Scalability Enhancements for Connection-Oriented Networks

E. Gauthier* J.-Y. Le Boudec

Laboratoire de Réseaux de Communication (LRC),
EPFL CH-1015 Lausanne

June 14, 1995

Abstract

We consider the issue of increasing the number of connections that connection oriented networks, such as ATM, can handle. We describe one step that arises at reducing connection awareness inside the network. To that end, connections between the same pair of access nodes are grouped together and made indistinguishable inside the network. The concept of dynamic virtual path trunks is introduced as a support mechanism, and it is shown how virtual path links can be setup and maintained without additional round-trip delays.

Keywords : connection awareness, multiplexed signaling, bandwidth re-negotiation.

1 Connection Limits for Network Nodes

Support for real time traffic in communication networks requires that a connection oriented philosophy be employed. Some network technologies are connection oriented right from their definition : this is the case for ATM [1, 2], or Narrowband ISDN [3]. Some others initially based on connection-less principle, like bridged LANs or Internet [4], are introducing resource reservation methods that end up using a concept of connection (or analogous concepts such as "soft state" for RSVP [5, 6]). As a result, and as multimedia communication becomes a mass market, the networking industry will be confronted with the requirement to build and operate connection oriented networks of increasing sizes.

Designing and operating connection oriented networks is not simple, because of the complexity required to support connections. Every connection established through the network is associated at every node with label swapping tables, with capacity reserved in queues, and with connection control blocks (or equivalent denominations) used by the signaling or control protocol ; connection establishment, tear-down, or simply keep-alive requires some processing [7]. Last but not least, every connection is visible to network management, and thus comes with a horde of various data structures to represent the connection static and dynamic attributes, together with measurement results for statistical purposes.

*contact person : gauthier@di.epfl.ch, tel. +41 21 693 6747, fax. +41 21 693 6610

Due to these elements, network nodes have limits on their connection handling capabilities. Such limits depend strongly on the node design, and generally depend on its configuration (such as amount of memory installed, or number of ports if some of the processing is distributed or the port cards). The limits can be expressed in a number of ways, representing limitations in both memory and processing capacity. For example, a local area ATM switch may be limited to support at most 10,000 connections in steady state, and at most 150 connection establishment attempts per second. (In general, exact limitations are expressed in a more complex way because the problem is multi-dimensional.) Of course, node capacity is not only limited by connections, but also by other factors such as link capacity and number of ports.

We believe that the success of connection oriented networks and especially the introduction of ATM to the end users will require that these limits be pushed as far away as possible. Ideally, we would like to be able to design network nodes that are not limited by the connection handling aspects ; to that end, we are developing the *scalability enhancements for connection-oriented networks* (SCONE) project. The goal of SCONE is to develop solutions that enable the use of many connections per end-user. We consider the four following directions for developing a SCONE solution :

1. **Basic performance improvement :** with faster processor, faster, larger and cheaper memory, performance of existing implementations can be improved.

2. **Hardware assist :** some functions traditionally implemented in software can gain performance if casted into silicon ; this would concern typically basic protocol processing functions such as timer management.

3. **Function distribution :** distributing functions onto several processors and associated memory is a well known, promising methods for increasing performance. Some network architectures lend them well to function distribution because they define functional blocks with loose coupling. For example, the ATM-forum P-NNI signaling method can be implemented with one control instance per port. Similarly, the proprietary control system of [2] defines functions for connection signaling, for bandwidth allocation that can be easily implemented as parallel modules.

4. **Architecture solutions :** by architecture solutions we mean protocols and methods that speed the network operation. Architecture solution can improve the performance of connection handling by reducing connection awareness wherever this is not necessary.

All these four directions are of course not exclusive and their effect at increasing the number of connections a network can support can be cumulated. In this paper we consider one architectural solution (Section 2). We do not claim to be exhaustive and there are many other aspects that need consideration, such as for example the use of precomputation rather than on demand computation for path selection [8, 9].

