Performance analysis of Capacity-aware State Aggregation for Inter-domain QoS routing

Venkatesh Sarangan
Computer Science Dept.,
Oklahoma State University,
Stillwater, USA, saranga@cs.okstate.edu.

Donna Ghosh, Raj Acharya
Dept. of Computer Sci. and Eng.,
Pennsylvania State University,
University Park, USA. (dghosh,acharya)@cse.psu.edu.

***Abstract*—Quality of Service (QoS) routing between domains is indispensable for deploying QoS in the Internet. State aggregation is a technique that makes QoS routing scalable to large networks. As the impreciseness in the advertised aggregate adversely affects the performance of the QoS routing process, it is important to investigate the accuracies of various aggregation schemes. In this paper, we study the efficacy of a new bandwidth aggregation scheme, and compare our scheme's performance with that of the existing schemes. Our analytical and experimental studies show that for active networks, the proposed aggregation scheme is more accurate, and gives a better routing performance than the current schemes.**

***Index Terms*— Inter-domain QoS routing, Topology aggregation, Network flows.**

I. INTRODUCTION

Topology aggregation is an important technique for reducing the message overhead involved in QoS routing. Large networks like the Internet are composed of different autonomous systems or domains. In topology aggregation, each domain constructs an aggregate that consists of two parts: (a) aggregate of the connectivity and (b) aggregate of the resource availability in the domain. After constructing such an aggregate, a domain advertises it to others. The routers in the internetwork will have detailed information about their own domain and aggregated information about other domains. Inter-domain routing decisions are based on the aggregated information while intra-domain routing decisions are based on the detailed information. Such a hierarchical approach reduces the overhead in QoS routing, making it scalable.

The topological aggregate has a compact representation when compared to the original network. Hence, the aggregation process usually results in an imprecise network characterization. The goal in topology aggregation is to create a network representation that is both concise and accurate.

A. Related Work

The ATM PNNI standard [1] has proposed techniques such as *mesh, star* and *star with bypasses* for aggregating the connectivity. The border routers (BRs) in the domain form the vertices in the aggregation and are connected using logical links. While the PNNI standard proposes techniques for aggregating the connectivity, it does not prescribe any policy for aggregating the network state. For state aggregation, various policies could be followed.

Consequently, topology aggregation involving bandwidth and delay have received considerable attention of the researchers. Various techniques have been proposed to either aggregate bandwidth alone [6], [7], [9], or bandwidth and delay together [8], [10]. Reference [6] evaluates the two commonly used bandwidth aggregation techniques - star and mesh. References [7] and [9] give some rules for assigning a QoS metric to the logical link. References [8] and [10] suggest approaches for simultaneously aggregating *delay* and *bandwidth*.

Though bandwidth and delay are independent parameters, under certain conditions, it is possible to reduce the delay constraint on a path into a bandwidth constraint [5]. Hence developing schemes that accurately aggregate the bandwidth information assumes significance. The current approaches for aggregating bandwidth find the "best path" in some sense, and assign the bandwidth available in the best path as the metric of the logical link[1]. These schemes aggregate just the individual path bandwidth information, and fail to provide a measure of the bandwidth available in a domain as a whole. *The drawback in the current bandwidth aggregation schemes is that they disregard the finite routing capacity of a domain as an aggregate.* Advertising a domain's routing capacity as an aggregate can improve the routing performance, than otherwise.

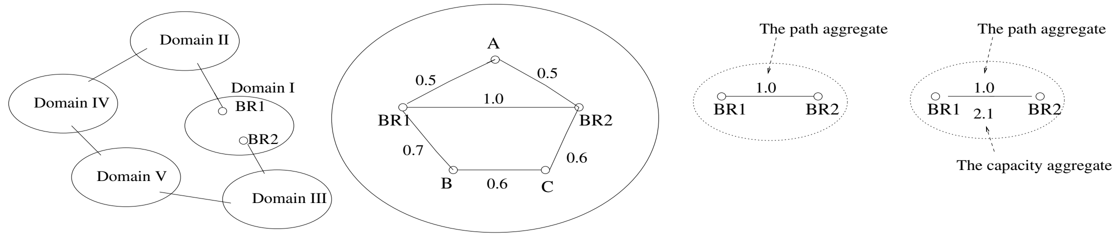
B. Contributions of this work

We had earlier proposed the use of routing capacity as an aggregate metric in [14]. In this paper, we extend our previous work and do the following:

1) We analytically study various alternatives for estimating a domain's routing capacity. *The study shows that the estimation scheme that we propose is more effective than advertising the topological capacity of a network as an aggregate.* We find that, unlike the topological capacity, the routing capacity advertised in our scheme is adaptive, and varies with the load on the network. Hence it gives a better performance.
2) Having identified a better scheme for estimating the routing capacity, we then experimentally compare our proposed capacity-aware aggregation with other existing bandwidth aggregation approaches. *The results demonstrate that, the proposed aggregation scheme increases the bandwidth admitted into the network.* We also find that *the routing process is more robust to the domain updates under the proposed scheme*. Hence for active networkscapacity aware aggregation is more suited than the current aggregation schemes.

