Network Working Group
Internet Draft
Expiration Date: December 2004
File Name: draft-ietf-bmwg-ospfconv-applicability-06.txt
June 2004

Vishwas Manral
Netplane Systems
Russ White
Cisco Systems
Aman Shaikh
University of California

Considerations When Using Basic OSPF Convergence Benchmarks
draft-ietf-bmwg-ospfconv-applicability-06.txt

Status of this Memo

Abstract

This document discusses the applicability of various tests for measuring single router control plane convergence, specifically in regards to the Open Shortest First (OSPF) protocol. There are two general sections in this document, the first discussing specific advantages and limitations of specific OSPF convergence tests, and the second discussing more general pitfalls to be considered when testing routing protocols convergence testing.

1. Introduction

There is a growing interest in testing single router control plane convergence for routing protocols, with many people looking at testing methodologies which can provide information on how long it takes for a network to converge after various network events occur. It is important to consider the framework within which any given convergence test is executed when attempting to apply the results of the testing, since the framework can have a major impact on the results. For instance, determining when a network is converged, what parts of the router's operation are considered within the testing, and other such things will have a major impact on what apparent performance routing protocols provide.

This document describes in detail the various benefits and pitfalls of tests described in [BENCHMARK]. It also explains how such measurements can be useful for providers and the research community.

NOTE: The word convergence within this document refers to single router control plane convergence [TERM].

2. Advantages of Such Measurement

- To be able to compare the iterations of a protocol implementation. It is often useful to be able to compare the performance of two iterations of a given implementation of a protocol to determine where improvements have been made and where further improvements can be made.

- To understand, given a set of parameters (network conditions), how a particular implementation on a particular device is going to perform. For instance, if you were trying to decide the processing power (size of device) required in a certain location within a network, you can emulate the conditions which are going to exist at that point in the network and use the test described to measure the performance of several different routers. The results of these tests can provide one possible data point for an intelligent decision.

  If the device being tested is to be deployed in a running network, using routes taken from the network where the equipment is to be deployed rather than some generated topology in these tests will give results which are closer to the real performance of the device. Care should be taken to emulate or take routes from the actual location in the network where the device will be (or would be) deployed. For instance, one set of routes may be

taken from an ABR, one set from an area 0 only router, various sets from stub area, another set from various normal areas, etc.

- To measure the performance of an OSPF implementation in a wide variety of scenarios.

- To be used as parameters in OSPF simulations by researchers. It may some times be required for certain kinds of research to measure the individual delays of each parameter within an OSPF implementation. These delays can be measured using the methods defined in [BENCHMARK].

- To help optimize certain configurable parameters. It may some times be helpful for operators to know the delay required for individual tasks so as to optimize the resource usage in the network i.e. if it is found that the processing time is x seconds on an router, it would be helpful to determine the rate at which to flood LSA's to that router so as to not overload the network.

3. Assumptions Made and Limitations of such measurements

- The interactions of convergence and forwarding; testing is restricted to events occurring within the control plane. Forwarding performance is the primary focus in [INTERCONNECT] and it is expected to be dealt with in work that ensues from [FIB-TERM].

- Duplicate LSAs are Acknowledged Immediately. A few tests rely on the property that duplicate LSA Acknowledgements are not delayed but are done immediately. However if some implementation does not acknowledge duplicate LSAs immediately on receipt, the testing methods presented in [BENCHMARK] could give inaccurate measurements.

- It is assumed that SPF is non-preemptive. If SPF is implemented so that it can (and will be) preempted, the SPF measurements taken in [BENCHMARK] would include the times that the SPF process is not running ([BENCHMARK] measures the total time taken for SPF to run, not the amount of time that SPF actually spends on the device's processor), thus giving inaccurate measurements.

- Some implementations may be multithreaded or use a multiprocess/multirouter model of OSPF. If because of this any of the assumptions taken in measurement are violated in such a model, it could lead to inaccurate measurements.

- The measurements resulting from the tests in [BENCHMARK] may not provide the information required to deploy a device in a large scale network. The tests described focus on individual components of an OSPF implementation's performance, and it may be difficult to combine the measurements in a way which accurately depicts a device's performance in a large scale network. Further research is required in this area.

- The measurements described in [BENCHMARK] should be used with great care when comparing two different implementations of OSPF from two different vendors. For instance, there are many other factors than convergence speed that need to be taken into consideration when comparing different vendor's products, and it's difficult to align the resources available on one device to the resources available on another device.

4. Observations on the Tests Described in [BENCHMARK]

Some observations taken while implementing the tests described in [BENCHMARK] are noted in this section.

