Int. J. Communications, Network and System Sciences, **2009, 7, 583-591**
doi:10.4236/ijcns.2009.27065 Published Online October 2009 (http://www.SciRP.org/journal/ijcns/).

A Scalable Architecture Supporting QoS Guarantees Using Traffic Engineering and Policy Based Routing in the Internet

Priyadarsi NANDA[1], Andrew SIMMONDS[2]
[1]*School of Computing and Communications, Engineering and IT*
University of Technology, Sydney, Australia
[2] *Mathematics, Computing and Technology, the Open University, England, UK*
Email: *pnanda@it.uts.edu.au, ajs2244@tutor.open.ac.uk*
Received February 25, 2009; *revised April* 10, 2009; *accepted June* 21, 2009

ABSTRACT

The study of Quality of Service (QoS) has become of great importance since the Internet is used to support a wide variety of new services and applications with its legacy structure. Current Internet architecture is based on the Best Effort (BE) model, which attempts to deliver all traffic as soon as possible within the limits of its abilities, but without any guarantee about throughput, delay, packet loss, etc. We develop a three-layer policy based architecture which can be deployed to control network resources intelligently and support QoS sensitive applications such as real-time voice and video streams along with standard applications in the Internet. In order to achieve selected QoS parameter values (e.g. loss, delay and PDV) within the bounds set through SLAs for high priority voice traffic in the Internet, we used traffic engineering techniques and policy based routing supported by Border Gateway Protocol (BGP). Use of prototype and simulations validates functionality of our architecture.

Keywords: QoS, BGP, Policy Based Routing, Traffic Engineering, Bandwidth Broker (BB)

1. Introduction

The success of the Internet has brought a tremendous growth in business, education, entertainment, etc., over the last four decades. With the dramatic advances in multimedia technologies and the increasing popularity of real-time applications, end-to-end Quality of Service (QoS) support in the Internet has become an important issue, which in this paper we address using Traffic Engineering and Policy Based Routing using BGP (Border Gateway Protocol), the core routing protocol of the Internet.

The Internet can be considered as a connection of Autonomous System (AS) domains, where each AS domain controls traffic routing in their own domain based on their own policies. These policies are defined to benefit the AS domain without consideration of other AS domains, which may result in policy conflicts while establishing a flow to achieve a certain degree of QoS on an end-to-end basis. Traffic Engineering concerned with resource allocation mechanisms has been widely studied [8,11–13] and also by us with a proposal for an integrated architecture bringing routing and traffic engineering along with resource management to support end-to-end QoS in the Internet [1]. The novelty of our scheme is mapping traffic engineering parameters into QoS paths available in the network and using policy routing to support end-to-end QoS. This is discussed in terms of the architecture of Figure 1 in Section 2 and how our schemes can be used to achieve some well known QoS objectives such as Delay, Throughput and Packet Delay Variation (PDV) for high priority voice traffic in the Internet. We conducted simulations to validate our results.

We introduce our architecture in Section 2 in order to guide the reader in understanding where traffic engineering and policy routing are used. In Section 3 we highlight the use of a Bandwidth Broker (BB), which is also part of our proposed architecture, to manage interdomain resources. Section 4 discusses our traffic engineering model reflecting the objectives for end-to-end QoS. Policy routing using Border Gateway Protocol

(BGP) is presented in Section 5. Simulation results to validate our model are discussed in Section 6 and finally our conclusion is given in Section 7.

2. An Integrated Architecture

In order to achieve a better service oriented model for the Internet, we propose a three layer policy based architecture for the Internet. The main functions of the architecture are presented in Figure 1.

One of the key components of our architecture is to separate out the control plane from the data forwarding plane by hierarchically grouping network management functions.

In this architecture, layer 3 end-to-end QoS, would be responsible for policy based routing and traffic engineering to dynamically provision bandwidth between different domains. Having determined the route, the layer 3 policy agent would inform the layer 2 of the preferred route. This route provisioning provides a connectivity overlay on top of the normal IP routing, such that if the route from Domain *A* to Domain *B* changes at the IP layer it is not necessary to change the overlay routing. The fall back position for a null layer 3 is that routes will be statically provisioned between individual domains so as to carry the flow to the destination domain.

Layer-2, Network Level QoS. The management unit in this layer is a Bandwidth Broker (BB) [2,3,14]. This interfaces to layer 1 and 3 devices, but also supports inter-domain resource control functions in cooperating with BBs in neighboring domains. Note that the policy function is an add-on to the BB function, i.e. with a null policy to accept everything, BBs can support end to end QoS, but any domain which wishes to implement network policies can do so to its benefit without affecting the functionality of the BB layer.

