The Use of Virtual Trunks for ATM/MPLS Control Plane Interworking

MFA Forum 9.0.0

MFA Forum Technical Committee
February 2006

Note: The user's attention is called to the possibility that implementation of the MPLS specification contained herein may require the use of inventions covered by patent rights held by third parties. By publication of this MPLS specification the MFA Forum makes no representation that the implementation of the specification will not infringe on any third party rights. The MFA Forum take no position with respect to any claim that has been or may be asserted by any third party, the validity of any patent rights related to any such claims, or the extent to which a license to use any such rights may not be available.

Editors:

Peter Busschbach
busschbach@lucent.com

Nikhil Shah
nbshah@lucent.com

For more information contact:

The MFA Forum

Suite 307
39355 California Street
Fremont, CA 94538 USA

Phone: +1 (510) 608-3997
FAX: +1 (510) 608-5917
E-Mail: info@mplsforum.org
WWW: http://www.mplsforum.org/

Full Notice

Revision History

Version	Change	Date
MFA 9.0.0	Initial version	February 2006

Table of Contents

1 Introduction

Many service providers derive significant revenues by offering and delivering services across dedicated Frame Relay and ATM networks. In the desire to reduce capital and operational expenditures, many of these same providers have embarked upon a strategy of convergence, where many per-service networks and their attendant technologies and services are migrated to a single IP/MPLS packet-switched network (PSN). It is envisaged that existing network services such as Frame Relay and ATM and emerging new services, including IP VPN, will operate over this multi-service converged PSN.

ATM networks employ dynamic signaling and routing protocols to expedite connection setup and recovery. PNNI is an example of one such protocol used to dynamically set up switched ATM connections across a native ATM network ([4]). When transitioning to a PSN, service providers will expect their existing ATM operations infrastructure to work with minimal modification. In particular, control protocols for routing and signaling must continue to work end-to-end.

The Internet Engineering Task Force (IETF) has developed pseudo-wire (PW) technology ([7]) to emulate services over a PSN; specifically, [2] defines how to emulate ATM connections over a PSN. These specifications, however, only describe the data plane behavior. They do not define how an ATM connection setup interacts with a PW setup.

The Virtual Trunk approach provides a mechanism in which pseudo wires function as trunks through which ATM switches can transparently establish and maintain Virtual Channel Connections (VCCs) and Virtual Path Connections (VPCs). This approach has two objectives:

- Minimize the amount of interactions between the ATM control plane and the MPLS control plane
- Enable a distributed interworking approach in which ATM control plane and PW control plane may be on different interworking devices. This enables separation of the MPLS network routing domains from the ATM routing domains.

Other methods enabling switched connection setup across an MPLS network have been defined in [6]and [9]. In these approaches each connection set up or tear down in the client control plane triggers the set up or tear down of a corresponding pseudo wire. The Virtual Trunk approach decouples control plane activity in the ATM and MPLS control planes, and pseudo wire set up or tear down is requested explicitly.

1.1 Purpose

This specification defines Virtual Trunks, their mapping to pseudo wires, and the way that ATM switches establish connections through them.

1.2 Scope and Requirements

The scope and requirements for the functions described in this document consist of:

- The architecture is general in nature. This document defines Virtual Trunks in the context of ATM control protocols, but analogous solutions can be developed for Frame Relay control protocols.
    - These protocols are not reduced to subsets of functionality.
    - There are no constraints on how PNNI hierarchy is deployed.

  - There is transparency for existing ATM and Frame Relay OAM mechanisms.

- This solution needs to be *scaleable* in a number of dimensions, to allow it to be used effectively in service provider networks, to support signaling rates commonly found in today's ATM networks, and to allow network growth and expansion. These dimensions include:

  - Performance: This approach must not reduce the signaling capabilities required in current networks.

  - Network resources: This approach must make effective use of network resources.

  - Future growth: As networks grow, the network resource usage to support this architecture must be similar to that required for the growth of current PNNI networks.

- This solution should be standards-based. It must use existing specifications and implementations to the greatest extent possible. In particular:

  - Minimal changes to current ATM and Frame Relay networks should be required to support the architecture.

  - Existing IETF mechanisms should be used whenever possible.

  - Any standards-based IP or MPLS PSN tunnel capable of carrying MPLS pseudo wires is not precluded.

  - Specific to MPLS, existing mechanisms and features commonly in use (especially for network resilience and efficiency) cannot be precluded from use. These mechanisms and features include traffic engineering, fast reroute, standby Label Switched Path (LSP), equal cost multi-path, penultimate hop popping, DiffServ support, the ability to multiplex multiple pseudo wires, and traffic types through the same PSN tunnels, etc.

  - The use of MPLS network resilience mechanisms is encouraged to allow ATM and FR VCs to survive switch or link outages within the MPLS network, if at all possible. ATM and FR-layer VC clearing should be avoided if at all possible.

- The solution should support multiple ATM service categories efficiently.

- The solution only supports point-to-point pseudo wires. This does not affect the ability to support point-to-multipoint ATM connections.

Definition of a Virtual Trunk control plane function based on dynamic signaling is for further study.

The following items are NOT within the scope of this document:

- Interworking between ATM and Frame Relay control planes [10].

- Interworking between ATM (or Frame Relay) and IP/MPLS control planes for setting up and managing switched connections between ATM (or Frame Relay) and IP/MPLS end-points and networks.

- L2TPv3 PWs [8]

1.3 Overview

Section 2 of this document contains definitions, references and abbreviations.

Section 3 introduces Virtual Trunks. Furthermore, it illustrates the control planes that play a role in the Virtual Trunk architecture. Finally, it provides reference models for the two implementation options that exist.

Section 4 defines Virtual Trunks and the associated data plane mechanisms in the case of a distributed implementation. Section 5 defines the same for an integrated implementation.

Section 6 defines how ATM connections are established through Virtual Trunks.

Section 7 defines the set up and maintenance of Virtual Trunks.

Section 8 summarizes the functions that need to be implemented on an existing ATM switch to support Virtual Trunks

Appendix A describes the impact that implementation of Virtual Trunks would have on existing ATM switches.

2 Definitions and Terminology

2.1 Definitions

Must, Shall or Mandatory — the item is an absolute requirement of this specification.

Should — the item is desirable.

May or Optional — the item is not compulsory, and may be followed or ignored according to the needs of the implementer.

2.2 Acronyms

Acronym	Definition
AINI	ATM Inter-Network Interface
ATM	Asynchronous Transfer Mode
BFD	Bidirectional Forwarding Detection
CCA	Client Connection Aggregation
IETF	Internet Engineering Task Force
IP	Internet Protocol
L2E	Layer 2 Edge
LDP	Label Distribution Protocol
LER	Label Edge Router

Acronym	Definition
LSP	Label Switched Path
ME	MPLS Edge
MP-BGP	Multi-Protocol BGP
MPLS	Multi-protocol Label Switching
NNI	Network-to-Network Interface
OAM	Operations, Administration and Management
PE	Provider Edge
PNNI	Private Network-to-Network Interface
PSN	Packet-Switched Network
PW	Pseudo wire
PWE	Pseudo-wire Emulation
PWE3	Pseudo-wire Emulation Edge-to-Edge (an IETF working group)
RSVP-TE	Resource Reservation Protocol – Traffic Engineering
RVPI	Relative VPI
UNI	User-to-Network Interface
VC	Virtual Connection
VCC	Virtual Channel Connection
VCI	Virtual Connection Identifier
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
VPN	Virtual Private Network
VT	Virtual Trunk
VTid	Virtual Trunk Identifier

2.3 Normative references

[1] *ATM Inter-Network Interface Specification Version 1.1*, ATM Forum af-cs-0125.001, September 2002

[2] Martini, L., et al. *Encapsulation Methods for Transport of ATM Cells/Frames Over IP and MPLS Networks.* IETF, draft-ietf-pwe3-atm-encap-10.txt. September 2005.

