VPN Topology Abstraction Service using Centralized Core Capacity Sharing Scheme

Ravishankar Ravindran[1] *Member, IEEE*, Changcheng Huang[1] *Senior Member, IEEE,* and
Krishnaiya Thulasiraman[2] *Fellow, IEEE*
(1. Carleton University, Ottawa, 2. University of Oklahoma, Norman)

Abstract—*This paper deals with the problem of sharing VPN service provider's core topology and link state information with its VPNs. This is provided as a service using topology abstraction principles. The topology abstractions are generated considering factors such as VPN SLA requirements, fairness and maximizing core network utilization. The paper deals with algorithms applicable to generate topology abstraction for the VPNs in a centralized manner. In this context we define a problem called the VPN core capacity sharing problem. We study this problem using algorithms based on multi-commodity flow theory. We observe from simulation analysis over several topologies that applying the maximum multi-commodity flow based partitioning scheme improves the call performance and network utilization statistics compared to a previously proposed topology abstraction scheme based on maximum concurrent flow theory.*

I. Introduction

VPN service providers (VSP) and IP-VPN customers have traditionally maintained service demarcation boundaries between their routing and signaling entities, and this has resulted in the VPNs viewing the VSP network as an opaque entity and therefore limiting any meaningful interaction between the VSP and the VPNs. In [1] we addressed this issue by enabling a VSP to share its core topology information with the VPNs as a service in a manner which is both practical and scalable. We used the notion of topology abstraction (TA), which serves as a means of sharing the core topology information as abstract graphs associated with QoS metric information. Our research deals with the abstraction of the available capacity associated with the links of the VSP's core network. The VSP generates the TA for each VPN based on TA SLA parameter set TS_k which has two key elements: *abstract topology type metric* (T_k) and *abstract topology refresh interval* (R_k). The T_k parameter allows a VPN to choose from one of either fully meshed, source-star, star or simple node abstract topologies which we analyzed in [1]. The R_k parameter determines the correctness of the abstract topology information at any point of time when compared to the state of the core network. Both these parameters are negotiated between the VPN and the VSP based on factors such as call performance expected from such a service and the criticality of the applications for which the TA service will be used. In [1] we evaluated this novel idea of providing TA service to VPNs in the context of an architecture based on the BGP/MPLS based IP-VPN solution [4].

The TA for the VPNs can be generated either in a decentralized or a centralized manner. In the decentralized approach the PE nodes serving the set of VPNs generate abstractions independent of one another. In a centralized case, a central manager first generates the partition subgraph for each VPN from which the abstract topology is derived.

In [2] we proposed three topology abstraction schemes applicable in a decentralized mode of TA generation. These abstraction schemes differ from one another with respect to degrees of aggressiveness in terms of exposing resources, which result in tradeoffs in terms of VPN call performance metrics and core network utilization. In [3] we studied the problem of generating TA in a centralized manner. We proposed an abstraction scheme based on maximum concurrent flow (MConF) theory, which used the approximation algorithm from [5] for online implementation.

In this paper we extend this study of centralized mode of TA generation by applying the maximum multi-commodity flow (MMCF) theory. We demonstrate the improvements achieved by this scheme over MConF based TA scheme using simulation analysis of a IP-VPN scenario implementing the TA service.

II. Graph Theory Notations

Graph Notations	Definition
$G(V,E)$	VSP core network
B	Set of all border (PE) nodes of the core network
C_k, P_k	Set of CE/PE nodes of VPN k, here $P_k \subseteq B$
$C_{k,b}$	Set of VPNs hosted by border node b
$G_{k,l}(V_k,E_k)$	Abstract topology of VPN k of topology type l
TS_k	TA SLA parameter set for VPN k
T_k / R_k	Abstract topology type subscription/Refresh interval of TA for VPN k
$S(V_k,E_k)$	Partition subgraph for VPN k
$t_{i,j}$ (or $t(e)$)	Total capacity of edge (i,j) or e
$r_{i,j}$ (or $r(e)$)	Residual capacity of edge (i,j) or e
U	Set of all VPNs provided TA service
U_b	Set of VPNs hosted on node b
$Z(x,y)$	Set of VPNs common to border nodes x and y
K	Set of all commodities in a multicommodity formulation
K_v	Set of all source-destination border node pair commodity of VPN v
$f_{mc}(k)$	Flow of commodity k corresponding to multicommodity flow (MMCF) formulation
$x_{i,j}^{v,k}$	Resource partitioned on edge (i,j) for VPN v for source-destination commodity k
$D(s_k, d_k)$	Demand corresponding to source-destination commodity k

