MPLS Working Group
Internet Draft
Expiration Date: Nov 2002

Peter Ashwood-Smith
Bilel Jamoussi
Don Fedyk
Darek Skalecki
Nortel Networks

May 2002

Improving Topology Data Base Accuracy with LSP Feedback in CR-LDP

draft-ietf-mpls-te-feed-04.txt

Status of this Memo

Abstract

One key component of traffic engineering is a concept known as constraint based routing. In constraint based routing a topology database is maintained on all participating nodes. This database contains a complete list of all the links in the network that participate in traffic engineering and for each of these links a set of constraints, which those links can meet. Bandwidth, for example, is one essential constraint. Since the bandwidth available changes as new LSPs are established and terminated the topology database will develop inconsistencies with respect to the real network. It is not possible to increase the flooding rates arbitrarily to keep the database discrepancies from growing. A new mechanism is proposed whereby a source node can learn about the successes or failures of its path selections by receiving feedback from the paths it is attempting. This information is most valuable in failure scenarious but is benificial during other path setup functions as well. This

fed-back information can be incorporated into subsequent route computations, which greatly improves the accuracy of the overall routing solution by significantly reducing the database discrepancies.

1. Introduction

Because the network is a distributed system, it is necessary to have a mechanism to advertise information about links to all nodes in the network [IS-IS], [OSPF]. A node can then build a topology map of the network. This information is required to be as up-to-date as possible for accurate traffic engineered paths. Information about link or node failures must be rapidly propagated through the network so that recovery can be initiated. Other information about links that may be useful for reasons of quality of service includes parameters such as available bandwidth, and delay. The information in this topology database is often out of date with respect to the real network. Available bandwidth is the most critical of these attributes and it can drift substantially with respect to reality due to the low frequency of link state updates that can be sustained in a very large topology. The deviation in the topology database available bandwidth is refered to as being optimistic if the database shows more available bandwidth than there really is, or pessimistic if the topology database shows less bandwidth than there really is. This distinction is important to enable an efficient algorithm to deal with optimistic databases without resorting to shorter flooding intervals.

One of the major problems for a constraint based routing system is dealing with changing constraints. Obviously, since bandwidth is one of the essential constraints, dealing with the rapid changes in reserved bandwidth poses some interesting challenges. In smaller networks, one can resort to higher frequency flooding but this obviously does not scale. The feedback mechanism is particularily useful in the case of link or node failures where the rapid change and notification of resource change is crutuail to the restoration time. Feedback is work conserving in this case since the availability of feedback information minimizes the extra burden of dealing with out of date topology and resource information.

The basic proposal is to add to the signaling protocol the ability to piggyback actual link bandwidth availability information at every link that the signaling traverses. This is done as part of the reverse messaging on success or failure (mapping, release, withdraw or notification). What this means is that every time signaling messages flow backwards toward a source to tell it of the success, failure or termination of a request, that message contains a detailed slice of bandwidth availability information for the exact

path that the message has followed. This slice of reservation information, which is very up to date, is received by the source node and attached to the source node's topology database prior to

making any further source route computations. The result is that the source node's topology database will tend to stay synchronized with the slices of the network through which it is establishing paths. This is nothing more than learning from successes and failures and represents an intelligent alternative to either waiting for floods or introducing non-determinism (guessing) into the source algorithms. It is important to note that the fed-back data is never re-flooded. It simply overrides flooded information for the purpose of route computation until a superceding flood or fed-back value arrives. As such, it is not actually inserted into a topology database, most likely it simply is linked to that database as an override used only by source route computations. Also the inclusion of feedback information is optional. At a minimum the blocked or failed link is required but if processing resources are scarce the additional feedback at other hops is optional.

Operating a constraint based routing system without such feedback is inefficient at best since a source node will continue to give out incorrect route over and over again until it gets an IGP update. This could be minutes away and as a result the worst case blocking time for a new route is the minimum repeatable flooding interval (often several minutes in big networks). Alternatives to feedback mechanisms involve adding some non-determinism (randomness) to the routing algorithm in the hopes that it will stumble onto a path that works. These sorts of approaches are seen in ATM dynamic routing systems, which do not have these forms of feedback.

In order to get a good understanding of how the feedback works, imagine a network with precisely one path (with sufficient unreserved bandwidth) available from the source to the destination. Further, imagine that the topology database at the source is significantly out of date with respect to the real network in that the source topology database sees sufficient bandwidth available on many different routes to the destination. This is termed being optimistic with respect to the network since the source thinks that more bandwidth is available than there really is.

