Building Intelligent Traffic Engineering Solutions

David Durham [*david.durham@intel.com*]
Priya Rajagopal [*priya.rajagopal@intel.com*]
John Vicente [*john.vicente@intel.com*]
Intel Corporation[1]

Abstract: In today's ever more complex IT networked computing environment, it is becoming increasingly necessary to control and automate traditional tasks of provisioning, monitoring and management of network bandwidth and resources. In this paper, we propose an intelligent traffic engineering system to deliver next-generation provisioning and management network services. The proposed solution is capable of providing autonomic control of networks to provision bandwidth, routing and QoS to meet user SLA expectations, and dynamically engineer and manage traffic for optimal stability and performance.

Key words: Policy Based Management, Topology Discovery, Network Analysis, SLA

1. INTRODUCTION

A robust and automated traffic engineering solution is critical to enabling intelligent, adaptive networks. This implies that the intelligent network environment would proactively or reactively respond to changing network conditions in a way that is seamless and transparent to customers. At the same time, the network must be able to satisfy customers' changing service requirements.

This paper introduces a framework for building such intelligent networks through automated traffic engineering and network services management.

[1] We would like to thank Gerhald Gross for the Latency and Burst Test Figures and his extensive work on active network analysis techniques. We would also like to acknowledge Greg Block, Chun Yang Chiu, Priya Govindarajan, Jac Noel and Lilin Xie for their notable contributions to the project.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35620-4_43
D. Gaïti et al. (eds.), *Network Control and Engineering for QoS, Security and Mobility*

We utilize intelligent traffic engineering mechanisms to guarantee the desired Quality of Service (QoS) to network users, maintaining contracted service level agreements. In order to be realistically deployable, however, these services cannot increase the operational expense or complexity of running a network. Automation of traffic engineering encompasses mechanisms to efficiently, expeditiously and reliably transfer data through the network infrastructure with minimal or no operator intervention. One direct consequence of this requirement is that the network must be consistently and selectively monitored and analyzed to determine the availability of various network resources. Based on this information, programmatic steps can be taken to ensure that the resources are automatically available to meet user requirements.

The paper is organized as follows – in the second section we present the component architecture of the proposed intelligent traffic engineering system. In the third section, we present the traffic engineering operational model based on a methodology-driven approach. Implementation of the Automated Traffic Engineering subsystems is presented in the fourth section. We present preliminary results from evaluation of our system in the fifth section. Finally, we summarize the paper and make some concluding remarks in the last section.

2. DISTRIBUTED NETWORK CONTROL SYSTEM

This section presents a distributed network management system that can be used to transform existing networks into intelligent, adaptive networks. The Distributed Network Control System is comprised of intelligent devices, herein referred to as Smart-Nodes (SN), distributed at strategic locations in the network, typically at the edges of subnets or autonomous networks. The SNs are configured, controlled and monitored by a central management station herein referred to as the Network Health Control Center (NHCC). The NHCC and SNs are synchronized using the Network Time Protocol (NTP).

Communication between the NHCC and the SNs is through the Common Open Policy Service (COPS) [COPS-RFC] protocol. The COPS protocol is responsible for transmitting the network control information from the NHCC to the SNs and sending statistical feedback data from the SNs to the NHCC. NHCC in one domain can communicate with peer NHCCs in other domains by using the Simple Object Access Protocol (SOAP) [SOAP-SPEC]. This peer-based framework provides a highly scalable and adaptive network management solution. In addition, the SOAP/XML based interface is easily rendered into different formats using suitable style sheets making for simple

integration with other software components, such as billing systems. The use of SOAP also avoids issues with firewalls while crossing domain boundaries. The SNs and the NHCC form a closed loop network control and feedback system that is capable of making dynamic traffic engineering decisions such as SLA provisioning, traffic rerouting, load balancing, and so on. The SNs and the NHCC are implemented in software and can easily be deployed into existing networking infrastructures.

A web-based Traffic Engineering Console (TEC) serves as the administrative console for the Distributed Network Control System for managing and provisioning the network.

