

Topology Abstraction As VPN Service

Ravi S. Ravindran¹, Changcheng Huang¹, K.Thulasiraman²

(1. Carleton University, Ottawa, 2. University of Oklahoma, Norman)

Abstract—VPN as a managed service enables the service provider to offer more demanding and revenue generating services. Some of the common managed VPN services known today include auto discovery, security, and also a potential ability to perform on demand signaling. In this paper we try to tackle an important problem of service providers namely topology dissemination to VPN customers seeking dynamic bandwidth service. Topology dissemination can easily be translated into a VPN service. In this regard we define this service and its associated SLA. Further we elaborate on four basic QoS enabled topology abstraction services: fully meshed, source rooted star, star, and simple node. The paper also describes a generic framework to generate these abstractions and presents simulation results comparing performance of each of these schemes.

Keywords: VPN, Topology Abstraction

I. INTRODUCTION

Topology aggregation is a problem well studied in the routing literature. This has been used mostly in the context of routing protocols for scalability reasons. One of the notable protocols that have employed topology abstraction successfully is PNNI [1]. For ease of analysis, the problem of topology abstraction can be broken down into two sub-problems. One is the type of abstraction that results in a desired abstract topology. Well-known abstract topologies in the context of PNNI are: fully meshed, star, and star with bypasses. The other sub-problem is the association of one or more link metric information to the abstracted topology. The objective of topology abstraction is to provide compact topologies while minimizing the error occurring due to aggregation of the desired link metric information. The problem of link metric abstraction can be classified into two categories: single metric abstraction and multiple metrics abstraction. Approaches that have tried to address the single metric abstraction problem, which has been shown to be fairly easy to solve, include algorithms proposed in [2] [3]. The non-triviality of the abstraction of the multiple metrics information arises due to the fact that it's difficult to capture the region of feasibility with a single set of coordinates in the multi-dimensional metric plane. Heuristics to abstract multiple paths and link metrics have been proposed in [4]. In [5][6] the abstraction problem for bandwidth and delay as link metrics has been solved. In [9] we introduced the idea of abstraction as applicable to switched networks using GMPLS as the control plane. In this paper we extend the same idea more formally and try defining it as a potential service offered by a VPN service provider (VSP). This, if realized, could be offered as an add-on service to well-defined regular VPN services like auto-discovery and security as proposed in [10][11].

The rest of the paper is organized as follows. In Section 2 we go into the details of what one means by topology abstraction as managed VPN service and the objectives we intend to achieve in this study. Section 3 discusses in detail topology abstraction schemes with their implication as a VPN service. Section 4 gives a brief introduction of realization to such a service. Section 5 discusses simulation results and performance analysis of a service provider scenario providing topology abstraction service to its VPN customers. For our discussions in this paper we follow the managed VPN reference model as discussed in [10].

II. VPN TOPOLOGY ABSTRACTION SERVICE

Creating virtual topologies that map virtual links to underlying physical paths through the VSP network is a well-studied problem in the literature, particularly in the context of overlay L2/L1 networks. This kind of provisioning is required because of the high probability of resource availability and redundancy associated with the VPN service layer agreements (SLA). In this study we explore a new type of VPN service. Here we assume the presence of a set of VPN customers for which the VSP does not pre-provision any resource before hand, but the provisioning takes place on demand. In the discussions below, we denote such a set of VPN customers requiring such a service as VPN_{abs} clients. VPN service of this kind, if offered by the VSP, would co-exist with the traditional guaranteed VPN service provided by the VSP. One way to realize such a service is for the VSP to adopt a flat model, and treat all VPN_{abs} customers alike and share the unprovisioned pool of resource by processing the bandwidth demands in the order received by the VSP. Intuitively, we note that in this kind of provisioning the probability of the call blocking performance on a longer time scale would be asymptotically the same across all the VPNs. Flat model would mean flat revenue model. A way to provide variable revenue scheme is to provide some choice to the VPN customer. One of the key reasons affecting the bandwidth on demand performance is the lack of any VSP topology information of the VSP core in the VPN clients domain which could have been used for making any routing decision. As one can imagine the best call blocking performance is achieved when the VPN_{abs} customer could be provided with all the topology information of the VSP, which we understand is an impractical proposition. Hence the key focus of this paper is to answer the question of exposing the core VSP topology to the VPN customer. We pursue this with the intuition that the VSP could define sets of topology abstraction of various granularities, where a better granular abstraction should result in better routing decision at the VPN customers end. We shall extend this idea to enable the VSP to develop a service model that could be exploited by the VSP to provide topology abstraction as a differentiated service to its

