TEKNOLOJİ, Volume 7, (2004), Issue 1, 161-169

TEKNOLOJİ

# A REGION-BASED APPROACH TO INTER-DOMAIN TUNNEL SCALABILITY

**İbrahim Taner OKUMUS*   Haci Ali MANTAR****
*Mugla University Technical Education Faculty, 48000, Mugla, Turkey
**Harran University, Şanliurfa, Turkey

## ABSTRACT

In this paper we present a scalability study on interdomain tunnels using Bandwidth Brokers. Our approach is based on collections of Autonomous Systems called regions. We analyzed our architecture through simulation. Our analysis results show that scalability of the approach depends on the size of the regions, and as the size of the region decreases, scalability of the approach increases. We show that our architecture increases state scalability as well as signaling scalability for Diffserv networks.

**Key Words:** Diffserv, Bandwidth Brokers, QoS Signaling.

## ALANLARARASI TÜNEL ÖLÇEKLENEBILIRLIGINE BÖLGE TEMELLI YAKLAŞIM

## ÖZET

Bu makalede bandgenişliği komisyoncular kullanilan alanlararasi tünellerin ölçeklenebilirliği üzerine bir çalişma sunuyoruz. Yaklaşimimiz otonom sistemlerinin biraraya gelmesinden oluşan bölgelere dayanmaktadir. Mimarimizi simulasyonlarla inceledik. İncelemelerimiz yaklasimimizin ölçeklenebilirliğinin bölgelerin boyuna bağli olduğunu ve bölgenin boyu küçüldükçe yaklaşimimizin ölçeklenebilirliğinin arttiğini gösterdi. Mimarimizin Diffserv ağlar için hem işaretleşme ölçeklenebilirliğini hem de durum ölçeklenebilirliğini arttirdiğini gösterdik.

**Anahtar Kelimeler:** Diffserv, Bandgenişliği Komisyoncuları, QoS İşaretleşme.

## 1. INTRODUCTION

As Internet Quality of Service (QoS) architectures start to mature, providing end-to-end QoS becomes more important. One early attempt to provide end-to-end QoS for the Internet was the RSVP/Intserv [1] architecture. This approach is based on microflow reservations. Resource reservations are intended to be end-to-end, and all the routers on the network need to keep the states of all the reservations they maintain. When this is combined with microflow management of RSVP/Intserv, scalability became a huge bottleneck for this approach.

The Differentiated Services (Diffserv) [2] architecture was proposed to overcome the problems faced with the RSVP/Intserv architecture. Using Diffserv, core routers have no knowledge of individual flows, but only know of a small set of classes of traffic. This pushes the complexity to the edge of the network, dramatically reducing the amount of state maintained by core routers, making them simpler and more manageable.

Diffserv alone does not provide end-to-end QoS; to solve this problem, Bandwidth Brokers [3] were proposed to augment the base Diffserv functionality. Bandwidth Brokers (BB) are centralized agents that provide admission control, resource management, and resource negotiation services to a Diffserv domain. Bandwidth brokers make policy decisions and configure Diffserv routers to implement the policy, thereby easing the load on the edge routers. In Diffserv, multiple flows of the same class are aggregated and treated as a single flow, and this feature of Diffserv is essential to provide end-to-end QoS in a scalable manner.

One of the responsibilities of a Bandwidth Broker is to manage intra-domain and inter-domain resources. Intradomain resources are relatively easy to manage because of the autonomy of the domain, and the Bandwidth Broker's control of its own domain. For inter-domain resource management, Bandwidth Brokers need to communicate and negotiate Bandwidth Brokers in neighboring domains.

The Simple Inter-Domain Bandwidth Broker Signaling (SIBBS) Protocol [4] is used for inter-domain Bandwidth Broker communication. This protocol is a simple request response protocol. There are two basic message types: Resource Allocation Request (RAR) and Resource Allocation Answer (RAA). RAR messages are used for requesting resources for a specific type of service or a specific class of QoS from another Bandwidth Broker. RAA messages are used for responding to the RAR messages, and response can be negative or positive. Results of RAR-RAA exchange can be reservation of resources specified in the RAR message.

In the Bandwidth Broker/Diffserv architecture, scalability can also become an issue depending on how the resources are managed. Having Bandwidth Brokers handle every individual reservation request is essentially the same as the Intserv management of microflow reservations, and suffers the same scalability problems.