2.1 Smart-Nodes

The SN runs driver-level software that is responsible for fast packet processing. This includes the DiffServ building blocks comprising a packet classification engine, token-bucket metering module, marking module and a traffic-conditioning module [DIFFSERV-RFC]. The SN includes techniques for deep packet inspection, fast cache lookup and range based classification. Routers, switches, servers, end-hosts and other network appliances can act as, or be "converted" into Smart-Nodes.

A wide range of per-flow and aggregate statistics are collected and reported periodically to the NHCC by individual Smart-Nodes. All statistical data is precision time-stamped. This data is translated to the format understood by the NHCC and communicated via the COPS protocol using an event-driven methodology.

2.2 Network Health Control Center

The NHCC is the central management station that is responsible for managing and controlling the SNs in its domain. It gathers statistical information from the SNs, analyzes and correlates the statistics. The NHCC choreographs and coordinates the behavior of the SNs and can infer and react to traffic conditions network-wide. This network-wide view enables the NHCC to make global traffic engineering decisions in near real-time. In addition, the NHCC is responsible for configuring the SNs with suitable management policies based on the current network conditions. This intelligent traffic engineering solution can be easily integrated into an existing network infrastructure by simply adding Smart-Nodes at key points in the network.

2.3 Traffic Engineering Console

To facilitate the underlying services supported by the SN and NHCC system, we developed a web-based Traffic Engineering Console (TEC). The TEC allows a network administrator to manage the network topology as connectivity graphs in either a *Logical* view, e.g., site-to-site; *Virtual* view, e.g., Smart-Node overlay topology; or the traditional *Physical* view, e.g., router or switch topologies. The TEC also facilities two modes (i.e., manual and automated) of provisioning, see Section 3.1. The manual mode of operation follows a step-wise methodology, taking into account baseline metrics and historical considerations from projected demand growth and application characterizations. The TEC also supports an automation mode where services in the network respond to network events and conditions and automatically adjust the provisioning model, through QoS and routing policies as well as adjustments in monitoring policies. To manage the provisioning complexities, the TEC provides traffic demand profiling services that allow the administrator to characterize business and application services on a higher abstraction level.

3. TRAFFIC ENGINEERING OPERATIONAL MODEL

3.1 Methodology

The provisioning functionality of the traffic engineering system is handled through the TEC, which supports both manual (user-driven) and automated (user-independent) provisioning invocations. Figure 1 illustrates the basis of this methodology as a finite state machine.

As stated earlier, the manual process is user-driven, following a six-step methodology. The *baseline* step enables the user to gather and analyze current provisioned policy state as well as measurement-based link state. The next step, *application characterization*, enables the user to define new traffic demand SLA or QoS policies on an abstract level for eventual distributed provisioning on a granular flow-level by the NHCC. The *provisioning* step allows the user to manually deploy policies, as defined in the characterization step, as well as to statically configure overlay routing if necessary. The *active network analysis* and *policy tuning* steps allow the user to take active network measurements to verify network or deployed policy state and then to refine or un-deploy policies to further tune or stabilize the network, based on unexpected dynamics of the newly

provisioned environment. Finally, the user enters the *reporting* phase, where the historical state of the environment is aggregated and presented for further analysis. The manual process may be followed again accordingly, or the user may put the system in auto-pilot mode, entering into the inner process cycles for automated provisioning and management.

In the automated mode, the system is considered to operate without user intervention. As such, only active network analysis services, policy (i.e., link/flow measurement threshold and provisioning policies) and tuning services are employed to manage against network dynamics. The policy-configurable short and long cycles allow the system to react (or counteract) to dynamic burst conditions as triggered by event thresholds.