When such an optimistic source selects its first path it will likely contain links that do not in reality have sufficient unreserved bandwidth. Therefore, the path is only established up to the link that does not have sufficient bandwidth. A notification message is formatted that contains the actual unreserved bandwidth for this blocking link which flows back toward the source, collapsing the partially created path as it goes. In addition, at every link that this notification traverses, the current unreserved bandwidth information for each corresponding link is appended to the vector of unreserved bandwidth along the path. In this manner, an accurate view of the slice through the network traversed is constructed.

Eventually this message arrives back at the source node, where the vector is taken and used to temporarily override the topology database for route computations. This node has just learned from its

mistake and is now slightly less optimistic with respect to the real network conditions.

Path selection can be attempted again but this time the node will not make the same mistake it made the previous time. The link in question, at which rejection occurred the first time, will not even be eligible this time around, so a source route computation is guaranteed to produce a different path (or none). The same procedure may be repeated as many times as is necessary, each time learning from its mistakes, until eventually no paths remain in the source topology to the destination, or a path is found that works. This tendency to converge either to a solution or determine that there is no solution is an important property of a routing system (it actually behaves a lot like a depth first search). This property is not present with flooding mechanisms alone since the source node must randomly hunt, or continually make the same mistakes, or abort until the next flood arrives.

In addition to feeding back bandwidth on failure, feedback on success is recommended. This has important consequences on our ability to spread load or to spill over to new links as existing links fill. It is true that spilling over to new links does not require feedback on success since a node could simply wait for a feedback on failure, but better load spreading can be achieved earlier.

Finally, when a path is torn down the release/withdraw messages also contain bandwidth information that can be fed back to override the source topology database. This is very important during failure scenarios where the links required for rerouting the path share common sub-segments with the failed path. Without the feedback, the common sub-segments may not indicate sufficient available bandwidth until an LSA flood is received which may mean many seconds without a connection. With feedback at least the database is up to date with respect to available bandwidth up to the point of failure in the path. Also since failure involves many paths tearing down and re-establishing this is the time that it is most critical to have an accurate view.

When preemption is being employed it is also extremely important that the topology database inconsistencies be small. If not, high setup priority LSPs may unnecessarily preempt lower holding priority LSPs to obtain bandwidth that, had they had a more up to date view of unreserved bandwidth, they would have been able to find elsewhere. Since preempted LSPs may in turn preempt other LSPs in a domino like effect, the results of such database inconsistencies can have wide reaching ripple like impacts. These feedback mechanisms help reduce these occurrences significantly.

There are a number of network conditions where feedback shows its value. One can think of a constraint-based network as being in one of three conditions. The first is called ramp-up, this is when the

rate of arriving reservations exceeds the rate of departing reservations. The second is called steady state, this is when the rate of arriving reservations is about the same as the rate of departing reservations. Finally, the ramp-down condition is that which has a greater rate of departing reservations than arriving reservations.

These three network conditions show distinctly different types of error in the topology databases. In particular an optimistic view of available bandwidth by a source node is characteristic of the ramp-up condition of a network. A pessimistic view of available bandwidth by a source node is characteristic of the ramp-down condition of a network. If one plots the average error in the topology databases with respect to the real network for the three different network conditions, one will see the error slowly go positive during ramp up, slowly go negative during ramp down, and drift slowly around 0 for the steady condition. The effect of flooding on this plot is to periodically snap the error back to 0 at flooding intervals. The effect of the feedback algorithm is to bring an optimistic error back to zero without having to wait for the flood interval. On average then, the feedback algorithm tends to halve the absolute error, keeping it mostly negative or pessimistic. This makes sense since a routing system will never give paths to links that it thinks do not have resources and as a result its pessimistic view of the world stays that way until it gets a flood. This relieves the IGP updates of the most urgent requirement of flooding when bandwidth is consumed. Availability of new bandwidth occurs when paths are released or new links become available. Floods already accompany new links. Significant releases of bandwidth can be broadcast at relatively low frequencies in the order of several minutes with little operational impact.

Extensive operational experience with this feedback protocol in proprietary Nortel Networks (pre-standard CR-LDP) products has shown it to work very well for networks up to 1000 nodes with significant flooding intervals damped to several minutes. Without this protocol, these networks would block setups for up to several minutes. With this protocol, the blocking in most cases is reduced to a small number of retry attempts which is usually sub-second depending mostly on the propagation delays in the network.