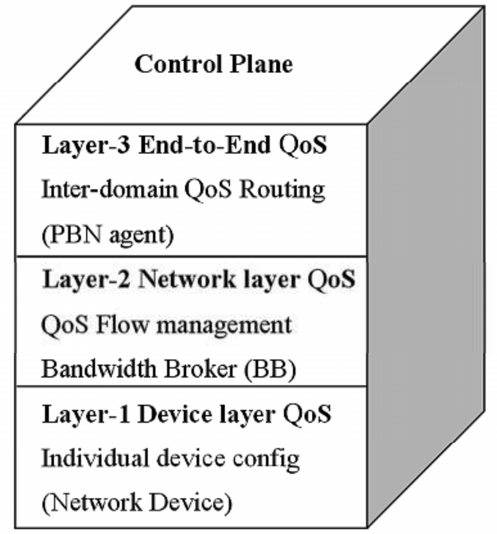


Figure 1. Logical view of the architecture.

The inclusion of null policies and layers is important to enable a gradual take-up of these tools in the Internet. It is not necessary for all domains to implement all levels before anything can work. We present the prototype of our BB design in Section 3 of this paper.

Layer 1, Device Level, is where network devices are configured to support the QoS levels agreed on in the higher levels, getting their instructions from higher layers in the architecture. One possible QoS mechanism being Differentiated Services (DiffServ) (RFC 2475) with Common Open Policy Service (COPS) [13] (RFC 2748, RFC 3084) and being used for signaling. Units in this layer are network devices such as routers and switches and the operation is purely intra-domain.

3. Bandwidth Broker (BB) Design

The conventional definition [2,3] of a Bandwidth Broker (BB) is an agent, running in an Autonomous System (AS), which manages resources within its own domain and with adjacent BB domains, to provide Quality of Service (QoS) support for traffic flows across multiple domains. BBs use hop-by-hop based routing to negotiate with other BBs (the inter-domain function) to provide agreed levels of service for selected traffic flows. Flows getting this preferential treatment will normally be expected to pay more, and this is expected to be a driver in sharing Internet resources as well as providing a revenue stream for Internet Service Providers (ISPs).

A BB controls the network devices in its own domain (the intra-domain BB function) which provide QoS functionality, such as routers and switches. Note that for scalability it is best if the core routers have as little to do as possible apart from forwarding packets, so there should be no interaction between a BB and the core routers. As no particular QoS mechanism is linked to the BB function, different domains can run different QoS mechanisms if they choose. As long as BBs can communicate with each other and agree on common definitions for the level of service required by different priority flows, then a consistent level of QoS support can be set up across different domains for a particular flow. When a new request for a particular QoS arrives, BBs pass the request from one to another, such that if resources are free all along the chain from source to destination then the request is allowed, else it is rejected.

We developed a prototype for a simpler BB architecture and signaling protocol which we believe can be implemented easily. A BB is a resource manager, the resource often being taken as simply bandwidth (BW), as in our prototype, but it could be high quality (e.g. low delay or low jitter or low loss links), buffers, or even low cost, low quality links. The six traffic classes we use for sake of example, in descending priority with binary values for the DiffServ field [15], are:

1) Network traffic – 11100000 (used for BB signaling)

2) Expedited Forward (EF) - 1011 10xx (used e.g. for VoIP)

3) Assured Forward Gold (AFg) - 0111 10xx, AF33

4) Assured Forward Silver (AFs) - 0101 10xx, AF23

5) Assured Forward Bronze (AFb) - 0011 10xx, AF13

6) Best Effort (BE) - 0000 00xx, default

RFC 2597 [16] defines the Assured Forward "Olympic" Per Hop Behavior (PHB) classes and RFC 3246 [17] the EF PHB class. A drop precedence of 3 was chosen for the AF values for compatibility with the deprecated TOS field of the IP packet header, giving flag settings for (D = 1) low delay, (T = 1) high throughput, and (R = 0) normal reliability.

The resources monitored in our implementation are simply additive, but statistical multiplexing could be used to carry more paying traffic over reserved links, as [18] suggests. Our current implementation is open loop, that is available resources are entered in a database (DB) and the BB subtracts resources from the available total as requests are granted, and adds resources when flows finish. Eventually the aim is to have closed loop control, by deploying a resource discovery mechanism to actually measure queue length, etc., e.g. as proposed by one of us using Fair Intelligent Admission Control (FAIC) [19].

The design philosophy we chose is one we believe is consistent with the design philosophy of the Internet: where we faced a design choice we chose the simplest solution, and we implement a minimum function set which can then be extended to provide added functionality.

4. Traffic Engineering Issues

An important objective of Internet traffic Engineering is to facilitate reliable network operations by providing proper QoS to different services through mechanisms which will enhance network integrity and achieve network survivability. The objective of traffic engineering measures in our architecture is to achieve load balancing between neighboring ASs using BGP parameters. By doing so, the architecture then optimizes resource utilization across multiple links, maps divergent QoS parameters to the paths which can support end-to-end service qualities, and avoids congestion hot-spots across the Internet.

In our architecture we used BGP routing to send traffic between domains. But BGP routing policies are not designed specifically to address traffic engineering issues in the Internet. Instead, they are designed to support routing policies determining network reachability between ASs. Obtaining a globally optimized routing path in the Internet is a difficult task due to different policy requirements. Our aim to achieve a scalable solution is based on the following assumptions while incorporating traffic engineering into the architecture:

1) The use of community attributes in policy routing to add extra policy information into the BGP path announcements, enabling traffic engineering to map different QoS parameters to the available paths computed using policy routing.