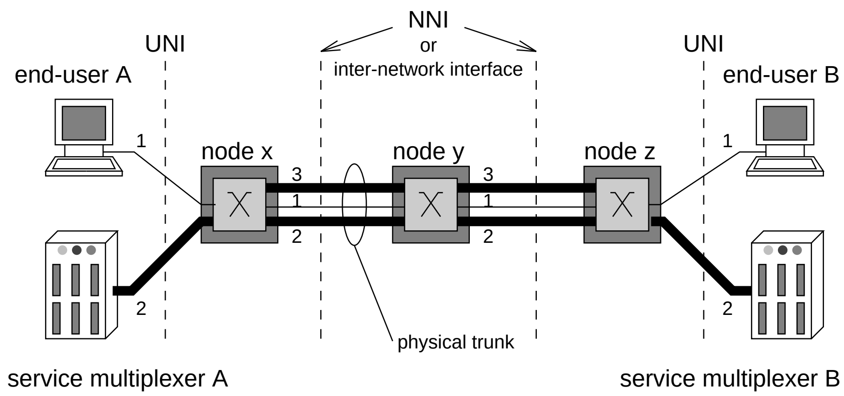

Figure 1: Example of VCCs, VPCs and VPTs : Virtual Channel Connection 1 is setup between systems at UNIs A and B ; it is made of 4 VC links (all numbered 1). Similarly Virtual Path Connection 2 is setup between systems at UNI A and B, and is made of 4 VP links. There also exists a Virtual Path Connection used as a trunk (VPT 3) between node x and z. The VPT is used internally and is not visible at UNIs.

2 Reducing Connection Awareness

In this paper we present a method that reduces connection awareness at transit nodes in an ATM network. The method is based on the following ideas :

- Connections that have same source and destination nodes are grouped together and made indistinguishable at transit points ;
- Virtual Path concepts are used for grouping connections.

The method is described for point-to-point connections. Application to multicast connections is under study. In this paper, we apply our ideas to a private network environment, and adopt the framework of the ATM Forum P-NNI working group, which assumes that signaling between nodes inside a private network uses a simple extension of the UNI signaling. We believe however that our method can equally be applied to public networks.

2.1 Virtual Path Trunks

The method is based on the concept of dynamic Virtual Path Trunks (VPTs). We use the term “ trunk ” to denote a link between adjacent nodes inside the network (as opposed to access links). VPTs are virtual path connections (VPCs) setup inside the network, by the network itself, for the sole purpose of reducing connection awareness inside the network. Once established, VPTs are used as trunks, namely, the network nodes connected by VPTs act as though they were physically adjacent. VPTs are not visible at the UNI. Figure 1 shows an example of VPT used by an end-to-end connection, and illustrates related concepts. Transit nodes that are not the end points of VPTs act as Virtual Path switches. We assume, as is usual in a private environment, that ATM switching nodes can act both as transit and access nodes, and can perform both virtual path and virtual channel switching [10]. We also assume that every VPT carries

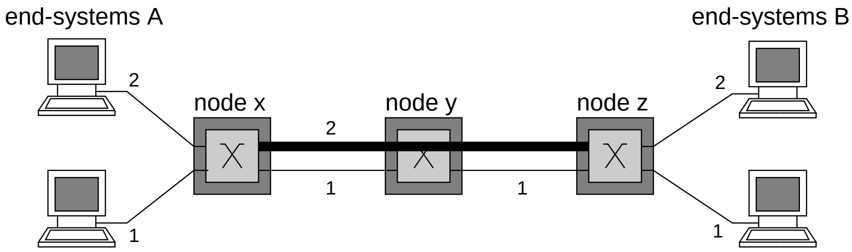

Figure 2: Virtual Channel Connection 1 is setup hop-by-hop, using only implicit VPCs ; VCC 1 is made of 4 VC links. Virtual Channel Connection 2 uses an already established Virtual Path Trunk ; VCC 2 is made of 3 VC links.

a signaling link (using for example VCI = 5). The signaling link needs to be established between the two entities at the end of the VPT.

In principle, VCCs can exist only inside a VPC, since the ATM cell header requires the VPI field to be populated. However, in many cases in private networks, especially at the UNI, there is no real VPC supporting the VCC, but, in contrast, there is one VC link per physical link. In such cases we speak of *implicit* VPCs. Implicit VPCs exist only in that a VPI value is allocated ; however, they need not be visible a separate network management entities, there is no need to establish, maintain or release them, and they have no associated resources. Figure 2 shows implicit VPCs.

VPTs could in principle be concatenated and more complex scenarios could be envisioned than shown on Figure 2 (with VPTs between transit nodes for example). In this paper, we consider only cases where VPTs are used between access nodes. In particular, VPTs apply to scenarios where end users establish Virtual Channel Connections ; the concept cannot be used to carry end-to-end Virtual Path Connections.