[3] Martini, L., et al. *Pseudowire Setup and Maintenance using LDP*. IETF, draft-ietf-pwe3-control-protocol-17.txt. June 2005.

[4] *Private Network-Network Interface Specification Version 1.1*, ATM Forum af-pnni-0055.002, April 2002.

[5] "Traffic Management Specification version 4.1," af-tm-0121.000, ATM Forum, March 1999

2.4 Informative references

[6] ATM Forum. af-cs-0197.000.pdf . *ATM-MPLS Interworking Signalling Specification Version 1.0.* August, 2003.

[7] Stewart, B., and Pate, P. *PWE3 Architecture.* IETF, RFC3985, March, 2005.

[8] "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC3931, Internet Engineering Task Force, March 2005

[9] MFA Forum *ATM and Frame Relay to MPLS control plane interworking: Client network-to-Client network*, MFA Forum 10.0.0, September 2006.

[10] "Network-to-Network FR/ATM SVC Service Interworking Implementation Agreement", FRF.18, Frame Relay Forum, Frame Relay Forum Technical Committee April 2000

3 Introduction to Virtual Trunks

This section introduces Virtual Trunks. It illustrates the control planes that play a role in the Virtual Trunk architecture. Finally, it provides reference models for two implementation options.

3.1 Pseudo-Wire Emulation and related terminology

In the pseudo-wire architecture ([7]), *Provider Edge* (PE) nodes encapsulate traffic received over an *Attachment Circuit* (AC) into a pseudo wire (PW) for transport over a *Packet-Switched Network* (PSN). PEs aggregate one or more PWs into a PSN Tunnel. In this architecture, the *client protocol* is the protocol that is emulated over the PW.

A *control plane* is the combination of routing and signaling protocols that switches use to determine how to forward traffic. In the case of a connection-oriented protocol like ATM, the control plane is used to set up, maintain and tear down connections.

The *client control plane* is the control plane used to manage the client protocol.

3.2 Virtual Trunks

Figure 1 shows the generic Virtual Trunk architecture. A Virtual Trunk is a logical connection between two client networks. Client Connection Aggregation (CCA) functional elements aggregate multiple client connections onto a Virtual Trunk. Pseudo-Wire Emulation (PWE) functional elements are responsible for the actions (such as encapsulation) required to emulate the Virtual Trunk over a PSN. There is a one-to-one correspondence between pseudo wires and Virtual Trunks.

From a PWE perspective, the Virtual-Trunk segment between CCA and PWE is an attachment circuit.

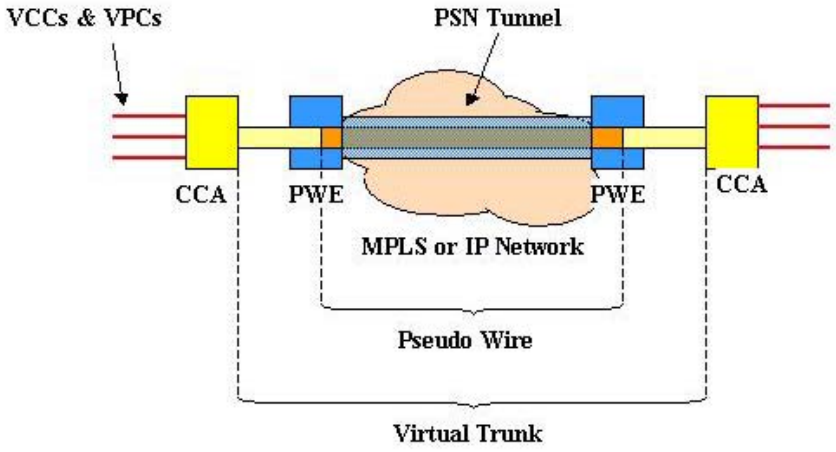

Figure 1: Generic Virtual Trunk Architecture

In the case of ATM, there already exists a mechanism that can be used for Virtual-Trunk capability: namely, virtual paths (VPs). An operator can establish a Virtual Path Connection (VPC) between ATM switches and emulate that VPC as a PW. Thus, Virtual Channel Connections (VCCs) can be transported across a PSN without exposing any knowledge of the individual VCCs to the PEs or to any other router in the MPLS network.

One drawback in associating a Virtual Trunk with a single VP is the inability to handle end-to-end client VPCs, i.e., VPCs cannot be carried within other VPCs. In order to extend the concept of Virtual Trunks to end-to-end client VPCs as well as to end-to-end client VCCs, this specification defines Virtual Trunks as consisting of a contiguous range of VPIs. Thus, Virtual Trunks form an encapsulation mode that is between VP mode (i.e. the encapsulation of a single VP) and port mode, in which case all traffic on a physical interface is encapsulated on one pseudo wire.

CCAs may exchange signaling and routing information through the Virtual Trunk and use this information to dynamically establish VCCs and VPC through that Virtual Trunk. The PWE functional elements are transparent to the VCCs and VPCs that are carried within the Virtual Trunk.

3.3 Control Planes

Figure 2 shows the control planes that play a role in the Virtual Trunk architecture. It also identifies the points where one control plane may interact with another.

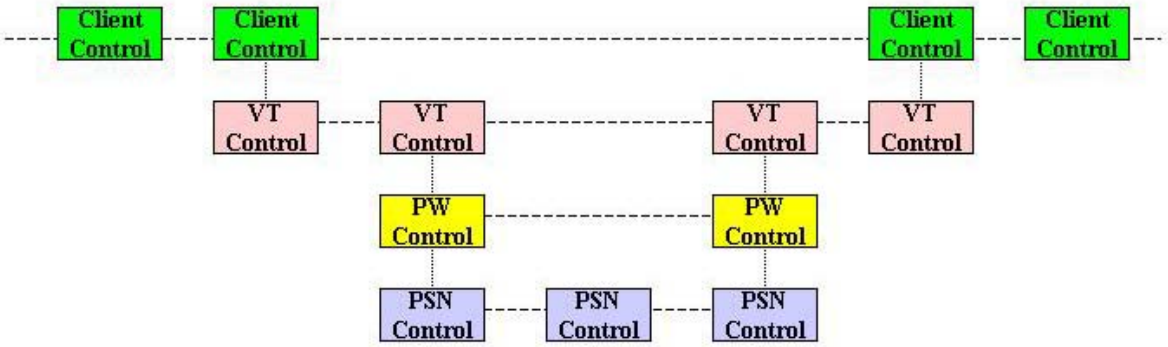

Figure 2 Control Plane Architecture

The control planes correspond to the aggregation stages that exist in the Virtual Trunk architecture.

