Analysis of link failures in an IP backbone

Gianluca Iannaccone, Chen-nee Chuah, Richard Mortier, Supratik Bhattacharyya, Christophe Diot

Abstract— **Today's IP backbones are provisioned to provide excellent performance in terms of loss, delay and availability. However, performance degradation and service disruption are likely in the case of failure, such as fiber cuts, router crashes, etc. In this paper, we investigate the occurence of failures in Sprint's IP backbone and their potential impact on emerging services such as Voice-over-IP (VoIP). We first examine the frequency and duration of failure events derived from IS-IS routing updates collected from three different points in the Sprint IP backbone. We observe that link failures occur as part of everyday operation, and the majority of them are short-lived (less than 10 minutes). We also discuss various statistics such as the distribution of inter-failure time, distribution of link failure durations, etc. which are essential for constructing a realistic link failure model. Next, we present an analysis of routing and service reconvergence time during a controlled link failure scenario in our backbone. Our results indicate that disruption to packet forwarding after link failures depends not only on routing protocol dynamics, but also on the design of routers' architectures and control planes. Thus our results offer insights into two basic components for defining network-wide availability, which we consider a more appropriate metric for service-level agreements to support emerging applications.**

I. Introduction

Service Level Agreeements (SLAs) used by today's Internet Service Providers (ISPs) are based on three simple metrics: packet loss, packet delay and "port" availability. The first two are computed network-wide and usually averaged over a one month period. In the third metric, the term "port" refers to the point at which a customer link terminates on an ISP's edge router. Port availability refers to the fraction of time that this port is operational, and therefore measures a customer's *physical* connectivity to an ISP's network.

Today's IP backbones are engineered to guarantee excellent performance for all three metrics described above. For example, a typical SLA may guarantee an average loss rate of 0.3%, an average delay of 55 msecs within the continental USA, and port availability of 99%.

Although such an SLA may be sufficient for traditional Internet applications (email, file transfer, web access, etc.), it may not be enough to support emerging applications such as Voice-over-IP (VoIP). These new applications are adversely affected if packets are dropped or delayed due to failures caused by optical fiber cuts, router reboots, maintenance windows, etc. Hence it is important to define a notion of "service availability" for these applications. Loosely speaking, service availability measures the fraction of time that a network can provide a service to a customer such as access to a web-site or voice calls.

The issue of availability is easier to understand for telephone networks with call admission control, where phone calls are blocked in the event of a problem such as equipment failure or high call volume, rendering the network unavailable. The situation is very different for IP networks where there is no admission control. A network service may become unavailable to a customer if most of the customer's packets are lost due to heavy congestion, or if routers inside the network are temporarily unable to find a route to a destination. Given that current IP backbones are more than adequately provisioned, the former is extremely unlikely. However, the route to a destination may indeed become unavailable when a failure occurs and the routing protocol has to recompute an alternate path around the failure. This is what is referred to as the routing protocol reconvergence time.

The first step towards defining and measuring service availability is to develop a detailed understanding of how often failures occur in a network, how long they last, and how each such failure impacts packet forwarding. Unfortunately very little is known about these issues for operational networks. In this paper, we attempt to address this deficiency by analyzing link failures in Sprint's operational IP backbone.

Our contribution is two-fold. In the first part of our work we study IS-IS routing updates [4] collected using a passive IS-IS "listener" over a four-month period. We analyze the frequency and duration of link failures as re-

G. Iannaccone (gianluca@sprintlabs.com), S. Bhattacharyya (supratik@sprintlabs.com) and C. Diot (cdiot@sprintlabs.com) are with Sprint Advanced Technology Laboratories, Burlingame (CA). C. Chuah (chuah@ece.ucdavis.edu) is with University of California, Davis. R. Mortier (mort@microsoft.com) is with Microsoft Research, Cambridge, UK. This work was done while C. Chuah and R. Mortier were visiting Sprint ATL.

ported in IS-IS Link State PDUs (LSPs). We also report various statistics such as mean inter-failure times for individual links, that form the basis for a realistic link failure model for a large network. Such a failure model is fundamental to effective network design and traffic engineering.

In the second part of our work, we examine how a typical link failure disrupts packet forwarding. The data for this study was collected by shutting down multiple network links during a maintenance window. We analyze router logs, IS-IS routing updates (from our listener), SNMP link utilization data, etc. to carefully isolate and identify the factors contributing to IS-IS protocol reconvergence time. In addition, we study the impact of our failure experiment on Voice-over-IP-like active probes that we inject into our network. By studying network failures and service disruption during a typical failure, we provide insights into the two aspects that are important for defining "service availability" for a network.