The rest of the paper is organized as follows. In section II, we explain the proposed aggregation scheme, followed by the description of the routing algorithm under section III. We compare the effectiveness of different schemes for estimating the routing

[1] We refer to this bandwidth as *path aggregate* in our work



(a) Domain Connectivity (b) Internal structure of domain I (c) Aggregate of domain I advertised to domain II in the current schemes. (d) Aggregate of domain I advertised to domain II in the proposed scheme.

Fig. 1. An illustrative example for motivating capacity-aware aggregation

capacity under section IV. In section V, we compare the performance of the proposed bandwidth aggregation scheme with that of the current aggregation schemes. We finally conclude in section VI.

II. Capacity aware bandwidth aggregation

A. In-adequacy of the path aggregate

The short-comings of the path aggregate can be easily understood through a simple example. The inter-network shown in figure 1(a) has five domains I, II, III, IV and V, with the internal structure of domain I as shown in figure 1(b). The numbers above the links indicate the link capacities in either direction. For simplicity, let us assume that the widest path is treated as the *best* path.

The connectivity aggregate for domain I will have two vertices, one for each border router $BR1$ and $BR2$. These vertices will be connected by a logical link, whose weight will be the bottleneck bandwidth in the widest path between $BR1$ and $BR2$. Accordingly, the aggregate advertised by domain I in the current schemes will be the one shown in figure 1(c). Let domain II generate five connection requests for domain III, each asking for 0.5 units of bandwidth. These requests should be routed through a feasible path to the destination domain. Let the requests be routed along the shortest feasible path, as it results in a better bandwidth utilization [6]. Based on the aggregate advertised by domain I, the shortest feasible path between II and III for these requests is through I. Hence, domain II will forward all the five requests to I. However, the resources available in domain I only allow it to support at most 2.1 units of traffic from II. As a result, the last of the five requests forwarded to domain I will be dropped.

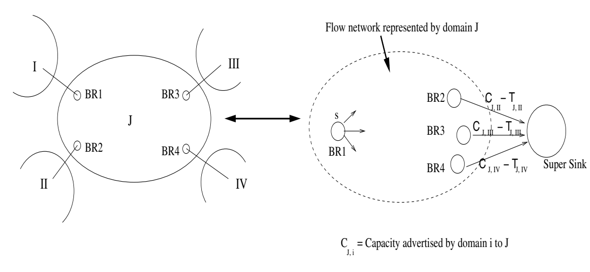
If domain II was made aware of the constraint that domain I can handle only 2.1 units of traffic, it could have routed the fifth connection through alternate paths that are feasible. Thus all the five connection requests might have been successful. The path aggregate advertised under the current schemes does not provide any information about this finite routing capacity, and as a result, the routing performance degrades. Advertising the routing capacity might improve the quality of the routing decisions. The proposed aggregate of domain I, as advertised to II in our scheme is shown in figure 1(d).

B. The proposed aggregation scheme

We assume that the routers within a domain maintain a detailed information about the bandwidth available in all the intra-domain links. In addition, a border router maintains the amount of traffic exchanged between its own domain and the neighboring domains that it connects to. Using this information, a border router periodically constructs and advertises its domain aggregate to other domains. The current schemes for inter-domain QoS routing assume that a domain maintains the aggregate of every other domain. Such a high degree of co-operation is very difficult in real-life, if not impossible. Hence, in this work, we assume that a domain is aware of the aggregate of its immediate neighbors alone. Such a limited co-operation is more practical.

We propose to construct a bandwidth aggregate for a domain having two constituents: (a) the capacity aggregate, which is the routing capacity of the domain, and (b) the path-aggregate, which is the bottle-neck bandwidth in the widest path. Different neighbors can view the resource availability in a domain differently. Hence, we estimate the resource aggregate of a domain with respect to each of the neighbors.