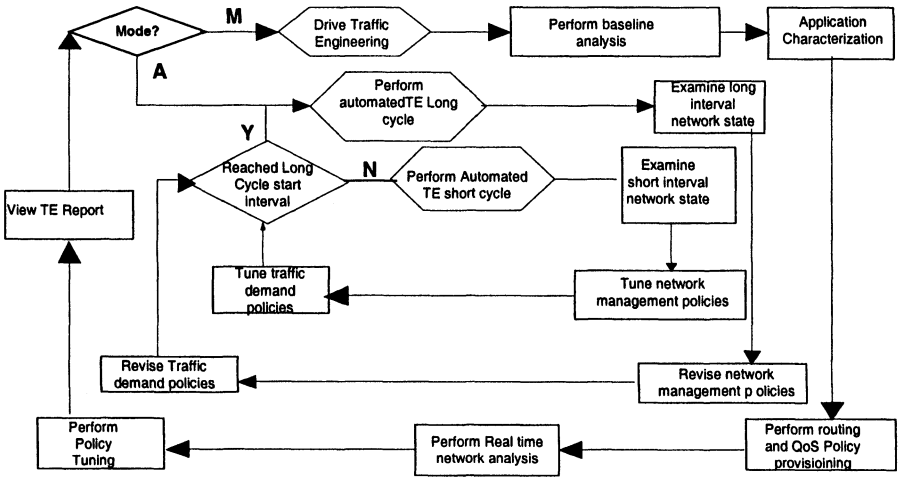

Figure 1. Policy-Based Traffic Engineering Methodology

4. TRAFFIC ENGINEERING SERVICES IMPLEMENTATION

This section describes the automated traffic engineering services that are available through the distributed network management system described in Section 2. The system is comprised of three major subsystems; Network Topology Discovery Service, Network Analysis Service, SLA Management & Verification Service. These three services operate in conjunction and in a completely automated fashion to ensure user-level service requirements are satisfied.

4.1 Automated Network Topology Discovery Service

The logical interconnection of SNs constitutes a virtual network that overlays the actual physical network. It is this virtual network topology that is meaningful and useful for making provisioning and other network management decisions. The Automated Topology Discovery subsystem is responsible for discovering the topology of the overlay network within the NHCC managed environment. The service captures and identifies all possible paths between SN pairs. There are two types of paths, namely *"Direct Paths"* and *"Tunnel Paths"* which are described next.

4.1.1 Identification of Direct Paths

A "Direct Path" is defined as a path between a pair of SNs as determined by trace-route. The NHCC sends trace-route commands to every SN within its domain instructing it to periodically run trace-route from itself to every other SN within the domain. The trace-route results are communicated from every SN in the domain to the associated NHCC. The use of trace-route has several advantages. First, there are also no security issues as would be associated with getting access to, for example, routing tables or network configuration files. Second, the path taken during trace-route tests also represents the actual path data with travel between the communicating SNs. Finally, the distributed trace-route process is completely automated, requiring no administrative configuration or input.

4.1.2 Identification of Tunnel Paths

While direct paths may exist between a SN pair, there may also be several non-overlapping paths between the same pair. These alternate paths are referred to as "Tunnel Paths". For these paths, the NHCC further analyzes the trace-route results from the SNs and determines all possible paths between every SN pair. The "Tunnel Path Identification" algorithm is based on the simple transitivity rule:

If there exist three non-intersecting paths: A to B, A to C, and C to B, then there exists a path from A to B via C.

The algorithm further eliminates redundant paths. The result of the analysis is the complete set of unique paths between every SN pair which are then stored in a Path database tree.

4.2 Automated Network Analysis Service

The Automated Network Analysis Service is used to determine the characteristics of the virtual network. It periodically runs active analysis tests across the direct and tunnel paths to obtain updated information about the paths such as maximum capacity, available capacity, latency, loss rate and jitter with minimal overhead impact to the existing network traffic. Since an alternate path is a concatenation of direct paths between two or more SN pairs, the analyzed direct path metrics are correlated to continuously update measurements of capacity, latency and jitter of the tunneled paths.

4.2.1 Determination of path capacity and jitter

The "*burst test*" is used for determination of path capacity and jitter between SN pairs. It consists of periodically sending a short burst of UDP packets from the source SN to the destination SN. Based on techniques described in [DISPTECH] and [PATHCR], the burst traffic is generated short enough so that it does not overfill forwarding queues (i.e., causing packet drops) and maintains constant inter-packet spacing at the source SN.

The burst test request policy is an aggregate of a traffic generation policy at source SN, monitor policy at source SN and drop policy at destination SN. The traffic generation policy at the source SN instructs the SN to periodically generate a 10msec burst of UDP packets of specified size to an unused port on the destination SN. The monitor policy at the source and the drop policy at the destination SN collects statistics pertaining to the burst traffic at the source and destination, respectively.