VPN_{abs} customers. An abstract topology is a topology comprising of virtual nodes and virtual links. As mentioned before a virtual link in our context does not map to any physical resources across the nodes or links in the VSP's network. The quality of abstraction and its management is part of the Service Layer Specification (SLS) agreed between the VSP and the VPN customer, which we discuss further in the next section. The relevance of VPN topology abstraction problem can also be seen in light of VPN routing models existing today. The VPN overlay model hides the internal network resulting in two completely different routing domains with no interaction between them at all. On the other hand, the VPN peering model, though not practically implemented, requires a full routing adjacency between the client edge router and VSP's Provider Edge (PE) node. The latter model seems to be the ideal one with respect to routing and signaling efficiency of the client nodes. But in reality the core provider usually will not wish to divulge its topology for reasons of trust. In fact the overlay model can be considered a special case of the peering model, achieved by disabling routing between the VSP edge nodes (PE) and the Client Edge (CE) nodes. The abstraction schemes proposed in this paper also tries to bridge the gap between these two extreme kinds of VPN routing models.

III. VPN ABSTRACTION SCHEMES

In the context of routing, the goal of abstraction is to reduce the routing overhead and to enable other domains to find a path through it to a remote destination. As a VPN service we use it to provide a scaled version of the topology information associated with QoS metrics to enable the VPN client node to make conscious routing decisions. The virtual topology information thus disseminated will be used by the client node to verify the feasibility of its QoS requirements, before sending the path request to the PE node. In a general sense one can state that a more granular topology would mean better probability of making a right routing decision.

To be treated as a VPN service, some of the basic variables we perceive to be part of the SLA are:

1. *Abstraction type*: This represents the abstract view of the core topology as seen by the VPN client. In theory though one could generate any arbitrary topology, but it is easier to generate topologies mapping them to the border nodes in a certain fashion. One key optimization objective for any form of abstraction is the complexity with respect to the topology updates which is proportional to the order of virtual nodes and links. Sparser virtual topologies could be generated using virtual nodes in the final abstraction along with the relevant border nodes.

2. *Abstraction link metrics*: These metrics is associated with the virtual links of the abstract topology. Abstracting single metric has been shown to be solvable problem. It is more challenging to abstract multiple metrics where the objective is to minimize the exposure of the infeasible region in the multi-dimensional metric plane.

3. *Abstraction probability metric*: This parameter relates to the correctness of the abstracted topology at any given point of time. This SLA parameter ensures the statistical guarantee associated with any form of abstraction. The VSP could adopt

several schemes to provide such a guarantee. One way to guarantee this is to define the probability as a function of the abstract topology update interval. This update interval defines the interval between any two successive abstract topology updates to the VPN client.

Other than the basic abstraction SLA parameters mentioned above, a real implementation would involve many other parameters. In this paper we explore four abstraction types: fully meshed, source routed, star and simple node abstraction schemes. For the link metric, we restrict ourselves to the bandwidth metric, though as mentioned above one could extend this to any number of link or path metrics. In our study to achieve a certain level of abstraction correctness, we use abstract topology update interval as a variable parameter.