III. VPN Capacity Sharing Problem

In a centralized mode of TA generation, the abstract topology for each VPN k is derived by the central manager from an intermediate subgraph $S(V_k,E_k)$ called as the partition subgraph. The abstract topology $G_{k,l}(V_k,E_k)$ is then derived from this partition subgraph. Here, we assume that the VSP adopts a model where the goal is to always expose the available capacity equally among all the VPNs that have subscribed to the same abstract topology type parameter T_k. This fairness policy guarantees that two VPNs sharing a pair of border nodes *x* and *y* and provided with the same TA type will have the same amount of bandwidth associated with the virtual link (*x, y*) in their respective abstract topologies.

A. Objectives of VPN Capacity Sharing

First, the goal of TA service is to enable better call performance of the VPNs making capacity requests. We use the *VPN success ratio* to measure VPN call performance. This parameter is a measure of making right bandwidth request decisions by a VPN using the abstraction provided to it by the VSP. The second objective is to ensure the best possible use of the VSP's available core capacity resource so that the VSP's network utilization is maximized. The third criterion we consider while computing subgraphs for the VPNs is to ensure that the schemes are fair to all the VPNs with respect to sharing the resources, in line with the fairness policy discussed earlier.

B. Definition of VPN Capacity Sharing Problem

Considering these objectives we define the *VPN capacity sharing problem* as follows:

VPN-Capacity Sharing Problem (VPN-CS):

Given a graph G(V,E) representing the core network of the VSP providing topology abstraction service to the set of VPNs, U, the objective is to compute fair partitions S(V_k,E_k) for each VPN k∈U of the network so as to maximize the VPN success ratio and core network utilization. If x_e^k *represents the resource identified as part of subgraph S(V_k,E_k) on edge e∈E and for VPN k∈U, then* $\sum_{k\in U} x_e^k \le t(e)$, *∀e∈E. Recall that t(e) is the total capacity of the link e.*

Using the relationship between the maximization objectives of the VPN-CS problem and their relation to total abstracted capacity, which is the sum of the link capacities in the partition graphs S(V_k,E_k) for k∈U, we formulate the VPN-CS problem using multi-commodity flow theory.

In [3] we studied this problem using the maximum concurrent flow (MConF) approach. The objective of the MConF formulation is to maximize the factor *β* such that there exists a flow that satisfies the demand *β*D*(s_i, d_i), ∀ *i*= 1, 2,…*k*. Here (s_i, d_i) represents a source-destination pair and the flow between s_i and d_i is considered as a commodity. We proposed a solution for the VPN-CS problem applying the approximation algorithm from [5]. For the MConF based partitioning heuristic, *D*(s_i ,d_i) is set to the maximum flow between the source-destination pair.

There is a drawback in applying the MConF based approach to solve the VPN-CS problem. The throughput achieved by the MConF formulation with the values of the commodities set to their maximum flow values can turn out to be conservative in some cases. This happens because the edges of the paths corresponding to a commodity's maximum flow are also shared by other commodities; this forces the same fraction of the demand to be satisfied for all the commodities resulting in conservative topology abstractions. This inefficiency can be addressed by maximizing the sum of the independent commodity flows. Hence we propose the use of maximum multi-commodity flow formulation for the VPN-CS problem.

C. Maximum Multi-Commodity Flow (MMCF)*Approach*

The node-link MMCF formulation is as follows:

$$Maximize \sum_{k\in K} f_{mc}(k)$$

Subject To:

$$\sum_{(i,j)\in E} x_{i,j}^k - \sum_{(j,i)\in E} x_{j,i}^k = f_{mc}(k) \quad \forall k \in K, i = s_k \qquad (1)$$

$$\sum_{(i,j)\in E} x_{i,j}^k - \sum_{(j,i)\in E} x_{j,i}^k = 0 \ \forall k \in K, i \ne s_k, i \ne d_k \qquad (2)$$

$$\sum_{(i,j)\in E} x_{i,j}^k - \sum_{(j,i)\in E} x_{j,i}^k = -f_{mc}(k) \quad \forall k \in K, i = d_k \qquad (3)$$

$$\sum_{k\in K} x_{i,j}^k \le t_{i,j} \quad \forall (i,j)\in E \qquad (4)$$

$$x_{i,j}^k \ge 0 \quad f_{mc}(k) \ge 0$$

We next propose the MMCF based subgraph partitioning scheme.

