded.

4 Adjusting Equal Cost Path Loadings

Adjustments are made to a next hop structure to reflect differences in loading on the paths as reported by the type LSA_OMP_LINK_LOAD Opaque LSA and type LSA_OMP_PATH_LOAD Opaque LSA. Section 3.4 describes the selection of a "critically loaded segment" which is used to determine when to make adjustments and the size of the adjustments. An adjustment to loading of a given set of equal cost paths is made when one of two conditions are true. Either the "critically loaded segment" has been reflooded, or a criteria is met involving 1) the difference between the equivalent load of the"critically loaded segment" and the lightest loaded path, 2) the equivalent load of the "critically loaded segment", 3) the type of destination, intr-area, inter-area, or external, and 4) the amount of time since the last load adjustment. The details of this conditional are described in Appendix B.

The reflooding algorithm is designed to be slightly less aggressive than the adjustment algorithm. This reduces the need to continuously flood small changes except in conditions of overload or substantial change in loading. Some overshoot may occur due to adjustments made in the absence of accurate knowledge of loading.

4.1 Load Adjustment Rate

In order to assure stability the rate of adjustment must be sufficiently limited. An adaptive adjustment rate method is used.

A "critcally loaded" segment for a next hop structure is determined as described in Section 3.4. When the type LSA_OMP_LINK_LOAD Opaque LSA or type LSA_OMP_PATH_LOAD Opaque LSA for this segment is updated or the criteria in Appendix B is met, load is shed from all paths in the next hop structure that include that segment toward all paths in the next hop structure that do not include that segment. A separate set of variables controlling rate of adjustment is kept for each path receiving load.

The number of paths usually exceeds the number of next hops. The distinction between paths which share a next hop is important if one of the paths sharing a next hop goes down (see Section 4.2). This distinction is only needed in making the computations. When moving the next hop structure into the data structures used for forwarding, paths which share a common next hop may be combined.

The following variables are kept for each path in a next hop structure.

1. The current "traffic share" (an integer, the range is 0 to 65355 for a CRC16 hash),
2. The current "move increment" used when moving traffic toward this path (an integer, the range is 0 to 65355 for a CRC16 hash),
3. The number of moves in the same direction, referred to as the "move count".

If there is no prior history for a path, then the move increment is initialized to about 1% or 650. The number of moves in the same direction is initialized to 0. No loading adjustment is made on the first iteration.

If the critcally loaded segment has changed, all paths now containing the critically loaded segment are first examined. The lowest move increment of any one of these paths is noted.

The move increment is adjusted for each path before any traffic is moved. One of the following actions is taken for each path.

1. If the path contains the critcally loaded segment its move increment is left unchanged.
2. If the path does not contain the critcally loaded segment but the critically loaded segment has changed and the path contains the prior critcally loaded segment, then first the move increment is replaced with the lowest move increment from any of the paths containing the critically loaded segment unless the move increment is already lower, then the move increment is cut in half.
3. If the path does not contain the critcally loaded segment and either the critically loaded segment has not changed, or the path does not contain the prior critcally loaded segment, then the move increment is increased.

The amount increase in the move increment is described in Appendix B.4. The increase is designed to minimize the possibility of dramatic overshoot due to to great an increase in adjustment rate.

The move increment is never less than one and the increase in move increment is never less than one. The move increment is never allowed to exceed the size of the hash space divided by the number of equal cost paths in the next hop structure.

The dramatic decrease in move increment when move direction is reversed and the slow increase in move increment when it remains in the same direction keeps the algorithm stable. The exponential nature of the increase allows the algorithm to track externally caused changes in traffic loading.

The traffic share allocated to a path not containing the critcally loaded segment is incremented by the move amount for that path and the traffic share allocated to the path or paths containing the the critcally loaded segment are reduced by this amount divided by the number of paths containing the critcally loaded segment. This adjustment is described in pseudocode in Appendix B.4.

This adjustment process is repeated for each path in a next hop structure. The new hash space boundaries are then moved to the forwarding engine.

4.2 Dealing with Link Adjacency Changes

Link failures do occur for various reasons. OSPF routing will converge to a new set of paths. Whatever load balance had previously existed will be upset and the load balancing will have to converge to a new load balanced state. Previous load balancing parameter should remain intact to the extent possible after the SPF calculation has completed. Adjustments for new or deleted paths in the SPF result are described here. These adjustments must be made after the best path criteria is relaxed as described in Section 3.1.