The rest of the paper is structured as follows. In Section II we describe the systems used to record the routing messages and the method used for identifying failures in the network. Section III presents our analysis of the failure events while Section IV describes the potential impact of a failure on data traffic. Section V concludes the paper and describes some future work.

II. Method

A. Collecting IS-IS Updates

We use the Python Routeing Toolkit (PyRT) [1] to collect IS-IS Link State PDUs (LSPs) from our backbone. PyRT includes an IS-IS "listener" that collects these LSPs from an IS-IS enabled router over an Ethernet link. The router treats the listener in the same way as other adjacent routers, hence it forwards to the listener all LSPs that it receives from the rest of the network. Since IS-IS broadcasts LSPs through the entire network, our listener is informed of every routing-level change occuring anywhere in the network. However, the listener is "passive" in the sense that it does not transmit any LSPs to the router. The session between the listener and the router is kept alive via periodic IS-IS keepalive (Hello) messages. On receiving an LSP, the listener prepends it with an header in MRTD format (extended to include timestamp of granularity finer than a second) and writes it out to a file.

The data presented in this paper was collected from a single listener at a Point-of-Presence (POP) within our backbone. However, we installed two more listeners at two different POPs at the other end of the backbone. All three listeners are synchronized using NTP stratum-1 servers. There are two advantages in having multiple listeners. First, we are able to cross-check the information that we collect at each listener. Second, we are able to determine the time that it takes for an IS-IS LSP to reach different ends of our backbone.

B. Processing ISIS Updates

Whenever IP-level connectivity between two directly connected routers is lost, each router independently broadcasts an "adjacency down" LSP through the network. When the connectivity is restored, each router broadcasts an "adjacency up" LSP. Note that the loss of connectivity at the IP level may be triggered by a variety of causes such as an optical fiber cut, router interface failure, IS-IS protocol malfunction, etc. We refer to each such event as a *failure event*.

Each failure event is recorded with the MRTD timestamp of the *first* LSP received at our listener that reports the failure. Both the LSPs reporting the loss of IP connectivity may not reach every router (and our listener) at the same time. Our approach of determining a failure event based on the first LSP received is conformant with how the IS-IS protocol reacts to such failures. As soon as a router receives the first LSP reporting an adjacency down, it considers the IP connectivity to be lost without waiting for the second LSP. Hence the first LSP is sufficient to trigger a route recomputation, which may lead to a disruption in packet forwarding. A failure event ends when our listeners receive an LSP from both ends of the link. This is conformant with how routers handle "adjacency up" LSPs - both LSPs must be received by a router before it considers the IP connectivity to be restored.

Our backbone is in constant evolution with new links being added and older ones being decommissioned every week. When a link is decommissioned, "adjacency down" LSPs are broadcast, but there is no subsequent "adjacency up" LSP. On the other hand, when IP connectivity is lost due to a problem with the optical fiber, restoration usually takes a few hours (and sometimes just a few minutes). Connectivity loss due to router or protocol problems is usually restored in less than an hour. In order to distinguish link decomissioning from valid failure events, we consider only those failure events for which we subsequently receive the two "adjacency up" LSPs within the next twenty-four hours.

Since our goal in this paper is to study link failures, we focus only on those LSPs that report a link failing or being restored after a failure. Note that the LSPs collected also contain information on IS-IS weight changes, router overload bit set/reset, etc. However, a discussion of these is beyond the scope of this paper.

[1] www.sprintlabs.com/Department/IP-Interworking/Routing/PyRT

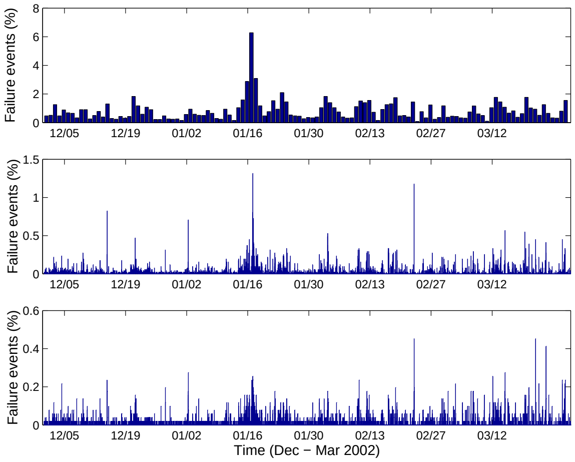

Fig. 1. Notifications of failure events over three time scales: one day (top), one hour (middle) and five minutes(bottom)