2.2 Dynamic Virtual Path Trunks

Assume, as an example, that end-system A, served by node x, is establishing a virtual channel connection to end-system B. Assume also that the routing and topology function has determined that B is served by node y and that the connection should be attempted through node y (the routing and topology function is outside the scope of this paper [11]). If no VPT exists between node x and y, then a hop-by-hop setup flow like in Figure 3 would be used (with implicit VPCs on all VC links). There, node y is active at setting up, maintaining and releasing the connection. In contrast, if a VPT already exists between node x and y, and has sufficient capacity to accommodate the new VCC, then the setup flow involves only access nodes x and y (Figure 4). In that case, node y does not see the new connection.

To make the presentation simpler, signals call-proceeding and connect-ack are not shown on the figures.

In current architectures, VPT must be created manually, in an *ad hoc* manner, and their capacity is permanently reserved. If there are no VPT with enough bandwidth to support a new VCC, either a new VPT should be created manually

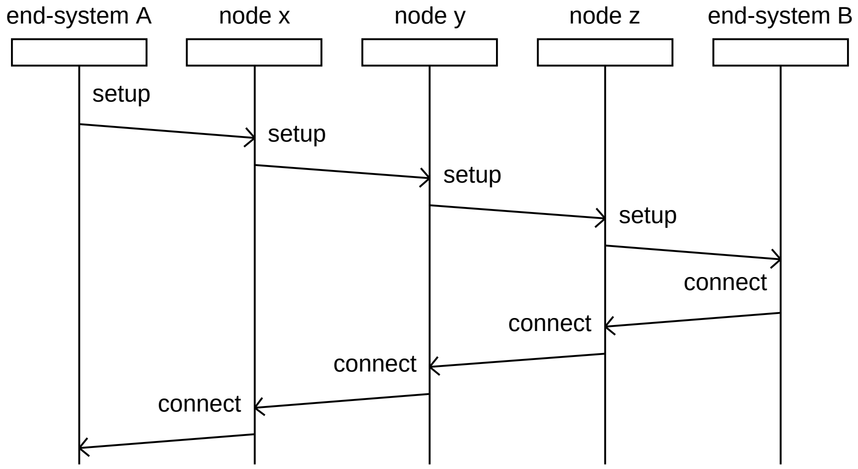

Figure 3: VCC establishment hop-by-hop

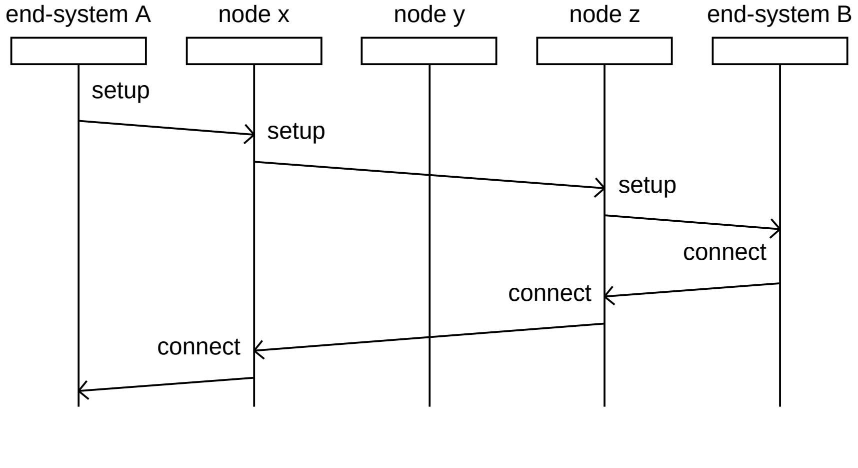

Figure 4: VCC establishment over an existing VPT between x and z

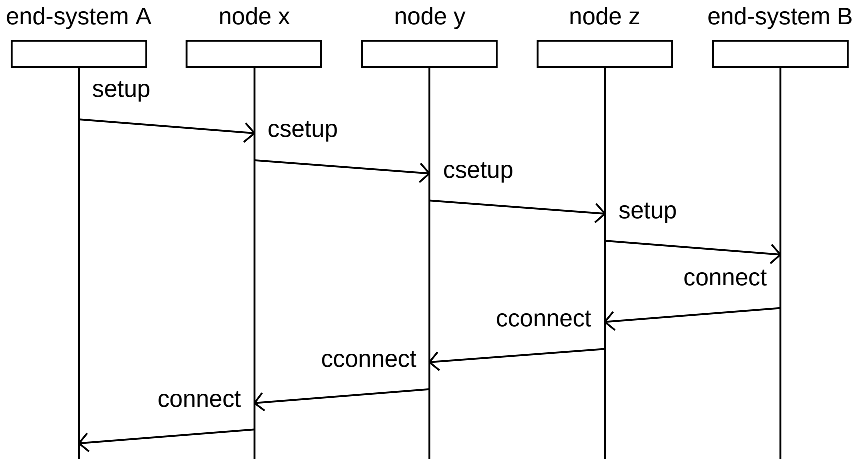

Figure 5: Combined VPT and VCC establishment

or the new VCC should be established hop-by-hop.

With SCONE, we would like to take advantage of VPTs in order to decrease connection awareness at transit point. The challenge is (1) to avoid adding overhead round-trips at VCC creation and (2) be able to use an existing VPT that does not have enough bandwidth reserved to accommodate the new connection.