2) That load balancing traffic with different policies across multiple available routes to the same destination is performed only when the policy co-ordination algorithm for a specific path fails.

Hence our proposed traffic engineering solution can be stated as parameter mapping to different QoS paths available in the Internet, using a policy co-ordination algorithm to resolve any policy conflicts between different ASs while selecting a QoS routing path. In order to be more specific on the issue of parameter mapping, we identified three important parameters related to real-time services such as VoIP application:

a) Bandwidth: When different bandwidth capacities are available in different AS domains for a specific policy in an end-to-end QoS path, the BW allocated is the BW of the AS with the minimum available BW. This minimum bandwidth also needs to satisfy the performance requirements for VoIP traffic in order for the path to be selected.

b) Delay: Two components of end-to-end delay are important for VoIP traffic: delay due to codec processing and propagation delay. ITU-T recommendation G.114 [4] recommends one way delay values less than 150 ms for most user applications, 150 to 400 ms for international connections, with more than 400 ms deemed to be unacceptable. ASs can indicate end-to-end delay in their own domain between edge routers. Hence, complete end-to-end delay for a QoS path would be the sum of all the delays offered by individual AS provided that the sum satisfied the delay requirements specified by G.114. An AS receiving the path announcement along with the delay value from its neighbor adds its own delay and then announces the sum to other ASs further along.

c) Packet Delay Variation (PDV): as it is now properly called rather than jitter, affects real time services, e.g., voice and video traffic. For non real-time voice and video traffic PDV can be removed by a buffer in the receiving device. However if the PDV exceeds the size of the PDV buffer, the buffer will overflow and packet loss will occur. PDV is caused by queuing and serialization effects on the packet path, and is defined by the IETF (RFC 3393) as the difference in delay between successive packets, ignoring any delays caused by packet loss. The one-way delay being timed from the beginning of the packet being sent at the source to the end of the packet being received at the destination. To clarify further, if consecutive packets leave the source AS domain with time stamps t1, t2, t3, …, tn and are played back at the destination AS domain at times t1', t2', t3', …, tn', then

$$\textbf{Maximum PDV} = \text{Max}\ \{\text{Abs}\ [(t_n' - t_{n-1}') - (t_n - t_{n-1})],\ \ldots,\ \text{Abs}[(t_2' - t_1') - (t_2 - t_1)]\} = \text{Max}\ \{\text{Abs}\ [(t_n' - t_n) - (t_{n-1}' - t_{n-1})],\ \ldots,\ \text{Abs}[(t_2' - t_2) - (t_1' - t_1)]\}$$

PDV can also be signed, where a positive PDV indicates that the time difference between the packets at the destination is more than that at the source, and vice-versa.

Hence, while mapping QoS parameters such as bandwidth (BW), Delay (d), and PDV (j) for a specific QoS path, traffic engineering considers the following, where $1 \leq i \leq k$ are the ASs involved in the end to end path:

$$\text{BW} = \text{Min}\ \{\text{BW1}, \text{BW2}, \ldots\ \text{BWk}\}$$
$$\text{Delay} = \text{Sum}\ (\text{d1}, \text{d2}, \ldots\ \text{dk})$$
$$\text{PDV} = \text{Max}\{\text{Abs}\ (\text{j1}, \text{j2}, \ldots., \text{jk})\}$$

And minimizing cost over all the announced path would be given by:

$$\text{Min}\ [\text{C1}|P_1 - A_1| + \text{C2}|P_2 - A_2| + \ldots\quad \text{Ck}|P_k - A_k|],$$

where P is the required policy parameter, A is the announced value of the policy parameter by a neighbor which exported the path and C is the cost associated with these parameters which determines the weight for them. Such costs are important to consider when different ASs have different QoS objectives to satisfy a given Service Level Agreement (SLA) for their customers. In a standard traffic engineering problem, the aim is to minimize the maximum utilization of links, whereas in our architecture it is to maximize the number of AS domains which support the above mentioned constraints. Hence traffic with different policies can be distributed among those paths, improving overall traffic engineering objectives by using the traffic engineering framework of Section 4 and the policy routing of Section 5.

4. Traffic Engineering Framework

The framework is based upon the fact that ASs must communicate with their neighbors to get a fair picture about which relationships they must hold with them in order to apply specific traffic engineering policies in their respective domains. At the same time, ASs must also protect themselves against route instabilities and routing table growth which may otherwise occur due to misconfigurations or problems in other ASs. Manually configuring routing will of course achieve optimum results if the routing is configured optimally. However, Internet routing is complicated so manually configuring routing will not achieve optimal routing in practice, and misconfigurations may well cause catastrophic failure to the Internet. Hence we seek an automatic solution. The components of our traffic engineering framework are presented in Figure 2.

The middle layer (network layer QoS) of our architecture presented in Section 2 has the necessary components for including network policies in traffic engineering.