4.2.1 Impact of Link Adjacency Changes

Links which are intermitent may be the most harmful. The OSPF "Hello" protocol is inadequate for handling intermitent links. When such a link is up it may draw traffic during periods of high loss, even brief periods of complete loss.

The inadequacies of the OSPF "Hello" protocol is well known and many implementations provide lower level protocol state information to OSPF to indicate a link in the "down" state. For example, indications may include carrier loss, excessive framing errors, unavailable seconds, or loss indications from PPP LQM.

Even where the use of a link is avoided by providing indication of lower level link availability, intermitent links are still problematic. During a brief period immediately after a link state attribute is initially flooded OSPF state can be inconsistent among routers within the OSPF area. This inconsistency can cause intermittent routing loops and have a severe short term impact on link loading. An oscillating link can cause high levels of loss and is generally better off held in the neighbor adjacency "down" state. The algorithm described in the [7] can be used when advertising OSPF type 1 or type 2 LSA (router and network LSAs).

Regardless as to whether router and network LSAs are damped, neighbor adjacency state changes will occur and router and network LSAs will have to be handled. The LSA may indicate an up transition or a down transition. In either an up or down transition, when the SPF algorithm is applied, existing paths from the router doing the SPF to specific destinations may no longer be usable and new paths may become usable. In the case of an up transition, some paths may no longer be usable because their cost is no longer among those tied for the best. In the case of down transitions, new paths may become usable because they are now the best path still available.

4.2.2 Handling the Loss of Paths

When a path becomes unusable, paths which previously had the same cost may remain. This can only occur on an LSA down transition. A new next hop entry should be created in which the proportion of source/destination hash space allocated to the now infeasible path is distributed to the remaining paths proportionally to their prior allocation. Very high loading percentages should result, triggering an increase in LSA_OMP_LINK_LOAD Opaque LSA flooding rate until convergence is approached.

4.2.3 Handling the Addition of Paths

When a new path becomes usable it may be tied for best with paths carrying existing traffic. This can only occur on an LSA up transition. A new next hop entry should be created in which the loading on the new path is zero. If such a path were to oscillate, little or no load would be affected. If the path remains usable, the shift of load to this path will accellerate until a balance is reached.

If a completely new set of best paths becomes available, the load should be split across the available paths, proportional to 10% of link capacity plus the remaining link bandwidth as determined by prior LSA_OMP_LINK_LOAD Opaque LSA values. The contribution of link capacity in the weighting should be configurable. See Appendix C.5.

Acknowledgements

Numerous individual have provided valuable comments regarding this work. Dave Ward made a very substantial contribution by pointing out that the best path criteria could be relaxed. Dave Ward, John Scudder, Tony Li, and Daniel Awduche have provided particularly valuable review and comments.

References

[1] R. Coltun. The ospf opaque lsa option. Technical Report RFC 2370, Internet Engineering Task Force, 1998. ftp://ftp.isi.edu/in-notes/rfc2370.txt.

[2] Atul Khanna and John Zinky. The revised ARPAnet routing metric. In *SIGCOMM Symposium on Communications Architectures and Protocols*, pages 45–56, Austin, Texas, September 1989. ACM.

[3] Steven H. Low and P. Varaiya. Dynamic behavior of a class of adaptive routing protocols (IGRP). In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, pages 610–616, March/April 1993.

[4] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM Computer Communication Review*, 27(3), July 1997.

[5] J. Moy. Ospf version 2. Technical Report RFC 2328, Internet Engineering Task Force, 1998. ftp://ftp.isi.edu/in-notes/rfc2328.txt.

[6] W. Stevens. Tcp slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. Technical Report RFC 2001, Internet Engineering Task Force, 1997. ftp://ftp.isi.edu/in-notes/rfc2001.txt.

[7] Curtis Villamizar, R. Govindan, and R. Chandra. Bgp route flap damping. Internet Draft (Work in Progress) draft-ietf-idr-route-damp-03, Internet Engineering Task Force, 5 1998. ftp://ftp.isi.edu/internet-drafts/draft-ietf-idr-route-damp-03.txt.

