

Bridging the Gap Between Service and Network Control

Eric Brendel
NetSocket
Plano, Texas

ebrendel@netsocket.com

Next generation network architectures promise an evolution of the Internet to one that is rich and diverse, capable of supporting multimedia real-time services across a diverse set of access technologies. Architectures being specified and developed for the various network access types are being specified by 3GPP/3GPP2 (IMS), Cablelabs (PacketCable), ETSI TISPA (NGN), MSF, and MEF. These architectures all have, in common, a SIP-based service control plane with an underlying network layer based on IP/MPLS. Session-based control allows providers to monitor, bill, authorize and manage service and subscribers. Convergence on a common IP/MPLS network provides for a cost-effective packet based technology that provides any-to-any communication and an access to the wealth of services being developed over the Internet.

A key component to these similar (IMS based) architectures is the notion of policy and QoS. Sessions requiring real-time guarantees will benefit from having QoS enabled throughout the network for that session. These architectures provide a framework for how sessions can be authorized, subscribers billed, and network resources assigned. In the specifications of these architectures there are defined functions, or devices, that are responsible for performing policy-based admission control and configuring network elements for QoS. This functionality or device is referred to differently in each of the architectures being specified. Some names given to this class of device include Policy Decision Function (PDF), Resource and Control System (RACS), Policy Server, and Bandwidth Manager.

Part of the duties of this policy device is to communicate policy and configure network elements to support the required QoS. However, a couple of problems associated with it become apparent.

1. One is that QoS mechanisms exist in the IP/MPLS network; however, there is no linkage between the session request and the mechanisms that provide the QoS in the underlying IP/MPLS network. The vision of most of these architectures is to rely on a set of defined protocols, DIAMETER and COPS, to communicate with the network elements to implement QoS in the underlying Layer 2 and Layer 3. However, most infrastructure devices deployed in Layer 2 and Layer 3 switching and routing networks either **do not** support either of these protocols or are unwilling to deploy them.
2. Another shortcoming of these architectures is the lack of support for end-to-end QoS. These architectures are primarily focused on configuring and managing QoS in point devices, primarily the access device that terminates the customer access connection. Configuring QoS in this device alone, ignores the majority of the IP/MPLS network where congestion and packet lost may occur. And

while QoS mechanisms exist in the IP/MPLS network, currently, there is no specification or implementation that includes the end-to-end path and coordinates QoS from the access device through the network to the destination.

The result is that operators are building a service control plane that requires flow-based QoS in the underlying network, but they lack the capability to configure the network to support the required QoS and monitor the network to ensure that the quality remains.

Perhaps, the real source of the problem is that the service control plane and the network (routing) control plane have not only been designed separately, but are operated separately. Two completely different industry mindsets conceived and orchestrated the two separate layers. The world of routers and switches (the network layer) was architected and deployed by “netheads.” Any-to-any “good-enough” packet delivery dominates this mindset. The services control layer was put together with the “bellhead” world of service setup and signaling as the main objective. Fine grained session-based control for billing, QoS, and management dominate this mindset.

Keeping service control separated from network control has its advantages (independent development, quicker deployment, less complexity). However, the evolving services and the operators that manage them clearly need real-time visibility into the underlying network. Network based admission control, QoS provisioning, path and session monitoring could all be of use to the operators and equipment in the service layer. Operators have some tools to support these, but there just is no easy way to provide it to the session control layer. The tools and the information are locked up in the routers.

Today, operators do the best they can to provide support to the services layer by using the tools that they have to monitor and control their network resources. For monitoring, they use passive monitoring techniques to obtain the network topology, available resources and routing state. These techniques usually involve SNMP polling of network elements and routers for traffic statistics and CLI-based scripts that perform a remote login to the router to obtain routing table dumps. However, these techniques suffer some major drawbacks. Primarily, they are not real-time, nor even quick. Obtaining routing table dumps and statistics from routers is cumbersome. The databases are very large and transmission of the databases requires time. Additionally, the routing state is likely to change during the acquisition leaving the data inconsistent. Once obtained, operator’s still need to be process, correlate, and analyze the data. By doing this the operator can collect a topology of the available resources and paths. Operators are forced to build costly, non-standard OSS systems that perform this analysis and interwork with other systems that do not adequately meet their requirements.

Assuming the operator has knowledge of available resources, QoS requests can be planned. However, QoS policy would need to be distributed across the network to all devices needing configurations updated. There is currently no easy, scalable way to do this. Sure, there exists lots of QoS technology out there (diffserv, rsvp-te, intserv, pen), but operators can not use them in a dynamic fashion. Today, operators are usually forced to manually configure network elements to reroute and classify traffic to achieve the desired result. Some more advanced operator's built and manages expensive systems to make the appropriate changes to the network. However, managing and building these special-purpose systems is clearly not in the best interests of the operator.

Perhaps a better solution lies in closer integration with the service plane with the underlying routing (network) control plane. Near real-time knowledge of the routes and paths established as well as the available resources can be readily available for session establishment and monitoring. Routing Control Platforms (RCP) can be an ideal point for integration of these domains.

The RCP has been proposed as a centralized BGP routing engine that provides a point within the network to automate *intelligent* decisions and monitoring capabilities. [1][2][3] It offers operators a interface to the state of the underlying network. It also provides an ideal point to inject routing information to control network paths specific applications. RCPs provide an advantage over traditional polling methods because they participate in the routing protocol sessions and can learn about changes in IP routing, network topology, and resource availability on a near real-time basis.

By combining a BGP based RCP with intra-domain topology based information obtained from the IGP and resource usage information (from traffic engineering extensions), an RCP would have enough information to assist service control. Add in the connection/service-oriented resource requests of IMS-like architecture with the network topology information, and session-based visibility of the underlying IP/MPLS network and resource state can be improved. Most policy distribution solutions have focused on distributing QoS through management plane protocols. Perhaps, an RCP based architecture can leverage the routing protocols to piggyback QoS relevant information. This visibility can improve the quality of evolving services by providing session control with better admission control, bandwidth management, path monitoring, network congestion notification, and fault mitigation

1. REFERENCES

- [1] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and implementation of a Routing Control Platform. In *Proc. Networked Systems Design and Implementation*, May 2005.
- [2] M. Caesar and J. Rexford. BGP policies in ISP networks. *IEEE Network Magazine*, October 2005.
- [3] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe. The case for separating routing from routers. In *Proc. ACM SIGCOMM Workshop on Future Direction in Network Architecture*, August 2004.