Tunneling can ease the scalability pressure on Bandwidth Brokers. A tunnel is an inter-domain reservation where one (or both) end of the reservation is not fully specified [4]. A tunnel is established from the edge of one domain to the edge of another domain, and tunnels are established unidirectionally. In this paper we refer to these tunnels as edge tunnels. A Bandwidth Broker establishes an edge tunnel to the destinations requested by its domain's customers. When we think of the size of the Internet today (as of March 01, 2003, there are 14742 autonomous systems in the Internet [5]), establishing an edge tunnel from a domain to all possible destinations is simply not scalable. Every Bandwidth Broker keeps the state of the reservations it originates, and also keeps the reservations that other domains initiated to its own domain. If the number of domains in the Internet is n, then every domain needs to keep at least (n-1) + (n-1) reservation states. In addition to this, the number of control messages required to establish and maintain these tunnels is proportional to the number of domains on the Internet. Moreover, when we think of the Bandwidth Brokers that sit at the core of the Internet, the state scales with $O(n^2)$.

In this paper we are proposing an architecture that increases the scalability of edge tunnels on a Bandwidth Broker supported Diffserv Internet. Our architecture represents the Internet as regions where every region consists of a number of transit and transit-only domains. Transit domains carry traffic whose endpoints are not connected to that domain; stub domains have directly-connected networks that may contain endpoints for communication. Our approach, described in detail in section 3, uses region affinity to reduce the number of tunnels maintained by each domain.

Section 2 gives a brief review of previous work. Our Proposed architecture is described in detail in section 3. We present the evaluation results in section 4 and give concluding remarks and future research directions in section 5.

## 2. RELATED WORK

The QBone Bandwidth Broker Architecture is described in [4]. In this model, a BB can handle reservations individually, or it can establish core tunnels to all possible destinations. Although the core tunneling case is specified in the document, the exact mechanism and details are left as research questions.

Guerin, et al., [6] proposed to use RSVP for aggregated QoS requests to increase scalability. Several possible choices are given in the document which includes RSVP tunnel-based aggregations. In this approach, tunnels are established between source destination pairs and traffic from source to destination is aggregated into the same tunnel.

Pan, et al. [7] proposes an inter-domain reservation protocol (BGRP) to increase the inter-domain reservation signaling and state scalability. This protocol uses destination-based sink trees to aggregate reservation requests. Reservation requests coming to a router are aggregated into the same tunnel if the destination is the same. The scalability of this approach depends on the destination definition, whether it is a network, or an autonomous system etc.

Mantar, et al., [8] proposed to use destination-based aggregation with BB. Inter-domain tunnels are established using BB and SIBBS protocol and on a domain, the same class of traffic destined to the same destination is aggregated into the same tunnel. This approach and the BGRP approach show significant improvement compared to the aggregated and nonaggregated RSVP cases for inter-domain reservations.

Our approach lies in between the two approaches. We are proposing to have tunnels established inside every region between every possible source destination pairs, and traffic destined to other regions will be aggregated into a single tunnel that goes to that region. Our basic assumption is that every source domain conditions the traffic originating from its domain based on destination before sending the traffic to the next domain. Inter-domain and intra-domain tunnels can be established as Label Switched Path (LSP) tunnels.

## 3. REGION-BASED INTER-DOMAIN CORE TUNNELING

In our architecture we represent the Internet in terms of regions. We consider the Internet as a network I = (N, L), where N is a finite set of nodes and L is a finite set of links. Every node n Є N is an autonomous system (AS). A region R = (V, E) is a connected graph such that V is a subset of N and E is a subset of L. Core regions consist of transit-only AS and transit AS. Stub regions consist of stub-domains and their transit service providers. Transit-only AS are the ones that only provides transit services for other domains and do not host any network. Transit networks hosts networks and also provides transit services for other domains. Stub-domains are the ones that only host networks and do not provide transit services to other domains.

Every domain in our architecture is a Diffserv domain that employs a Bandwidth Broker (BB). Figure 1 shows the representation of a network in our architecture.