Following graph theoretical notations are used in our discussion: $G(V,E)$ represents the core of the VSP, B represents the set of border PE nodes of the core network and C represents all the CE nodes served by the VSP. Each link $e_{ij} \in E$ is associated with bandwidth bw_{ij} . For a VPN_{abs} customer i subscribing to abstraction service $VG_{abs,i}(V_i, E_i)$ represents the VPN virtual topology of VPN_{abs} i . Here V_i includes the set of VPN CE nodes where $CE_i \subset C$, and the set of virtual PE nodes i.e. PE_i , hence $PE_i \subset B$ and $V_i = (PE_i \cup CE_i)$. The set of nodes $v \in PE_i$ are those nodes $u \in B$ which support at least one instance of the VPN instance i . A virtual link $e \in E_i$ is associated with abstract link bandwidth metric $bw_{abs(i,j)}$. This bandwidth represents the maximum available unreserved resource between PE border nodes i and j . This definition can also be generalized to multiple virtual link metrics. To construct the virtual topology, the PE node derives the virtual links from the fully meshed virtual graph, associating maximum bandwidth values from shortest-widest path tree from itself to the other nodes $u \in PE$, which then can be stored in a special abstract topology database. To propagate a particular kind of abstraction for a particular VPN_{abs} i subscribing to the abstraction service, the PE nodes utilize the pre-computed tree to first prune those node and links that do not belong to the concerned VPN. It then chooses only those virtual links from the fully meshed virtual graph $VG_i(V_i, E_i)$, such that $\forall e_i(u,v) \in E_i$, with $u,v \in PE_i$, i.e. nodes u and v support the same VPN instance. By doing this we derive a fully meshed sub-graph comprising of only those PE nodes belonging to the same VPN instance. Transformations are later applied on this fully meshed abstraction to generate the desired abstract topology as agreed between the VSP and the VPN customer.

Applying the above formulation we evaluate four different types of VPN abstraction services: a) Fully Meshed Abstraction, b) Source Rooted Abstraction, c) Star Topology Abstraction and d) Simple Node. Though we restrict our discussion to these topologies, nothing prevents the VSP from adapting arbitrary abstract topologies of interest. In the following we elaborate on different cases of abstractions referring to Fig 3.1, which shows a VPN service provider (VSP) providing VPN abstraction service to two different VPN clients, VPN-A and VPN-B. N1-N6 refers to the PE nodes of the VSP.

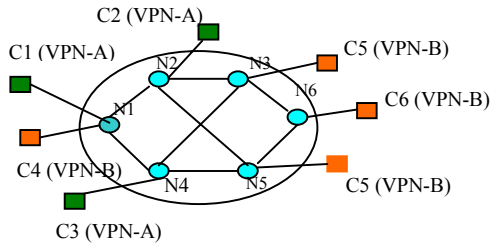


Fig. 3.1: VSP example

Fully Meshed Abstraction Service: This kind of abstraction is shown in Fig. 3.2 with respect to VPN-B. Here the VPN client edge node is provided a virtual topology that is a full mesh of virtual links. Each of the virtual links is associated with the maximum bandwidth available between the two PE nodes belonging to the same VPN. One of the drawbacks with this abstraction is the $O(PE_i^2)$ virtual link updates that need to be sent out to the VPN client nodes. Though this is a scaling issue with respect to hierarchical routing, this might not be a factor of concern with VPN routing, since typically the number of VPN PE nodes supporting a VPN i is small compared to the total number of PE nodes in the core network, i.e. $PE_i \ll V$. The total message complexity in this case is $O(PE_i^2 + CE_i)$. $O(CE_i)$ accounts for the message complexity of updating VPN remote access link resource information. $O(CE_i)$ remains the common complexity component for remaining cases as well.

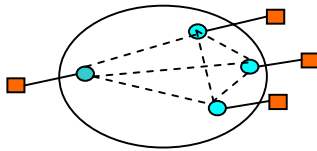


Fig. 3.2: Full Mesh Abstraction

Source Rooted Abstraction Service: This abstraction shown in Fig 3.3, has been proposed in [4]. Here each of the VPN's subscribing to this service would be given a source-rooted tree. The central node of the star is the PE node on which the VPN client node has a direct connection. The other nodes of the star are the PE nodes on which clients of the same type of VPN exist. Directed virtual links could be employed to reduce the virtual topology message update complexity. The total message complexity using this scheme is $O(PE_i + CE_i)$.