We explain our aggregation methodology with reference to the schematic internetwork shown in figure 2. Domain J is con-



(a) An example Internet (b) Equivalent flow graph representation for routing capacity estimation

Fig. 2. A portion of an internetwork

nected to four neighboring domains I, II, III and IV through the border routers $BR1$, $BR2$, $BR3$, and $BR4$ respectively. Domain J should advertise to each of its neighbors, its routing capacity, and the bottle-neck bandwidth in the widest path. Let us calculate the routing capacity that has to be advertised to domain I. The *routing capacity* $C_{I,J}$ of domain J with respect to a neighbor I, is defined as the maximum amount of traffic that I can route to J, provided the traffic from all other neighbors of J remain unchanged.

The traffic from domain I enters domain J through border router $BR1$. The incoming traffic can leave domain J through any of the other border routers $BR2$, $BR3$ or $BR4$. Hence $BR1$ acts as the source S for the traffic, and the border routers $\{BR2, BR3, BR4\}$ act as the sinks T. $C_{I,J}$ is the sum of two quantities: (a) $T_{I,J}$, the average traffic sent by I through J over the previous advertisement interval, and (b) $\Delta C_{I,J}$, the total free bandwidth available in J at the instant of estimation. The border routers maintain the traffic values, and hence $T_{I,J}$ can be readily calculated. The $\Delta C_{I,J}$ value is estimated by modeling domain J as a flow graph, with $BR1$ acting as the source, and $\{BR2, BR3, BR4\}$ acting as the sinks. The maximum flow value on this flow graph is determined using 'Ford-Fulkerson' algorithm, and is taken to be the $\Delta C_{I,J}$ value.

The path aggregate is the bandwidth of the widest path in the domain, and is estimated through well-known techniques [11]. Domain J thus periodically constructs the capacity and the path aggregates, and advertises to domain I. A similar procedure is adopted by J for advertising its aggregate to II, III, and IV.

Associated Overhead: In the proposed approach, each BR should solve for the maximum-flow in-order to estimate the routing capacity. On a domain with V routers and E links, this takes $O(VE^2)$ time using Ford-Fulkerson algorithm [4]. This time complexity can be reduced by employing other sophisticated algorithms for solving the maximum flow problem.

III. Routing

In our approach, for reasons discussed under section II-B, a domain has the aggregate information of its neighbors alone. As a result, we develop a *distributed* routing strategy called *Restricted Selective Flooding (RSF)*, to find a feasible path from the source domain to the destination domain.

A domain is said to be *feasible* if it has sufficient resources to support a connection requesting b bandwidth units. The feasibility of a neighbor d_j as evaluated by domain d_i is given by an indicator function $I_{j,i}(b)$. Domain d_j is feasible with respect to d_i only if $I_{j,i}(b) = 1$. The indicator function is defined as:

$$I_{i,j}(b) = \begin{cases} 1, & \text{if } T_{i,j} + b \leq C_{i,j} \text{ and } b \leq P_{ij}, i \neq j \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

where C_{ij} is the routing capacity advertised by d_j to d_i. T_{ij} is the amount of traffic flowing from d_i to d_j at the instant of the request arrival, and P_{ij} is the widest path bandwidth advertised by d_j to d_i.

Remark: We note here that satisfying $b \leq P_{ij}$ in equation 1 does not guarantee that $T_{i,j} + b \leq C_{i,j}$ will be satisfied. Referring back to the example scenario discussed in section II-A, we have the following values for the aggregates: $C_{II,I} = 2.1, P_{II,I} = 1.0$. Initially, before the arrival of the five requests, $T_{II,I}$ is zero. Once the first four requests each demanding 0.5 bandwidth units are routed, $C_{II,I}$ and $P_{II,I}$ still remain at the previous values of 2.1 and 1.0 respectively. However, $T_{II,I}$ increases from 0 to 2.0 units. The fifth request demanding a bandwidth of $b = 0.5$ satisfies $b \leq P_{II,I}$, but fails to satisfy $T_{II,I} + b \leq C_{II,I}$, signifying that domain I cannot handle the request.

A. Restricted Selective Flooding (RSF)

The principle of RSF is that, the shortest path to the destination domain in terms of *domain-hops* is followed. If somewhere along the shortest path, a domain not capable of supporting the request is encountered, the search branches off at that point, and other possible paths are searched.