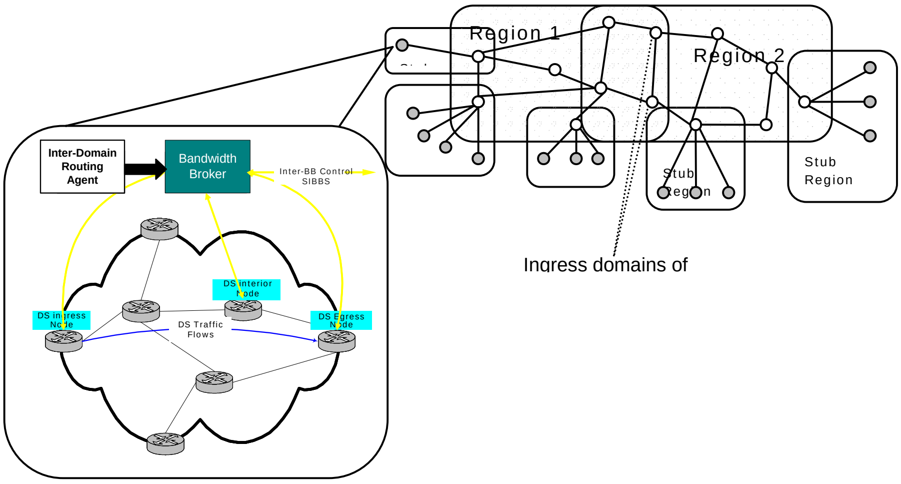

Figure 1 Inter-Domain and Intra-Domain representation of an internetwork.

We can increase scalability for inter-domain tunneling by reducing the number of states kept at each BB. The straightforward method of managing tunnels requires $O(n^2)$ state at each BB for an Internet that has n number of autonomous systems. Some approaches focus on considering all autonomous systems but reducing the state to be O(n) [7], [8]. Our method maintains an $O(N^2)$ approach, but focuses on reducing N, the number of autonomous systems for which we maintain state. Clearly, for an Internet with n autonomous systems, we must ensure that $N \leq \sqrt{n}$, on average, for our approach to be competitive.

In our architecture, every domain inside a particular region establishes a tunnel to every other domain inside that same region, and also to ingress domains of neighboring regions. Thus, the number of states a BB keeps, for a region R with k neighboring regions is approximately $N = ||R|| + k$. Recall that we want $N^2 \leq n$ , or, in other words, the total number of states kept at each BB to be less than the number of nodes in the overall network. Our simulation results, given in detail in section 4, show that this holds. The path to establish the tunnel is decided by the region-based inter-domain QoS routing protocol introduced in our earlier work [9]. A connection starting from a region and ending in another region uses multiple tunnels to reach the destination. The end-to-end path will be the concatenation of regional tunnels for that connection.

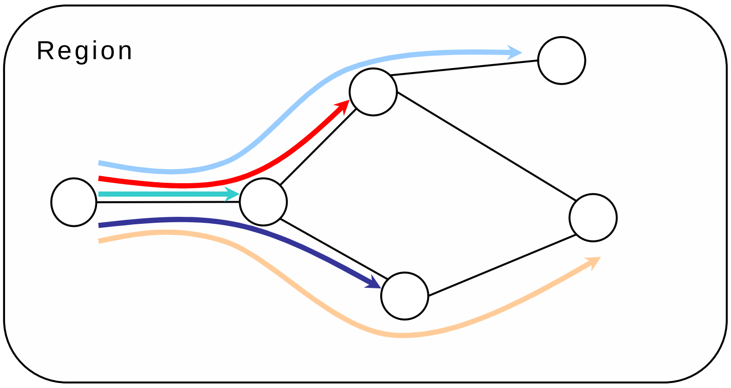

Figure 2 Every domain establishes a tunnel to every other domain in its own region.

Establishing and maintaining regional tunnels increase the state and signaling scalability at Bandwidth Brokers, and the degree of scalability depends on the size of the region. Inter-domain tunnel establishment involves inter-BB communication through the Simple Inter-Domain Bandwidth Broker Signaling (SIBBS) protocol. SIBBS can be used for either per-flow reservations or for aggregate reservations, i.e., core tunneling. Because using SIBBS for per-flow reservations incurs the same scalability penalties as Intserv, we use SIBBS for core tunneling.

In our architecture, each BB establishes tunnels to every other domain in its own region. Tunnel sizes are determined based on expected demand from the neighboring domains, and can be dynamically adjusted in time [10].