III. Study of failure events

We limit our study of failure events to the inter-POP level, i.e. failures occuring on links that connect different POPs in our network. There are three reasons for this. First, the POP-to-POP topology is much more stable over time than the internal topology of each POP, which is frequently modified to accomodate new customers. Hence, most of the failure events at the intra-POP level are likely to be due to *planned* reconfiguration and/or addition/removal of links. Such events are less relevant than *unplanned* failure events in determining how to improve service availability. Secondly, traffic engineering policies are determined primarily by the POP-to-POP topology. Finally, intra-POP link failures have a much smaller impact on traffic given that backbone routers inside the same POP are always connected in a full mesh and thus allow "localization" of the effect of a failure event.

A. *Temporal Analysis of Failures*

Figure 1 shows the distribution of failure events that occur over a four month period period (December 2001 to April 2002) on three timescales: one day, one hour and five minutes[2]. We observe that failure events are fairly well spread out across days, and even over the course of a single day. There were few days in mid-January when the number of failures was significantly higher, particularly one day which accounted for 6% of total number of failures events. Although we are still determining the reason behind this, we suspect that there may have been a widespread outage on that day due to multiple fiber cuts.

It is not possible to identify the factors causing failures from IS-IS routing updates. However, the duration of a failure may provide some hints about the possible cause. In Figure 2, we plot the cumulative distribution of failure durations over the four-month period. We find that only 10% of failures last longer than 20 minutes. These are possibly caused by fiber cuts and/or equipment failures/upgrades. Note that the longest duration for a failure is 24 hours since we disregard all failure events where IP connectivity is not restored within 24 hours (Section II). About 40% of the failures last between one minute and 20 minutes. These are possibly caused by router reboots, software problems, transient equipment problems, short maintenance operations on equipments or optical fiber, etc. Interestingly, about 50% of all failure events last less than a minute. While the cause behind this is still under investigation, one possible reason is for a router to mistakenly consider an adjacency to be down when it is not actually down. This could happen if the router CPU is overloaded and fails to process the IS-IS keepalive messages that are used to detect the loss of an adjacency. Furthermore, it is possible that multiple failure events on a single link within a short span of time could in fact be the oscillatory effect of a single fault or problem.

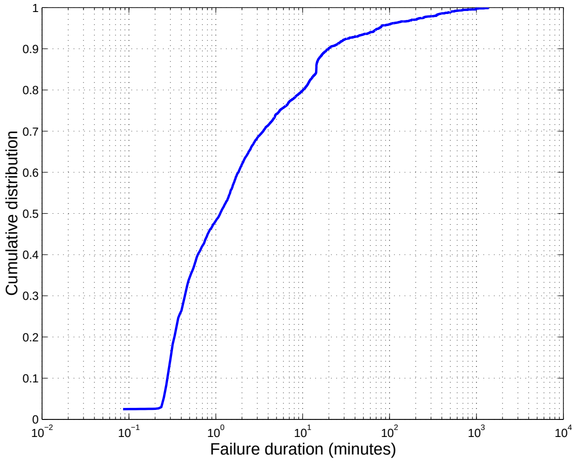

Fig. 2. Duration of failure events

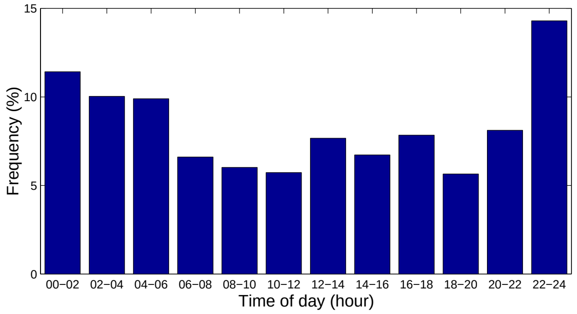

Fig. 3. Frequency of failures during 4 hours time windows

It is also interesting to compare the number of failure events due to scheduled maintenance and those that are

[2] For proprietary reasons we are unable to provide absolute numbers.