D.MMCF Based Partitioning Scheme for VPN-CS Problem

In order to reduce the complexity of the execution time and memory requirement of solving the multi-commodity flow problem, we apply the MMCF formulation on an aggregated set of source-destination commodity pairs. For this, we define a commodity as a source-destination border node pair (*s,d*) that has a potential of a VPN bandwidth request from *s* to *d*. This is the case if the source-destination border nodes *s* and *d* host at least one common VPN; in this case, it will satisfy the criteria |(U_s ∩U_d)|≥ 1. We define the set of VPNs with a common (*s,d*) pair as *Z*(*s*,*d*). This aggregation reduces the commodity complexity to *O*(|*B*|^2). Once the resource allocation for the commodities is achieved by solving the MMCF problem, we then fairly distribute the resources corresponding to the aggregate commodity among the VPNs in the set *Z*(*s*,*d*), from which the partition subgraph is generated. Based on this discussion, we next propose the MMCF based centralized heuristic to address the VPN-CS problem.

Maximum Multi-commodity Flow Based Partitioning Scheme

Input: *G*(*V,E*), *B*, *U*, U_b ∀*b*∈*B*

Output: *S*(V_v, E_v), $G_{v,l}$(V_v,E_v) ∀*v* ∈*U*

***Step 1*:** For each potential border node pair (*s,d*) , identify the set *Z*(*s,d*) . Define commodity set *K* ={(*s, d*): |*Z*(*s,d*)|>0 };

***Step 2*:** For the commodity set *K*, solve the maximum multi-commodity flow problem. The solution $x_{i,j}^k$ represents the resource associated with commodity *k* over edge (*i,j*) ∈*E*;

***Step 3*:**For each commodity pair (s_k,d_k)∈*K* , we divide the edge flows fairly among the VPNs in the set *Z*(s_k,d_k) as follows,. We set $x_{i,j}^{v,k}$ = $x_{i,j}^k$ /|*Z*(s_k,d_k)| ∀ (*i,j*) ∈*E* and $x_{i,j}^k$ >0;

(Here $x_{i,j}^{v,k}$ is the virtual capacity assigned to VPN *v* on link (*i,j*) for VPN source-destination pair $(s_{k,,}d_k) \in K_v$.)

***Step 4*:** For all VPN $v \in U$, create subgraph $S(V_v, E_v)$, where $V_v \subset V$ and $E_v \subset E$ is the set of all links such that $x_{i,j}^{v,k} > 0$ for at least one commodity $k \in K_v$ of VPN *v*. The capacity of each link (*i,j*) in $S(V_v, E_v)$ is the sum of the flows $x_{i,j}^{v,k}$ $\forall\, k \in K_v$;

***Step 5*:** For each VPN *v*, distribute $S(V_v, E_v)$ to P_v ;

***Step 6*:** Nodes P_v uses $S(V_v, E_v)$ to generate $G_{v,l}(V_v, E_v)$ for each VPN .

IV. Simulation Study

Here we compare the performance of the proposed MMCF based partitioning scheme with the MConF based partitioning scheme from [3] using simulation analysis. For online implementation, [5] proposed the fastest ε-approximation algorithm for the MMCF problem which we apply in Step 2 of the MMCF based scheme. For analysis, we set (1-ε) value to 0.3; this achieves FPTAS execution time of less than 1s.

Performance Metrics:

We compare the performance using the following metrics.

Success Ratio:

$$Success\ Ratio = \frac{Number\ of\ calls\ correctly\ accepted + Number\ of\ calls\ correctly\ rejected}{Total\ number\ of\ calls}$$

Network Utilization:

$$Average\ Network\ Utilization = \frac{Aggregate\ utilized\ link\ capacity}{Aggregate\ link\ capacity}$$

Abstraction Efficiency:

$$Abstraction\ Efficiency = \frac{Aggregate\ com\mathrm{mod}ity\ flow\ over\ all\ links}{Aggregate\ capacity\ of\ all\ links}$$

Simulation Setup:

The simulation analysis was conducted using OPNET on a random network topology and two European networks. Because of the correlation of performance characteristics for the three topologies, we report results only for the random network topology. The simulation models a VSP providing TA service over a BGP/MPLS based L3 VPN solution [4]. In addition, a central manager that executes the partitioning solutions proposed for the VPN-CS problem was also implemented.