These feedback algorithms have been particularly beneficial in cases of failure recovery during which the network is in a sudden condition of ramp-up. Since a large number of reservations must be remade, it is highly likely that the bandwidth reservation limits of certain key links in the network will be exceeded. Without feedback, the rerouting must block until a flood arrives telling us of the situation at those key links at which time rerouting can continue.

With feedback, the rerouting simply continues until a feedback indicates that a link is full. In addition since reservation-balancing algorithms are also often used, feedback allows the

balancing algorithms to make better distribution decisions based on immediate feedback.

We have also explored through simulation and implementation a variety of mechanisms to deal with the pessimistic error in the database. One simple proposal is to use selective forgetting. In this algorithm, a reserved bandwidth value slowly drops back to zero over a relatively short time interval. The theory being that you shift the network back to an optimistic state (by forgetting your pessimism) where the feedback algorithm will again correctly operate. These algorithms have not shown any great advantage and are actually non-optimal when the error is purely optimistic.

Other algorithmic permutations explored include such variations as:

Feeding-back to all intermediate nodes, information learned from control messages upstream of that intermediate node.

Feeding back in both directions so that both the source and destination node's databases stay synchronized.

Allowing a request to continue to its destination despite there being insufficient bandwidth at some intermediate hop. Then, rejecting the request with a full bandwidth vector slice all the way to the destination instead of just to the point of rejection.

Our simulations have not show significant benefits relative to the simpler algorithm proposed here. However, it is an interesting research topic to explore and quantify the different feedback algorithms and their impacts on blocking times so we do not want to discourage the interested reader from exploring these concepts more fully.

2. Adding feedback TLVs to CR-LDP

Two new TLVs are optionally added to the CR-LDP mapping, notification, and withdraw messages. There may be an arbitrary number of these TLV in any order or position in the message. It is recommended that they be placed such that they can be read and applied to override the topology database by scanning the message forwards and walking the topology database from the point where the last link feedback TLV left off.

Each TLV consists of the eightunreserved bandwidth values for each holding priority 0 through 7 as IEEE floating-point numbers (the units are unidirectional bytes per second). Following this are the IP addresses of the two ends of the interface. Two TLVs are possible, one for IPV4 and one for IPV6 addressing of the link.

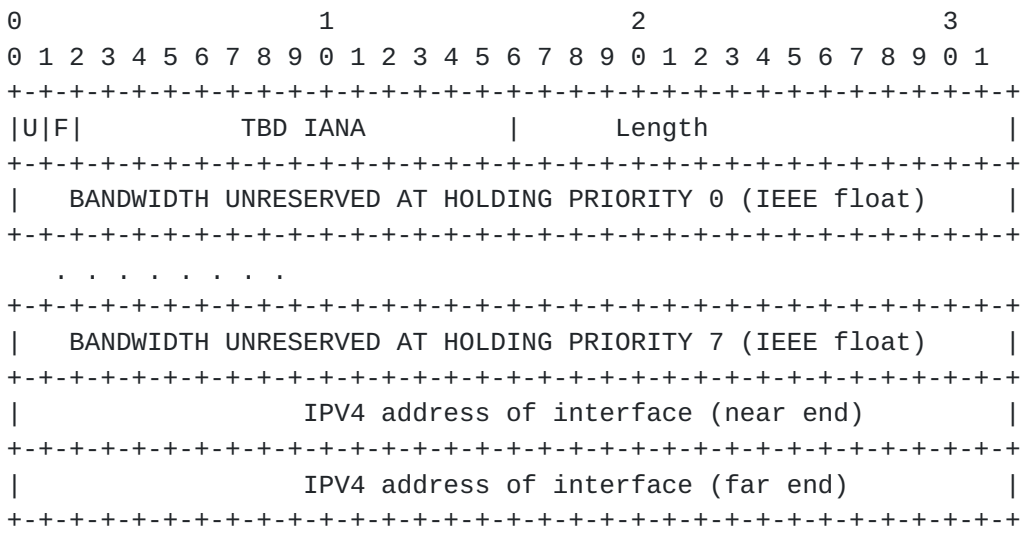
Note: the feedback TLVs may also optionally be included in query or query-reply messages in response to bandwidth update queries from an LER. Details of this mechanism are provided in [QUERY].