One issue in tunnel establishment is finding the path for the tunnel. In our region-based, inter-domain QoS routing protocol, each domain is represented as a single routing agent. These Inter-Domain Routing Agents (IRDA) communicate with each other to collect reachibility information. A BB of a domain communicates with its own IDRA to get the QoS path to a destination. An IDRA is capable of calculating a QoS path to any destination on the network, but works more efficiently if the destination is in its own domain. This reduces the signaling messages required to get the path to destinations that are not in the same region as the source.

Once a BB gets the path information to the destinations in its own region, the BB initiates reservation procedures to set up the tunnel through the calculated path. Reservations are done using the SIBBS-TE protocol. The BB sends RAR messages to the next BB on the explicit path. The RAR message includes authentication, authorization information as well as QoS parameters for the path. Every BB on the path processes the RAR message, relaying it to the next BB on the path if the request conforms to local policy. The destination domain also processes the RAR message, and if it is acceptable, prepares an RAA message. The RAA message follows the inverse path of the RAR message. When the source domain BB gets the RAA message, path setup is complete. This process is repeated for every domain in the source domains' region.

Our basic focus is on transit and transit-only domains; we omit stub domains from the core view of the Internet. One of the reasons for us to focus on transit domains is that tunnel states kept at each stub domain are far fewer than the tunnel states kept at transit domains. Stub domain Bandwidth Brokers need only to track state for tunnels with an endpoint in that domain, while BB for transit domains need to keep tunnel states not only for its own tunnels, but also for transit tunnels that only pass through the transit domain. If we look at the current size of the Internet, every transit domain has on average 7 stub domains. This means a transit domain will have on average 8 times the number of states kept at stub domains. This is the lowest limit on the number of states kept at a transit domain. If that transit domain also carries traffic for another transit domain, the number of tunnel states kept at a transit domain BB will increase exponentially. Because of this, if we can decrease the number of states kept at transit domains, we will solve the major scalability problem in terms of inter-domain tunnels.

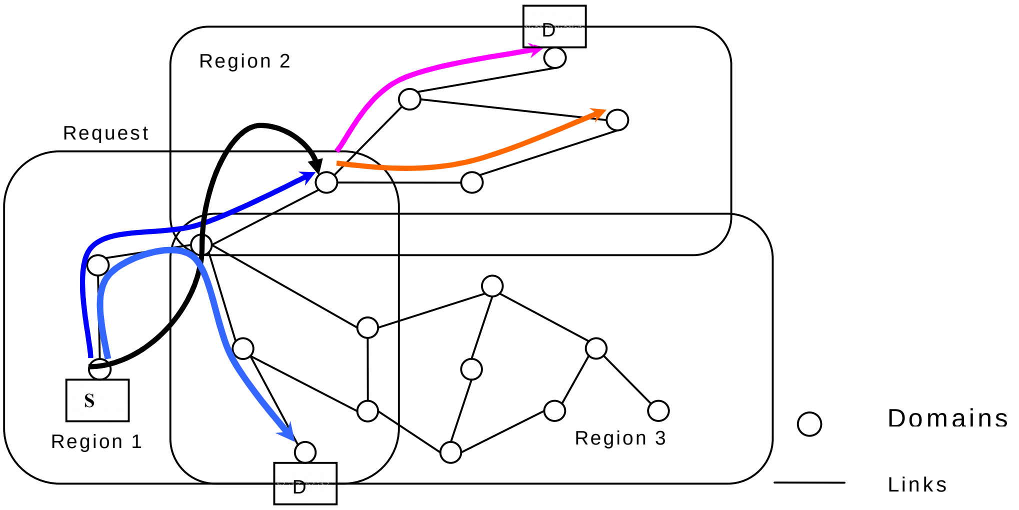

Figure 3 In-region and out-of-region tunneling.