- PSN Control serves to establish, maintain and tear down PSN Tunnels (e.g. MPLS LSPs) between PEs.
- Each PSN Tunnel can carry multiple PWs. PW Control serves to establish, maintain and tear down individual PWs.
- There is a one-to-one correspondence between a VT and a PW. However, because a VT can extend beyond the reach of a PW, and because there is VT specific information (unrelated to the corresponding PW) that may need to be exchanged between the two VT end-points, a separate control plane is required. The VT control plane serves to establish, maintain and tear down VTs.
- A VT can carry multiple client connections (e.g. ATM VPCs and/or VCCs). The client control plane serves to establish, maintain and tear down individual client connections.

The following sections discuss the individual control planes and the interaction between control planes.

3.3.1 Client Control Plane

The role of the client control plane is the establishment, maintenance and tear down of client connections. It may consist of routing protocols for topology and address discovery, and/or signaling protocols for connection setup and tear down.

In the case of Virtual Trunks, client control protocols are tunneled transparently through the VTs. This can include client routing protocols such as PNNI. The solution does not preclude the use of statically provisioned routing information or manually-configured connections, e.g. the solution can be used in conjunction with AINI, in which case routing information is statically configured and connections are set up dynamically by using the AINI signaling protocol.

3.3.2 Virtual Trunk Control Plane

This specification assumes that Virtual Trunks are configured statically. The use of dynamic mechanisms is for further study.

3.3.3 Pseudo-Wire Control Plane

This specification uses the PW control mechanisms defined in [3] as the initial pseudo-wire control plane in all scenarios. Future use of other PW control plane mechanisms is not precluded.

When the pseudo-wire control plane is used, at a minimum it is responsible for set up and tear down of PWs between LERs. This requires exchange of PW-related information between the PEs, such as the PW labels, the encapsulation used to transport client data in the PW, and some identification of the attachment circuits to which the PW is bound at the PEs.

3.3.4 PSN Control Plane

The PSN Control Plane is outside the scope of this specification.

3.3.5 Interaction between control planes

The VT architecture is designed with the basic principle of keeping the control planes independent. Therefore, one control plane could evolve without any implications for the other control planes. Nevertheless, at various points in the network, one control plane may interact with the control plane above or below itself. For example, based on traffic demands in the client control plane, an ATM switch that functions as an end-point of a VT could request the VT control plane to allocate more bandwidth.

3.4 Implementation Options

The CCA and PWE functionality introduced above could be located in two physically separate devices or as two logical components in a single physical device.

3.4.1 Distributed Implementation

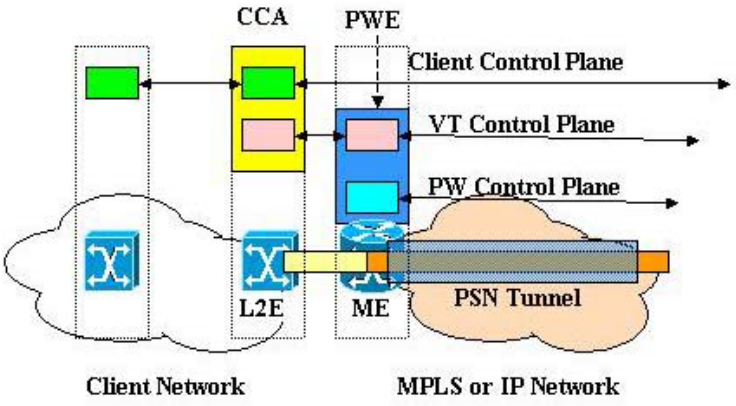

Figure 3: Distributed Implementation

Figure 3 shows a distributed implementation. In this implementation, CCA functionality is located in a Layer 2 Edge (L2E) device and PWE functionality is located in an MPLS Edge (ME) device.

Thus, the distributed implementation of Virtual Trunks offers a complete solution for end-to-end operation of a dynamic client control plane in such a way that:

- The L2E devices support the client control plane but do not need to support Pseudo-Wire Emulation (PWE), IP or MPLS
- The MEs support PWE, IP and MPLS, but do not need to support the client control plane.

To maintain the current levels of scaling expected in an ATM core (e.g., call setup rates and number of VCs) and not burden the ME with maintaining per-VC state, the MEs should be able to determine the Virtual Trunk membership and QoS treatment of a given ATM connection without participating in ATM connection setup.

3.4.2 Integrated Implementation

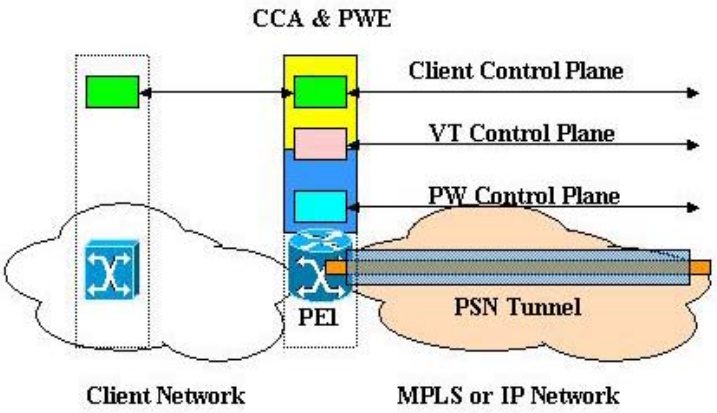

Figure 4 Integrated implementation

Figure 4 shows an integrated implementation. In this implementation, CCA and PWE functionality are integrated into one physical device, the PE.

The two implementations are interoperable. A Virtual Trunk can be established between a PE, which supports both CCA and PWE functionality, and an ME – L2E pair where the functionality is distributed, as shown in Figure 5.

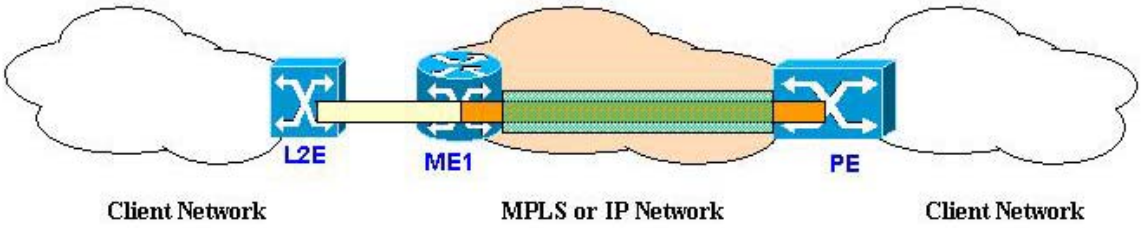

Figure 5 Combination of a distributed and an integrated implementation

4 Virtual Trunk data plane procedures – distributed implementation

This section defines ATM Virtual Trunks and associated data plane mechanisms in the case of a distributed implementation (see 3.4.1) where the interface between L2E and ME is an ATM interface.

4.1 Definition of Virtual Trunks on an ATM interface

On an ATM interface, a Virtual Trunk is a contiguous set of VPIs. Thus, the VPI space on an ATM physical interface is divided into sub-sets, each representing a VT.

Two parameters are now introduced: Virtual Trunk identifier (VTid) and Relative VPI (RVPI). The VTid identifies the range of VPIs that constitute the VT. The RVPI identifies a "relative" VPI within that range, i.e. if a certain VT is defined as the range of VPIs between a lower bound L and an upper bound U, i.e. VT = {L, U}, then for a VP, with VPI=K, within that trunk, RVPI = K - L. For example, if a Virtual Trunk is defined as the set of VPIs {32, 63}, then VPI = 39 corresponds with RVPI = 7.