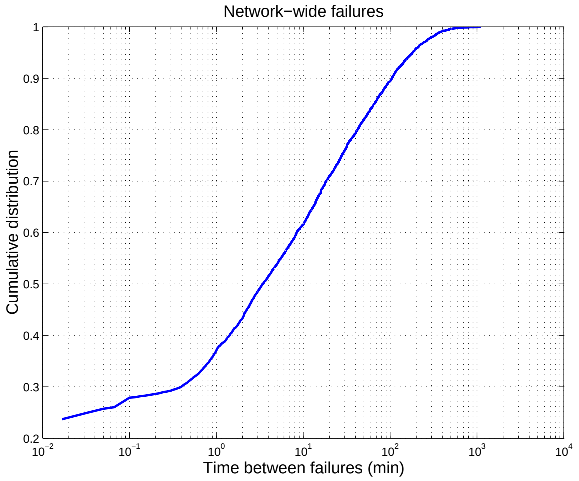

Fig. 4. Cumulative distribution of times between failures over the entire network

unplanned or accidental, since it is desirable to eliminate (or at least minimize) the latter. Maintenance windows are scheduled during late night/early morning; hence a breakup of failure events by the time of day sheds light on this issue. In Figure 3 we show all the failures (over 4 months) grouped in two-hour bins by time of day (where the timezone is US Eastern Standard Time). We observe that about 47% of the failure events occur between 10 PM EST and 6 AM EST. If we take into account the three hour time difference between the east and west coasts of the USA, then this is the time window during which most maintenance windows are scheduled. Although all failure events during this period are possibly not scheduled maintenance, the fact that this time period accounts for almost half of all failures indicates that maintenance activities do account for a significant portion of the failure events that we observe. Note also that failure events during this period are likely to have less of an impact on traffic, because the backbone is relatively lightly loaded at night.

B. *Towards a Link Failure Model*

An expected outcome of our analysis of failure events is the construction of a failure model that captures how failures occur in a large operational network. There is a large number of questions that we need to answer in order to do so. For example: (i) what is the distribution of failure durations? (ii) what is the distribution of the time between successive failure events? (iii) are all links identical in terms of failure characteristics? (iv) can we model the failure of each link independently of all other links? In this section we discuss these issues and present initial results on some of them.

The distribution of failure durations has already been discussed in the previous section. We start by looking at the distribution of time between failures. Figure 4 shows the cumulative distribution of the time between two consecutive failure events over the entire network. This implies that a lot of failures happen close to each other. This is not surprising, given that our network is known to experience periods of widespread outage when multiple links in the network are brought down by one or more fiber cuts. This is also in keeping with our earlier speculation about oscillatory problems on a single link manifesting itself as bursts of very short failure events. On the other hand, the mean time between failures indicates that there are at least some links which experience very few failures spaced well apart in time.

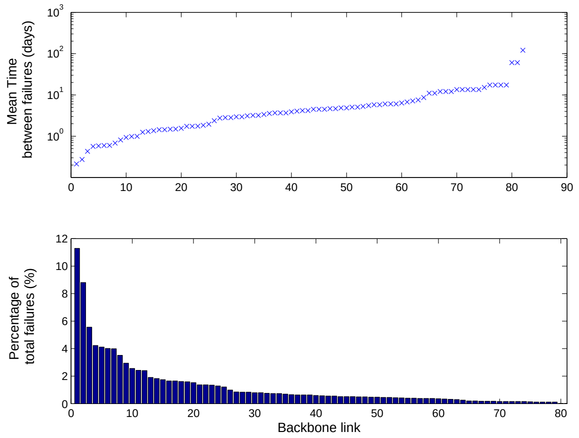

Fig. 5. Failure and mean time between failures per link

Figure 5 shows the mean time between successive failures for every link and the percentage of total failures affecting each link. We observe that links differ widely in their failure characteristics. The mean time between failure for a given link may be as low as a few minutes. Such a link is likely to experience bursts of short failures quite frequently, possibly due to faulty router or optical components. On the other hand there are links that hardly ever fail.

In Figure 5 we also show the percentage of failures per link, sorted in decreasing order. We find that there are three highly failure-prone links that together account for almost 25% of the failures. This shows how our analysis can help network operators in identifying chronically problematic links or router interfaces.