4.1. Measuring the SPF Processing Time Externally

The most difficult test to perform is the external measurement of the time required to perform an SPF calculation, since the amount of time between the first LSA which indicates a topology change and the duplicate LSA is critical. If the duplicate LSA is sent too quickly, it may be received before the device under test actually begins running SPF on the network change information. If the delay between the two LSAs is too long, the device under test may finish SPF processing before receiving the duplicate LSA. It is important to closely investigate any delays between the receipt of an LSA and the beginning of an SPF calculation in the device under test; multiple tests with various delays might be required to determine what delay needs to be used to accurately measure the SPF calculation time.

Some implementations may force two intervals, the SPF hold time and the SPF delay, between successive SPF calculations. If an SPF hold time exists, it should be subtracted from the total SPF execution time. If an SPF delay exists, it should be noted in the test results.

4.2. Noise in the Measurement Device

The device on which measurements are taken (not the device under test) also adds noise to the test results, primarily in the form of delay in packet processing and measurement output. The largest source of noise is generally the delay between the receipt of packets by the measuring device and the information about the packet reaching the device's output, where the event can be measured. The following steps may be taken to reduce this sampling noise:

o   Increasing the number of samples taken will generally improve the tester's ability to determine what is noise, and remove it from the results.

o   Try to take time-stamp for a packet as early as possible. Depending on the operating system being used on the box, one can instrument the kernel to take the time-stamp when the interrupt is processed. This does not eliminate the noise completely, but at least reduces it.

o   Keep the measurement box as lightly loaded as possible.

o   Having an estimate of noise can also be useful.

The DUT also adds noise to the measurement. Points (a) and (c) apply to the DUT as well.

4.3. Gaining an Understanding of the Implementation Improves Measure- ments

While the tester will (generally) not have access to internal infor- mation about the OSPF implementation being tested using [BENCHMARK], the more thorough the tester's knowledge of the implementation is, the more accurate the results of the tests will be. For instance, in some implementations, the installation of routes in local routing tables may occur while the SPF is being calculated, dramatically impacting the time required to calculate the SPF.

4.4. Gaining an Understanding of the Tests Improves Measurements

One method which can be used to become familiar with the tests described in [BENCHMARK] is to perform the tests on an OSPF implemen- tation for which all the internal details are available, such as [GATED]. While there is no assurance that any two implementations will be similar, this will provide a better understanding of the

tests themselves.

5. LSA and Destination mix

In many OSPF benchmark tests, a generator injecting a number of LSAs is called for. There are several areas in which injected LSAs can be varied in testing:

- The number of destinations represented by the injected LSAs

  Each destination represents a single reachable IP network; these will be leaf nodes on the shortest path tree. The primary impact to performance should be the time required to insert destinations in the local routing table and handling the memory required to store the data.

- The types of LSAs injected

  There are several types of LSAs which would be acceptable under different situations; within an area, for instance, type 1, 2, 3, 4, and 5 are likely to be received by a router. Within a not-so-stubby area, however, type 7 LSAs would replace the type 5 LSAs received. These sorts of characterizations are important to note in any test results.

- The Number of LSAs injected

  Within any injected set of information, the number of each type of LSA injected is also important. This will impact the shortest path algorithms ability to handle large numbers of nodes, large shortest path first trees, etc.

- The Order of LSA Injection

  The order in which LSAs are injected should not favor any given data structure used for storing the LSA database on the device under test. For instance, AS-External LSA's have AS wide flooding scope; any Type-5 LSA originated is immediately flooded to all neighbors. However the Type-4 LSA which announces the ASBR as a border router is originated in an area at SPF time (by ABRs on the edge of the area in which the ASBR is). If SPF isn't scheduled immediately on the ABRs originating the type 4 LSA, the type-4 LSA is sent after the type-5 LSA's reach a router in

the adjacent area. So routes to the external destinations aren't immediately added to the routers in the other areas. When the routers which already have the type 5's receive the type-4 LSA, all the external routes are added to the tree at the same time. This timing could produce different results than a router receiving a type 4 indicating the presence of a border router, followed by the type 5's originated by that border router.

The ordering can be changed in various tests to provide insight on the efficiency of storage within the DUT. Any such changes in ordering should be noted in test results.

6. Tree Shape and the SPF Algorithm

The complexity of Dijkstra's algorithm depends on the data structure used for storing vertices with their current minimum distances from the source, with the simplest structure being a list of vertices currently reachable from the source. In a simple list of vertices, finding the minimum cost vertex then would take O(size of the list). There will be O(n) such operations if we assume that all the vertices are ultimately reachable from the source. Moreover, after the vertex with min cost is found, the algorithm iterates thru all the edges of the vertex and updates cost of other vertices. With an adjacency list representation, this step when iterated over all the vertices, would take O(E) time, with E being the number of edges in the graph. Thus, overall running time is:

O(sum(i:1, n)(size(list at level i) + E).