Once every BB establishes a tunnel to every other domain in its region, the BB simply puts the incoming traffic destined for domains in the same region into the appropriate tunnel, provided that the capacity is available. This situation is represented with the reservation request from S to D2 in Figure 3. S has a pre-established tunnel to D2 and simply forwards the new traffic into this tunnel. If the destination of the traffic is in another region, the BB needs to communicate with the corresponding region's BB to get capacity for the traffic. In order to do that, BB does not need to communicate with a series of BB to setup a tunnel to that destination. The BB has already created a tunnel to the ingress domain of that region. That ingress domain also already has established tunnels to possible destinations in its own region and the ingress domains of neighboring regions. Therefore, the source BB simply sends an RAR message to the BB of ingress domain of the corresponding region, and if necessary the ingress domain's BB forwards that RAR to another ingress domain's BB of another region. In Figure 3, if a traffic is destined to D1 from S, S sends a request to ingress domain BB of Region 2 and if the capacity is available, traffic will follow the tunnel from S to ingress domain of Region 2 and then into the tunnel to D1. In the evaluation section, we will show that with this region-based approach the maximum number of regions crossed by traffic to reach a destination is 3. This increases the signaling scalability to establish a tunnel to any destination in the Internet. Instead of sending signaling messages to every BB of every domain on the QoS path, the BB just sends the RAR message to the ingress domains of the regions.

## 4. EVALUATION RESULTS

We evaluated our approach using the ns2 simulation tool [11], with topologies derived by the BRITE topology generator [12] using a heavy-tailed Waxman model [13]. We evaluated our architecture using three different network sizes: 100 nodes, 1500 nodes, and 3000 nodes. All topologies consist of only transit domains.

We simulated each topology using a variety of region sizes to determine the effect of the size of the region on the number of tunnel states kept at each BB. For a 100 node topology, we simulated one region, five regions, and 11 regions, and used the one-region result as the basis for our comparison. The average number of tunnel states kept at each BB using only one region for the entire network gives an upper limit for this topology. The five-region and 11-region simulation results are compared to the results with one region. For the 1500 node topology, we simulated for 24 regions, 44 regions, and 60 regions. We did not simulate the 1500 node topology for one region because of the long simulation time for this configuration; instead, we used 24 region results as the basis for our comparison. For 3000 nodes we used a single configuration with 128 regions.

For evaluation purposes, we calculated a QoS path for every region with a fixed amount of bandwidth from every domain to every other domain inside the same region. For routing purposes, ingress domains of neighboring regions are also kept in the same regional link state area. This ensures that calculating a QoS paths to neighboring domains are automatically calculated by the routing algorithm. These calculated paths are AS-level paths, which means that the results of the calculation for a specific destination is a series of AS starting from the origin domain and ending at the destination domain. Our basic assumption is that at the initial tunnel establishment phase, capacity is available to establish all the tunnels for a specific QoS class. After calculating all the possible paths, we analyzed the number of paths passing through a domain.

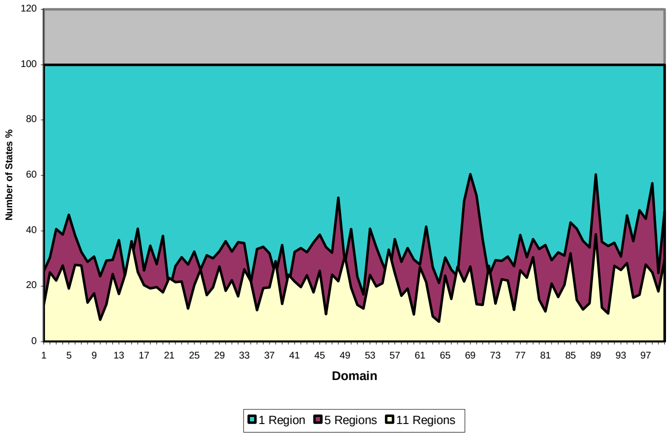

Figure 4 Number of States kept at each node in 100 node network for different region sizes.

Figure 4 shows the number of states kept at each BB on 100 node simulation for different region sizes. Number of states in 1 region topology is taken as 100% since this configuration is the base configuration for our comparison and figure shows the number of states in terms of percentage. For one region of 100 nodes, which is the entire topology, the average number of tunnels established by the BB is 436.7. For five regions, the average region size is 38.6, which is 61.4% less than the base configuration of one region. The average number of tunnel states kept at each BB for his configuration is 148.58. The five region configuration has 66% fewer tunnel states than the base configuration. For this configuration, the maximum number of regions to reach to any destination is one. The 11-region configuration's region size is 18.5, which is 81.5% less than the base configuration region size; 89.33 tunnel states kept per BB, reaching our goal of being less than the number of nodes in the network. This suggests a reduction of 79.5% over the base configuration's average number of states and 39.9% reduction over the five region configuration's average number of tunnel states. The 11-region configuration requires on average 1.25 regions to reach any destination in the topology.

