Dynamic VPN Architecture for Group Working and Orchestrated Distributed Computing

Yiran Gao[1], Chris Phillips [1], John Bigham[1],

Abstract: This paper describes a new service architecture that extends the use of a Multi-Protocol Label Switching (MPLS) infrastructure for the formation and operation of dynamic communities, incorporating value-added resources allied to the tasks being performed. The approach introduces a new operator-owned control-plane entity called the Dynamic VPN Manager (DVM) for managing customer communities and liaising with the operator's infrastructure. The communities take the form of dynamic VPNs that can be used to support extranet-based on-demand services compatible with "grid computing" and other distributed, collaborative activities. The significance of the approach is further strengthened by supporting methods that allow automated business processes to dynamically request communication infrastructure and processing resources that are compatible with state-of-the-art commercial standards in distributed computing.

1. Introduction

VPNs already carry advanced IP applications such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), videoconferencing, and other mission-critical applications that require guaranteed performance. Typically the formation and maintenance of VPNs within an operator's domain is achieved using manual intervention to configure the various network resources. This is acceptable for long-term connectivity such as inter-connection between enterprise campuses. However, new forms working are likely to have resource usage demands that do not fit well with this arrangement. In these cases it is desirable to establish community relationships that may last minutes, or hours through to weeks. Indeed, seeing this limitation, a number of Dynamic VPN initiatives have recently begun to emerge [1-9]. To extend the flexibility of VPNs in terms of connectivity with QoS support or in terms of their rapid, on-demand, creation, adjustment and removal, a novel unit, referred to as the Dynamic VPN Manager (DVM) is be combined with enhancements to the MPLS signalling framework, to deliver the new functionality [10]. The proposed system enables activities that are currently performed manually to be augmented and even replaced by an automated infrastructure capable of inter-operator negotiation as well as supervising local MPLS VPN configuration management.

2. The Dynamic VPN Architecture

An illustration of the architecture is given in Figure 1. As far as possible the dynamic architecture makes use of existing forwarding and signalling methods, or schemes that are under consideration in IETF RFC documents[11][12]. The principle addition is the inclusion of a new piece on equipment within each Autonomous System (AS), referred to as a Dynamic VPN Manager (DVM) and its associated communications protocols.

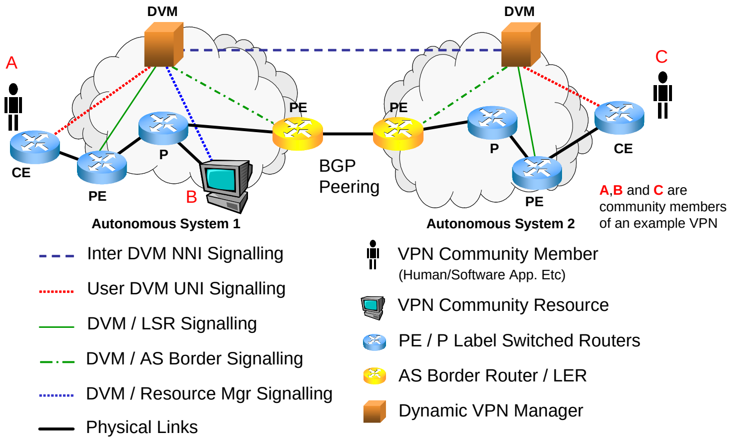


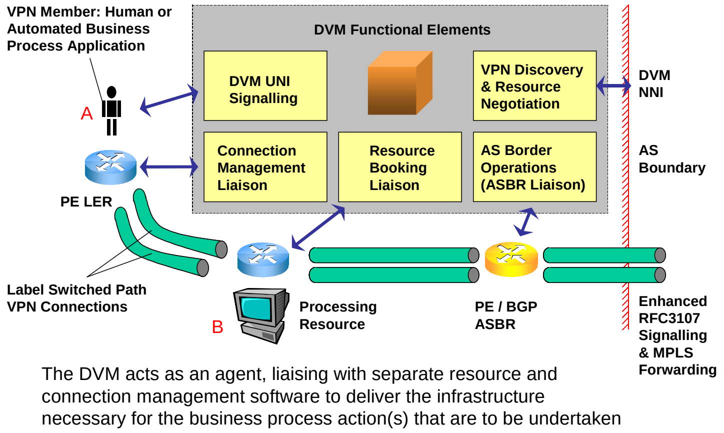
Figure 1: Dynamic VPN Architectural Overview