Consider the schematic internet shown in 2(a). Let domain J get a connection request for b bandwidth units to a destination domain d_n that is several domain hops away. Domains I, II, III and IV are the immediate neighbors of J. The request is routed in *RSF* as follows:

1) Domain J chooses to route the request along the shortest domain-hop path to the destination domain d_n. Since the shortest domain-hop path is based only on connectivity, it is easy to obtain the shortest domain-hop path.
2) Let I be the next hop in the shortest domain-hop path. Before forwarding the request to I, domain J checks if $I_{J,I} = 1$. If so, J forwards the request to I.
3) If $I_{J,I} = 0$, then it means that domain I is not feasible, and hence cannot support the new request. Then domain J performs the check for all the other neighbors (but for the domain it received the request from), and forwards the request to all of them that satisfy the above check. If none of the neighbors satisfy the check, the request is dropped.
4) A neighbor, say I of J on receiving the request, would forward the request along its own shortest domain-hop path to the destination domain. Before forwarding, I checks the indicator function's value for its next-hop domainand the procedure mentioned in the above steps get repeated until the destination domain is reached.
5) Since this is a distributed strategy, it is possible that the destination receives multiple copies of the request. The destination acknowledges only for the first copy and the duplicates are rejected.
6) The *ack* reserves resources along the way back to the source.

Each connection request carries along with it the list of domains it has visited so far. Through this mechanism loops are detected, and the requests that go around in loops are discarded.

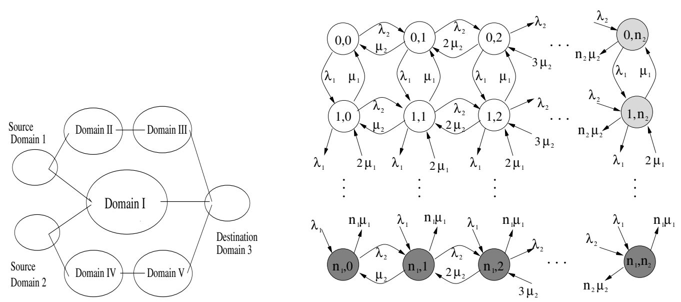
B. Intra-domain Routing

For routing requests within a domain, we use a 'source routing' strategy. The *ingress* router on receiving a request, determines the next-hop domain(s) and the appropriate *egress* router(s) using the *RSF* strategy. The ingress router then finds the *shortest feasible path* [2] to the egress router(s), and routes the request along the chosen path(s). If there is no feasible path inside the domain, the request is dropped.

IV. Proposed Aggregate vs Topological Capacity

The topological capacity is the *total installed capacity* of a domain to handle traffic. The topological capacity does not change with time, and hence it is sufficient, if it is estimated only once. Thus it is an attractive candidate for being considered as the routing capacity of a domain.

In our proposed scheme for estimating the routing capacity, we take into consideration the traffic routed by other neighbors. As the traffic pattern can change with time, we have to periodically estimate the routing capacity, adding more computational burden on the border routers. Hence, one may wonder what difference would it make, if the topological capacity is advertised as the routing capacity, instead of the proposed aggregate. In order to answer this question, we analytically study the performance of the two alternatives. Let the scheme that advertises the topological capacity be referred to as TOP, and the proposed scheme be referred to as PRO. The two schemes are evaluated on the network shown in figure 3(a) using RSF as the routing strategy.



(a) Kite topology used in performance analysis.

(b) Markov Chain representation of the primary path.

Fig. 3. Figures for Performance analysis

In the network considered, domains 1 and 2 are stub domains that generate connection requests to domain 3. Domains I, II, III, IV, and V are the transit domains. Connections from domains 1 and 2 share the resources of domain I. The resources in domains II and III (domains IV and V) are exclusively used by connections from domain 1 (domain 2). While the above topology may not completely model the complex traits of the Internet, the analysis does provide good insight into the results that follow.

A. Assumptions and Notations

We assume that the host domains 1 and 2 generate connection requests as per Poisson processes with rates λ_1 and λ_2 respectively. The holding times of domain 1 and domain 2 calls are assumed to be exponentially distributed with means $1/\mu_1$ and $1/\mu_2$ respectively. Let domain I have the capacity to hold N calls simultaneously. Let domains II and III have identical capacities to hold L_1 calls. Similarly, let domains IV and V be identical and have a capacity for L_2 calls.

Domains 1 and 2 generate unit bandwidth requests that always satisfy the *path aggregate constraint* of domains I, II, III, IV and V. This assumption alleviates the influence of the path aggregate on the performance, and helps us to clearly characterize the behavior of the two routing capacity estimates. Let n_1 and n_2 be the capacity of domain I advertised to domains 1 and 2 respectively. Let m_1 and m_2 be the capacities of domains II and IV as advertised to domain I.