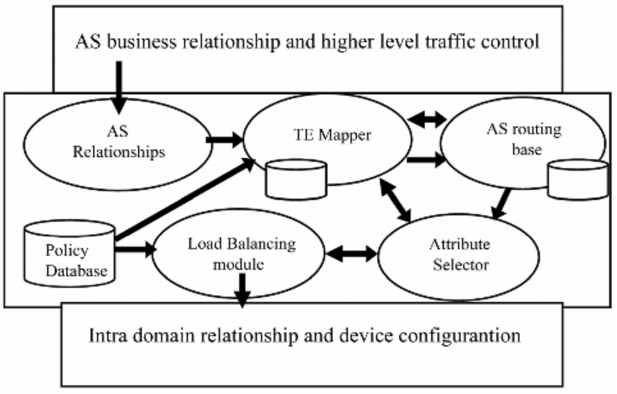

AS business relationship and higher level traffic control
AS Relationships
TE Mapper
AS routing base
Policy Database
Load Balancing module
Attribute Selector
Intra domain relationship and device configurantion

Figure 2. Framework of traffic engineering.

AS relationships play an important role supporting QoS in the Internet. But obtaining data on such relationship is a difficult task, as ASs such as ISPs may not reveal such data to their competitors. Hence we propose to use a measurement based approach where an ISP ranks ASs based on the frequency of their presence in the routing table. A heavily used AS in the path list is one where some kind of traffic engineering should be applied if selected for next hop forwarding. For example the decision of selecting local preference is very much local to an ISP in order to balance its outgoing traffic (selecting the path to forward packets to the next ISP). On the other hand, an AS which is used less frequently is less congested and has a better chance of providing QoS resources [5].

Traffic Engineering Mapper (TEM) has a repository that holds AS relationships and the hierarchy for interconnectivity between various ASs. TEM is responsible for directing those relationships to the Attribute Selector as well maintaining a list of those attributes once selected. Because the routing table holds information regarding import and export policy filters, as well the attributes associated with them, TEM also investigates their validity in the AS routing base. One of the export rules based on the business relationship between ASs is for the TEM to enforce the provider to send all routes (customer as well as provider routes) that the provider knows from its neighbors. Alternatively, TEM could ensure that peer or provider routes are not sent when sending routes to another provider (i.e. just send customer routes). TEM is an essential component of traffic engineering framework.

Finally, the decision on traffic engineering is taken by the Load Balancing module which receives necessary inputs regarding which attributes are to be applied and to which paths they must be applied. The policy database holds policy information regarding how the AS may change routing for its own domain. Also included in the policy database is information on a list of remote ASs which are also customers of this AS, and pricing structures imposed by the SLAs of its providers. Such information is given to the load balancing module which then takes a final decision on traffic engineering. The process is the same for both importing and exporting a route between neighboring ASs.

Several efforts on finding solutions to BGP based traffic engineering and AS relationships have been explored in the past [6–9]. While the authors described some drawbacks of BGP in the first instance and then proposed their schemes on better management of BGP for traffic engineering, our approach is different as we consider the relationship between ISPs as a central issue in defining necessary traffic engineering policies for the Internet, and add a community policy attribute to BGP to solve this issue. Hence our proposal builds on BGP to provide a solution. Policy routing using BGP is presented in the following section of this paper.

5. Policy Routing

Routing protocols play an important role in exchanging routing information between neighboring routers. Such information may be used to update routing tables and to share information about network status so that traffics to appropriate destinations will be set up quickly, efficiently and achieve the required QoS between end systems. Different types of routing protocols are in widespread use across the Internet. Apart from determining optimal routing paths and carrying traffics through the networks, these routing protocols should have additional functionalities such as resource discovery, policy mapping and policy negotiation mechanisms to support network policies, traffic engineering and security.

BGP is a path vector protocol that uses AS path information between neighboring routers in different AS domains to determine network reachability. Such network reachability information includes information on the list of ASs and the list of AS paths. One of the important features supported by BGP is policy routing, where an individual AS can implement network policies to determine whether to carry traffic from different users (mostly users from other ASs) with diverse QoS requirements. Such network policies are not part of BGP, but provide various criteria for best route selection when multiple alternative routes exist and help to control redistribution of routing information, resulting in a rich support by BGP for policy routing and traffic engineering in the Internet.

Current Internet Traffic Engineering depends heavily on both Intra and Inter Domain routing protocols using network policy in order to configure the routers across various domains. The support for policy based routing using BGP can provide source based transit provider selection, whereby ISPs and other ASs will then route traffic originating from different sets of users through different connections across the policy routers. Also QoS support for Diffserv networks can be supported using policy routing through the use of the DiffServ field in the IP packets. Hence, a combination of traffic engineering for load balancing across network links offered by destination based routing, and policy based routing, can enable implementation of policies that distribute traffic among multiple paths based on traffic characteristics.

Policy routing in the Internet can be based on the following principles:

1) Each AS to take action on routing based upon information received from neighbors. Such decision process is central within each AS.

2) Neighbors are free to negotiate any policy conflict by adjusting their traffic parameters and waiting for confirmations from all the domains involved in routing.

3) Incorporation of a direct relationship between network level flow management and traffic engineering objectives.