[1] Queen Mary, University of London, United Kingdom

Each DVM is responsible for managing VPN communities under its jurisdiction. This involves regulating the creating and membership of a given VPN. However, the DVM is not responsible for MPLS connection management and the formation of Label Switched Paths (LSPs). This action is left to separate connection management software that existing within the operator's domain. The DVM's role is to identify the member end-points and to request the connection management function to interconnect them via LSPs that have the desired QoS characteristics. These requests are made between the DVM and the LSRs originating the LSPs within the operator's AS.

Users, be they human or CE-based software entities, can request to create or join a VPN using DVM User-Network Interface (UNI) signalling. Assuming the DVM accepts the request, it then coordinates the setting up of LSPs between the users / entities as well as managing the use of certain processing resources that can be thought of as additional community members. The aim is to exploit an environment in which individual users can reserve / access on-demand computers, databases and experimental facilities simply and transparently, without having to consider where those facilities are located. It is possible that operators could provide grid-computing resource "farms" within their Autonomous System that users can connect to and exploit as part of their dynamic VPN. This provides a means for operators to not only to obtain revenue from the transport infrastructure but also by leasing the "value added" resources that form part of a specific community's grid computing needs. For example, data storage facilities, high performance computing and access to specific databases could all be components of the community's infrastructure requirements.

The DVM acts as an agent, liaising with separate resource and connection management software to deliver the infrastructure necessary for the business process action(s) that are to be undertaken. In order for the DVM to provide the management and coordination of the dynamic VPNs, it comprises a number of functional elements. The key functional elements of the DVM are shown in Figure 2.


VPN Member: Human or Automated Business Process Application
DVM Functional Elements
DVM UNI Signalling
VPN Discovery & Resource Negotiation
DVM NNI
A
Connection Management Liaison
Resource Booking Liaison
AS Border Operations (ASBR Liaison)
AS Boundary
PE LER
Label Switched Path VPN Connections
B
Processing Resource
PE / BGP ASBR
Enhanced RFC3107 Signalling & MPLS Forwarding

The DVM acts as an agent, liaising with separate resource and connection management software to deliver the infrastructure necessary for the business process action(s) that are to be undertaken

Figure 2: Dynamic VPN Manager Functional Overview

3 Testbed Experiment

To evaluate the architecture, a small test scenario was successfully implemented on eight PCs, representing VPN user entities, CE, PE and P routers as well as a DVM, as shown in Figure 3. Further details of the testbed arrangement are given in [13].

The DVM is implemented as a web service. Its interface is advertised using WSDL and clients written in any language can invoke the operations advertised by the WSDL, as the requests comes as SOAP messages compliant with the advertised interface. Requests to the DVM from web service clients are served by the Axis server and this calls Java methods (corresponding to the WSDL descriptions) to e.g. set up a VPN between points of attachment. The Java methods call the JNI (Java Native Interface) and invoke the appropriate C functions of the DVM itself. Via a web browser, a user can communicate over HTTP with its Web Service Client located on a Customer Edge (CE) router. The web service client is a JSP running in the Tomcat web server. The WS client could communicate directly with the DVM using SOAP over HTTP, but in the scenario chosen several such web service clients communicate with a BPEL processing engine (which is also a web service) also located in the CE that choreographs the interaction with the DVM. The location of the BPEL engine does not need to be in the CE. This is simply for convenience. The BPEL process forwards appropriate VPN requests to the DVM over the IP infrastructure again as SOAP messages over HTTP. Effectively it communication with the DVM is via a Web Service Choreography Interface with the DVM's WS server.

