Technical Memorandum

BT TYMNET, Inc.
Technology Research and Planning

Title: **Virtual Link Models for Protocol Encapsulation**

Author(s): William R. Soley.................B41........6210.......wrs@tymnet.com

Issued: January 14, 1990

Revised: February 21, 1991 12:38 pm

File: /u/wrs/doc/VirtualLinkModels.ps

Sensitivity:

Abstract: A *virtual link* is a data-link connection that uses one or more network connections through an underlying network in place of a physical connection. Diagrams show the virtual link in terms of the OSI Reference Model. Two possible virtual link models are presented: (1) a *Point-to-point Model*, and (2) a *Multi-point Model*. Some observations are made about the impact of each model on routing update messages (assuming RIP/IP). Examples of the models and their effect on address association are given using transmission of IP datagrams over X.25 Virtual Circuits as an example. The multi-point model is found preferable because of fewer network addresses, reduced routing update traffic, and reduced configuration complexity.

1. Introduction

Generally, a data communication network is made up of systems (nodes) which are interconnected by a communication medium. In wide area networks, this medium is typically a transmission line. The service provided by the line is enhanced by the data-link layer to create a data-link connection (link). The service provided by the link is further enhanced by the network layer and higher layers. Figure 1 shows an example of a “real” link in terms of the OSI Reference Model.[1]

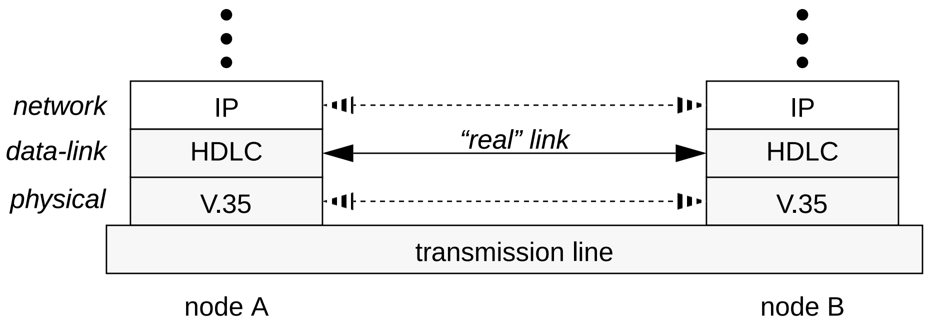

Figure 1. A “Real” Link based on a Transmission Line

For administrative, economic or technical reasons, it is sometimes desirable to interconnect nodes of one network by using the services of an underlying network in place of the physical connection which might otherwise be provided by a transmission line. Figure 2 shows an example of a “virtual” link as two concatenated subsystem hierarchies. Protocol data units of the top hierarchy are said to be *encapsulated* within those of the lower hierarchy. The lower hierarchy provides services to the data-link layer of the upper hierarchy which would normally be provided by a physical layer. To avoid confusion, the word *underlying* is used herein to refer to the lower hierarchy (*e.g.* underlying network layer).

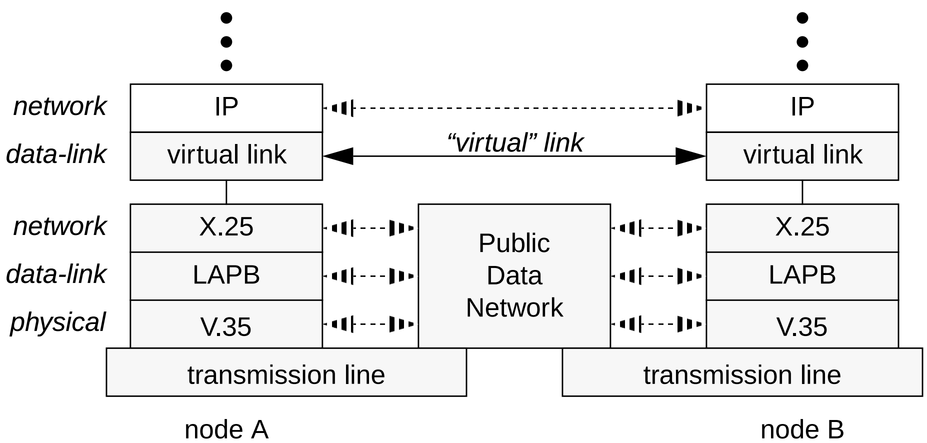

Figure 2. A “Virtual” Link based on an Underlying Network Connection

A virtual link must provide the same services at its interface (to the network layer) as would a real link (*e.g.* HDLC) so as to allow a network entity (*e.g.* IP) to use them interchangeably. Likewise, a virtual link must make use of the underlying network interface (*e.g.* X.25) as would a transport entity so as to allow a single underlying network entity to serve them both.