Routing traffic across several routers in the same domain to support QoS between the edges of the network is relatively easy to achieve, as we can gather knowledge on QoS paths and select edge routers administrated by a single network entity. But inter-domain QoS path selection is difficult to achieve and to demonstrate how we can approach such a problem, we present the policy routing framework in Figure 3. We assume that the intra-domain QoS path computations are already optimized based on the local knowledge of intra-domain routing protocol and this information is already stored in layer-2 of our architecture.

Standard BGP routing process involves applying an import policy onto routes received from neighbors, deciding the best route based on BGP routing decision process [10] and then applying export policy to the computed routes before announcing to neighbors. Such a process does not take all policy decisions into account, particularly while computing the routing paths in support of QoS in the Internet. The inter-domain route selector which is central to the routing module within an AS domain receives path announcements from the neighbors through the inbound route announcement. Apart from applying standard BGP decision process on selecting certain route advertisement from its neighbor, the route

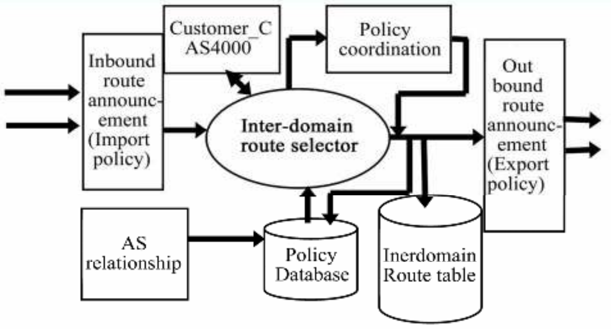

Customer_C AS4000
Policy coordination
Inbound route announcement (Import policy)
Inter-domain route selector
Out bound route announcement (Export policy)
AS relationship
Policy Database
Inerdomain Route table

Figure 3. Interaction between routing components for policy based routing.

selector needs further consultation for QoS path selection by interacting with the following components:

- It is important to decide which types of neighbor (e.g., provider, customer or peer) the route advertisement came from and based on that, the AS will then decide whether to announce the path to its neighbor. Such relationships are held in a policy database which then inputs the information to the route selector.
- The route selector gets path information within its own domain by communicating with the intra-domain QoS path repository. Actions such as changing values for LOCAL_PREF, MED, IGP Cost, and Pre-pending AS_PATH results in directing incoming traffic to a specific edge router.
- The decision process also needs to consider which QoS policies are supported by the AS domain which sent such path announcements. For this, each AS, which can support different policies in relation to QoS services (e.g., Premium, Gold, Silver, Bronze), adds a "COMMUNITY" policy attribute along with the path announcement.
- In case of policy mismatch i.e., advertised policies by neighbor does not match with the AS's own policy, the route selector will apply "policy co-ordination algorithm" (Subsection 5.1) to resolve such conflict.

Finally routes selected by either the route selector without any policy mismatch, or applying policy co-ordination algorithm in case of any policy conflict, are further announced to ASs through outbound route announcement. The announced route is stored in the AS's inter-domain routing table.

5.1 Policy Co-Ordination Algorithm

An algorithm performing such functions is presented below:

```
Get list of policies from neighbor
For each neighbor policy {
  Compare policy support with own policy list
  If match {
    Set values and put policy in End-to-End (E-E) list
  }
  Else (no match) {
      Tag policy as non-confirmed and put policy in
  Temp list
    }
  } (All policies checked)
  For all policies in the Temp list {
  Check if another route satisfies policy constraints
  If match {
      Set values and put policy in E-E list
    } (End the process of policy comparison)
    Else (no match) {
    For all policy mismatch {
      Adjust own policy and apply traffic engineering
  parameters for new policy
      Select the ones which contribute to maximum
   revenue
      Announce all paths to neighbors in the list
  }
      Set values and put policy in E-E list
    } (End the process of policy adjustment)
  } (Temp list emptied)
```

Finally in order to validate our algorithm and functional models, we conducted a series of experiments using OPNET based simulation to take into account the effect of traffic engineering and policy routing which are presented in the next section of this paper.

6. Simulation Results

In order to validate our algorithm and functional models we performed a series of experiments and obtained various statistics from the simulation. The topology and the default routing paths between customers A, B and C are presented in Figure 4 below:

A-B - - - , A-C ━━ and B-C -·-·

As presented in Figure 4, the network is created by configuring all default values into the devices and network reachability test is performed to ensure end-to-end connectivity between each AS domains in the network. Once these are performed, based on routing table entries, we performed our analysis on how BGP paths are recorded between different AS domains without any policy but with its default routing decision process. The complete network diagram is presented in Figure 5 below which also presents end-to-end connectivity between all the domains.

Our second scenario in Figure 6 demonstrates the effect of our proposed policy mechanism compared with the base-line scenario in Figure 4. The end-to-end path between customers now have different routes as a result of policy enforcements across all the AS domains.