Our conclusion is that links differ widely in their failure characteristics, and a link failure model has to account for this. Our data may be useful in this regard. For example, in a network simulation study, the mean time between failures for each one of the links in the simulated network can be assigned to conform to the distribution in Figure 5.

In order to answer question (iv), we need to develop a deeper understanding of the causes behind failure events and the physical topology of a network. If link failures were mostly due to router interface problems, then it would be reasonable to model the failure process for each link independently. However, given that a single fiber cut can potentially affect multiple links, we believe that failures of certain subsets of links are likely to be correlated, while the different subsets may be independent of each other (e.g. links over disjoint fiber paths). This issue remains to be studied in depth.

To summarize, we have presented initial results from our analysis of link failures. Our analysis provides some hints about the possible causes behind link failures. We also provide insights into a number of questions that are central to building a link failure model. From an operational point of view, our analysis is useful for identifying and fixing chronic problem points in the network. However, much deeper analysis remains to be done before a usable link failure model is finally built.

IV. Impact of a typical failure

Another important goal of our work is to examine how a typical link failure impacts an operational network both at the *control* and *data* plane. For the former, we study IS-IS protocol dynamics and re-convergence properties in response to link failures. The second component refers to service availability, i.e. whether packet forwarding is disrupted. This is crucial in determining what SLA performance we can offer to VoIP or VPN customers.

A. Experimental Setting

We sent two-way packet probes from a machine on the U.S. East Coast to one on the U.S. West Coast, across the backbone network. The shortest path taken by these probes traverses three different POPs. The probes are 200 byte long UDP packets and are sent every 5 ms. Two DAG cards [3] capture and timestamp the packets on their way to the end systems.

During a maintenance window (around midnight), we shut down two backbone links connecting two POPs along the path of the packet probes. The links were brought up after 22 minutes. Then, we waited for the network to stabilize and repeated the same experiment shutting down all three links connecting the two POPs. The effect of this is to force the traffic to go through two different cities.

We analyze how the network reacts to the following two categories of events:

- *LinkDown:* A link is down, kicking off IS-IS convergence procedure to recompute the shortest paths for all route entries. Traffic may be lost due to unresolved routes in the interim until a new alternative path is found.
- *LinkUp:* The link that previously failed has recovered. Again the routers have to recompute shortest paths to all destinations. The traffic originally carried by this link will be re-routed from the respective alternative paths to the primary shortest paths.

To analyze the IS-IS protocol dynamics, we collected the routing data using the PyRT listeners described in Section II. We also record the following two logs from the routers at the two ends of the links: the *SPF log* and the *system log*. The SPF log tells us the start-time, duration and the LSP that triggers a specific Shortest Path First (SPF) computation at the router. The system log records the control-plane activities at the router along with a timestamp, e.g. when a specific interface is declared as “down” or “up”, and when the IS-IS stack is notified.

In addition to the packet probes, traceroute was run continuously during the experiment to determine the exact path followed by the traffic we are interested in.

B. Experimental Results

In this experiment, we study how long it takes for the network to stabilize after a LinkUp or LinkDown event, specifically: *routing convergence delay* and *service disruption duration*. Routing convergence delay is defined as the duration between a link failure/recovery and the instant when new routing tables are available to the router. Service disruption time refers to the duration between the time at which packet forwarding stops and the time when it resumes.

In [1], the authors have identified three components that contribute to IS-IS convergence delay: failure detection, LSP propagation and SPF computation. However, we found that there are additional steps involved in the service restoration process that depend on the router architecture and the design of the router control plane. All the contributing factors are summarized as follows:

1. Detection of an interface being down or up. A router may be directly notified by the hardware or it may identify that a link is down due to the loss of a sequence of IS-IS Hello packets.
2. Initial delay before the IS-IS stack is notified of the link status change. In Cisco routers this is defined by the *carrier-delay* timer value [2]. This timer is used to filter out very short and transient link flaps.
3. Initial wait to generate a new LSP that informs other routers about the event. This is determined by another timer: *lsp-gen interval*. This timer permits the rate-limiting of LSP message generation and prevents LSP message transmission and processing from consuming excessive router resources.