[1] CCITT, *Reference Model of Open Systems Interconnection for CCITT Applications*, Rec. No. X.200, Geneva, 1985.

When more than two virtual link peers are based on the same underlying network, it becomes necessary to consider a more detailed model of the virtual link. Two models presented here are a *point-to-point* model, and a *multi-point* model.

The examples herein show transmission of IP datagrams over X.25 Virtual Circuits as specified by RFC-877[2], but the models presented are as general as possible. Comer and Korb[3] describe the CSNET IP-to-X.25 interface, the concepts of which may naturally be extended to other combinations of protocols. In particular, their paper discusses methods for providing a datagram service based on an underlying connection oriented service.

It is important to standardize the virtual link model to allow interoperating of nodes connected by virtual links in the same way one expects to be able to interoperate nodes connected by standard data-link protocols such as HDLC. An example of this problem is systems using RFC-877 encapsulation. Many vendors have implementations which follow the RFC, but some are unable to interoperate because the RFC leaves the virtual link model undefined. For example, Sun Microsystems' SunLink X.25 Internet Router[4] product can only interoperate with cisco Systems' Gateway Server[5] in certain degenerate configurations. The principle incompatibility between the systems is the virtual link model: Sun uses a point-to-point model, while cisco uses a multi-point model. Sun and cisco are mentioned only as an example to highlight the importance of choosing a model and nothing is implied about how closely these implementations may fit the models presented here. [Since this paper was written, both cisco and Sun have released new versions of software (8.0 and 7.0, respectively) which claim to support both models described here.]

2. Point-to-point Model

In this model, the virtual link behaves like a data-link layer based on a point-to-point transmission line. Figure 3 shows address associations in an example 3-node subnet based on the point-to-point model. The underlying network connections may be permanent or on demand such as described in [3].

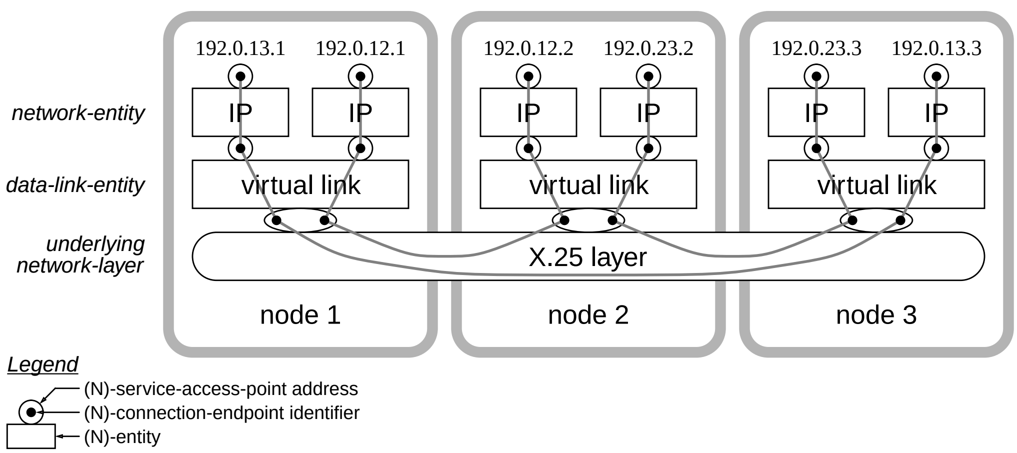

Figure 3. Example of Address Association in the Point-to-point Model

A point-to-point virtual link entity maps each potential underlying network connection endpoint (see Figure 3) into a data-link service access point (interface). In an IP[6] network, the network entity maps each data-link interface to one or more network addresses. As a result, a subnet of n nodes fully interconnected by point-to-point virtual links requires a minimum of $n^2 - n$ network addresses. If RIP[7] routing is employed (not a good choice in this case, but

[2] J. T. Korb. *A Standard for the Transmission of IP Datagrams Over Public Data Networks*, RFC 877, Purdue U., Sep. 1983.