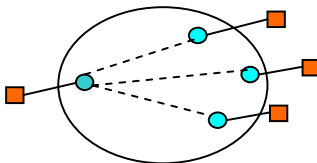


Fig. 3.3: Source Star Abstraction

Star Abstraction Service: This is shown in Fig 3.4. In this scheme the service provider core is virtualized as a virtual node. For a given instance of VPN i , and for a given PE node u , the spoke nodes connect a node $u \in PE_i$ to the virtual node v bi-directionally. To reduce the topology updates one could only flood the links useful in the clients path computation i.e from u to v and v to w where $w \in PE_i$ and $w \neq u$. The bandwidths of the virtual links are averaged from the fully meshed virtual link bandwidth information. In the following equation we assume m is the size of PE_i set.

$$bw_{abs(u,v)} = \sum (bw_{abs(u,k)}) / (m-1), \forall k \in PE_i, u \neq k$$

$$bw_{abs(v,w)} = \sum (bw_{abs(k,w)}) / (m-1), \forall k \in PE_i, k \neq w.$$

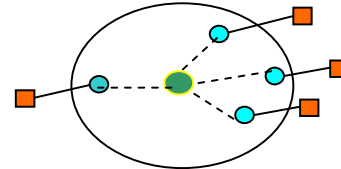


Fig. 3.4: Star Abstraction

Simple Node Abstraction Service: This is the least granular of all the abstraction services. Here the core network is completely virtualized as a single node. The client node is only provided updates on the remote VPN access link available bandwidth. The complexity of this update is $O(CE_i)$ where CE_i is the number of CE's belonging to a VPN instance i .

IV. VPN ABSTRACTION REALIZATION

In this section we try to discuss some of the key ideas with respect to implementing such an abstraction service. Fig 4.1 shows the novel building blocks that would be required to realize such a service. The following proposed modifications would only involve some basic software updates in the control plane layer of a router to realize the abstraction service fully.

In this regard we identify three key VPN components in a PE node with regard to implementing abstraction service: the *Abstract topology database*, *Virtual Topology Computing algorithm* and *SLA VPN Database*. The VPN abstraction topology database is analogous to the Virtual Routing table used to separate each VPN context as referred in [10]. The SLA VPN Database is a commonly shared database that stores the SLA's corresponding to a particular instance of VPN, and information that is necessary to form the abstract topology. Each of the abstraction schemes would be generated using specialized algorithms depending on the virtual link metric association agreed as part of the SLA. As shown in Fig. 4.1, the virtual topology-computing algorithm would use the core network LSDB from the routing database of the VSP to compute the desired abstract topology. The re-computation of the virtual topology could be triggered periodically or when there is a significant change core links in bandwidth information. Also the rate of computation depends on dynamic nature of the VSP's VPN customers. As mentioned earlier the rate of updating the abstract topology is a parameter negotiated as part of the SLA. We later study the effect of the abstract topology update parameter on various abstraction schemes.

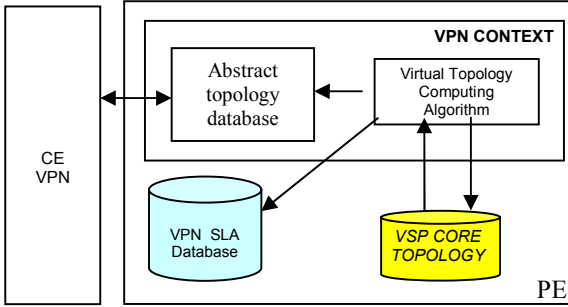


Fig 4.1, VPN Abstraction Topology Framework

V. SIMULATION STUDY

In this section we highlight the simulation results of the comparison study of four different types of VPN abstraction schemes discussed in this paper. The simulation was implemented using OpNet, on a mid size node topology generated using the Waxman's model [14] with an average node degree of 4. 10% of the total nodes of the core topology nodes were chosen randomly as the PE nodes. Each of the PE nodes was configured to handle five different VPN nodes. Each of these VPN's subscribes to a fully meshed (FM), source rooted (SRT), star abstraction (ST), simple node (SN) abstraction service. One of the VPNs was not provided any abstraction service. This is denoted as (NA) in our results. The last kind of VPN service was adopted to show the improvement one might achieve even with minimum amount of topology information. For performance evaluation statistics from each of these VPN service cases has been compared. For comparison purposes we use two different metrics, crankback ratio and success ratio. Success ratio is a measure of making a right routing decision using the abstraction provided by the VSP. This includes the case of computing a feasible path, as well as the case of rejecting a path locally at the VPN client's end. The correctness of a rejected path is verified by re-computing the path with the exact state of the network. For the case where the client node computes a feasible path, the path would crank back, if there were no feasible paths in the core. The second performance metric is the crankback ratio, defined as the ratio of number of calls that have been cranked back to the number calls for which the VPN client node has returned a feasible path. Note that though crankback ratio and success ratio are correlated they are not exactly complementary.