Note that this is possible that the routing and topology function determines that the route x-y-z should be attempted, even though VPT x-z exists and has the capacity to accommodate the new connection : in such cases, the capacity is available, but was not allocated yet to the VPT.

With SCONE, VPT can be created automatically by the network and their reserved bandwidth can be modified dynamically. Hence, if a new VCC should be established, SCONE offers two new functionalities :

- **combined VPT and VCC setup :** the establishment of the new VPT and the new VCC are combined together so that it takes no longer than establishing a new VCC hop-by-hop and it is much faster than establishing the VCC after having established a VPT ; the combination of VPT and VCC setup and connect signals, called csetup and cconnect respectively, is illustrated in Figure 5.

- **simultaneous VCC setup and VPT bandwidth increase:** if an existing VPT has not enough bandwidth, this function establishes a new VCC as fast as if the VPT had enough bandwidth as shown in Figure 6 ; two new signals are introduced for this purpose : reservreq which increases the reserved VPT bandwidth in each node along the VPT and reservack which acknowledges successful reservation of all nodes as specified in the fast reservation protocol (FRP) [12]; the dotted region indicates that signals setup and reservreq are both sent and received simultaneously.[1]

[1] Note that in Figure 6 the setup is forwarded to end-system B once both reservreq and setup are received at node z.

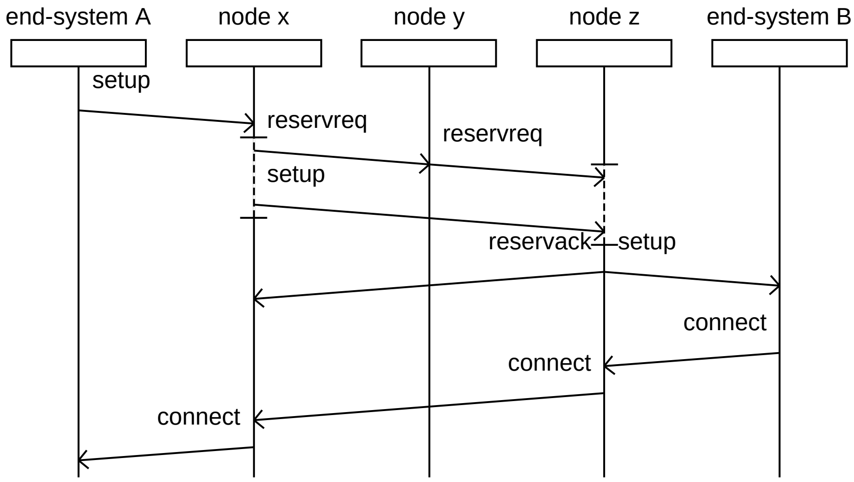

Figure 6: Simultaneous VPT new bandwidth reservation and VCC establishment : signals within the same dotted region are simultaneous.