If client control protocols are carried within a VT, they are transported in the lowest VPI within the range of VPIs that constitute that VT. If *x* is the lowest VPI within the VT's range of VPIs, then VPI/VCI = *x*/5 is used for ATM signaling and VPI/VCI = *x*/18 for PNNI routing. By definition, the RVPI of the VP with the lowest VPI equals 0. Therefore, expressed in terms of RVPI, the ATM signaling channel is *0*/5 and the PNNI routing channel is *0/18*.

Note: PW emulation of a single VP is a special case of a Virtual Trunk where the VPI range consists of only one value. All mechanisms defined for Virtual Trunks also apply to this case.

4.2 Mapping a Virtual Trunk into a Pseudo Wire

An ME maps a Virtual Trunk into a pseudo wire, using N-to-one cell mode encapsulation, as defined in [2]. There is a one-to-one correspondence between a VT and a PW.

When an ME receives an ATM cell, it determines the VPI range (and therefore the VT) to which the cell's VPI belongs. Based on the association between VTs and PWs, the ME thus derives the PW Label from the VPI of the incoming ATM cell.

If the cell's VPI does not match any of the VPI ranges known to the ME, and if it does not match the VPI of any otherwise provisioned VP, the ME shall drop the cell.

For efficiency reasons, the ME may encapsulate multiple ATM cells in a single MPLS frame. The ME shall only concatenate cells that belong to the same ATM service category and, in cases where the Cell Loss Priority (CLP) bit is significant, that have the same CLP. MEs may reorder cells belonging to different service categories due to parallel (per-service category) concatenation of cells into MPLS frames. Cells belonging to any one particular client VPC or client VCC shall remain in order with respect to other cells belonging to the same client VPC or client VCC.

Since cells on a Virtual Trunk may arrive out of order at the destination switch, sequence numbers shall not be used in the case of Virtual Trunk encapsulation.

4.3 VPI Range as local parameter

For the overall operation of an ATM-MPLS network, it is necessary that the VT's VPI Range is a local parameter that can be modified by the MEs. Consider the example shown in Figure 6.

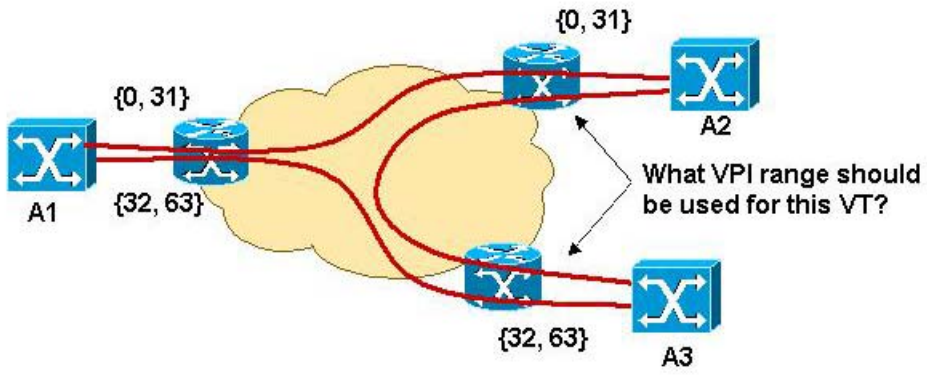

Figure 6 Example of the problem that arises when VPI Ranges cannot be modified by MEs

Assume that there are three ATM switches, A1, A2 and A3. Assume that the range of VPIs {0, 31} is allocated to the VT between A1 and A2 and the range {32, 63} to the VT between A1 and A3. If the VPI range has to be the same at both ends of the VT, i.e. if the MEs cannot modify the VPI range, the VPI range allocated to a VT between A2 and A3 would have to fall outside {0, 31} because those values are already used by A2, and outside {32, 63} because those values are already used by A3.

In general, the process of allocating VPI ranges to VTs becomes complex and inefficient if the MEs cannot modify the VPI values. In the example given above, the VPI space would be used more efficiently if the VT between A2 and A3 were allocated different VPI ranges on different switches, e.g. by {32, 63} on A2 and by {0,31} on A3, with the MEs translating a VPI in the {0, 31} range to the corresponding VPI in the {32, 63} range, and vice versa.

VPI translation must be done in such a way that the RVPI remains constant (the rationale for this requirement follows from section 6), e.g. VPI = 39 in the {32, 63} range corresponds to RVPI = 7 and will therefore be translated to VPI = 7 in the {0,31} range.

The translation of VPI values by MEs is purely a matter of replacing one VPI value with another. MEs do not need to maintain the VP state to perform this process.

4.4 Translating VPI ranges at an ME

A general description of VPI range translation is shown in Figure 7.

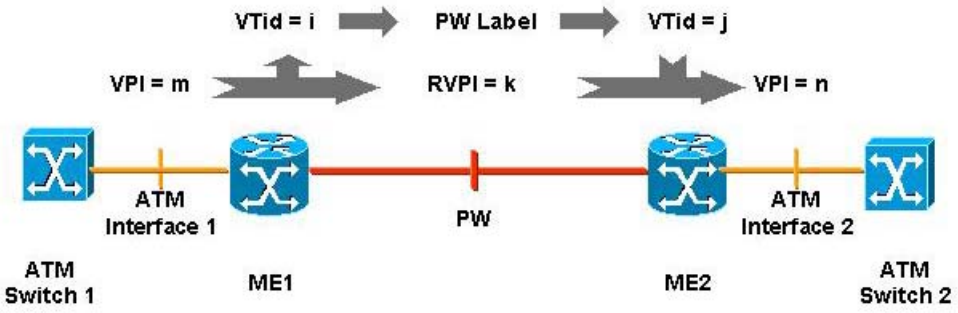

Figure 7 Translating the VPI Range

Assume that the range of VPIs {0, 31} is used on ATM interface 1, and {32, 63} on ATM interface 2. When ME1 receives cells with VPI = 7, it determines that the RVPI = 7 (i.e. 7-0). When ME2 receives the MPLS packet with RVPI = 7, it translates it into the corresponding VPI: 39.

In this particular example, note that ME1 and ME2 actually do not have to touch the RVPI value. They only need to modify the higher-order bits as illustrated in Figure 8

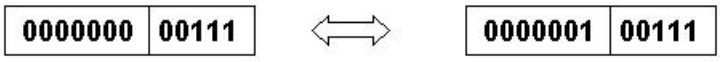

Figure 8 Translating VPI = 7 to VPI = 39

However, in the more general case, in which the choice of a VPI range is unconstrained, ME1 and ME2 will have to translate a VPI on interface 1 into the corresponding RVPI, which in turn is translated into the VPI on interface 2. In that case the procedure is as follows:

Consider again the model shown in Figure 7. When ME1 receives ATM cells from ATM switch 1, it must do the following:

- Derive the VTid and RVPI from the VPI in the ATM cell header
- Select the PW Label based upon the VTid
- Replace the original VPI value with the RVPI value

When ME2 receives an MPLS packet from ME1, it must do the following:

- From the PW label, derive the VTid that is to be used on the interface with ATM switch 2

- Based upon the VTid and RVPI value (which is the VPI value of the de-capsulated ATM cells), calculate the VPI value that is to be used on interface 2. Write this value into the VPI field(s) of the header(s) of the ATM cell(s) that are to be sent to ATM switch 2

4.5 Bit-aligned VTs.

The general definition of VTs, as explained in sections 4.1 and 4.2, with VPI ranges selected without any constraints, may be difficult to implement on MEs. As alluded to in section 4.4, the ME implementation could be simplified if the boundaries of VPI ranges are chosen in such a way that the RVPI coincides with the lower bits in the VPI field. Figure 9 illustrates this concept.