[3] D. E. Comer, J. T. Korb. *CSNET Protocol Software: the IP-to-X.25 Interface*, ACM SIGCOMM Symposium on Communications Architectures and Protocols, March 1983.

[4] Sun Microsystems, *SunLink X.25 System Administration Guide*, §6, Part No. 800-2455-05, Dec. 1988.

[5] cisco Systems, *Gateway System Manual*, Menlo Park, CA, July 1988.

[6] J. Postel (ed.). *Internet Protocol — DARPA Internet Program Protocol Specification*, RFC 791, USC-ISI, Sep. 1981.

useful as an extreme example), the additional routing overhead in the underlying network due to routing updates is 160 bits/entry × $(n^2 - n)$ entries/message × $(n - 1)$ messages/update (assuming the underlying network does not directly support multi-cast) × $1/30$ updates/node-second × n nodes = $5.33(n^4 - 2n^3 + n^2)$ bits/second. Both the number of network addresses required and the routing overhead can be reduced by not fully connecting the nodes with virtual links, however this requires some nodes to relay traffic (which they would otherwise not handle) between nodes which are not directly connected by a virtual link.

Configuring a point-to-point virtual link subnet requires each node to keep a table having one row for each of its neighbors in the subnet, each row containing three parameters as follows:

(1) data-link address of this end (usually the same as the network address, *e.g.* an IP address)
(2) data-link address of the other end
(3) underlying network address of the other end (*e.g.* an X.25 address)

This table is different for every node in the subnet so they may not share a common file. The configuration complexity is 3 parameters/row × $(n - 1)$ rows/table × n tables/subnet = $3(n^2 - n)$ parameters/subnet.

3. Multi-point Model

In this model, the virtual link behaves like a data-link layer based on a multi-point transmission line. If the underlying network service provides a multi-point connection, there is a one-to-one correspondence between the underlying network connection and the virtual data-link connection. Otherwise, the virtual link subsystem must simulate the behavior of a multi-point connection. Once again, underlying network connections may be permanent or on demand such as described in [3].

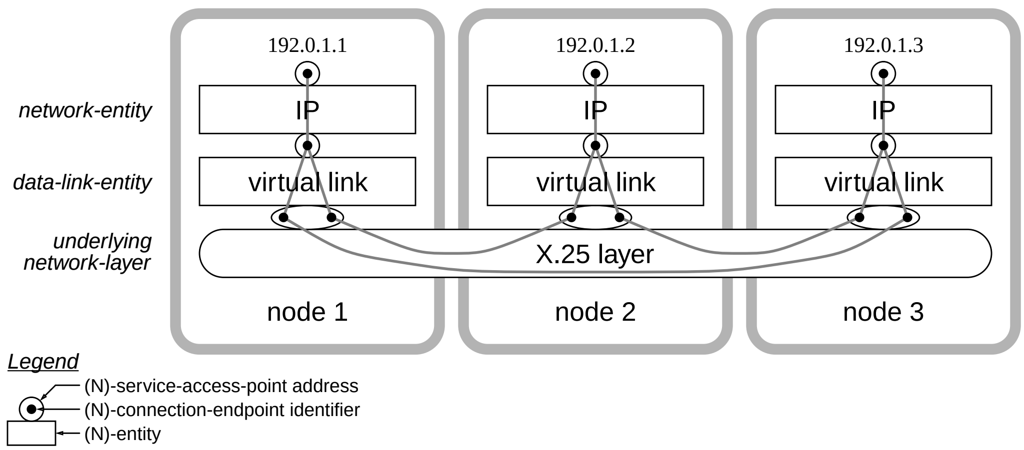

Figure 4. Example of Address Association in the Multi-point Model

A multi-point virtual link entity maps each underlying network (see Figure 4) into a data-link service access point (interface). In an IP network, the network entity maps each data-link interface to one or more network addresses. As a result, a subnet of n nodes interconnected by a multi-point virtual link requires a minimum of n network addresses. If RIP routing is employed, the additional overhead in the underlying network due to routing updates is 160 bits/entry × 1 entry/message × $(n - 1)$ messages/update × $1/30$ updates/node-second × n nodes = $5.33(n^2 - n)$ bits/second.