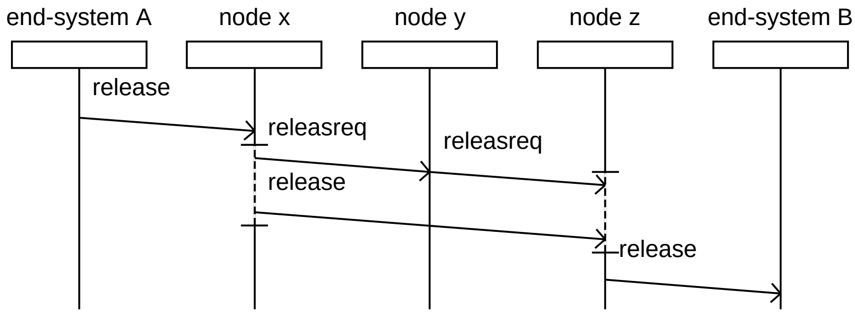

Figure 7: Simultaneous VCC release and VPT bandwidth decrease

To make the presentation simpler, signals validation-request and reservation-complete are not shown in the figures.

From a connection handling point of view, it is worth noticing that modifying the bandwidth of one VPT is considerably simpler than setting up a new connection. In particular, it can be assumed in some implementations that FRP used for that purpose is handled by the switch hardware [12].

Similarly, if a VCC should be released, SCONE offers two new functionalities :

- **simultaneous VCC release and VPT bandwidth decrease :** as shown in Figure 7, this new function takes no longer than a simple VCC release ; signal release requests that resources reserved by VCC be returned to VPT while signal releasreq decreases VPT bandwidth by the amount reserved by the VCC ;

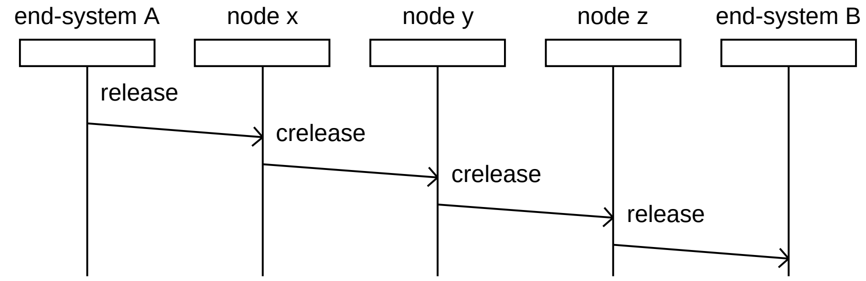

Figure 8: Combined VPT and last VCC release

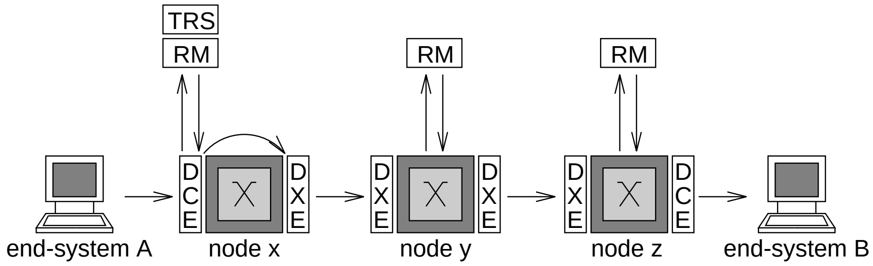

Figure 9: Functional model for connection handling

- **combined VPT and last VCC release :** the release of the VPT and of its last VCC are combined together so that it takes no longer than releasing a VPT and it is much faster than releasing the VPT after having released its last VCC ; the combination of VCC and VPT release signals, called crelease, is shown in Figure 8.

Again to make the presentation simpler, signals release-complete and releasreq-complete are not shown on the picture.

In essence, the method proposed here consists in replacing bundles of individual VCCs that follow the same route by single VPTs with variable capacity. We assume the functional model illustrated in Figure 9 for connection handling. The signaling function is called DCE on the network side of the UNI, and DXE at the P-NNI. Resources requested by setup flow are owned by Resource Managers (RM) ; routes are computed by the "Topology and Route Selection" (TRS). The SCONE method has a lot of potential for routing in a transparent way large number of VCCs. In particular, application to re-routing scenarios (for example after failure) is for future study. The new signaling functionalities are described in detail in the following section.