Jitter is computed as a sum of "phase jitter" and "inter-packet jitter". Phase jitter refers to a difference between the average inter-packet departure and arrival times. Inter-packet jitter refers to the magnitude by which the inter- arrival spacing of each packet at the destination is distributed about the average inter-arrival spacing at the destination. This magnitude may be measured as a standard deviation about the average. Thus,

$$PhaseJitte\ r(u\ \text{sec}) = (ADS - AAS)$$

where, ADS = Average inter-packet departure times; AAS = Average inter-packet arrival times.

$$InterPacketJitter(u\ \text{sec}) = \sqrt{\frac{n\sum x^2 - \left(\sum x\right)^2}{n^2}}$$

where, n = Number of packets – 1; x = inter-packet spacing.

The bandwidth computed is the maximum path bandwidth at a given instant. Thus, if TS_{last} is the last packet arrival time corresponding to the burst, TS_{first} is the first packet arrival time corresponding to the burst, and $Bytes_{total}$ is the total number of bytes received over the same time period, then the instantaneous capacity is computed as:

$$Rate\ (bps) = \frac{Bytes_{total} * 8}{TS_{last} - TS_{first}}$$

The instantaneous values of jitter and bandwidth are averaged and historical data is maintained to facilitate trending. The Path database tree which holds the virtual topology information is overlaid with the aggregated statistics to support proactive analysis and reporting.

An example of burst analysis is shown in Figure 2.

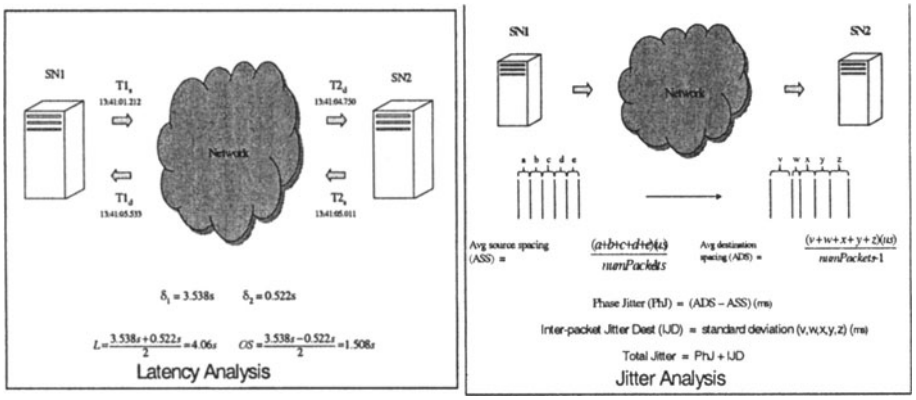

Figure 2. Examples: Jitter and Latency Analysis

4.2.2 Determination of path latency

The "latency test" is used for estimating path latency of the various paths between SN pairs. It is comprised of sending a UDP echo-request packet periodically from the source to the destination SL.

The latency test is an aggregate of traffic generation policy and monitor policies at the source SN and monitor policies at the destination SN. The traffic generation policy instructs the source SN to periodically generate a UDP echo-request to port 7 (i.e., echo port) on the destination SN. The UDP packet is echoed back from the destination SN. The monitor policies at the source SN and destination SN monitor the UDP burst traffic. The results from the source and destination SNs are collected and analyzed by the NHCC to infer the instantaneous latency along the measured path.

There are two types of latency measurements. The first is "*RTT Latency*" which is computed as the time it takes the UDP packet to complete a round trip from the source to the destination and back to the source again. This assumes that the path is symmetrical. The second is "*AsyncLatency*" which is computed as the time it takes the UDP echo request packet to be transferred from the source to the destination. In addition, the clock offset between the source and destination SNs is also computed.

Thus,

$$RTTLatency(u\,\text{sec}) = (TS4 - TS1)$$

$$AsyncLatency(u\,\text{sec}) = \frac{|\Delta 2 + \Delta 1|}{2}$$

$$ClockOffset(u\,\text{sec}) = \frac{|\Delta 2 - \Delta 1|}{2}$$

where, TS1 = timestamp of the UDP packet when it is sent by source SN; TS2 = timestamp of the UDP packet when it arrives at destination SN; TS3 = timestamp of the UDP packet when it is echoed back from destination SN; TS4 = timestamp of the echoed UDP packet when it arrives at the source SN; $\Delta 1$ = TS2 – TS1; and $\Delta 2$ = TS4 – TS3.