The performance of the two estimates for the routing capacity is studied in terms of the bandwidth rejected in the network in the time interval between two successive domain updates.

B. RSF's behavior on the topology

From figure 3(a), we see that between domains 1 and 3, the shortest path is $1 \to I \to 3$. We call this path as the *primary path*, P. The path is $1 \to II \to III \to 3$, is an alternative to the primary path, and is referred to as the secondary path, S_1. Between the domains 2 and 3, we have the same primary path, while the secondary path S_2 is $2 \to IV \to V \to 3$.

In the above settings, RSF behaves in the following manner. A domain 1 request will be forwarded by default to domain I. If $I_{1,I}(.) = 0$, (it can happen if $T_{1,I} = C_{1,I}$), then the request will be forwarded to domain II. If both $I_{1,I}(.) = 0$ and $I_{1,II}(.) = 0$, the request will be dropped as there is no other path to forward the request. Similarly, a domain 2 request by default will be forwarded to domain I. If $I_{2,I}(.) = 0$, the request will be forwarded to domain IV. If both $I_{2,I}(.) = 0$ and $I_{2,IV}(.) = 0$, the domain 2 request will be dropped.

C. Bandwidth rejected in the primary path

The primary path is a loss system with N resource units to which requests from domains 1 and 2 arrive. Let (r_1, r_2) denote the primary path's state of having r_1 domain 1 calls, and r_2 calls of domain 2 in the primary path. The maximum number of domain i, $i = 1, 2$ calls that can be routed through the primary path is limited by the advertised capacity aggregate. If $\mathbf{n} = (n_1, n_2)$, the state space for (r_1, r_2) can be written as:

$$\Omega_{\mathbf{n}} = \{(r_1, r_2) : r_1 + r_2 \leq N, r_i \leq n_i, i = 1, 2\} \quad (2)$$

Since the arrival processes are Poisson and the call holding times are exponential, the primary path can be modeled as a *two-dimensional Continuous Time Markov Chain (CTMC)*, with each state characterized by the tuple $\mathbf{r} = (r_1, r_2)$. The state diagram for this chain is shown in the figure 3(b). The steady-state distribution $\boldsymbol{\pi}$ of this Markov chain can be obtained using the method outlined in [13], and is given by:

$$\boldsymbol{\pi}(\mathbf{r}) = \frac{1}{G} \frac{\rho_1^{r_1}}{r_1!} \frac{\rho_2^{r_2}}{r_2!}, \ \ \mathbf{r} \in \Omega_{\mathbf{n}} \quad (3)$$

where $G := \sum_{\mathbf{r} \in \Omega_{\mathbf{n}}} \frac{\rho_1^{r_1}}{r_1!} \frac{\rho_2^{r_2}}{r_2!}, \ \ and \ \ \rho_i = \lambda_i/\mu_i.$

A call forwarded by domain i, $i = 1, 2$ to the primary path will be lost inside domain I, when this arriving call sees $N - r_i$ calls of the other domain $j \neq i$, in the primary path. Since domain i call arrivals at the primary path is Poissonian, we can write the bandwidth rejected in the primary path, BR_P as,

$$BR_P = \lambda_1 \sum_{r_1 = N - n_2}^{n_1 - 1} \boldsymbol{\pi}_{\mathbf{n}}(r_1, N - r_1) + \lambda_2 \sum_{r_2 = N - n_1}^{n_2 - 1} \boldsymbol{\pi}_{\mathbf{n}}(N - r_2, r_2) \quad (4)$$

D. Bandwidth rejected in the Secondary paths

An arriving domain i, $i = 1, 2$ call that finds $r_i = n_i$ active calls in the primary path is termed as an *overflow*, and is forwarded to the secondary path S_i. The process controlling these overflows is nothing but the *CTMC* that governs the primary path's state. Hence, the overflows from domains 1 and 2 are *Markov Modulated Poisson Processes (MMPP)* in nature, and their parameters can be obtained from the state transitions given in figure 3(b).