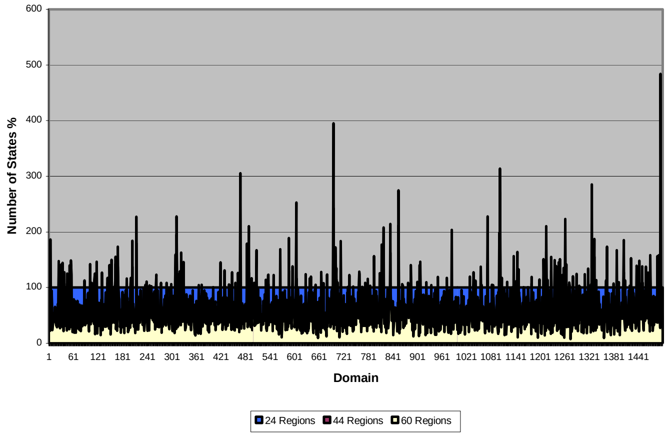

Figure 5 Number of states kept at each node in 1500 node network for different region sizes.

Figure 5 shows the number of states kept at each BB for 1500-node simulations for different region sizes. For this topology we used the 24-region simulation results as a basis for our comparison. The 24-region configuration has an average of 121.75 domains inside a region, and an average of 899.7 tunnel states kept at each BB. For this configuration, the average length of regional path is 1.01. The 44-region configuration has on average 69.3 domains inside a region, which is 43% less than the 24 region configuration. The average number of states kept at each BB for this configuration is 512.8. This suggests a 43% decrease from the 24 region configuration. For the 44-region configuration, average size of the regional path length is 1.28. The 60-region configuration's average region size is 50.65. This is 58.4% less than the 24-region configuration and 26.9% less than the 44-region configuration. The average number of tunnel states kept at each BB for this configuration is 373.96, which is 58.4% less than the 24-region tunnel states and 27% less than the 44-region configuration. For the 60-region configuration, the maximum number of regions to pass to reach any destination in the topology is on average 2.3. Figure 6 shows average number of states for 100 node and 1500 node simulation for different region sizes we used during the simulations. For our 3000 node simulation, the average number of states is 358.

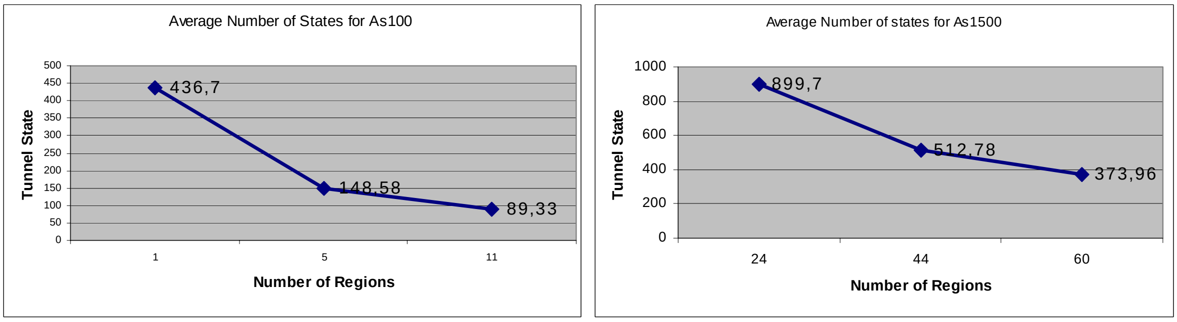

Figure 6 Average number of states for 100 node and 1500 node simulations for different region sizes.

These evaluation results suggest that as the size of the region decreases, number of states kept at each BB also decreases. Relative amount of decrease in number of states is very closely follows the relative amount of decrease in region size. For 1500 node, these numbers are approximately equal. Note that as the number of regions goes up, the average number of states maintained at each BB drops, and fairly quickly drops below the overall network size, suggesting that we have fulfilled our condition that $N \leq \sqrt{n}$; this data is summarized in Table 1; entries fulfilling our condition have the number of states in italics.