Security Considerations

The use of the Opaque LSAs described in this document do no impact the security of OSPF deployments. In deployments which use a strong OSPF authentication method, and require signatures on LSA from the originating router, no leveraging of a partial compromise beyond a localized disruption of service is possible. In deployments which use a strong OSPF authentication method, but do not require signatures on LSA from the originating router, compromise of a single router can be leveraged to cause significant disruption of service with or without the use of these Opaque LSA, but disruption of service cannot be achieved without such a compromise. In deployments which use a weak OSPF authentication method, significant disruption of service can be caused through forged protocol interactions with or without the use of these Opaque LSA.

Author's Addresses

Curtis Villamizar
ANS Communications
<curtis@ans.net>

A Data Formats

```
+----------------+----------------+----------------+----------------+
|   Link State Advertisement Age  |   Options      |   LSA Type     |
+----------------+----------------+----------------+----------------+
|   Opaque Type  |   Opaque ID                                      |
+----------------+----------------+----------------+----------------+
|   Advertising Router                                              |
+----------------+----------------+----------------+----------------+
|   Link State Advertisement Sequence Number                        |
+----------------+----------------+----------------+----------------+
|   LSA Checksum                  |   LSA length                    |
+----------------+----------------+----------------+----------------+
|   Version      | Reference Type | Packing Method | BW Scale       |
+----------------+----------------+----------------+----------------+
|   Reference to a Type 1-5 LSA (32 or 64 bits, see below)          |
+----------------+----------------+----------------+----------------+
|   Packed Loading Information (variable length, see below)         |
```

```
+---------------+---------------+---------------+---------------+
```

The "LSA Age", "Options", and "LSA Type" are part of the Link State Advertisement Format described in Appendix A of RFC–2328. The LSA Type is is 10, signifying an Opaque LSA with Area local scope, as defined in RFC–2370. RFC–2370 splits the Link State ID field into two part, Opaque Type, and Opaque ID. The Opaque Type contains either the value LSA_OMP_LINK_LOAD or LSA_OMP_PATH_LOADas described in Section 2. The "Advertising Router", "Link State Advertisement Sequence Number", "LSA Checksum", and "LSA length" are part of the Link State Advertisement Format described in Appendix A of RFC–2328. The remainder of the packet is specific to Opaque Type LSA_OMP_LINK_LOAD or LSA_OMP_PATH_LOADLSAs.

Opaque Type The Opaque Type will contain the value LSA_OMP_LINK_LOAD or LSA_OMP_PATH_LOADas described in Section 2. Numeric values will be requested from IANA.

Opaque ID The Opaque ID will contain an integer which will be unique per router and interface, virtual interface, or MAC address for which loading is reported. These numbers are only of significance to the advertising router except as a means of identification of subsequent LSAs.

For a LSA_OMP_LINK_LOAD Opaque LSA, the "Opaque ID" must contain a 24 bit integer that is unique to the link or virtual link. The method of assignment of these 24 bit integers is a local matter. A router must be capable of uniquely identify an interface using a 24 bit number.

For a LSA_OMP_PATH_LOAD Opaque LSA, the "Opaque ID" must contain a 24 bit integer that is unique to a summary LSA or AS-external LSA advertised by the same router. The method of assignment of these 24 bit integers is a local matter. A router must be capable of uniquely identify a summary LSA or AS-external LSA using a 24 bit number.

Version The version number is 1.

Reference Type The Reference Type indicates the type of LSA that is being referenced in the "Reference to a Type 1-5 LSA" field.

Packing Method The Packing Method is an integer that indicates how the "Packed Loading Information" field is formated.

BW Scale Bandwidth numbers must be scale by shifting the 32 bit integer left by this amount. If this value is non-zero, integers larger than 64 bits must be used.

Reference to a Type 1-5 LSA This field contains the 32 bit "Link State ID" field of a Type 1-5 LSA. Type 1-5 indicate:

1. Router-LSAs
2. Network-LSAs
3. Summary-LSAs (IP network)
4. Summary-LSAs (ASBR)
5. AS-external-LSAs

Loading information for a Type 1 LSA, a Router-LSA, is sent as aLSA_OMP_LINK_LOAD Opaque LSA. For a Type 1 LSA the "Link State ID" field is followed by a 32 bit "Link ID". This identifies a single link. There are four types of Links.