A domain 1 overflow, that sees m_1 active calls in the secondary path S_1, will have an indicator function $I_{1,II}(.)$ value of 0. This is because, the advertised capacity of domain II to domain 1 is m_1. Hence, this overflow will not be forwarded to the secondary path, and will be dropped in domain 1 itself. Similarly, a overflow from domain 2 that sees m_2 active calls in S_2 will be dropped in domain 2. The secondary path S_i can be modeled as a $MMPP/M/m_i/m_i$ system, and the probability of an overflow getting dropped is nothing but the blocking probability of this loss system. Using the approach outlined in [12], the blocking probability p_{S_i} in the secondary path S_i, $i = 1, 2$ can be calculated. The total bandwidth rejected at the two secondary paths is then given by

$$BR_S = p_{S_1}.\hat{\lambda}_1 + p_{S_2}.\hat{\lambda}_2 \quad (5)$$

 0-7803-8794-5/04/$20.00 © 2004 IEEE

where $\hat{\lambda}_i$ is the average rate of call arrivals at the secondary path S_i. The secondary path S_1 receives calls only when $r_1 = n_1$ in the primary path. Hence, $\hat{\lambda}_1 = \lambda_1 \sum_{r_2=0}^{n_2} \boldsymbol{\pi}_{\mathbf{n}}(n_1, r_1)$. Similarly, $\hat{\lambda}_2 = \lambda_2 \sum_{r_1=0}^{n_1} \boldsymbol{\pi}_{\mathbf{n}}(r_1, n_2)$.

E. Bandwidth Rejected under the two schemes

The total bandwidth rejected in the two schemes TOP and PRO is the sum of BR_P and BR_S, and can be calculated once the values n_1, n_2, m_1, and m_2 are determined.

As the installed capacity of domain I is N, under TOP, it advertises a routing capacity value of N to domains 1 and 2. In other words, $n_1 = n_2 = N$. Similarly, II and IV respectively advertise routing capacities of $m_1 = L_1$ and $m_2 = L_2$ to domains 1 and 2. With these values of n_1, n_2, m_1 and m_2, the total bandwidth rejected in network in TOP can be calculated.

In PRO, the value of domain I's routing capacity advertised to neighbor $i = 1, 2$ is the sum of two quantities: (a) the average traffic sent by i through I, and (b) the free bandwidth available in I at the instant of estimation. Since the domains generate connection requests at a constant rate, the average traffic of domain i sent through I will be ρ_i erlangs, provided $\rho_1 + \rho_2 < N$. We will restrict our analysis to the above condition, since $\rho_1 + \rho_2 > N$ corresponds to the unlikely scenario of the network being severely under-provisioned. When $\rho_1 + \rho_2 < N$, the average free bandwidth available in the network at the instant of estimation, will be $N - (\rho_1 + \rho_2)$. Hence the routing capacity advertised to domain 1 will be $n_1 = \rho_1 + N - (\rho_1 + \rho_2) = N - \rho_2$. Similarly, the routing capacity advertised to domain 2 will be $n_2 = N - \rho_1$. Since domains II and IV are exclusively used by the connections of 1 and 2, it is straightforward to see that the values of m_1 and m_2 will be respectively L_1 and L_2.

F. Numerical Results

We observed the bandwidth rejected in the network under TOP and PRO by maintaining the total system load $\rho = \rho_1 + \rho_2$ at a constant value, and varying the individual domain loads. We then repeated the procedure for different system load (ρ) values. The idea behind this experiment is to study how the total network load, and the load distribution among the individual domains affect the bandwidth rejected under the two schemes.

Figure 4 shows the performance of the two schemes. The results presented here are for two system loads $\rho = 12$, and $\rho = 18$. The values of N and L used in the experiment are respectively 20 and 15. From the graph, we can infer the following:

1) At light loads ($\rho = 12$), there is not much difference in the performance of TOP and PRO.
2) At high loads ($\rho = 18$), we see that the proposed aggregation scheme PRO gives a better performance than the topological capacity TOP.
3) For a given total system load value ρ, the performance of both the schemes is almost in-sensitive to the load distribution among the individual domains.

The above observations can be explained as follows: In TOP, domain I advertises a routing capacity of N to the source domains 1 and 2. Hence, each source domain thinks that the entire capacity N is available exclusively for its traffic. A domain is not made aware of the fact that this N is being utilized by the other domain's traffic too. As a result, there is contention for bandwidth along the primary path, and requests get dropped. In PRO, the presence of other domain's traffic is indicated to i by advertising a lower routing capacity $N - \rho_j,\ j \neq i$. resulting in fewer contentions along the primary path. The performance difference between PRO and TOP is more pronounced at high system loads, when more contentions occur.

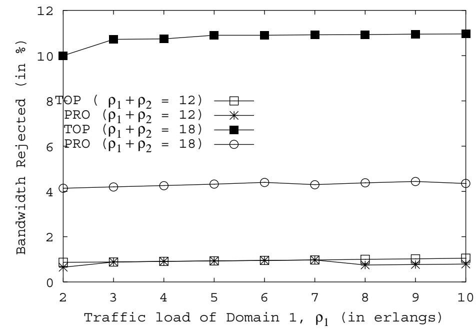


Fig. 4. Performance of TOP and PRO with respect to system loads. Other parameters used are $N = 20,\ L = 15$.