2.1 Bandwidth directionality considerations

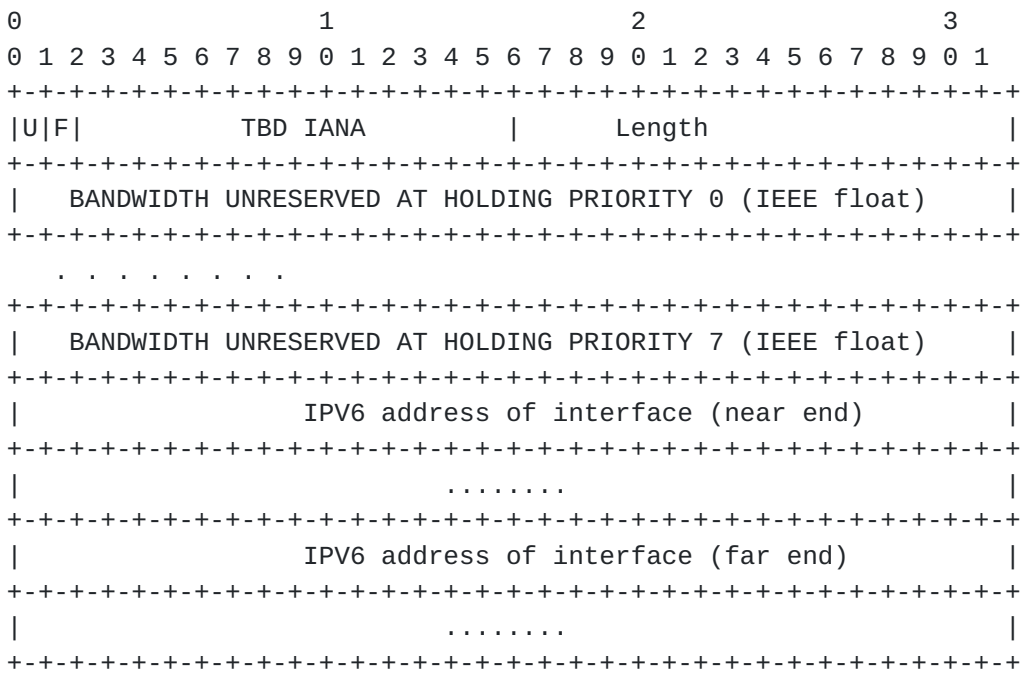
The order of the two addresses in the feedback TLV implies the direction in which the bandwidth is available. For example if the first address is A and the second address is B the bandwidth is unreserved in the A to B direction.

It is possible for an implementation to provide both the A to B direction and the B to A direction as part of the same feedback message. This is done by simply including a TLV with A, B as the addresses of the link and a different TLV with B, A as the addresses of the link. Should CR-LDP evolve to be able to support bi-directional traffic flow and reservations it is expected that bi-directional feedback would also be implemented via this mechanism.

3. IPV4 specified link feedback TLV



4. IPV6 specified link feedback TLV



5. Detailed Procedures

On receipt of a withdraw, notification, query-reply, or mapping message pertaining to a request made by CR-LDP (as opposed to LDP), a feedback TLV of the appropriate format for the interface over which the message was received is inserted into the message before forwarding it back to the source of the request. The eight bandwidth values are filled in with the outgoing bandwidth available on this

interface for each of the eightholding priorities in bytes per second. Finally the interface's address and far end address are placed in the TLV.

On receipt of a CR-LDP request message, which cannot be satisfied. A notification message is formatted normally. The eight bandwidth values are filled in with the outgoing bandwidth available on this interface for each of the eight holding priorities in bytes per second. Finally, the interface's address and far end address are placed in the TLV.

On receipt of a CR-LDP request message which has been satisfied and which results in a mapping being generated. No feedback TLV is added since the previous node will insert the proper TLV when it receives the reverse flowing mapping.

When an LDP session goes down either because of a link failure, TCP/IP timeout, keep alive timeout, adjacency timeout etc. Other LDP sessions in the module must generate either notification, withdraw or release messages for LSPs that traversed the LDP in question. In the case that the LSP was created by CR-LDP and that a withdraw or notification is about to be generated, LDP will insert a feedback TLV for the interface which just went down that contains 0's for all the bandwidth values and attach to it the proper interface addresses. Where LDP FT procedures [LDP-FT] are in use, LSPs that are protected by FT procedures should not be torn down until after session reestablishment has failed. During LDP re-establishment time new connections may be queued and delayed for the re-establishment time. If signaling delay is undesirable feedback may be used to report zero bandwidth. In this case, if LDP is successfully re-established a Link LSA should be triggered if sufficient amount bandwidth is available.

When the LDP session that originated a CR-LDP label request receives a mapping that contains feedback TLV's it is recommended that these bandwidth values supersede the corresponding values in the node's topology database for source route computations. Doing so permits this node to immediately synchronize its topology with respect to the real bandwidth reservations along the path that was just established.