3 SCONE Architecture for ATM

We now introduce the different functional blocks involved in setting up and releasing VPTs and VCCs. We will only study in detail the DXE function since

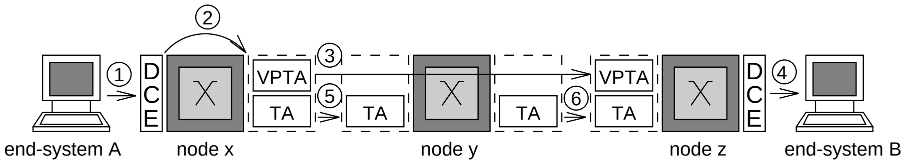

Figure 10: Connection Request Flows : DXE functions are dashed.

this is the only function affected by the addition of dynamic virtual path trunks to current architectures. A DXE supports all signaling carried by a particular trunk and is composed of the two signaling agents :

- a *trunk agent* (TA) that supports the signaling with the next adjacent node ;
- a *virtual path trunk agent* (VPTA) that supports the signaling with the other extremity of each VPT starting at that trunk.

3.1 Connection Request Example

Figure 10 shows the request of a point to point VCC, where each circled number corresponds to one of the following steps.

1. A connection setup is issued by end-system A across the UNI to its network access node x ; the signal is received at x by the DCE which supports ATM signaling with A.

2. DCE function forwards the connection request to the DXE function which supports all signaling down the computed route x-y-z.

3. If VPT x-z exists and is selected to accommodate the new connection, the VPTA creates a *connection unit* (CU) to represent the P-NNI signaling state of the new connection. The VPTA uses the VPT signaling link to transmit a setup to the peer VPTA at the other extremity of the VPT. When the setup is received by the peer VPTA, a peer CU is created and receives the connection request.

   The VPTA is also responsible for allocating VPT bandwidth to the new connection.

4. DCE transmits the connection setup to the destination end-system B.

5. If a VPT x-z should be created to support the new connection, the VPTA creates two CUs : one for the new VPT and one for the new VCC. Both setup messages are combined in a single csetup which is forwarded to the next adjacent node using the TA signaling link.

   In the case where the bandwidth for the selected VPT must be increased to accommodate the new connection, the VPTA creates a *fast reservation protocol unit* (FRPU) to initiate the FRP signaling along the VPT [12]. The FRPU uses the implicit trunk VPC to transmit a reservreq cell to the next adjacent node.

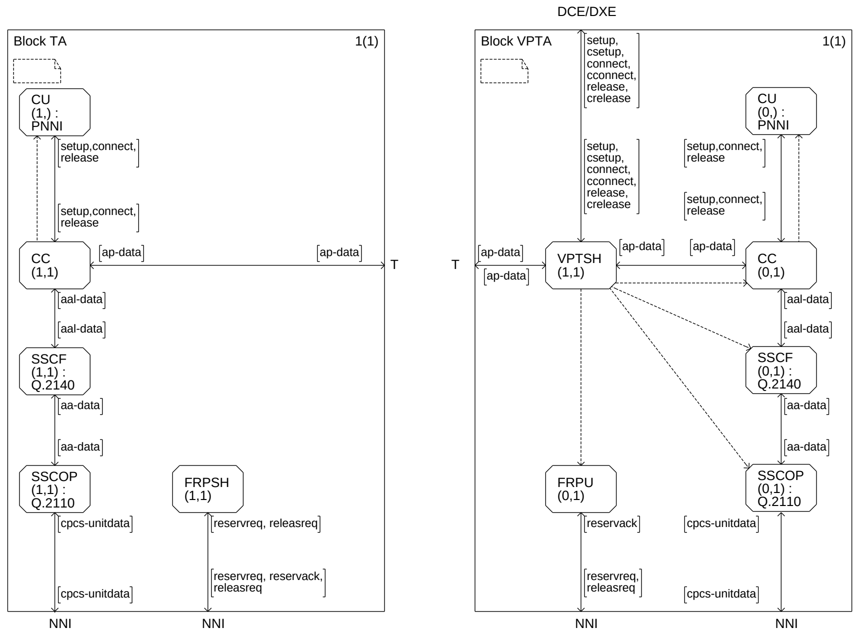

Figure 11: Trunk and Virtual Path Trunk Agents : signals are listed between brackets, plain arrows indicate signals direction, process names are followed by their initial and maximum number of instances and sometimes by the process type, dashed arrows indicate process creation (a dashed arrow is directed from the parent process to the child process), and channel identifiers are indicated outside the block frame.

6. The processing at transit nodes is much simpler. A reservreq signal is received by the *fast reservation protocol signaling handler* (FRPSH) entity which is responsible for reserving the new VPT bandwidth. The FRPSH then forwards the reservreq to the next adjacent node using the implicit trunk VPC.