An example of latency estimation is shown in Figure 2.

4.3 Automated SLA Provisioning and Verification

The third subsystem of the automated traffic engineering system performs SLA provisioning and management. A customer would typically specify the QoS requirements in the form of a Service Level Specification (SLS) [SLS-SPEC]. This subsystem uses the updated information from the Automated Network Analysis to determine the most current network conditions.

4.3.1 SLA provisioning and path selection

SLA provisioning is the process of allocating resources along suitable paths in the network to meet service level requirements of customer traffic. When an SLA request is made, the subsystem translates the service level requirements into quantifiable metrics such as throughput, latency, jitter, and loss rate depending on the traffic type. Once the requirements of the SLA traffic have been quantified, the appropriate path (i.e., direct or tunnel) is selected based on the requirements of the SLA traffic type as well as the current path conditions as determined by the active network analysis

subsystem and knowledge of other SLAs that have been provisioned along the path during the same time period.

After path selection, the SLA is provisioned by installing suitable policies on each of the SNs along the path. If a tunnel path is selected, then the SLA traffic must be tunneled through the alternate path. Currently, IP-based tunneling is used; however, the model can be extended to incorporate other tunneling schemes (e.g., MPLS) as well. Once the SLA traffic is appropriately provisioned along a path, the path metrics are updated to reflect this.

4.3.2 SLA verification and re-provisioning

Once the SLA traffic has been provisioned along a selected path, it is necessary to iteratively "verify" the SLA. SLA verification is defined as the process of validating whether SLA specifications have been met for the duration of the SLA. The NHCC passively monitors the SLA traffic to determine if the SLA traffic is performing per specification. By gathering statistical data from the source and destination SNs, it is possible to infer that there have been some changes in the network conditions (e.g., path or link failures or link capacity changes).

The Automated Topology Discovery and the Automated Network Analysis subsystem periodically monitor and update network status to the SLA provisioning module. If the provisioned traffic does not meet the SLA requirements, the SLA is appropriately re-provisioned. The re-provisioning logic determines if there are any alternate paths available that can satisfy the new SLA requirements. If there are multiple SLAs provisioned along the "problem" path, then all the SLAs along that path may need to be re-provisioned. Every SLA is assigned a "*SLA precedence*". SLA precedence determines the relative priority of the SLA with respect to other SLAs and is used as criteria for determining the SLAs that need to be removed or reassigned lower service in favor of higher precedence SLAs.

5. RESULTS

In this section, we present some preliminary results from our evaluation of the prototype system. The experimental setup is shown in Figure 3. It consists of three subnets with Smart-Links at the edges of each subnet. The NHCC resides on one of the subnets. By default all the links are of 100Mbps capacity. We first tested the performance of the active path analysis tests that were described in Sections 4.2.1 and 4.2.2. These tests form the basis of the automated network analysis subsystem. Figure 3 shows the results of

running the latency test. The plot shows both the RTT Latency estimates as well the Async Latency estimates. The paths were symmetrical, and hence the Async Latency estimate is approximately half the RTT latency estimate as expected. The burst test was conducted between one pair of SNs with a packet size = 1518 bytes and burst size of 61 packets per burst period (i.e., 10msec). The bursts were generated at an inter-burst period of 30 seconds.