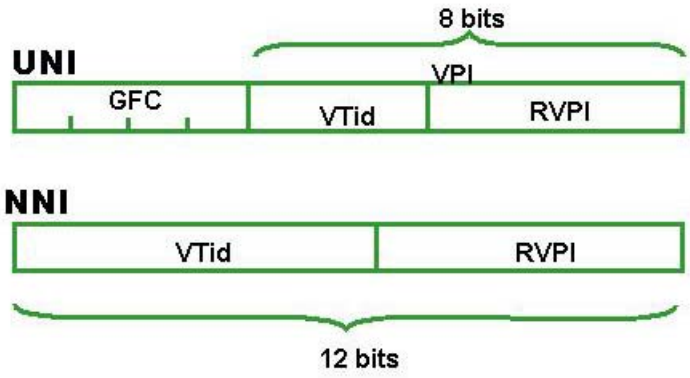

Figure 9 Bit-aligned VTs

When the lower bit values coincide with the RVPI, the upper bits can be used to identify the VT. For example, if a VT is defined on an NNI interface as the VPI range {32, 63}, then, for all VPIs within this range, the lower 5 bits are equal to the corresponding RVPI, and the upper bits are equal to 0000001. Hence, the ME could take a bit mask of the lower bits of the VPI to derive the RVPI, and a bit mask of the upper bits of the VPI to derive the VTid. Since the VTid maps directly to a PW label, the process of mapping ATM cells to VTs, and subsequently to PWs, becomes straightforward.

Bit-aligned VTs, in which the lower k bits coincide with the RVPI, result in a division of the VPI space on a specific interface into blocks of size 2^k. The lower bound of each range equals $n*2^k$, the upper bound $((n+1)*2^k)-1$ (where n is an integer).

An ME must support bit-aligned VTs. An ME may be able to support VTs that consist of more flexible ranges of VPIs.

- What this means for the operator:

  The operator decides, per physical interface, how many VPs need to be supported on Virtual Trunks; e.g. if VTs should support 32 VPs, the operator would provision the ME to recognize the lower 5 bits

as RVPI and the upper 7 bits (or 3 in the case of a UNI interface) as VTid. The operator would provision the ATM switch to allocate 32 VPs to each Virtual Trunk.

- What it means for the ME:

  The operator configures the ME to interpret the lower 5 bits as RVPI and the upper bits as VTid. For each ATM cell it receives from its adjacent ATM switch, it performs a bit mask to derive the VTid and maps that to the PW label. Similarly, for each (encapsulated) ATM cell the ME receives through the MPLS core, it maps the PW Label to the VTid and writes this value into the corresponding bits of the VPI of the cell it sends to its adjacent ATM switch.

- What it means for the ATM switch:

  The ATM switch is not aware of the notion of bit masks. It should be possible for the operator to configure the ATM switch with VPI ranges in chunks chosen by the operator; e.g. the operator should be able to force the ATM switch to choose the following ranges: {0, 31}, {32, 63}, {64, 95}, etc. Apart from this, the ATM switch behaves in the exact same way as described in the previous sections.

In general, an operator will select the same VPI range size per VT on every physical interface that connects ATM switches to the MPLS core. However, under certain circumstances (e.g. if one interface is a UNI and the other is an NNI) an operator might want to allocate a larger range on one interface than on another; e.g., as illustrated in Figure 10 it is possible that the operator allocates 32 VPIs on the interface between ATM switch 1 and ME 1 and 16 VPIs on the interface between ME2 and ATM switch 2.

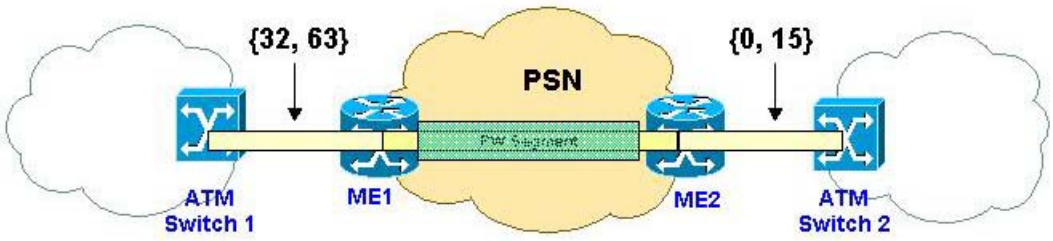

Figure 10 Reconciling differently-sized VPI ranges

To cope with this situation, the range of usable VPIs within a VT must be negotiated between the two end-points of the VT. This can be achieved through manual or dynamic techniques that are defined in the Virtual Trunk control plane.

4.6 QoS

A standard VPC has a traffic contract that includes a service category and a conformance definition [5]. In this specification the combination of service category and conformance definition is referred to as a

"Service Class". The QoS treatment[1] provided to a packet as it traverses the MPLS core must be sufficient to meet the packet's QoS requirements. A VT may have one of two QoS characteristics: single-QoS or multi-QoS.

4.6.1 Single-QoS VTs

A Single-QoS VT is one in which all constituent VPCs receive the same QoS treatment in the MPLS core. Through provisioning or otherwise, the ME will associate a VT with a certain QoS treatment, and it will map the corresponding PW to a PSN tunnel that provides the required QoS treatment.

Support for Single-QoS Virtual Trunks is **mandatory**.

An operator deploying a Single-QoS VT has two options:

1. Combine VPCs of more than one Service Class into the VT. In this case the operator shall associate with the VT a QoS behavior that is sufficient to meet the most stringent QoS requirements of any constituent VPC even if some of the VPCs or VCCs on that VT have less stringent needs, e.g. treat the entire VT as if it carries CBR traffic even if that is true for only a subset of the VPCs
2. Associate a VT with a specific Service Class and force the ATM switch to transport only VPCs and VCCs with that Service Class on the VT.

Option 2 results in more efficient use of the MPLS core. However, it may require the operator to set up multiple VTs between two ATM switches and allocate VPI ranges per class of service.

4.6.2 Multi-QoS VT

4.6.2.1 Definition

A Multi-QoS VT is a VT whose constituent VPCs do not all share the same Service Class, and in which VPCs of different Service Classes receive appropriately different QoS treatment in the MPLS core. Within a Multi-QoS VT, the QoS treatment that a packet receives depends upon the Service Class of the VPC to which it belongs.

Support for Multi-QoS Virtual Trunks is optional.

In Multi-QoS VTs, contiguous ranges of RVPI per Service Class are established (i.e. provisioned or signaled). The association between a range of RVPI and a Service Class is established at the ATM edge. The association between a range of RVPIs and an MPLS network QoS treatment is established at the MPLS edge. In the extreme, a range could be as small as a single RVPI.

This specification places no restrictions on the partitioning of the RVPI space within a Multi-QoS VT; for example, an RVPI range need not be a power of 2, and two RVPI ranges need not be equal in size. Following the initial partitioning of the VT's RVPI space, it is permitted to provision additional ranges from the unused RVPI space as required. A given Service Class may be associated with more than one RVPI range.

---

[1] The term "QoS Treatment" does not include drop precedence. Individual packets accorded the same QoS treatment may be differentiated on the basis of drop precedence.