Table 1 Summary data for network size, number of regions, and number of states.

| Nodes | Regions | Avg. states |
|---|---|---|
| 100 | 1 | 436.7 |
| | 5 | 148.6 |
| | 11 | 89.3 |
| 1500 | 24 | 899.7 |
| | 44 | 512.8 |
| | 60 | 374.0 |
| 3000 | 128 | 357.6 |

To increase the scalability in terms of the number of tunnel states kept at each BB, we need to decrease the region size. Another measure for inter-domain tunneling is the path setup time. If the destination is inside the same region as the source BB, than the request can be immediately answered and accepted if there is available capacity in the tunnel to that destination. If the destination is in another region, how this will effect the path setup time and the signaling scalability? Answer to this question lies in the average number of regions to pass through to reach a destination in the topology. Maximum number of regions to pass to reach any destination for any of the configuration is 3. For most of the configurations, average of this number is much less than 2. This means that if a source BB gets a resource allocation request to the farthest destination, RAR message needs to go to at most 3 BB which are the BB of the ingress domains of those regions. This suggests that region-based approach does not add significant stress on the path setup time and the signaling scalability while increasing the state and the signaling scalability.

## 5. CONCLUSION AND FUTURE WORK

In this paper we presented a region-based inter-domain tunneling approach. In our architecture a network is divided into regions that consist of autonomous systems. Inside a region every domain establishes a tunnel to every other domain inside the same region. Inter-regional traffic is forwarded via a tunnel that is established to the ingress domain of the corresponding neighboring region.

Our evaluation results show that that region-based approach significantly increases the state scalability compared to the base approach of tunnels between every possible pair of AS. Region size is the determining factor in terms of scalability. As the region size decreases, scalability increases. The amount of improvement in scalability and in region size is approximately the same. Our approach also increases the signaling scalability. Traffic originating and sinking inside the same region does not cause any inter-domain signaling messages to be sent. Traffic originating from a region and sinking in another region requires RAR messages which are sent only to the ingress domains of the regions that are on the path of the traffic.

As a future work, we will investigate the effect of using LSP tunnels combined with this architecture on scalability of the approach.

## REFERENCES

1. Wroclawski J, "The **Use of RSVP with IETF Integrated Services**", RFC 2210, Sep 1997.
2. Blake S et al. "**An Architecture for Differentiated Services**", RFC 2475, Dec 1998.
3. Internet2 Bandwidth Broker Working Group, ``**QBone Bandwidth Broker Architecture**." http://qbone.internet2.edu/bb/.
4. "**QBone Bandwidth Broker Architecture**", work in progress, http://qos.internet2.edu/wg/documents-informational/20020709-chimento-etal-qbone-signaling/.
5. **Asia Pacific Network Information Center**, http://www.apnic.net/mailing-lists/bgp-stats/index.shtml
6. Guerin R, Blake S, Herzog S, "**Aggregating RSVP-based QoS Requests**", internet draft, Nov 1997.
7. Pan P, Hahne E, and Schulzrinne H., "**BGRP: A Tree-Based Aggregation Protocol for Inter-domain Reservations**", Journal of Communications and Networks, Vol. 2, No. 2, June 2000, pp. 157-167.
8. Mantar H A, Hwang J, Okumus I T, Chapin S J, "**Inter-domain Resource Reservation via Third-Party Agent**", Fifth World Multi-Conference on Systemics, Cybernetics and Informatics 2001, Jun 2001, Orlando, FL, USA.
9. Okumus I T, Chapin S J, Mantar H A, Hwang J, "**Inter-Domain QoS Routing on Diffserv Networks: A Region Based Approach**", submitted for publication.
10. Mantar H A, Hwang J, Okumus I T, Chapin S J, "**Edge-to-edge Resource Provisioning and Admission Control in Diffserv Networks**" **IEEE SoftCOM 2001**, October, 2001 Split, Dubrovnik (Croatia) Ancona, Bari (Italy).
11. "**The Network Simulator - ns-2**", http://www.isi.edu/nsnam/ns/
12. "**Boston University Representative Internet Topology Generator (BRITE)**", http://www.cs.bu.edu/brite/
13. Medina A, Lakhina A, Matta I, and Byers J, "BRITE: An Approach to Universal Topology Generation." In Proceedings of the International Workshop on Modeling, **Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01**, Cincinnati, Ohio, August 2001.