For the sake of analysis, all the VPNs are assumed to subscribe to the source-star abstraction [1]. The mean call inter-arrival time and the mean holding time are assumed to be exponentially distributed, with capacity request being uniformly distributed between [1-500] units. The load offered to the network is varied by varying the mean call holding times in the range of [10, 1000]s keeping the mean inter-arrival time constant at 100s. The constant abstract topology update interval R_k is also initialized to the same value as the mean inter-arrival time of the call requests. The simulation was set to collect statistics for 30 independent replications, with each replication generating over 3000 calls to achieve an absolute error of less than 1% and C.I of 95%. Due to space limitation only two graphs containing simulation results are presented in Fig.1(a) and (b). More detailed analyses are in [6].

Observations based on Fig. 1(a) and (b):

Here we briefly summarize the improvement achieved by employing MMCF based partitioning scheme.

The MMCF based partitioning algorithm performed on an average 5% better than the MConF based approach with respect to the success ratio metric. This gain in performance is due to gain in abstraction efficiency. We observed that MMCF's performance in terms of abstraction efficiency was on an average 20% better than the MConF based scheme. The better call performance and abstraction efficiency of the MMCF based approach also resulted in better network utilization in the range of 5% for various load conditions over the MConF based approach.

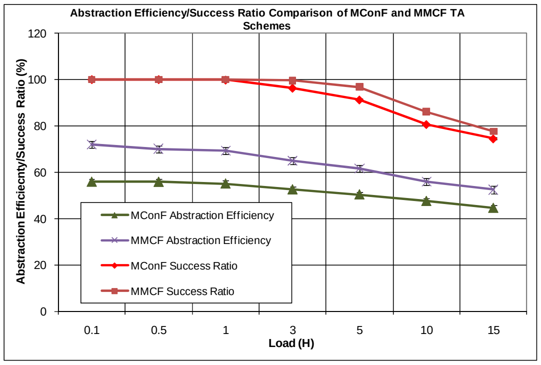

Fig. 1(a): Comparison of Abstraction Efficiency and Success Ratio

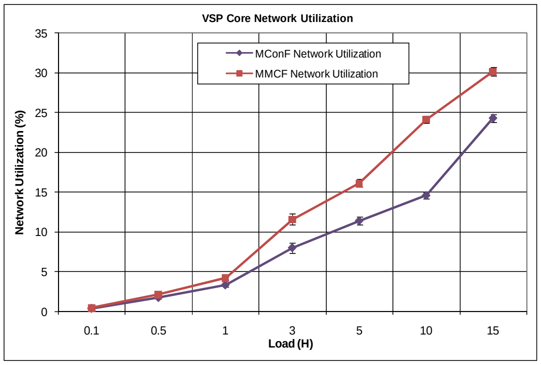

Fig. 1(b): Comparison of Network Utilization

V.Conclusions

In summary, the study in this paper demonstrated how MMCF based partitioning heuristic can be used to improve the call performance and network utilization when TA is generated in a centralized manner for TA service. This is compared to the case when MConF based partitioning scheme is applied.

References

[1] Ravi Ravindran, Changcheng Huang, and K.Thulasiraman, "Topology Aggregation as a VPN Service", Proc. of IEEE ICC, pp. 105-109, 2005.

[2] Ravi Ravindran, Changcheng Huang, and K.Thulasiraman, "A Dynamic Managed VPN Service: Architecture and Algorithms", Proc. of IEEE ICC, pp. 664 – 669, 2006.

[3] Ravi Ravindran, Changcheng Huang, and K.Thulasiraman, "A Dynamic Managed VPN Service: Capacity Sharing Based On Maximum Concurrent Flow Theory", Proc. of IEEE ICC, Vol. 2, pp.211 – 216, 2007.

[4] E. Rosen and Y. Rekter, "BGP/MPLS Virtual Private Networks", RFC 4364, IETF, 2006.

[5] Lisa K. Fleischer, "Approximating Fractional Multicommodity Flow Independent of the Number of Commodities", Proc. of IEEE, Annual Symposium of Foundations of Computer Science, pp. 24-31,1999.

[6] Ravi Ravindran, "Topology Abstraction Service for IP Virtual Private Networks", Ph.D. Thesis, Carleton University, April, 2009.