4.6.2.2 Mechanisms at the Ingress L2E

The ATM edge shall switch ATM VPCs, originating in the ATM network, of a given Service Class and PNNI Designated Transit List (DTL), onto an available VPI within the RVPI range supporting that Service Class for the appropriate VT. The mechanism for selecting the appropriate RVPI and VT is device-specific.

4.6.2.3 Mechanisms at the Ingress ME

The ME shall map VPCs falling within a particular RVPI range to the appropriate MPLS network QoS treatment.

5 Virtual Trunk data plane procedures – integrated implementation

In the case of an integrated implementation (see section 3.4.2), the interface between CCA and PWE is internal to a PE. Since there is not an ATM interface carrying multiple virtual trunks in this configuration, there is no need to partition VPI address space. This has the following consequences:

- The number of VPs per VT that a PE can support is only limited by the size of the VPI space or by internal limitations of the PE itself.
- There is no need for VPI translation between CCA and PWE. Therefore, RVPI = VPI

5.1 Encapsulation

The procedures defined in section 4.2 apply.

5.2 ATM Control Interface

ATM signaling and routing channels are carried in VPI=0.

5.3 QoS

In case a PE that supports both CCA and PWE functionality interfaces with an L2E – ME pair in which the functionality is distributed, the QoS procedures defined in section 4.6 apply, i.e. the VT is either a single-QoS VT or a multi-QoS VT, in which case the different service classes are identified by specific RVPI (i.e. VPI) ranges.

In case both end-points of a VT are PEs that support both CCA and PWE functionality, the PEs could carry multiple service classes over a VT in a more flexible fashion. Since each PE supports both client control plane and Pseudo-Wire Emulation, it is, in contrast to section 4.6.2.1, not necessary to use contiguous ranges of RVPIs per Service Class. Instead, the service class of each individual VCC or VPC (as negotiated during client connection setup) can be directly mapped to the appropriate MPLS network QoS treatment. Support of this functionality is optional.

5.4 Usable VPIs within a VT

The range of usable VPIs within a VT is negotiated between the two end-points of the VT (refer to the last paragraph of section 4.5). In case a PE that supports both CCA and PWE functionality interfaces with an L2E – ME pair in which the functionality is distributed (as shown in Figure 5), the range of VPIs that can be used to establish VCCs and VPCs through the VT is limited by the size of the VPI range

configured for the distributed side of the VT. The VPI range on the integrated side of the VT should always start at 0.

6 Signaling Procedures between ATM switches Connected Through a VT

When a VT is established between two ATM switches, those switches may use the standard routing and signaling channels in the VP, with RVPI = 0 to exchange ATM routing and signaling information. They use the signaling channel to set up and/or release VPCs and VCCs in the Virtual Trunk.

Since the VPI Range assigned to a VT is a local parameter, the VPI value used for a specific connection on one switch may differ from the VPI value used on the other switch. However, the corresponding RVPI value is identical on both switches. Therefore, the two ATM switches shall use the RVPI value when setting up specific VPCs or VCCs.

The PNNI spec [4] introduces the Virtual Path Connection Identifier. As explained in 6.5.2.2.3 of [4], the VPCI only has significance with regard to a given signaling virtual channel. Its value is not necessarily equal to the actual VPI of the connection, but the two interfacing ATM switches understand the relationship between the VPCI and the actual VPI values. AINI is similar to PNNI in most respects, and uses the VPCI in the same way [1].

Therefore, when ATM switches use PNNI or AINI to set up VPCs and VCCs through a VT, they shall use regular signaling procedures, with the RVPI value used as the VPCI. If signaling protocols other than PNNI or AINI are used, similar mechanisms shall be used. The exact definition is protocol-dependent, and is not covered in this IA.

Upon connection establishment, each ATM switch shall translate – as described in section 4.4 - the RVPI into the actual VPI value that it will use in the data plane.

7 Virtual Trunk Setup and Maintenance

This specification assumes that Virtual Trunks are configured statically. The use of dynamic mechanisms is for further study.

7.1 Virtual Trunk Configuration

In the case of manual configuration, the operator needs to provision the following parameters for each VT:

7.1.1 Parameters relevant to an L2E and its adjacent ME in the case of a distributed implementation

- VPI range (lower-bound VPI and upper-bound VPI)
    - In case an ME supports only bit-aligned VTs, it might need to be provisioned with the VTid (see 4.5)

7.1.2 Parameters relevant to the CCA functional elements at either end of a VT

- The maximum number of VPs that can be established on the VT

  - It is possible that for a specific VT the actual maximum number is lower than the difference between upper-bound VPI and lower-bound VPI. (See 4.5 and 5.4)

- If applicable, the ATM routing and signaling protocols to be used on the VT

7.1.3 Parameters relevant to the PWE functional elements

The two PWE functional elements need to establish a pseudo wire to carry the VT. The PWEs shall use the procedures defined in [3] as follows.

- The MEs shall use the PWid FEC element in the LDP signaling messages.
- The MEs shall use a PW Type value 0x001E (ATM Virtual Trunk)

The operator shall provision each PWE functional element with:

- The loopback address of its peer
- The PWid value that identifies the PW

7.1.4 Parameters relevant to both the CCA and the PWE functional elements

- QoS parameters associated with the Virtual Trunk

7.2 Virtual Trunk OAM

Virtual Trunks emulate a direct interface between two switches. If two ATM switches are connected through a direct connection, e.g. a SONET or SDH path, a failure in the SONET/SDH layer will result in a failure notification (Path AIS) to the ATM switch downstream of the failure. In turn, the ATM switch will generate F4 AIS for all affected VPCs and F5 AIS for all affected VCCs. In addition, in case a signaling protocol, such as PNNI, was used for connection establishment, the ATM switch will perform the actions specified by that protocol in order to restore or release the affected connections.

Similarly, if a failure brings down a Virtual Trunk (i.e. after other protection mechanisms like MPLS Fast Reroute have been exhausted without clearing the failure), the ME downstream of the failure will notify the adjacent ATM switch. To signal a persistent VT failure on the ATM interface between ME and ATM switch, an ME shall send F4 AIS on the VP with the lowest VPI of the range (i.e. RVPI=0, VCI=4). Since this VP is not an end-to-end VPC, but merely a collection of individual VCCs (the VCC used for signaling being one of them), reception of F4 AIS will unambiguously signal a VT failure. Any other failure would either lead to F5 AIS for the individual VCCs on this VP, or to a failure notification in a server layer like SONET/SDH.

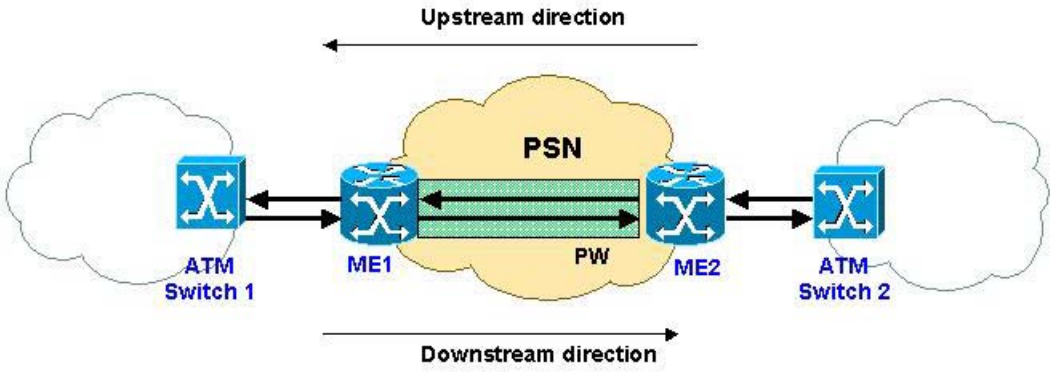

Figure 11 OAM Reference model

7.2.1 Defects and consequent actions

Defects could occur in different places. Please refer to Figure 11. This section considers *uni-directional* failures that affect the transmission in the *downstream* direction.