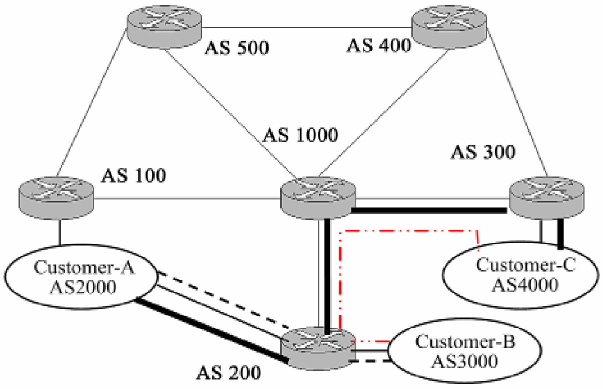
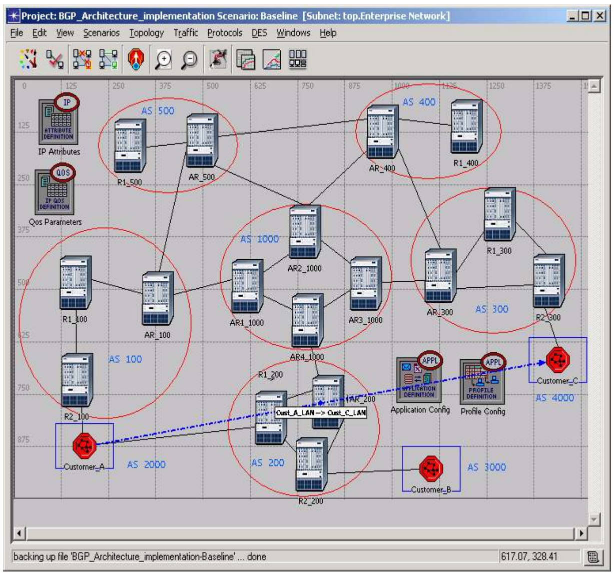


Figure 4. Simulation topology and default routing paths.


Project: BGP_Architecture_implementation Scenario: Baseline [Subnet: top.Enterprise Network]

backing up file 'BGP_Architecture_implementation-Baseline' ... done

617.07, 328.41

Figure 5. End-to-end network configuration.

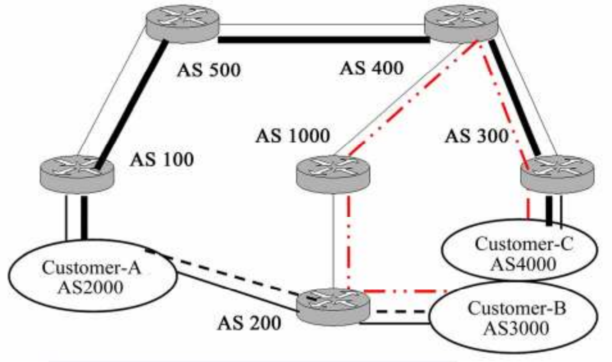


Figure 6. End-to-end path using policy.

The results show effect of our proposed Community attribute for selecting specific QoS domains using BGP routing process between end nodes. Such a scheme not only balances traffic distribution across inter-domain links but also fine tunes traffic engineering for better provisioning of QoS between end domains. However, the scheme does increases complexity in BGP decision process due to extra information involving the community attribute.

Traffic was generated from a G_711 interactive voice source with duration of 1 hour and several experiments were conducted to demonstrate the quality of voice traffic on an end-to-end basis. We assigned a DSCP value of B8 (EF=184) to the VoIP traffic which is then mapped to a BGP community value of 0x00640184 to ensure voice quality is maintained strictly between end domains. A series of graphs representing QoS parameters for VoIP applications are presented through Figure 7 (a-d).

While sending QoS aware applications in the Internet such as VoIP, we are mainly concerned about maintaining delay budget within the limit set for QoS assurance. The plots in Figure 7 (a-d) represent Packet Delay Variation (defined as jitter by OPNET), end-to-end delay, variance of the end-to-end delay and BGP updates, averaged over a 10 minute period for the scenario with policy routing enabled on all the routers running BGP. The actual VoIP traffic starts after 2 minutes and is deliberately set to make sure that BGP timer values are taken into account.

In our experiment, plot (a) demonstrates the variation in packet end-to-end delay (PDV) and shows that it is kept to low bounds (-0.3 µs to +0.1 µs), in spite of activating multiple QoS and routing policy configurations across the whole network. The PDV is influenced by packet scheduling and queuing strategy implemented across the routers (layer-1 functions) in order to support QoS within and across various domains. PDV is reported as the maximum absolute time difference between the instances when successive packets are received at the destination minus the time difference between the instances when these packets are sent at the source, averaged over 10 minutes, which is equivalent to the IETF definition assuming constant packet processing times at the destination.

The end to end delay for VoIP traffic is maintained at a value ≤ 50.4 ms (plot b), well within the SLA of 150 ms, while PDV converges to less than 0.1 µs (Plot a).