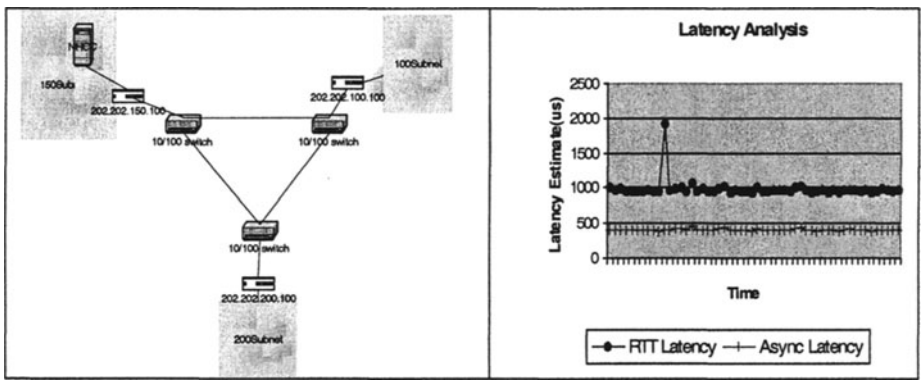

Figure 3. TestBed SetUp and Results From Latency Test

Figure 4 shows the results of the burst test analysis. It is observed that the bandwidth as obtained from the burst test, is on average, approximately 100Mbps, which is the maximum path capacity. The same burst test was also used to determine the jitter along the path. The total jitter estimate is on average approximately 425us.

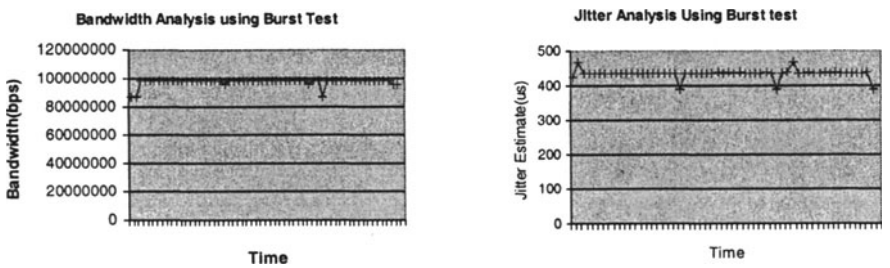

Figure 4. Burst Test Analysis: Bandwidth and Jitter Estimates

Preliminary testing of the Automated Topology Discovery and Automated Network Analysis processes on a LAN testbed, as illustrated in

Figure 3, and over a similar, lower-speed WAN topology indicated changes to the topology or link capacity can take up to 3 minutes to detect and complete the discovery process and then, to perform the network analysis on the updated topology. The system is designed such that the topology discovery and the network analysis processes are mutually exclusive. However, optimizations to the prototype have been identified that would significantly improve the convergence time after network changes.

A prototype version of Automated SLA Management scheme has also been implemented and tested on the testbed.

6. CONCLUSION

Intelligent networks which are capable of adapting to changing network conditions are becoming increasingly necessary in order to rapidly and efficiently adapt to changing service level needs of customer applications and their demands. This paper presents a distributed traffic engineering system consisting of Smart-Nodes choreographed by a Network Health Control Center, which together constitute a network control and feedback system provisioned through a web-based Traffic Engineering Console. A closed-loop methodology for employing automated and manual traffic engineering is introduced; the cyclic procedures support network topology discovery, real-time network analysis and SLA provisioning and verification. The distributed framework and TE methodology have been implemented, evaluated and deployed over a multi-site Trials WAN.

REFERENCES

[SOAP-SPEC] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. F. Nielsen, S. Thatte, D. Winer. "Simple Object Access Protocol (SOAP) 1.1", May 2000.

[COPS-RFC] J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Rajan, A. Sastry. "The Common Open Policy Service Protocol", RFC 2748, January 2000.

[DIFFSERV-RFC] Blake, et al. "Architecture For Differentiated Services", RFC 2475, December 1998.

[SLS-SPEC] D. Goderis, S. V. D. Bosch, Y. T'joens, O. Poupel, C. Jacquenet, G. Memenios, G. Pavlou, R. Egan, D.Griffin, P. Georgatsos, L. Georgiadis, P. V. Heuven. "Service Level Specification Semantics and Parameters", Internet draft, February 2002.

[DISPTECH] C. Dovrolis, P. Ramanathan, D. Moore. "What Do Packet Dispersion Techniques Measure?" In IEEE INFOCOMM, April 2002.

[PATHCR] A. Downey. "Using PathChar To Estimate Internet Link Characteristics", In ACM SIGCOMM, September 1999.

[TAILGAT] K. Lai, M. Baker, "Measuring Link Bandwidths Using a Deterministic Model of Packet Delay." Proceedings of the ACM SIGCOMM 2000 Conference, Stockholm, Sweden, August 2000.