Under both the schemes, the number of contentions that occur in the network depends on the total load alone. The load distribution among the individual domains does not alter the contentions significantly. Hence for a given network load, we do not observe noticeable changes in the bandwidth rejected, when the individual domain loads are varied .

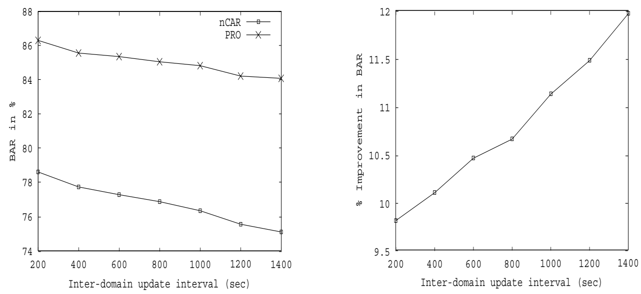
Thus we see that while TOP may be a simpler scheme to implement, it does not give a good performance like PRO.

V. Experimental Results

Having identified that PRO is a better scheme than TOP for estimating the routing capacity, we now experimentally compare PRO with other aggregation schemes. Current bandwidth aggregation schemes are not routing capacity aware, and they advertise just the path aggregate alone. Henceforth, we shall refer to such schemes as $nCAR$ in our discussions. PRO and $nCAR$ are studied using RSF as the routing strategy. Their performance is observed on *generic* network topologies obtained through the GT-ITM package [3]. The simulations were done using OPNET, a commercial network simulation software. The schemes are compared based on the bandwidth admission ratio (BAR) achieved.

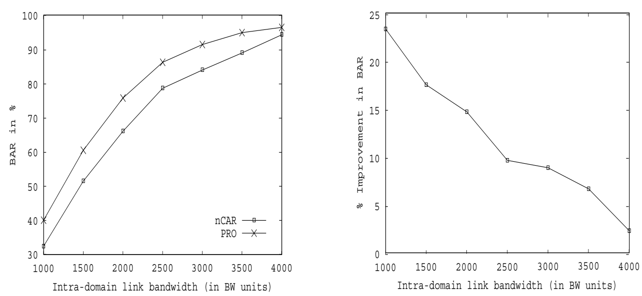
The results reported in this paper are from a 16 domain internetwork with roughly 300 routers. The network had three classes of traffic, with each class requesting 20, 30 and 40 units of bandwidth respectively. The connection requests arrive as per a Poisson process, and the call durations are drawn from an exponential distribution. The results shown are the average of ten simulation runs.

1) Sensitivity to Domain Updates: In inter-domain QoS routing, it is important that the advertised aggregate accurately reflects the resource availability in the domain over extended periods of time. If not, the routing performance will deteriorate under long inter-domain update intervals. The graphs in figure 5 show the BAR of the PRO and $nCAR$ schemes with respect to varying inter-domain update intervals.



(a) BAR of PRO and $nCAR$ (b) Percentage improvement in BAR of PRO over $nCAR$

Fig. 5. Performance of PRO and $nCAR$ with respect to inter-domain update intervals. Parameters used are: Inter-domain link BW = 6000 units, Intra-domain link BW = 1500 units, Arrival rate of all classes = 1/6 sec^{-1}, Mean holding time of all classes= 200 sec.



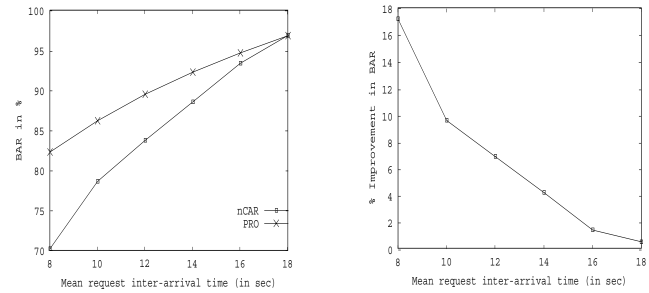
(a) BAR of PRO and $nCAR$ (b) Percentage improvement in BAR of PRO over $nCAR$

Fig. 6. Performance of PRO and $nCAR$ with respect to network bandwidth. Parameters used are: Inter-domain update interval = 200 sec, Arrival rate of all classes = 1/6 sec^{-1}, Mean holding time of all classes= 200 sec.