When the LDP session that originated a CR-LDP label request receives a notification that contains feedback TLV's it is recommended that these bandwidth values supersede the corresponding values in the node's topology database for source route computations. Doing so permits this node to immediately synchronize its topology with respect to the real bandwidth reservations along the path that just failed to establish. The source node may then re-compute a path knowing that the computation will take into account the failure if it was caused by the topology database being in error with respect to the real network state.

6. IGP considerations

Implementations MUST NOT permit bandwidth information learned by this feedback mechanism to be re-flooded via IS-IS, OSPF or any other IGP. The bandwidth information learned via these feedback mechanisms is to be used ONLY for source route computations on the nodes that are directly on the path that fed back the bandwidth. Normally only the source node of the LSP, or perhaps intermediate gateway nodes will use this information. It is however permitted for intermediate nodes that are forwarding this feedback information to store it for their own local source route computations.
There is a possibility of a race condition between the bandwidth information that is received via feedback and that, which is received via a normal IGP flood. While there may be a discrepancy between the two, both are within a few 100 milliseconds of being correct. Solutions to allow us to determine which information is most up to date (say by adding a sequence number) do not add any significant benefit. Constraint based, source routed systems will always have errors in the local topology database with respect to the real network. These errors can be reduced through reduced flooding intervals, path following feedback and selective flooding but realistically the errors cannot be reduced below the second or so range. As a result propagation delay order race conditions are noise with respect to the average expected errors. An implementation SHOULD therefore consider the most recently received update (IGP or feedback) as being the most up to date.

7. Future considerations

Constraint based routing systems such as CR-LDP will in the future offer other forms of constraint than simply reserved bandwidth. Actual utilization levels, current congestion levels, number of discrete channels/wavelengths available etc. are all possible constraints that change rapidly and which must be taken into consideration when computing a route. It is expected that this mechanism will be used to feedback these and other new forms of link constraining data.

8. RSVP-TE consideration

Nothing precludes the use of such feedback mechanisms with a similar TLV structure in the RSVP-TE Resv and other reverse flowing messages although repeatedly applying unchanged feedback should be avoided. This could be accomplished by a simple rule that only permits feedback information on the original RESV, not on subsequent refreshes. This document only covers the CR-LDP protocol.

9. Intellectual Property Consideration

The IETF has been notified of intellectual property rights claimed

in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

10. Security Considerations

This document raises no new security considerations for CR-LDP, RSVP-TE or MPLS in general.

11. Acknowledgments

The authors would like to thank Keith Dysart for his guidance, and Jerzy Miernik for helping implement these concepts and bringing them to life. The authors' would like to aknowledge Dave Allan for his comments and suggestions.

12. References

[CR-LDP] Jamoussi, B. et al., "Constraint-Based LSP Setup using LDP", RFC 3212, January 2002.

[LDP] Andersson, L. et al., "LDP Specification", RFC 3032, January 2001.

[IS-IS] Li, T., Smit, H., "Extensions to IS-IS for traffic engineering", Internet Draft, draft-ietf-isis-traffic-04.txt, August 2001.

[OSPF] Katz,D., Yeung, D., Kompella, K., "Traffic Engineering Extensions to OSPF," draft-katz-yeung-ospf-traffic-06.txt, February 2001.

[QUERY] Ashwood-Smith, P., Paraschiv, A., "MPLS LDP Query Message Description", Internet Draft, draft-anto-ldp-query-04.txt, May 2002.

[LDP-FT] Farrel, A et al., "Fault Tolerance for LDP and CR-LDP", Internet Draft, draft-ietf-mpls-ldp-ft-02.txt, October 2001

13. Author's Addresses

Peter Ashwood-Smith
Nortel Networks Corp.
P.O. Box 3511 Station C,
Ottawa, ON K1Y 4H7
Canada
Phone: +1 613-763-4534
petera@nortelnetworks.com

Bilel Jamoussi
Nortel Networks Corp.
600 Technology Park Drive
Billerica, MA 01821
USA
phone: +1 978-288-4506
jamoussi@nortelnetworks.com

Darek Skalecki
Nortel Networks Corp.
P.O. Box 3511 Station C,
Ottawa, On K1Y 4H7

Don Fedyk
Nortel Networks Corp.
600 Technology Park Drive
Billerica, MA 01821

Canada
Phone: +1 613-765-2252

USA
Phone: +1 978-228-3041

dareks@nortelnetworks.com dwfedyk@nortelnetworks.com

Full Copyright Statement