The simulation setup had link bandwidths for the access as well as the core initialized to 100 units. The bandwidth call requests from the VPN client nodes were modeled as Poisson arrivals, with mean call rate of 0.05calls/sec. The holding times of the calls are exponentially distributed with average holding time of 120sec. The VPN abstraction topology update interval was configured as 20secs. The core topology update interval follows the standard intra-domain routing flooding interval of 5secs. The bandwidth requirements follow a uniform distribution of [10, 50]. A higher range of bandwidth request is chosen to bring out visible differences in the performance of each of the abstraction schemes. Fig 5.1 compares the crank

back ratio for each of the VPN's subscribing to the different abstraction services. Here we see that for the VPN with no core topology information, the crank back ratio almost reaches 100%. On the other hand among VPN's subscribing to some kind of abstraction service, the simple node's performance is the worst with 60% calls cranking back. An important result is the improvement in the performance of blocking achieved by providing even minimal topology information that could be as coarse as in simple node abstraction. This is relevant to our discussion since most of the VPN services today offer no core topology service. The best-known information is the reachability service provided to the VPN clients. Fig. 5.2 shows the comparison of the success ratio for the various abstraction schemes. Here one would notice (as expected) the source star and fully meshed schemes have better performance than the star abstraction, but no obvious difference in the performance between source star and fully meshed abstractions. By intuition one would expect the fully meshed abstraction to be the best of all the three abstractions, which has also been reported in [8] with respect to PNNI. This might not hold well in the VPN abstraction context, which can be attributed to the following reasons. In the context of VPN abstraction service, the path computation at the VPN clients end is only a feasibility check for existence of a path satisfying the QoS requirement, as against finding the end-to-end path through which the call is routed. Secondly with increase in core topology information, the client routing could potentially compute multiple paths between any two border nodes, which could lead to greater probability of making erroneous routing decision, a highly likely case when the client's abstract topology information is outdated there by deprecating the performance of a more granular abstraction service.

Fig 5.3 shows the effect of increasing the abstract topology update interval. For the results presented here we increased the interval from 20 to 100 sec. The graph shows lower success ratio for each of the VPN abstraction cases, which can be attributed to lack of the latest core resource status. One should also notice that even a more granular service like fully meshed service does not prove to be of any special advantage. From the results in Fig 5.1 and 5.2, one can observe that, the VSP could use the same abstraction scheme with different abstraction update interval to provide various grades of abstraction services, hence making abstraction update intervals a very crucial SLA parameter.