We note that PRO has higher BAR values than $nCAR$ for all update intervals. This proves that *the advertised capacity aggregate helps a domain to have a better estimate of the resource availability in its neighbors.* Figure 5(b) shows the relative improvement in BAR achieved by PRO over $nCAR$. The relative improvement increases as the routing updates become less frequent. This implies that the *routing process is more robust to the frequency of the domain updates under capacity aware aggregation, than under non-capacity aware schemes*. Thus PRO is more suited for active networks than $nCAR$ schemes.

2) Sensitivity to Network resources and Traffic pattern: Figures 6(a) and 6(b) show the performance achieved by PRO over $nCAR$ for varying intra-domain link bandwidths. During this experiment, the ratio of the inter-domain link bandwidth to the intra-domain link bandwidth was kept constant at 4. From figure 6(b), we see that the performance difference between the capacity and non-capacity aware schemes in more pronounced when there are not enough resources in the network. As the resource availability becomes higher and higher, gains achieved by the capacity aware aggregation schemes with respect to their non-capacity aware counterparts go down. This behavior is due to the fact that, in the presence of more resources, the quality of aggregation does not greatly affect the routing performance.

Figures 7(a), and 7(b) show the performance of the PRO and $nCAR$ for varying request arrival rates. We see that, as the arrival rate increases, the relative gain achieved by PRO over $nCAR$ increases. This is because, with increased arrival rate, bandwidth available inside a domain changes frequently. The capacity aggregate helps a domain to keep track of these changes in its neighbors to a reasonable extent. Such a functionality is not possible with the path aggregate.



(a) BAR of PRO and $nCAR$ (b) Percentage improvement in BAR of PRO over $nCAR$.

Fig. 7. Performance of PRO and $nCAR$ with respect to traffic pattern. Parameters used are: Inter-domain link BW = 6000 units, Intra-domain link BW = 1500 units, Inter-domain update interval = 100 sec, Mean holding time of all classes= $200sec$.

VI. CONCLUSIONS

In this paper we have studied the performance of a new capacity aware bandwidth aggregation scheme. The proposed scheme captures the finite capacity of a domain to route traffic, and hence is more accurate than the existing approaches. Analytical and experimental results show that routing using our capacity aware scheme is more robust to inter-domain updates, and has a better performance for a wide range of traffic conditions. Also, unlike the topological capacity, the routing capacity advertised in our scheme is adaptive and varies with network conditions. As a result, it gives a better performance than topological capacity, and hence is a better aggregate.

REFERENCES

[1] Private network-network interface specification. version 1.0. *ATM Forum Technical report af-pnni-0055.000*, March 1996.

[2] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi. Qos based routing: a performance perspective. In *ACM SIGCOMM '98*, 1998.

[3] K. Calvert, M. Doar, and E. Zegura. Modeling internet topology. *IEEE Communications Magazine*, June 1997.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, 1990.

[5] R. Guerin and A. Orda. QoS routing in networks with Inaccurate information: theory and algorithms. In *INFOCOM'97*, 1997.

[6] F. Hao and E. Zegura. On Scalable QoS Routing: Performance Evaluation of Topology Aggregation. In *INFOCOM 2000 Proceedings*, 2000.

[7] A. Iwata, H. Suzuki, R. Izmailow, and B. Sengupta. QoS aggregation algorithms in hierarchical ATM networks. In *IEEE Proceedings of the ICC'98*, 1998.

[8] T. Korkmaz and M. Krunz. Source-oriented topology aggregation with multiple QoS parameters in hierarchical ATM networks. *ACM Transactions on Modeling and Computer Simulation*, 19(2):384–395, November 2000.

[9] W. Lee. Topology aggregation for hierarchical routing in ATM networks. In *ACM SIGCOMM'95 Proceedings*, 1995.

[10] K.-S. Lui and K. Nahrstedt. Routing With Topology Aggregation in Delay-Bandwidth Sensitive Networks. *IEEE/ACM Transactions on Networking*, 12(1):17–29, February 2004.

[11] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. In *ICNP '96, Atlanta*, 1996.

[12] K. S. Meier-Hellstern. The analysis of queue arising in overflow models. *IEEE Transactions on Communications*, 37(4):367–372, April 1989.

[13] K. Ross. *Multiservice Loss models for Broadband Telecommunication Networks*. Springer-Verlag, NY, 1995.

[14] V. Sarangan and R. Acharya. A study on using network flows in hierarchical QoS routing. In *IEEE Globecom 2001, San Antonio*, 2001.