1. At ME1
   When ME1 detects a failure that affects an entire Virtual Trunk, e.g. a failure of the physical interface between ATM Sw1 and ME1, which cannot be restored through local protection switching, it should generate F4 AIS on RVPI=0 of the affected Virtual Trunk towards ME2. If ME1 does not generate AIS, it shall send a PW Status notification for the pseudo wire that carries the affected Virtual Trunk. The PW Status shall indicate "Local Attachment Circuit (ingress) Receive Fault". (See [3] for more information on the use of PW Status TLV in PW control)

2. At ME2
   When ME2 detects a defect that affects the pseudo wire related to a specific Virtual Trunk, which cannot be restored through local protection switching, ME2 shall generate F4 AIS on RVPI=0 towards ATM switch 2.

   ME2 could detect the defect itself, e.g. when it detects a SONET failure; when it receives a PathErr for the PSN tunnel (RSVP-TE built LSP) that carries the PW; or when it detects a PW failure. Alternatively, ME2 might be informed about the defect by ME1 through a PW Status message. Finally, ME1 and ME2 might run a Fault Detection protocol like BFD or Y.1711 through the PW. Failure to receive the respective periodic messages will lead ME2 to declare a defect.

   Note: ME2 transparently forwards all ATM cells carried over the VT. Therefore, if ME1 inserts F4 AIS on RVPI=0, ME2 transparently forwards it to ATM switch 2.

3. At ATM switch 2
   Upon reception of F4 AIS on RVPI=0, or if ATM switch 2 detects a failure on the interface between ME2 and ATM switch 2 that affects an entire Virtual Trunk and which cannot be restored through local protection switching, ATM switch 2 shall consider the VT to be down. It shall generate, in the downstream direction, F4 AIS for all affected VPCs and F5 AIS for all affected VCCs. In addition, it

shall generate F4 RDI on RVPI=0 in the reverse (upstream) direction. Finally, in case a signaling protocol (such as PNNI) was used for connection establishment, the ATM switch will perform the actions specified by that protocol in order to restore or release the affected connections.

Upon the reception of F4 RDI on RVPI=0, ATM switch 1 shall not inject an ATM fault notification like AIS or RDI on the VPCs and VCCs that are affected by the VT failure. Instead, the downstream end-point of each affected VCC or VPC is expected to generate an upstream RDI in response to the downstream AIS signal generated by ATM switch 2.

In the case of an integrated implementation, where ME2 and ATM switch 2 are integrated into a single network element, that network element shall be able to interpret both F4 AIS on RVPI=0 and PW Status Notification, indicating "Local Attachment Circuit (ingress) Receive Fault", as indications that the Virtual Trunk is down. It shall generate, in the downstream direction, F4 AIS for all affected VPCs, and F5 AIS for all affected VCCs. In addition, it shall generate F4 RDI on RVPI=0 in the reverse (upstream) direction. Finally, in case a signaling protocol (such as PNNI) was used for connection establishment, the ATM switch will perform the actions specified by that protocol in order to restore or release the affected connections.

In case a failure causes transmission in both directions of a VT to be interrupted, the F4 RDI on RVPI=0 sent by one ATM switch will obviously not be received by its peer. However, in that case both ATM switches will detect the VT failure independently. They shall both act as described above.

8 Summary of requirements

This specification describes two implementation options. Since the two options are interoperable, a manufacturer might decide to implement one or the other. In the case of the distributed implementation (section 3.4.1), a product that supports the L2E functionality must be complemented with a product that supports the ME functionality in order to get a complete solution.

The previous sections address several aspects of the impact of Virtual Trunks on L2E, ME and PE. This section summarizes the requirements for each of these network elements

8.1 Requirements for the distributed implementation

8.1.1 Requirements for the Layer 2 Edge Device

An L2E shall have the ability to treat a Virtual Trunk as a logical interface over which it can set up VPCs and VCCs.

A Virtual Trunk is defined by a contiguous range of VPIs. The L2E must allow the operator to specify specific constraints on the VPI range allocated to a particular VT. In addition, an L2E may allocate VPI ranges autonomously, based on considerations that are outside the scope of this specification.

On each VT, the lowest VPI within the range of VPIs that constitute a VT carries the signaling and routing channels. In the case of PNNI routing and signaling, VPI/VCI = *x*/5 is used for signaling and VPI/VCI = *x*/18 for routing, where *x* is the lowest VPI within the VT's range of VPIs. Which routing and/or signaling protocols are used on a particular VT can be determined using manual or dynamic techniques defined in the Virtual Trunk control plane.

To set up VPCs and/or VCCs through the VT using PNNI or AINI signaling, two L2Es that are connected through a VT shall use the RVPI value as VPCI in the signaling messages.

When a new VPC/VCC has been established for a specific RVPI, an L2E shall use the actual VPI (that corresponds with that RVPI) for subsequent processing of ATM cells (switching, queuing, performance monitoring, etc).

The number of usable VPIs within a VT is not necessarily equal to the size of the VPI range allocated to that VT. Instead, the maximum number of VPIs is determined by manual or dynamic techniques defined in the Virtual Trunk control plane. If L is the lowest VPI within the range of VPIs that constitute a VT and M is the maximum number of VPIs usable within the VT, an L2E shall use the range {L, L+M-1} as the range of valid VPIs on that VT.

L2Es shall use regular Connection Admission Control procedures based on the bandwidth allocated to the VT. The amount of bandwidth allocated to the VT is determined by manual or dynamic techniques defined in the Virtual Trunk control plane. Alternatively, an L2E may autonomously modify the amount of bandwidth allocated to the VT based upon actual traffic demands or other considerations that are outside the scope of this specification.

When an L2E receives F4 AIS for the VPI that corresponds to RVPI = 0, it shall interpret that as an indication that an upstream failure in the VT occurred. It shall consider the VT as failed and shall generate F4 AIS for all affected VPCs, and F5 AIS for all affected VCCs. In addition, in case a signaling protocol (such as PNNI) was used for connection establishment, the ATM switch will perform the actions specified by that protocol in order to restore or release the affected connections. All other failure notifications (i.e. F4 AIS for VPIs other than the one that corresponds to RVPI=0 or F5 AIS for any VCC on the VT) shall be treated according to regular ATM standards.

8.1.2 Requirements for the ME

An ME must be able to map an ATM cell it receives from an adjacent L2E to a Virtual Trunk and thus to a pseudo wire. At a minimum, an ME must support bit-aligned VTs, where specific bit positions in the VPI field directly map to VTid and RVPI, as shown in Figure 9. An ME may be able to recognize VTs that consist of more flexible ranges of VPIs.

An ME shall use N-to-one cell mode encapsulation to encapsulate ATM cells. It may support cell concatenation. When using concatenated cell mode, an ME must ensure that cells that belong to the same VPC or VCC remain in order. Furthermore, the ME must ensure that only cells that require the same QoS treatment in the MPLS core are concatenated in the same packet. As a consequence, for Service Classes for which the CLP bit is relevant, only cells with the same CLP bit value may be concatenated in the same packet.