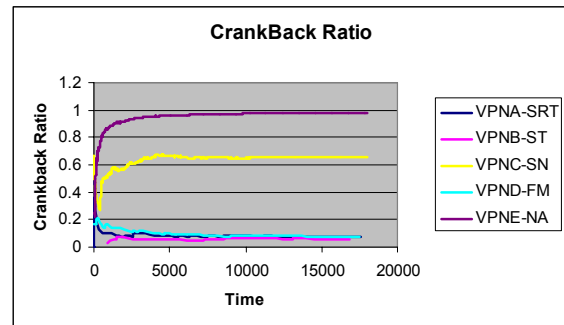


Fig 5.1

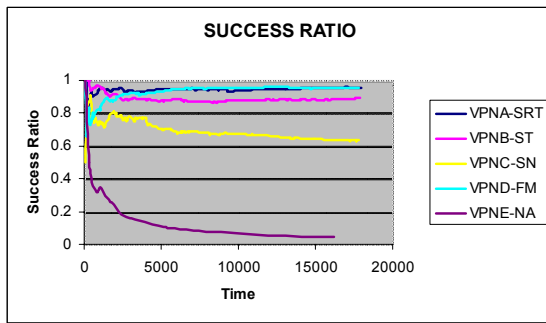


Fig 5.2

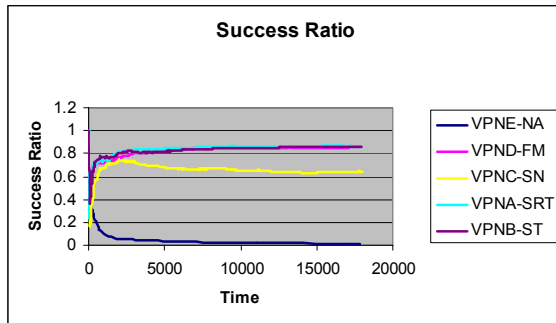


Fig 5.3

In addition simulation was also conducted to study the combined effect of poor abstraction (star) with high update intervals versus good abstraction (source star) with low update intervals. This simulation was conducted with the same arrival rate for all VPN's. It was observed that the stale abstraction information had a dominating affect on the performance. We observed that when the difference in the update intervals is greater than 10secs, the performance of the star abstraction exceeds that of the source star abstraction. This indicates the effectiveness of the combination of abstraction type and virtual topology update interval in differentiating abstraction service.

From these results we conclude that topology abstraction can certainly be realized as VPN service. But our initial presumption of more granular services providing better blocking performance is discounted by the simulation results, as is seen in the case of source rooted and fully meshed schemes. We also observe a wide gap in the performance as we increase the abstraction granularity from a simple node to a fully meshed abstract topology. Our further study would entail to find better intermediate abstraction scheme that could further fill this gap.

VI. CONCLUSIONS

In this paper we explored the problem of topology abstraction as a VPN service. In this context we studied four basic VPN abstraction topologies, namely, fully meshed,

source rooted star, star and simple node abstractions, and discussed the complexity of each type of abstraction. In addition to the virtual topology, other variables of an SLA would be virtual link metrics, and virtual topology update intervals. Architecture to realize this system has also been discussed. Our simulation study shows that providing even minimum topology information can drastically improve the blocking probability for current state of VPN model. For the different cases of abstraction the simulation shows that the simple node scheme suffers from higher crankback ratio than the star, source star and the fully meshed schemes. The simulation study also shows that increasing the virtual update interval results in decrease in performance of the abstraction schemes, making it a good variable parameter to create service differentiation. In addition, the virtual update interval in combination with abstraction schemes can be used to create abstraction service with varying performance. We also note that the proposed problem studied in this paper could be further ramified to issues related to VPN traffic engineering, which is a problem we intend to study further.

VII. REFERENCES

- [1] Private network-network interface specification version 1.0 (PNNI). Technical report, The ATM forum technical committee, Mar. 1996. af-pnni-0055.000.
- [2] W.Lee, "Topology Aggregation for hierarchical routing in ATM networks", ACM SIGCOMM Computer Commun. Rev., vol 25, Apr 1995, pp. 82-92.
- [3] W.Lee, "Spanning tree method for link state aggregation in large communication networks", in Proc. IEEE INFOCOM, 1995, pp, 82-90, Feb, 2001.
- [4] Turgay korkmaz and Marwan Krunz, "Source-Oriented Topology Aggregation with Multiple QoS parameters in Hierarchical Networks", ACM Transactions on Modeling and Computer Simulations, Vol 10, No. 4, Pages 295-325, October 2000.
- [5] King-Shan Lui, Klara Nahrstedt, Shigang Chen, "Routing with Topology Aggregation in Delay-Bandwidth Sensitive Networks", IEEE/ACM Transactions on Networking, VOL. 12, No. 1, February 2004.
- [6] Yong Tang, Shigang Chen, "QoS Information Approximation for Aggregated Networks", IEEE ICC, 2005.
- [7] Fang Hao, Ellen W.Zegura, "On Scalable QoS Routing: Performance Evaluation of Topology Aggregation", IEEE INFOCOM 2000.
- [8] Z.Wang and J.Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications," *IEEE Journal on Selected Areas in Communications*, Vol. 14, no.7, September 1996, pp.1228-1234.
- [9] Ravi Ravindran, Peter Ashwood-Smith et al, "Multiple Abstraction Schemes for Generalized Virtual Private Networks", IEEE-CCECE-2004, Niagara.
- [10] IETF draft, Eric Rosen et al, "BGP/MPLS IP VPN's"
- [11] IETF draft, Paul Knight et al, "Network Based VPN using virtual routers".
- [12] B.M.Waxman, "Routing of multipoint Connections", Journal on Selected Areas in Communications", vol. 6, pp. 1617-1622, December 1988.