1. Point-to-point connection to another router

2. Connection to a transit network
3. Connection to a stub network
4. Virtual link

Normally loading information is provided for a Link Type 1. Loading information may also be provided for a Link Type 2 or 3. Loading information cannot be provided to a Link Type 4.

Loading information is not provided for Type 2 LSAs, Network-LSAs.

Loading information for Type 3 and Type 4 LSAs, Summary-LSAs for IP networks in another area and Summary-LSAs for ASBRs, is sent as a LSA_OMP_PATH_LOAD Opaque LSA as described in Section 2.2. For a Type 3 and Type 4 LSA there is no information in the Reference following the "Link State ID".

Loading information for Type 5 LSAs, AS-external-LSAs, is sent as a LSA_OMP_PATH_LOAD Opaque LSA as described in Section 2.2. For a Type 5 LSA there is no information in the Reference following the "Link State ID".

Packed Loading Information The format of the Packed Loading Information depends on the value of the Packing Method field. Currently only the value 1 is defined.

The following format is used when the Packing Method field contains 1. The LSA must be ignored if values other than 1 are found in Packing Method.

```
+---------------+---------------+---------------+---------------+
|  In Scaled Link Capacity in kilobits per second               |
+---------------+---------------+---------------+---------------+
|  In Link Loading Fraction     | In Link Drop Fraction (packets) |
+---------------+---------------+---------------+---------------+
|  Out Scaled Link Capacity in kilobits per second              |
+---------------+---------------+---------------+---------------+
|  Out Link Loading Fraction    | Out Link Drop Fraction (packet) |
+---------------+---------------+---------------+---------------+
```

The Scaled Link Capacity is an unsigned integer in kilobits per second. If this would be larger than a 32 bit integer, the value should be shifted to the right until the top bit is in the 32 bit number MSB and the number of bit shifts recorded in the BW Scale field

The Link Loading Fraction is a 16 bit unsigned integer from 0 to 0xffff representing capacity in bytes being used relative to capacity in bytes per the measurement interval. The hex number 0x10000 would represent unity, but if full loading has been acheived or due to counting or truncation error, greater than full loading, substitute 0xffff. The Link Drop Fraction is a 16 bit unsigned integer from 0 to 0xffff representing number of packets dropped relative to the number of packets received. This value can be derived from the change in two MIB-2 counters (ifOutDiscard¡¡16)/ifInPacket. The hex number 0x10000 would represent unity (all of the packets being dropped) so 0xffff must be substituted.

B Concise Statement of the Algorithms

An OSPF router may play one of two roles, or both. An interior routers and edge routers will flood loading information. A router may choose not to flood information if it is beleived that there is no way that the interface could become congested. An ingress router will process loading information and if it has equal cost paths will balance load across those paths.

The description of algorithms is broken down into the following subsections.

Flooding Loading Information Appendix B.1

Building Next Hop Structures Appendix B.2

Processing Loading Information Appendix B.3

Adjusting Loading Appendix B.4

The algorithms are defined in the following section in pseudocode.

B.1 Flooding Loading Information

It is assumed that counters are large enough to avoid multiple overflow (ie: 64 bit counters are used for high speed interfaces) and that counter overflow is easily detected is compensated for in counter deltas. It is assumed that ifInDiscard and ifOutDiscard accurately counts all queueing drops.

The following counters are sampled at 15second intervals: ifInOctets, ifOutOctets, ifInPacket, ifOutPacket, ifInDiscard and ifOutDiscard. The value if ifInSpeed and ifOutSpeed is assumed to be constant. Some state must be stored. The previously used value of each raw counter is needed to compute deltas. State variables InFilteredUtil, OutFilteredUtil, InLoss, OutLoss, InEquivLoad and OutEquivLoad must be saved. The last time a reflooding occurred must also be stored.