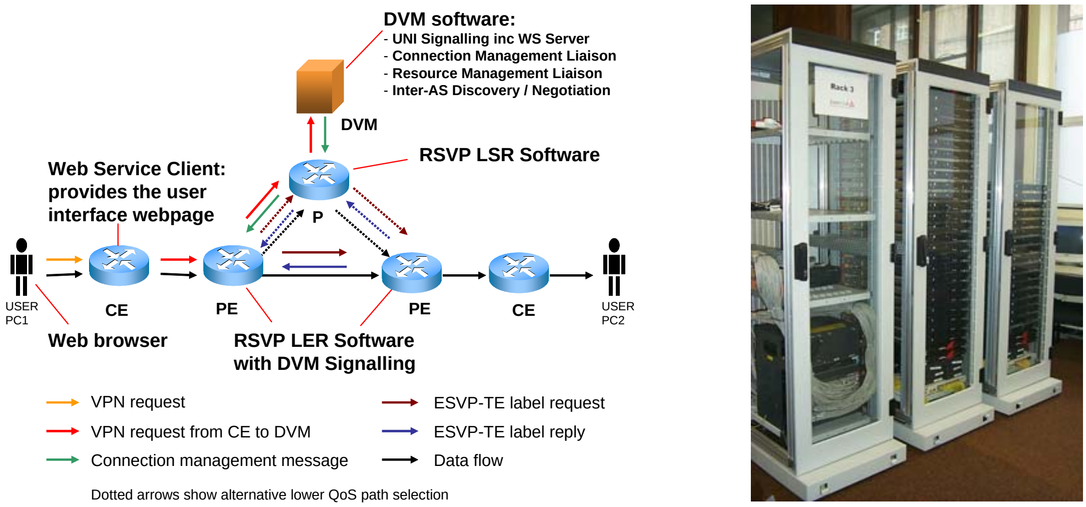

DVM software:
- UNI Signalling inc WS Server
- Connection Management Liaison
- Resource Management Liaison
- Inter-AS Discovery / Negotiation
DVM
RSVP LSR Software
Web Service Client: provides the user interface webpage
P
USER PC1
CE
PE
PE
CE
USER PC2
Web browser
RSVP LER Software with DVM Signalling
VPN request
VPN request from CE to DVM
Connection management message
ESVP-TE label request
ESVP-TE label reply
Data flow
Dotted arrows show alternative lower QoS path selection

Figure 3: Dynamic VPN Test Scenario and the actual Testbed

As already mentioned, the DVM is responsible for controlling VPN membership. It is also responsible for liaising with separate operator owned connection and resource management to selectively interconnect community members via MPLS LSPs between their Provider Edge (PE) router points of access to the core network. It also determines access to operator or third-party resource islands that it can make accessible to specific communities. In the test scenario, a user entity joins a community and is then able to communicate over a dynamically configured LSP with a remote member of the same community. This is achieved by the DVM liaising with connection management software located in the ingress PE router, which sets up an LSP between the identified PE endpoints. The path taken by the LSP is chosen by the connection management software based on the QoS constraints identified by the DVM. In one instance this is directly between the PE routers (when a latency sensitive pathway is requested) or, alternatively, via an intermediate P router, when a longer path can be tolerated. However, it is important to note that the DVM does not itself select the path; this is a connection management task. There is therefore a useful decoupling between the user community constraints the means by which the operator chooses to meet them.

4 Current Activities

The DVM is responsible for scheduling and reserving communications and processing resources for complex jobs expressed as a workflow. For example a job may require data to be processed, then combined with further data and subsequently processed again before being delivered to a storage facility. The location of these processing activities could be distributed. Given the performance requirements, it may also be beneficial to set aside underlying communications channels (such as LSPs) at the appropriate time(s) to ensure data is moved quickly between the processing points. The processing and storage resources are formed into a VPN community. However, one of the key challenges remains determining how long these resources will be used for. This requires the use of prediction and possibly contingency planning. Current research is focusing on this point using OPNET Modeler™.