[7] C. Hedrick. *Routing Information Protocol*, RFC 1058, Rutgers U., June 1988.

Configuring a multi-point virtual link requires a table having one row for each node in the subnet, each row having two parameters as follows:

(1) data-link address (usually the same as the network address, *e.g.* an IP address)
(2) underlying network address (*e.g.* an X.25 address)

This table is identical for every node in the subnet so they may share a common file. The configuration complexity is 2 parameters/row $\times n$ rows/table $\times 1$ tables/subnet $= 2n$ parameters/subnet.

4. Comparison

The point-to-point model, if the nodes are fully-connected, requires a large number of addresses, $O(n^2)$ in large subnets, which may be a problem under some numbering plans. More numerous addresses generally means more routing overhead. In an IP network, each of these addresses appear in every routing table, and routing updates (using RIP) consume up to $O(n^4)$ resources. In addition to these resource issues which impact performance, the additional configuration complexity may also impact reliability by increasing the probability of configuration errors. Use of the point-to- point model in a design therefore precludes its use in large fully-connected subnets. The definition of a *large* subnet depends on the available bandwidth and the definition of acceptable performance.

	point-to-point	***multi-point***
number of network addresses	$n^2 - n$	n
RIP update overhead (bps)	$5.33(n^4 - 2n^3 + n^2)$	$5.33(n^2 - n)$
configuration complexity	$3(n^2 - n)$	$2n$

The point-to-point model, by explicitly specifying each underlying network connection in the configuration table, allows greater administrative control. The network administrator may omit selected underlying network connections from the table creating a non-uniform topology which may better suit policy. A side effect of constraining the topology is to reduce the resource requirements allowing larger subnets to be feasible under this model. The constraints must be periodically reviewed and possibly adjusted to track changing policies and usage patterns creating a significant engineering burden. Contractual agreements, tariffs, cost and characteristics of the underlying network, government regulations, security policy and intra-company cost accounting are some examples of situations which may dictate constraining the subnet topology. If the total number of simultaneous underlying network connections per node is limited by the underlying network implementation to significantly less than $n - 1$, performance improvements might be achieved by manually constraining the subnet topology based on a priori knowledge of traffic patterns.

The multi-point model requires fewer resources in large subnets. It is simple to configure, and its configuration file may be shared by all nodes in a subnet. The multi-point model does not inherently possess the configuration flexibility of the point-to-point model which allows manual constraint of the subnet topology. This feature may be recaptured by designing the virtual link subsystem to allow more than one multi-point virtual link per underlying network interface. Figure 5 shows an example of one node with two multi-point virtual links based on the same underlying network. Because a subnet based on a collection of degenerate (i.e. having only two endpoints) multi-point virtual links behaves identically to a subnet based on a collection of point-to-point virtual links configured in the same topology, a subsystem designed thusly may interoperate with subsystems based on the point-to-point model.

5. Summary

Virtual link subsystem designs based on the multi-point model require fewer resources and are simpler to configure than those based on the point-to-point model. To allow interoperability between these two models, and to provide enhanced configuration flexibility, designs based on the multi-point model should support multiple virtual links per underlying network interface.

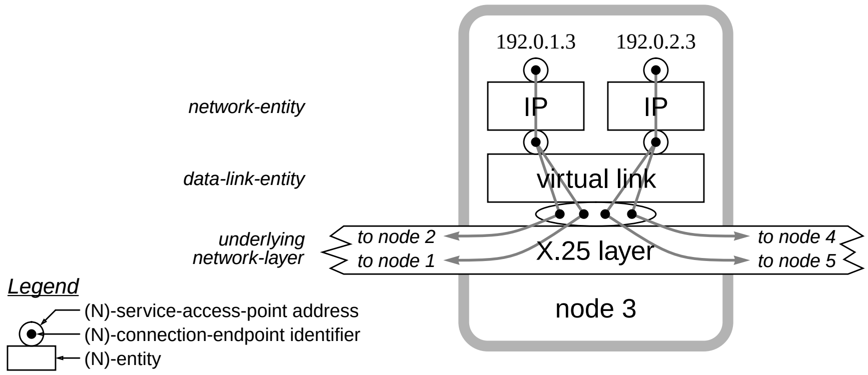

Figure 5. Address Association with Multiple Multi-point Links per Underlying Network