The input and output utilizations are expressed as fractions using ifInOctets, ifOutOctets, ifInSpeed, and ifOutSpeed. Call the raw 15second fractional utilizations InRawUtil and OutRawUtil. Compute the following filtered values for both In and Out, making sure to save the previous values.

```
PrevFilteredUtil = FilteredUtil;
if (RawUtil < FilteredUtil) {
    FilteredUtil -= (FilteredUtil >> 3);
    FilteredUtil += (RawUtil >> 3);
} else if (RawUtil > FilteredUtil) {
    FilteredUtil -= (FilteredUtil >> 1);
    FilteredUtil += (RawUtil >> 1);
}
```

InLoss and OutLoss is computed using the ratio of the deltas of Discard to Packet SNMP counters. Next compute an “equivalent loading” for the purpose of determining whether to reflood.

```
PrevEquivLoad = EquivLoad;
if (Loss < 0.005) {
    EquivLoad = FilteredUtil;
} else {
    if (Loss <= 0.09) {
        LossComp = 10 * sqrt(Loss);
    } else {
        LossComp = 3;
    }
    EquivLoad = FilteredUtil * LossComp;
}
```

A square root is somewhat time consuming to compute, so a table lookup can be done to avoid this computation. Increments of 0.1% loss would yield an 90 entry table. A 128-512 entry table would be adequate. The table can be sized so a shift and mask can be used to index it. The computation could then be done with a table lookup, a shift, and an integer multiply. At most this needs to be done for links with nonzero loss once every 15 seconds.

Then decide whether to flood based on the greater of the relative change in InEquivLoad or OutEquivLoad and on the time elapsed since the last reflooding (taking care not to divide by zero).

```
Diff = max(abs(InEquivLoad - InPrevEquivLoad)
               / InPrevEquivLoad,
           abs(OutEquivLoad - OutPrevEquivLoad)
               / OutPrevEquivLoad);
Load = max(InEquivLoad, OutEquivLoad)
Elapsed = Now - LastReflood;
if (((Load > 1.00) && (Diff > 0.05) && (Elapsed >= 30))
    || ((Load > 1.00) && (Diff > 0.02) && (Elapsed >= 60))
    || ((Load > 1.00) && (Diff > 0.01) && (Elapsed >= 90))
    || ((Load > 1.00) && (Elapsed >= 180))
    || ((Load > 0.90) && (Diff > 0.05) && (Elapsed >= 60))
    || ((Load > 0.90) && (Diff > 0.02) && (Elapsed >= 240))
    || ((Load > 0.90) && (Diff > 0.01) && (Elapsed >= 480))
    || ((Load > 0.90) && (Elapsed >= 600))
    || ((Load > 0.70) && (Diff > 0.10) && (Elapsed >= 60))
    || ((Load > 0.70) && (Diff > 0.05) && (Elapsed >= 120))
    || ((Load > 0.70) && (Diff > 0.02) && (Elapsed >= 480))
    || ((Load > 0.70) && (Elapsed >= 900))
    || ((Load > 0.50) && (Diff > 0.10) && (Elapsed >= 60))
    || ((Load > 0.50) && (Diff > 0.05) && (Elapsed >= 300))
    || ((Load > 0.25) && (Diff > 0.25) && (Elapsed >= 120))
    || ((Load > 0.25) && (Elapsed >= 1200))
    ) {
        /* we passed one of the tests so flood it */
        LastReflood = Now;
        ...
```

If the decision is made to reflood an LSA according to the test above, then input and output FilteredUtil and Loss must be packed into an LSA and flooded.

The 15second timer interval should be jittered by a random number in the range of plus or minus 5 seconds (obviously taking the actual time interval into account in computing rates).

B.2 Building Next Hop Structures

The OSPF SPF calculation is done as per RFC–2328. The arrival of loading information does not require new SPF calculations since neither reacheability or costs are changed.

The SPF calculation yields the shortest paths from the given node to all other routers and networks in the OSPF area. In some cases multipaths will already exist. For all destinations, every feasible hop is examined, and paths through next hops that simply provide an improvement are added, as described in Section 3.1.

After completing the SPF calculation and relaxing the best path criteria, intra-area multipath sets are recorded as next hop structures. If a previous SPF had been in use, loadings are transfered to the new set of data structures and links are added or removed as needed as described in Section 4.2.

After recording the intra-area next hop structures, the existing set of inter-area next hop structures and the set of external route next hop structures is examined. Paths are added or removed from next hop structures as needed, as described in Section 3, Section 3.3, and Section 4.2.

Inter-area and external routes map onto the intra-area routing. These therefore share the same set of paths and the same next hop structure as the intra-area route to the nearest ABR or ASBR. Next hop structures may be needed to reach any one in a set of ABRs or ASBRs if 1) there are multiple ABRs equally distant to a summary route or 2) multiple ASBRs equally distant advertising an external route at the same cost, 3) relaxing the best path criteria for an intra-area destination results in going to a next-hop which does not share the same closest ABR or ASBR.