OPNET Modeler™ is also being used to explore the scalability of the architecture. As the DVM is a centralised controller, it may be necessary to support multiple DVMs within a domain or to operate it as a distributed, concurrent server whereby child processes are created to manage individual VPN communities as needed. These children could be located on different computer(s) from the main access point for scheduling and coordination. Another aspect of the architecture being considered is the security of the DVM, as it would appear to be a single point of vulnerability for Denial of Service attacks. However, once again there is scope to distribute its functionality or to provide a "farm" of DVM processors to spread the load and improve robustness.

5. Conclusion

The DVM concept provides the ability to sell on-demand connectivity together with leased access to value added resources such as CPU time or data storage, which should help to bring a new source of revenue to the carrier. An automated VPN management system based on the DVM is described that eases the manual burden and the associated risks of erroneous configuration, where the process is automated using a web service workflow engine. This is superficially illustrated in a test application described using a prototype based on a public domain forwarding engine [14]. Future work aims to extend the resource management, discovery and negotiation and security management to roaming community members.

Acknowledgment

The authors gratefully acknowledge the support of BT through a short-term fellowship with the Security Research Centre, BT Group CTO as well as the invaluable advice of Dr Liwen He.

References

[1] Kindred, D.; Sterne, D. “Dynamic VPN communities: implementation and experience”, DARPA Information Survivability Conference & Exposition II, 2001. DISCEX '01. Proceedings Volume 1, 12-14 June 2001 Page(s):254 - 263 vol.1

[2] Y. Jia et al, “Dynamic resource allocation in QoS-enabled/MPLS supported virtual private networks and its Linux based implementation”, Electrical and Computer Engineering, 2002. IEEE CCECE 2002, Volume 3, 12-15 May 2002 Page(s):1448 - 1454 vol.3

[3] Y. Jia et al, “Design and testbed implementation of adaptive MPLS-DiffServ enabled virtual private networks”, Electrical and Computer Engineering, 2003. IEEE CCECE 2003, Volume 2, 4-7 May 2003 Page(s):965 - 968 vol.2

[4] Refer to: http://www.leetnet.org/ (URL accessed 20th June 2005)

[5] Refer to: http://www.nrns.ca/DRDC.htm (URL accessed 20th June 2005)

[6] P. Lago, R. Scandariato, “A TINA-based solution for dynamic VPN Provisioning on heterogeneous Networks”, Proc. IEEE Telecommunications Information Networking Architecture Conference (TINA'2000), Paris, France, 13-15 Sep. 2000.

[7] Rebecca Isaacs, and Ian Leslie,“Support for Resource-Assured and Dynamic Virtual Private Networks”. IEEE JSAC, Vol. 19, No. 3, March 2001.

[8] Fujita, N. et al, “Scalable overlay network deployment for dynamic collaborative groups”, Proc. 2005 Symposium on Applications and the Internet, Page(s):102 – 109, 2005.

[9] L. Andersson, T. Madsen, “Provider Provisioned Virtual Private Network (VPN) Terminology”, IETF Request for Comments: 4026, March 2005.

[10] C. Phillips *et al*, “Flexible Distributed Testbed for High Performance Network Evaluation”, IEEE TridentCom, 2006, Barcelona, March 2006.

[11] E. Rosen, Y. Rekhter, “BGP/MPLS VPNs”, IETF Request for Comments: 2547, March 1999.

[12] Y. Rekhter, E. Rosen, “Carrying Label Information in BGP-4”, IETF Request for Comments: 3107, May 2001.

[13] C. Phillips *et al*, “Managing dynamic automated communities with MPLS based VPNs”, Journal of the Institution of British Telecommunication Engineers, (to appear) Vol 24 No 2, 2006.

[14] Refer to: http://www.cl.cam.ac.uk/Research/SRG/netos/netx/ (URL accessed 16th February 2006)