4. LSP flooding across the network. Note that LSP flooding is also subject at each router to the *lsp-gen interval*.
5. Delay between the arrival of an LSP and the start of SPF computation. In Cisco routers this is determined by the *spf-interval* timer. The role of this timer is to aggregate multiple closely spaced LSP messages and perform only one SPF computation that incorporates all the changes.
6. SPF computation and update of Routing Information Base (RIB). The RIB contains the next hop address for each destination prefix a router has knowledge of via BGP or ISIS messages. Note that the next hop may be multiple hops away from the router.
7. Pushing new routing entries to the Forwarding Information Base (FIB) at the linecards. The FIB contains specific local information on how to serve incoming packets, i.e. the output interfaces for any given destination prefix.

By our definition, the routing protocol is said to have converged after the new routing table is ready (Step 6), i.e., the IS-IS convergence delay is the time taken to complete Step 1 through 6. However, the router will not know how to forward the traffic until Step 7 is completed. Hence, the service convergence delay is the sum of protocol convergence delay plus the time taken to update the FIBs.

Our experimental results are based on the use of Cisco default values for all the timers and other parameters (e.g. *lsp-gen interval*). Interestingly, results show that show that failure detection is mostly done at the hardware level and only takes less than a 100 milliseconds, but the default carrier-delay (Step 2) is 2 seconds for LinkDown and 12 seconds for LinkUp. Step 3 and 4 add insignificant delay: between 90 to 110 ms.

In Step 5, the default value for *spf-interval* is 5.5 seconds. The SPF computation itself takes between 100 and 400 ms for a topology of more than 600 nodes and includes the update of the RIB. Putting all these numbers together, we found that the routing convergence take 5.1-5.9 seconds for LinkDown event, and 17.5-17.6 for LinkUp event.

To determine the service disruption time and compute the time needed to perform step 7, we analyze the sequence number and arrival time of the packet probes. The gaps in the sequence number indicate packet losses. Figure 6 shows the sequence number versus time during the first LinkDown event (with two links shut down). We notice a 6.6 seconds gap where all the packets are dropped due to unresolved routes following the link failure.

Thus, there is an additional 1.5 seconds delay before traffic forwarding resumes after the routing protocol has converged (5.1 seconds). This is the cost of updating and pushing to the linecards the information in the FIB.

In summary, IS-IS timers and the FIB update are two most significant components that contribute to the service disruption time. If we tune the IS-IS timer values for *carrier-delay* and *spf-interval* to O(ms), the overall convergence time can be reduced to about $2 - 3$ seconds. Since these timers were originally used to absorb transient flaps, we need to evaluate the tradeoff between fast convergence and overall network stability to determine optimal timer values. Nevertheless, there is a serious need to revisit router architectural design and implementation issues before sub-second fail-over can be achieved.

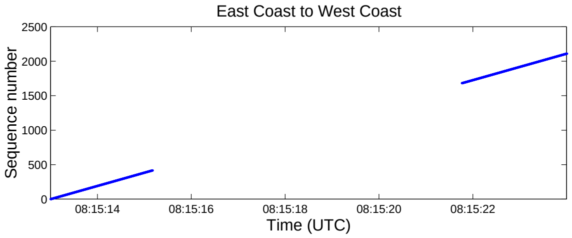

Fig. 6. Sequence number of VoIP probe packets

V. Conclusions

In this paper, we studied the two main issues that form the basis for defining a precise measure of service availability in Sprint's IP backbone. First we have analyzed the characteristics of network-wide link failure events as derived from IS-IS routing updates. Then we analyzed the service disruption caused by a typical link failure event in the backbone. To the best of our knowledge, our work is the first systematic study of the failure behavior of IS-IS in an operational network.

Future work will proceed in two directions. First, we aim to study link failure events in greater depth in order to define a realistic failure model for an operational IP network. Second, we intend to progress towards defining a measure of network service availability for emerging applications such as VoIP. This will require combining various kinds of information such as (i) statistics on failure events on a network-wide basis and per-link basis, (ii) impact of failures on data traffic, (iii) knowledge of traffic demands on the failed links, (iv) knowledge of IS-IS primary and backup paths in our backbone. We also aim to define a basis for point-to-point SLAs in our network which will be based on the availability of service between two given POPs in our network.

References

[1] C. Alaettinogly, S. Casner. ISIS routing on the Qwest backbone: A recipe for subsecond ISIS convergence. NANOG 24, 2/2002.
[2] Cisco Systems. IOS configuration guide, 1998.
[3] J. Cleary et al. Design principles for accurate passive measurements. *Passive and Active Measurement Workshop*, 4/2000.
[4] D. Oran. OSI IS-IS intra-domain routing protocol. RFC 1142.