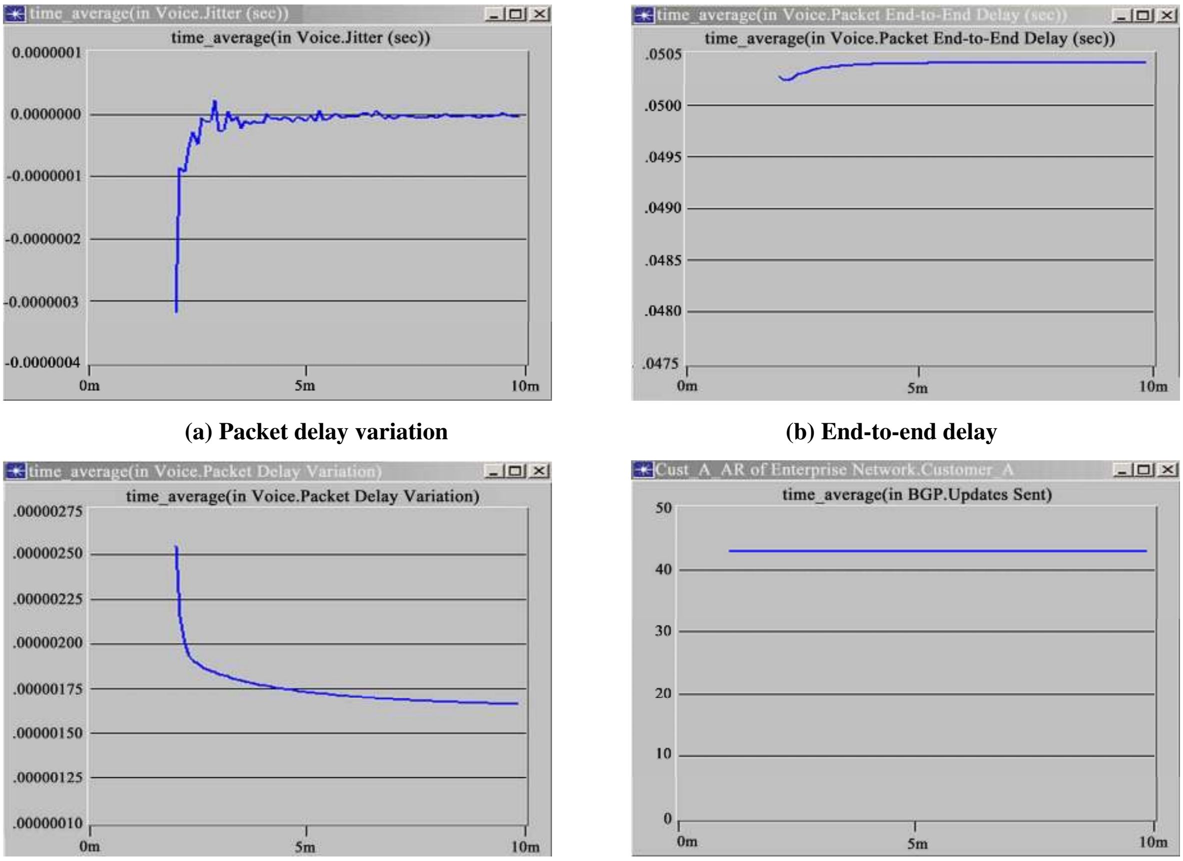
Plot c shows that the variance of the end-to-end delay falls to less than 1.75 µs after 5 min. This is confusingly defined as Packet Delay Variation (PDV) by OPNET, but we will use the IETF definition for PDV.

Plot d presents number of BGP updates. In our simulation the access router in Customer_A network (Customer_A_AR) is the one where most policies related to load balancing and traffic engineering are enforced. For this reason we collected the BGP updates sent by this router which contains either new routes or unfeasible routes or both in the system. In our case this access router sent 43 updates at 69 s due to strong policy enforcement.

As shown above, voice traffic sent between Customer_A network and Customer_C network experienced QoS parameters well within our design limits. However these parameters could be further improved by carefully selecting other QoS strategies within individual domains.

7. Conclusions

In this paper we demonstrated the effect of Internet traffic engineering and use of policy routing to achieve end-to-end QoS for high priority Voice traffic, in the context of our high level architecture of Figure 1. We also presented simulation results to demonstrate how we achieve automatic load balancing between different service providers using a BGP community policy attribute and the policy co-ordination algorithm of Subection 5.1.



(a) Packet delay variation

(b) End-to-end delay

(c) End-to-end delay variance

(d) BGP updates at Customer_A_AR

Figure 7 (a-d) VoIP QoS measurement.

This is substantially different from the default routing which does not select the AS domains based on QoS requirements for an application. Such results are evidence that our scheme improves end-to-end QoS requirements for high priority voice traffic particularly when many other applications are running simultaneously in the Internet.

The objective of our design is how BGP can be used to select QoS domains for QoS support. For this reason we are mainly concerned with AS domain traffic behavior contributing to policy routing and traffic engineering.

8. References

[1] P. Nanda and A. J. Simmonds, “Policy based QoS support using BGP routing,” International Conference on Communications in Computing, CIC’06, Las Vegas, Nevada, USA, CSREA Press, ISBN 1–60132–012 –4, pp. 63–69, June 26–29, 2006.

[2] B. Teitelbaum, “QBone bandwidth brokerarchitecture,” [Online]Available:http://qbone.internet2.edu/bb/bboutline2.html.