Next hop structures may also be needed to offload paths in adjacent areas or external paths. The following conditional is used to determine whether a next hop structure should be added for a SummaryLSA.

```
    if (IntraAreaLoad < 85%
        && SummaryLSALoad > 90%
        && SummaryLSALoad - IntraAreaLoad > 15%) {
            /* add a next hop structure */
            ...
```

The conditional for an external route is the same, except the intra-area load would be a more internal load, intra-area, or Summary LSA, and the 90% threshhold is increased to 95%.

The following conditional is used to determine is an existing separate next hop structure for a Summary LSA or external route should be deleted.

```
    if (MoreInternalLoad > 98%
|| MoreInternalLoad - MoreExternalLoad > 20%) {
            /* delete a next hop structure */
            ...
```

B.3 Processing Loading Information

Adjustments to loading may be triggerred by one of two events. When a new loading LSA is received, if the LSA corresponds to the most heavily loaded link for a next hop structure, then the next hop structure should be readjusted immediately. The last time each next hop structure has been readjusted must be maintained and periodically readjusted. Timer events are handled as follows.

```
foreach NextHopStruct ( AllNextHopStructs ) {
    Elapsed = Now - LastReadjust[NextHopStruct];
    MinLoaded = MinEquivLoad(NextHopStruct);
    MaxLoaded = MaxEquivLoad(NextHopStruct);
    AbsDiff = MaxLoaded - MinLoaded;
    if (((Elapsed >= 60))
         && (AbsDiff > 0.045) && (MaxLoaded > 0.95))
        || ((Elapsed >= 90))
            && (AbsDiff > 0.03) && (MaxLoaded > 0.95))
        || ((Elapsed >= 120))
            && (AbsDiff > 0.01) && (MaxLoaded > 0.97))
        || ((Elapsed >= 240))
            && (AbsDiff > 0.005) && (MaxLoaded > 0.98))
        || ((Elapsed >= 90))
            && (AbsDiff > 0.05) && (MaxLoaded > 0.90))
        || ((Elapsed >= 120))
            && (AbsDiff > 0.03) && (MaxLoaded > 0.90))
        || ((Elapsed >= 180))
            && (AbsDiff > 0.01) && (MaxLoaded > 0.90))
        || (Elapsed >= 300))
            ) {
               /* we need to readjust this multipath set */
               ...
```

This loop and conditional results in a subset of the available next hop structures being adjusted based on the timer. The same effect may be accomplished by determining when a next hop structure will need to be adjusted if no further flooding changes arrive and queueing next hop structures on lists according to how long they will remain idle.

B.4 Adjusting Loading

A next hop structure will need to be adjusted when one of the two criteria in the prior section is met. The adjustment procedure itslef relies upon the following stored state.

```
NextHopStruct {
    LastReadjust;
    PrevCriticalSegment;
    TotalPaths;
```

```
        SetofPaths (
            Path;
            bit HasCriticalSegment,
                HasPrevCriticalSeg;
            TrafficShare;
            MoveIncrement;
            MoveCount;
        );
    };
```

Before the path move increments are adjusted, a preliminary pass is made over the next hop structure. This pass notes which paths contain the prior critical segment, notes which paths contain the current critical segment and counts the number of paths containing the current critical segment.

```
    NumberWithCritical = 0;
    MinRateWithCritical = 65536;
    foreach Path ( SetofPaths ) {
        SetOrClear HasCriticalSegment;
        SetOrClear HasPrevCriticalSeg;
        if (HasCriticalSegment) {
            ++NumberWithCritical;
            if (MoveIncrement < MinRateWithCritical)
                MinRateWithCritical = MoveIncrement;
        }
    }
```

Next the move increments for each path is adjusted.

```
    foreach Path ( SetofPaths ) {
        if (HasCriticalSegment)
            continue;
        if (!HasPrevCriticalSeg) {
            ++MoveCount;
            if (MoveCount > 4) {
                Increase = MoveIncrement
                    / (2 * (1 + NumberWithCritical));
            } else {
                Increase = MoveIncrement
                    / (4 * (1 + NumberWithCritical));
            }
            if (Increase < 1)
                Increase = 1;
            MoveIncrement += Increase;
        } else {
            if (MoveIncrement > MinRateWithCritical)
                MoveIncrement = MinRateWithCritical;
            MoveIncrement /= 2;
            MoveCount = 0;
```

```
        }
        if (MoveIncrement < 1)
            MoveIncrement = 1
        if (MoveIncrement > 65635)
            MoveIncrement = 65635;
    }
```