   A csetup signal is just forwarded to the next adjacent node using the TA signaling link. When the csetup is received by the destination DXE at the other extremity of the VPT, the VPTA creates two peer CUs : one for the new VPT and one for the new VCC.

In the case a VCC is forced to be established hop-by-hop, the setup propagates through the TA signaling links and a new CU is created in the TA of both extreme nodes x and z.

3.2 More on Signaling Agents

We now describe in detail the TA and the VPTA. Figure 11 shows the specification of both agents in graphical-SDL. The *virtual path trunk signaling handler* (VPTSH), inside the VPTA, is the function that actually creates the protocol stack responsible for one of the VPT signaling link ; this protocol stack is composed of :

- a *connection coordinator* (CC) that creates the CUs of the connections carried by a particular VPT ;
- a *service specific coordination function* (SSCF) which serves as an interface to the underlying SSCOP ;
- a *service specific connection oriented protocol* (SSCOP) that provides, among other services, assured data transfer and keep-alive functions with the peer entity.

The DCE/DXE channel is used inside the node for signaling between a DCE and a DXE or between two DXEs. The VPTA uses the T channel to transmit and receive signals, such as csetup, through the TA signaling link. The NNI channel is used for signaling between nodes along the computed route.

Initially, the TA function contains one instance of all processes since we assume the trunk signaling link to be already established. However the VPTA function contains initially only the VPTSH since no pre-setup VPT signaling link is assumed.

To make the figure simpler, signals call-proceeding, connect-ack, alerting, status, status-enquiry, notify, validation-request, reservation-complete, reservation-denied, release-complete and releasreq-complete are not shown. Moreover, signal lists ap-data, aal-data, aa-data and cpcs-unitdata are not further detailed also for sake of simplicity.

SCONE specifications can be obtained in program-SDL as well as for SDT3.0 via anonymous ftp at `lrcftp.epfl.ch` in the directory `/pub/scone/`.

4 Conclusion

We consider the issue of increasing the number of connections that connection oriented networks, such as ATM, can handle. We describe one step that arises at reducing connection awareness inside the network. To that end, connections between the same pair of access nodes are grouped together and made indistinguishable inside the network. The concept of dynamic virtual path trunks is introduced as a support mechanism, and it is shown how virtual path links can be setup and maintained without additional round-trip delays.

Application to multicast connections is under study, and so is the application of VPTs to re-routing.

References

[1] J.-Y. Le Boudec, "The Asynchronous Transfer Mode : A Tutorial," *Computer Networks and ISDN Systems*, vol. 24 (4), pp. 279–309, May 1992.

[2] M. Peters, "Advanced Peer-to-Peer Networking : Intermediate Session Routing vs. High Performance Routing," *IEEE Systems Journal*, 1993.

[3] W. Stallings, *ISDN, An Introduction.* Macmillan Publishing Company, ISBN 0-02-415471-7, 1989.

[4] R. Perlmann, *Interconnections.* Addison-Wesley, ISBN 0-201-56332-0, 1993.

[5] C. Topolcic, *Experimental Internet Stream Protocol, Version 2, (ST-II)*. IETF, 1990.

[6] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zapalla, "RSVP : A New Resource ReSerVation Protocol," *IEEE Network*, September 1993.

[7] L. Gün and R. Guérin, "Bandwidth Management and Congestion Control framework of the Broadband Network architecture," *Computer Networks and ISDN Systems*, vol. 26 (1), pp. 61–78, 1993.

[8] J.-Y. Le Boudec and T. Przygienda, "A Route Pre-computation Algorithm for Integrated Services Networks," Technical Report 95/113, DI-EPFL, CH-1015 Lausanne, Switzerland, February 1995.

[9] O. Crochat, J.-Y. Le Boudec, and T. Przygienda, "Path Selection in ATM using Route Pre-Computation," Technical Report 95/128, DI-EPFL, CH-1015 Lausanne, Switzerland, May 1995.

[10] J.-Y. Le Boudec, E. Port, and L. Truong, "Flight of the FALCON," *IEEE Comm Mag*, pp. 50–56, February 1993.

[11] J.-Y. Le Boudec and T. Przygienda, "Routing Metric for Connections with Reserved Bandwidth," *EFOC-N*, 1994.

[12] P. Boyer and D. Tranchier, "A Reservation Principle with Applications to the ATM Traffic Control," *Computer Networks and ISDN Systems*, vol. 24, pp. 321–334, 1992.