So, everything boils down to the size (list at level i).

If the graph is linear:

```
root
 |
 1
 |
 2
 |
 3
 |
 4
 |
 5
 |
 6
```

and source is a vertex on the end, then size(list at level i) = 1 for all i. Moreover, E = n - 1. Therefore, running time is O(n).

If graph is a balanced binary tree:

```
    root
   /    \
  1      2
 / \    / \
3   4  5   6
```

size(list at level i) is a little complicated. First it increases by 1 at each level up to a certain number, and then goes down by 1. If we assumed that tree is a complete tree (like the one in the draft) with k levels (1 to k), then size(list) goes on like this: 1, 2, 3,

Then the number of edges E is still n - 1. It then turns out that the run-time is O(n^2) for such a tree.

If graph is a complete graph (fully-connected mesh), then size(list at level i) = n - i. Number of edges E = O(n^2). Therefore, run-time is O(n^2).

So, the performance of the shortest path first algorithm used to compute the best paths through the network is dependant o the construction of the tree The best practice would be to try and make any emulated network look as much like a real network as possible, especially in the area of the tree depth, the meshiness of the network, the number of stub links versus transit links, and the number of connections and nodes to process at each level within the original tree.

7. Topology Generation

As the size of networks grows, it becomes more and more difficult to actually create a large scale network on which to test the properties of routing protocols and their implementations. In general, network emulators are used to provide emulated topologies which can be advertised to a device with varying conditions. Route generators either tend to be a specialized device, a piece of software which runs on a router, or a process that runs on another operating system, such as Linux or another variant of Unix.

Some of the characteristics of this device should be:

- The ability to connect to the several devices using both point-to-point and broadcast high speed media. Point-to-point links can be emulated with high speed Ethernet as long as there is no hub or other device in between the DUT and the route generator, and the link is configured as a point-to-point link within OSPF [BROADCAST-P2P].

- The ability to create a set of LSAs which appear to be a logical, realistic topology. For instance, the generator should be able to mix the number of point-to-point and broadcast links within the emulated topology, and should be able to inject varying numbers of externally reachable destinations.

- The ability to withdraw and add routing information into and from the emulated topology to emulate links flapping.

- The ability to randomly order the LSAs representing the emulated topology as they are advertised.

- The ability to log or otherwise measure the time between packets transmitted and received.

- The ability to change the rate at which OSPF LSAs are transmitted.

- The generator and the collector should be fast enough so that they are not bottle necks. The devices should also have a degree of granularity of measurement at least as small as desired from the test results.

8. IANA Considerations

This document requires no IANA considerations.

9. Security Considerations

This document does not modify the underlying security considerations in [OSPF].

10. Acknowledgements

Thanks to Howard Berkowitz, (hcb@clark.net) and the rest of the BGP benchmarking team for their support and to Kevin Dubray(kdubray@juniper.net) who realized the need of this draft.

11. Normative References

[BENCHMARK]
Manral, V., "Benchmarking Basic OSPF Single Router Control Plane Convergence", draft-bmwg-ospfconv-intraarea-09, June 2004

[TERM]Manral, V., "OSPF Convergence Testing Terminology and Concepts", draft-bmwg-ospfconv-term-08, May 2004

[RFC2119]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997

12. Informative References

[INTERCONNECT]
Bradner, S., McQuaid, J., "Benchmarking Methodology for Network Interconnect Devices", RFC2544, March 1999.

[FIB-TERM]
Trotter, G., "Terminology for Forwarding Information Base (FIB) based Router Performance", RFC3222, October 2001.

[BROADCAST-P2P]
Shen, Naiming, et al., "Point-to-point operation over LAN in link-state routing protocols", draft-ietf-isis-igp-p2p-over-lan-03.txt, August, 2003.

[GATED]
http://www.gated.org

13. Authors' Addresses

Vishwas Manral
Netplane Systems
189 Prashasan Nagar
Road number 72
Jubilee Hills
Hyderabad, India

vmanral@netplane.com

Russ White
Cisco Systems, Inc.
7025 Kit Creek Rd.
Research Triangle Park, NC 27709

riw@cisco.com

Aman Shaikh
University of California
School of Engineering
1156 High Street
Santa Cruz, CA 95064

aman@soe.ucsc.edu

14. Full Copyright Statement

15. Intellectual Property