Then traffic share is adjusted.

```
    foreach Path1 ( SetofPaths ) {
        if (!Path1.HasCriticalSegment)
            continue;
        foreach Path2 ( SetofPaths ) {
            if (Path2.HasCriticalSegment)
                continue;
            Move = Path2.MoveIncrement / NumberWithCritical;
            if (Move < 1)
                Move = 1;
            if (Move > (65536 - Path2.TrafficShare)) {
                Move = 65536 - Path2.TrafficShare;
                Path2.MoveIncrement = Move;
            }
            if (Move > Path1.TrafficShare)
                Move = Path1.TrafficShare;
            Path2.TrafficShare += Move;
            Path1.TrafficShare -= Move;
        }
    }
```

The traffic shares for paths sharing a common next hop are then summed and the appropriate information is transferred to the forwarding data structures.

C Configuration Options

Many of the capabilities described here must be configurable. The ability to enable and disable capability subsets is needed. Many parameters used by the algorithm should also be configurable.

C.1 Capability Subsets

There should at least be the ability to enabled and disabled the following.

```
    default description of capability
      ON   Flooding any loading information
      ON   Flooding loading information for specific links
```

```
-    Relaxing best path criteria
-    Adjusting traffic shares (default to even split)
OFF  Flooding loading information for Summary LSA
OFF  Flooding loading information for specific Summary LSA
OFF  Flooding loading information for external routes
OFF  Flooding loading information for specific external routes
OFF  Considering loading information for Summary LSA
OFF  Considering loading information for specific Summary LSA
OFF  Considering loading information for external routes
OFF  Considering loading information for specific external routes
```

Flooding and considering Summary LSA and external route loading information should be disabled by default. Flooding loading information should be enabled by default. Relaxing best path criteria and adjusting traffic shares could be enabled or disabled by default, depending on vendor preference.

C.2 Parameters for Equivalent Load Calculation

The following values affect the computation of equivalent load.

```
default       description of parameter
  10     The value of K in ``equiv-load = load * K * sqrt(loss)''
  0.5%   The minimum loss rate at which to compensate for loss
  9%     The maximum loss rate above which compensation is fixed
```

C.3 Parameters for Relaxing the Best Path Criteria

The following parameter affects the number of next hops and paths added as a result of relaxing the best path criteria. For example, increasing the mtric difference to 2 would require the next hop to be a metric of "2" closer than the current distance to the destination, and reduce the number of paths added.

```
default       description of parameter
  1      The metric difference required to relaxing best path
```

C.4 Parameters for Loading Outside of the OSPF Area

The following parameters affect the creation of separate next hop structures to compensate for loading on Summary LSA and external routes when the those loadings are high and intra-AS loadings are substantially lower.

```
default       description of parameter
  15%    The loading difference to consider a more external load
```

```
             over a more internal load
      85%    The maximum internal loading where a more external load
             will become eligible for consideration
      90%    The minimum loading in which a Summary LSA will be
             considered over a an intra-area loading
      95%    The minimum loading in which an external route will be
             considered over a an intra-area loading
      20%    The load difference at which an external load will be
             removed from consideration due to being well under the
             internal load.
      94%    The maximum value used for in place of loading for a
             Summary LSA when performing traffic share adjustment.
      98%    The internal loading where a Summary LSA will be
             removed from consideration over the internal load
      90%    The maximum value used for in place of loading for a
             external route when performing traffic share adjustment.
      98%    The internal loading where a external route will be
             removed from consideration over the internal load
```

Limiting the compensation that will be made to accommodate external loading is consistent with the reason for considering external routes. Rarely does a business go out of its way to improve the performance of their competitor's product, a network service or otherwise. Avoiding congestion in a peer's network is doing a favor for one's own customers by not sending their traffic into known areas of congestion but only if it does not significantly impact congestion in one's own network.