[3] K. Nichols, V. Jacobson, and L. Zhang: “A two-bit differentiated services architecture for the internet,” IETF RFC 2638, July 1999.

[4] ITU–T Recommendation G.114, One way transmission time, 1996.

[5] A. D. Yahaya and T. Suda, “iREX: Inter–domain QoS automation using economics,” IEEE Consumer Communications and Networking Conference, CCNC’06, USA, January 2006.

[6] N. Feamster, J. Winick, and J. Rexford, “A model of BGP routing for network engineering,” SIGMETRICS/ Performance’04, New York, USA, June 12–16, 2004.

[7] D. O. Awduche, A. Chiu, A. Elqalid, I Widjaja, and X. Xiao, “A framework for internet traffic engineering,” Draft 2, IETF, 2002.

[8] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure, “Internet traffic engineering with BGP,” Quality of Future Internet Services, Springer, 2003.

[9] G. Di Battista, M. Patrignani, and M. Pizzonia, “Computing the types of relationships between autonomous systems,” IEEE INFOCOM, 2003.

[10] Y. Rekhter and T. Li, “A border gateway protocol 4 (BGP-4),” Internet draft, draft-ietf-idr-bgp4-17.txt, work in progress, January 2002.

[11] B. Quoitin, C. Pelsser, O. Bonaventure, and S. Uhlig, “A performance evaluation of BGP-based traffic engineer-

ing," Int'l. J. Network Mgmt, 2005.

[12] R. Yavatkar, D. Pendarakis, and R. Guerin, "A framework for policy-based admission control," RFC 2753, January 2000.

[13] S. Salsano, "COPS usage for Diffserv resource allocation (COPS-DRA)," Internet Draft, October 2001.

[14] P. Nanda and A. Simmonds, "Providing end-to-end guaranteed quality of service over the Internet: A survey on bandwidth broker architecture for differentiated services network," CIT'01, 4th International Conference on IT, Berhampur, India, pp.211–216, 20–23 December 2001.

[15] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers," IETF RFC 2474, Dec. 1998.

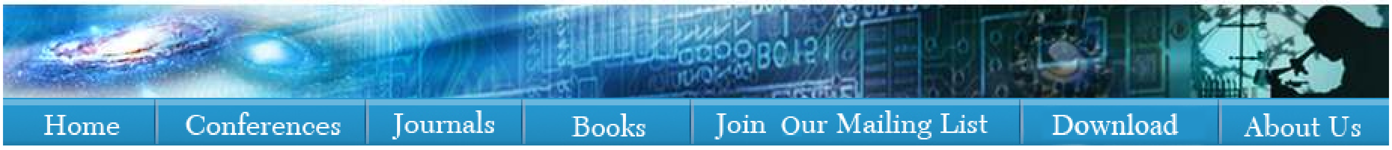
[16] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group," IETF RFC 2597, June 1999.

[17] B. Davie, *et al*., "An expedited forwarding PHB (per-hop behavior)," IETF RFC 3246, March 2002.

[18] P. Pan, E, Hahne, and H. Schulzrinne, "BGRP: Sink-tree-based aggregation for inter-domain reservations," Journal of Communications and Networks, Vol. 2, No. 2, pp. 157–167, June 2000.

[19] M. Li, D. B. Hoang, and A. J. Simmonds, "Fair intelligent admission control over DiffServ network," ICON'03, 11th IEEE International Conference on Networks, Sydney, Australia, ISSN: 1531–2216, pp. 501–506, 28 Sept–1 Oct 2003.

Scientific Research Knowledge is Power



Home | Conferences | Journals | Books | Join Our Mailing List | Download | About Us

Indexing | **View IJCNS Articles** | **Aims & Scope** | **Editorial Board** | **For Authors** | **Paper Submission** | **Contact Us**

Int'l J. of Communications, Network and System Sciences RSS

ISSN Print: 1913-3715 ISSN Online: 1913-3723 CNPIEC Serials Number: 734B0148

Website: http://www.scirp.org/journal/ijcns

Aim & Scope

IJCNS is an international refereed journal dedicated to the latest advancement of communications and network technologies. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these fast moving areas. The journal publishes the highest quality, original papers included but not limited to the fields:

- Ad Hoc and mesh networks
- Antenna and Circuit
- Coding , detection and modulation
- Cognitive Radio
- Efficient MAC and resource management protocols
- MIMO and OFDM technologies
- Network protocol, QoS and congestion control
- Network security
- Sensor networks
- Signal processing and channel modeling
- Simulation and optimization tools
- 3G and 4G technologies
- UWB technologies
- Wave propagation and antenna design

We are also interested in short papers (letters) that clearly address a specific problem, and short survey or position papers that sketch the results or problems on a specific topic. Authors of selected short papers would be invited to write a regular paper on the same topic for future issues of the IJCNS.

LOGIN

Email:

Pwd:

Sign In

New Users?

Forget My Password?

Search GO

News

- Download IJCNS Vol. 1, No. 1, February 2008
- Download IJCNS Vol. 1, No. 2, June 2008
- index information

Related Conferences

- WiCOM

Microsoft Internet Explorer is recommended if this page is not displayed properly !