An ME must be able to map the PW Label of an MPLS packet it receives through the MPLS core to a Virtual Trunk. Furthermore, an ME must be able to forward ATM cells to the destination ATM switch with a new VPI value based on that Virtual Trunk id and the RVPI. At a minimum, an ME must be able to do this for bit-aligned VTs, where specific bit positions in the VPI field directly map to VTid and RVPI, as shown in Figure 9. An ME may be able to do this for VTs that consist of more flexible ranges of VPIs.

An ME shall generate F4 AIS on the VP that corresponds to RVPI = 0 when it detects a persistent failure on the VT.

8.2 Requirements for the PE in the case of an integrated implementation

A PE shall have the ability to treat a Virtual Trunk as a logical interface over which it can set up VPCs and VCCs.

A PE shall use N-to-one cell mode encapsulation to encapsulate ATM cells. It may support cell concatenation. When using concatenated cell mode, a PE must ensure that cells that belong to the same VPC or VCC remain in order. Furthermore, a PE must ensure that only cells that require the same QoS treatment in the MPLS core are concatenated in the same packet. As a consequence, for Service Classes for which the CLP bit is relevant, only cells with the same CLP bit value may be concatenated in the same packet.

On each VT, VPI=0 carries the signaling and routing channels. In the case of PNNI routing and signaling, VPI/VCI = *0*/5 is used for signaling and VPI/VCI = *0*/18 for routing. Which routing and/or signaling protocols are used on a particular VT can be determined using manual or dynamic techniques defined in the Virtual Trunk control plane.

To set up VPCs and/or VCCs through the VT using PNNI or AINI signaling, two L2Es that are connected through a VT shall use the actual VPI value as VPCI in the signaling messages.

The maximum number of VPIs that can be used within a VT is determined by manual or dynamic techniques defined in the Virtual Trunk control plane. If M is the maximum number of usable VPIs, a PE shall use the range {0, M-1} as the range of valid VPIs on that VT.

A PE shall use regular Connection Admission Control procedures based on the bandwidth allocated to the VT. The amount of bandwidth allocated to the VT is determined by manual or dynamic techniques defined in the Virtual Trunk control plane. Alternatively, a PE may autonomously modify the amount of bandwidth allocated to the VT based upon actual traffic demands or other considerations that are outside the scope of this specification.

When a PE receives F4 AIS on VPI=0, or if it receives a PW Status notification indicating "Local Attachment Circuit (ingress) Receive Fault", it shall interpret that as an indication that an upstream failure in the VT occurred. It shall consider the VT failed and it shall generate F4 AIS for all affected VPCs and F5 AIS for all affected VCCs. All other failure notifications (i.e. F4 AIS for VPIs other than the one that corresponds to RVPI=0 or F5 AIS for any VCC on the VT) shall be treated according to regular ATM standards.

Annex A: Support of both Virtual Trunks and individual VCs on the same ATM interface

NORMATIVE

Optionally, ATM switches and MPLS edge devices may support both Virtual Trunks and other methods of control-plane interworking, such as those defined in [6] or [9]. An example is shown in Figure 12

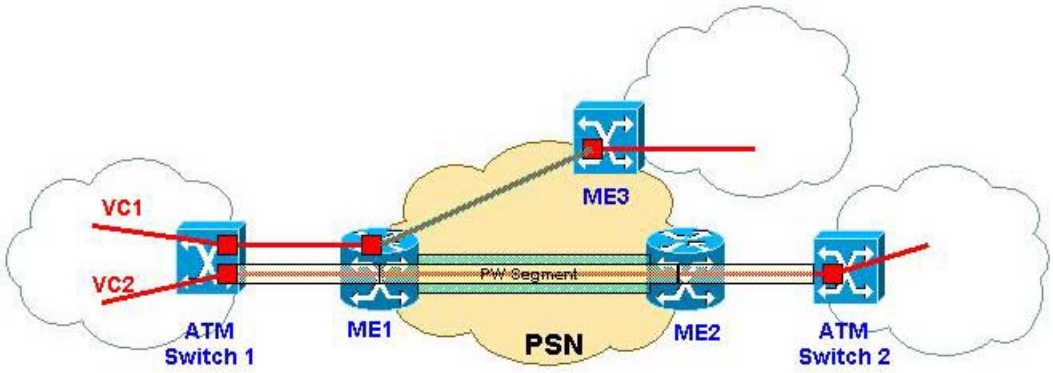

Figure 12 Network configuration

ME1 in Figure 12 functions as an LER in the architecture defined in [6] or [9]and as an ME in the Virtual Trunk distributed implementation as defined in section 3.4.1.

If a device chooses to support both Virtual Trunks and traditional VCs on the same ATM interface, then ATM switch 1 and ME1 must meet the following conditions:

1. VPI=0 may not be part of any of the VPI ranges allocated to the Virtual Trunks on the interface. Instead, VPI=0/VCI=18 is used to carry PNNI routing information and VPI=0/VCI=5 is used for PNNI signaling messages related to the establishment of individual VPCs and VCCs.
2. When assigning a VPI value to an individual VPC or VCC, both ATM switch 1 and ME1 must avoid the ranges that have been assigned to Virtual Trunks or that have been reserved for VT use.

Figure 12 shows two VCs. The red squares identify the nodes that participate in the establishment of each VC.

VC1 is an individual VC. To establish VC1, signaling messages are exchanged among ATM switch 1, ME1, and ME3. As part of the establishment of VC1, a pseudo wire is set up between ME1 and ME3 so that there is a 1-1 mapping between VC and PW.

VC2 is set up through a Virtual Trunk. To establish VC2, ATM switch 1 and ATM switch 2 exchange signaling messages. These signaling messages are transported over the signaling channel that is embedded in the VT (i.e. over RVPI=0). Setup of VC2 is transparent to ME1 and ME2.

Appendix A: Impact on ATM switch implementations

(INFORMATIVE)

Virtual Trunks are introduced to provide a flexible and scaleable mechanism for the interworking of a legacy ATM infrastructure over an MPLS core without requiring the introduction of new network elements that support full-blown ATM, full-blown MPLS and Pseudo-Wire Emulation. Obviously, that goal can only be achieved if the implementation of Virtual Trunks does not require any upgrades to existing ATM hardware and if the changes to existing software are relatively minor. This section lists the primary development items for an ATM switch.

The following are the primary new features required on an ATM switch to support Virtual Trunks

1. The introduction of a Virtual Trunk as a logical interface. The extreme case of a single VP as Virtual Trunk is already supported on most switches. However, such an implementation does not achieve all the benefits of a generic VT implementation.
2. North-bound interfaces to enable an operator to provision and manage Virtual Trunks
3. The use of RVPI values for connection set up and the mapping of RVPI values to the actual VPI values that are used on the external interfaces and internally for switching, queuing, performance monitoring, etc.
4. Connection admission control based on the bandwidth allocated to the Virtual Trunk
5. The ability to treat F4 AIS on RVPI = 0 as an indication that the VT is down and to perform the necessary consequent actions.

Apart from these items, there are very few changes required to the rest of the software. Except for signaling procedures in which RVPI is used, all other actions in the ATM switch use the actual VPIs, just as they would if VTs were not used. Hence, there are no changes required to existing ATM hardware.

END OF DOCUMENT