Limiting the compensation for Summary LSA loading and external route loading avoids triggering the hysteresis criteria where a separate next hop structure is deleted if an internal loading exceeds a fixed threshhold. In effect the loading on the Summary LSA loading and external route loading is ignored if internal loadings exceed a given threshhold, since the Summary LSA loading or external route loading will no longer be considered as the critical segment. If internal loading reaches a point where even with load balancing internal paths exceed the higher threshhold, the next hop structure will be removed.

C.5 Parameters for Initial Loading of Paths

When determining the initial loading on a new set of paths, where the destination was previously unreachable, or none of the previous paths appear in the new next hop structure, the following weighting is used.

```
weight = (link-capacity * 0.10)
       + (link-capacity * (1 - utilization))
```

The contribution of link capacity in the weighting should be configurable.

```
default        description of parameter
  10%    The fraction of total link capacity to consider in
         addition to the reported unused link capacity.
```

C.6 Parameters associated with Flooding and Traffic Share

Parameters associated with flooding rate, move increment and traffic share adjustment should not be configurable by the user or should be well hidden so as only to be accessible to developers. Adjustment of these parameters can slow convergence or increase overshoot. If the parameters are sufficiently altered instability could result. While it is likely that some improvements could result from further tuning, experimentation on live networks is not encouraged.

D Algorithm Performance

A number of simulations have been performed to test the OSPF-OMP algorithms. In these simulations the algorithm has been demonstrated to be very stable. This work has not been formally published yet but is currently available at http://engr.ans.net/ospf-omp.

The simulations done to date have only modeled behavior of load balancing intra-area routes. This would be applicable to networks in which external routing was mapped onto IGP routing with a single OSPF area. Passing loading information between areas, allowing loading in one area affect an adjacent area, has not been simulated. Similarly passing loading information with external routes and affecting loading in a peer AS has not been simulated.

D.1 Conversion from Loss to Equivalent Load

The current adjustment for loss in the equivalent load is based on the premise that traffic is dominated by TCP and that TCP flows sufficiently unconstrained by other bottlenecks and of sufficient duration exist to keep the aggrgate traffic flow elastic. This is considered a very safe assumption for Internet traffic. Enterprise networks may have a higher contribution of incompressible traffic (traffic not conforming to a form of congestion avoidance such as TCP congestion avoidance described in RFC–2001).

The assumed relationship between packet loss and link utilization is based on the work of Mathis et al [4]. The constants in this relationship cannot be determined as they depend on delay bandwith product of TCP flows, the number and duration of TCP flows, and whether TCP flows are severely constrained elsewhere.

The objective is to estimate the offered load, which cannot be measured directly when links are at full utilization, using the link utilization and loss. The load adjustment algorithm should remain stable as long as the first derivative of the estimator over offered load remains positive. If the first derivative is negative within some region, then oscillation will occur within that range of operation. The greatest risk of this occurring is in routers where active queue management is not used (ie: where drop-tail queues are used) and in particular where buffering is too small. In such cases, as offered load increases just beyond full utilization, loss increases somewhat, but utilization can drop substantially (typically to about 90increases, the estimator of offered load may decrease, causing the link to appear less loaded than another. The rather aggresive compensation for loss is intended to insure that this effect either does not occur, or occurs only within a very narrow range of offerred load at just over full utiization. If the derivative is negative within a narrow range, oscillations can occur only within that range, and the oscillations are well bounded.

D.2 Performance as traffic loads change

This work has considerable advantages over other approaches, particularly traffic engineering approaches that involve adjustment of virtual circuit layouts based on historic traffic figures. The advantage is the ability to adjust loading gradually as actual offerred traffic deviates for expected. The advantage is even greater when compared to dynamic virtual circuit layouts, using PNNI for example, since these algorithms have proven to often converge on very suboptimal layouts.

Simulations demonstrating behavior in these particular cases can be found at http://engr.ans.net/ospf-omp/ramp.html.

D.3 Convergence after a major perturbation

Simulations have been performed in which link failures are examined. Without relaxing the best path criteria, OSPF OMP is quite dependant of the set of link metrics to create a set of equal cost paths that will allow the most heavily loaded portions of the topology to be offloaded. When links fail, the set of metrics often are far from ideal for the remaining topology. Relaxing the best path criteria significantly improves the response to link failure.

Simulations are available at http://engr.ans.net/ospf-omp/fail.html.