          Considerations When Using Basic OSPF Convergence Benchmarks

Abstract

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

1. Introduction

   There is a growing interest in testing single router control plane
   convergence for routing protocols, and many people are looking at
   testing methodologies that 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 one attempts 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 the apparent
   performance that routing protocols provide.

This document describes in detail 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: In this document, the word "convergence" refers to single
router control plane convergence [TERM].

2. Advantages of Such Measurement

   o   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 in order to determine where improvements have been made
        and where further improvements can be made.

   o   To understand, given a set of parameters (network conditions),
        how a particular implementation on a particular device will
        perform. For instance, if you were trying to decide the
        processing power (size of device) required in a certain location
        within a network, you could emulate the conditions that will
        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 yield results that 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.

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

   o   To be used as parameters in OSPF simulations by researchers. It
        may sometimes 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].

   o   To help optimize certain configurable parameters. It may
        sometimes be helpful for operators to know the delay required
        for individual tasks in order to optimize the resource usage in
        the network. For example, if the processing time on a router is

found to be x seconds, determining the rate at which to flood
LSAs to that router would be helpful so as not to overload the
network.

3. Assumptions Made and Limitations of Such Measurements

   o   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].

   o   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 an implementation
       does not acknowledge duplicate LSAs immediately on receipt, the
       testing methods presented in [BENCHMARK] could give inaccurate
       measurements.

   o   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, thus giving inaccurate measurements.
       ([BENCHMARK] measures the total time taken for SPF to run, not
       the amount of time that SPF actually spends on the device's
       processor.)

   o   Some implementations may be multithreaded or use a
       multiprocess/multirouter model of OSPF. If because of this any
       of the assumptions made during measurement are violated in such
       a model, measurements could be inaccurate.

   o   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 that
       accurately depicts a device's performance in a large-scale
       network. Further research is required in this area.

   o   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 vendors' products. One
       difficulty is aligning the resources available on one device to
       the resources available on another.

4. Observations on the Tests Described in [BENCHMARK]

   Some observations recorded 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 because the amount of
   time between the first LSA that 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 being tested actually begins
   running SPF on the network change information. If the delay between
   the two LSAs is too long, the device 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 tested device; multiple tests with various delays
   might be required to determine what delay needs to be used to measure
   the SPF calculation time accurately.

   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 being
   tested) 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 receipt of information about the
   packet by 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 to remove
        it from the results. This applies to the DUT as well.

   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. This
        applies to the DUT as well.

o   Having an estimate of noise can also be useful.

The DUT also adds noise to the measurement.

4.3. Gaining an Understanding of the Implementation Improves
      Measurements

Although the tester will (generally) not have access to internal
information 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 that can be used to become familiar with the tests
described in [BENCHMARK] is to perform the tests on an OSPF
implementation for which all the internal details are available.
Although 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:

o   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.

o   The types of LSAs injected

      There are several types of LSAs that would be acceptable under
      different situations; within an area, for instance, types 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.

o   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 algorithm's ability to handle large numbers of
     nodes, large shortest path first trees, etc.

o   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
     being tested. For instance, AS-External LSAs 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. Therefore, routes to the external
     destinations aren't immediately added to the routers in the
     other areas. When the routers that already have the type 5s
     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 5s originated by that
     border router.

     The ordering can be changed in various tests to provide insight
     into 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; the simplest structure is a list of vertices currently
   reachable from the source. In a simple list of vertices, finding the
   minimum cost vertex would then 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 minimum cost is found, the algorithm iterates through all the
   edges of the vertex and updates the 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, the overall running time is:
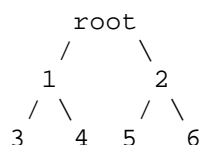
   O(sum(i:1, n)(size(list at level i) + E).

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

If the graph is linear,

```
    root
     |
     1
     |
     2
     |
     3
     |
     4
     |
     5
     |
     6
```
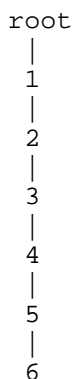
and source is a vertex on the end, then size(list at level i) = 1 for
all i. Moreover, E = n - 1. Therefore, running time is O(n).

If the 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 it goes down by
1. If we assume that the tree is a complete tree (as shown above)
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 the 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).

Therefore, the performance of the shortest path first algorithm used
to compute the best paths through the network is dependent on the
construction of the tree. The best practice would be to try to 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 that can be
advertised to a device with varying conditions. Route generators
tend to be either 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 as follows:

o   The ability to connect to 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 between the DUT and the route generator, and the
     link is configured as a point-to-point link within OSPF
     [BROADCAST-P2P].

o   The ability to create a set of LSAs that 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 to inject varying numbers of
     externally reachable destinations.

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

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

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

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

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

8. Security Considerations

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

9. 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 for this document.

10. Normative References

   [BENCHMARK]   Manral, V., White, R., and A. Shaikh, "Benchmarking
                      Basic OSPF Single Router Control Plane Convergence",
                      RFC 4061, April 2005.

   [TERM]   Manral, V., White, R., and A. Shaikh, "OSPF
                      Benchmarking Terminology and Concepts", RFC 4062,
                      April 2005.

   [OSPF]   Moy, J., "OSPF Version 2", STD 54, RFC 2328, April
                      1998.

11. Informative References

   [INTERCONNECT] Bradner, S. and J. McQuaid, "Benchmarking Methodology
                   for Network Interconnect Devices", RFC 2544, March
                   1999.

   [FIB-TERM]   Trotter, G., "Terminology for Forwarding Information
                   Base (FIB) based Router Performance", RFC 3222,
                   December 2001.

   [BROADCAST-P2P] Shen, Naiming, et al., "Point-to-point operation over
                   LAN in link-state routing protocols", Work in
                   Progress, August, 2003.

Authors' Addresses

     Vishwas Manral
     SiNett Corp,
     Ground Floor,
     Embassy Icon Annexe,
     2/1, Infantry Road,
     Bangalore, India

     EMail: vishwas@sinett.com

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

     EMail: riw@cisco.com

     Aman Shaikh
     AT&T Labs (Research)
     180 Park Av, PO Box 971
     Florham Park, NJ 07932

     EMail: ashaikh@research.att.com