

Understanding Session Establishment and Route Exchange in an MPLS–Enabled ATM Core

Document ID: 10471

Introduction

Prerequisites

Requirements

Conventions

Network Diagram

Adjacency Creation

Completion of the Routing Table

TVC VPI/VCI Details for Routes in the Routing Table

Related Information

Introduction

This document explains how a router establishes adjacency with its ATM Label Switch Router (LSR) in a Multiprotocol Label Switching (MPLS) enabled ATM core. It looks at how routes are exchanged and the labels used for this; here we use the virtual channel and path identifiers (VPI/VCI) of a Tag Virtual Circuit (TVC). It uses a series of debug commands to illustrate the steps involved.

Prerequisites

Requirements

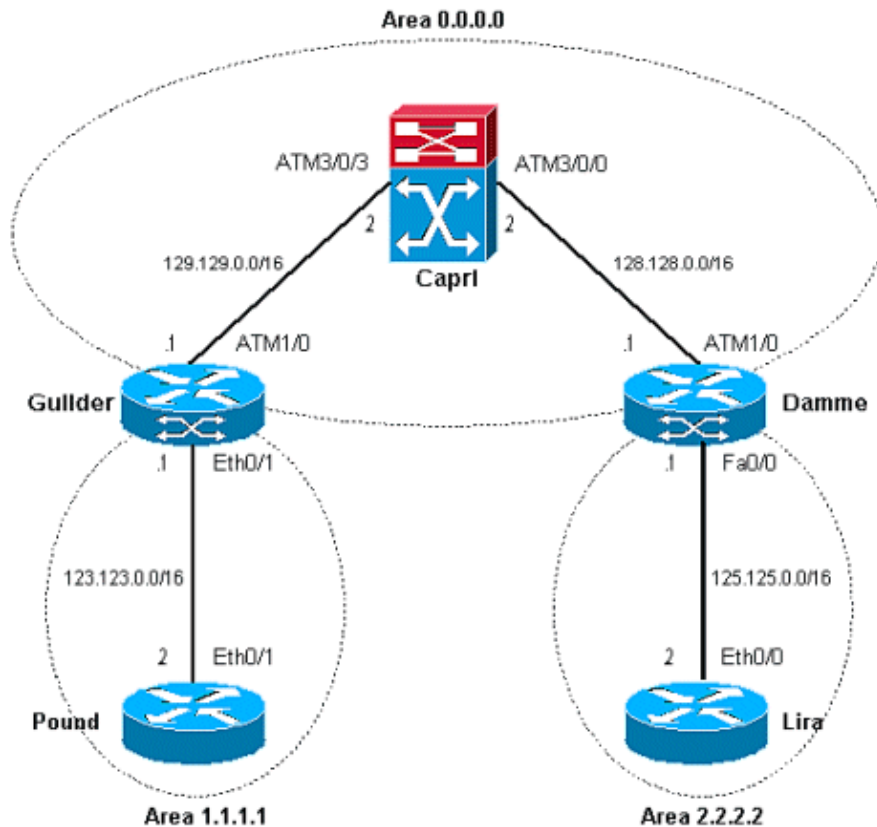
This document uses Cisco 3600 Series routers that run Cisco IOS® Software Release 12.0(7)T and OC–3 (optical carrier) interfaces. The ATM LSR used is an 8540 Multiservice Switch Router (MSR).

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Network Diagram

The network diagram shows the configuration used to illustrate the process. Related ATM sample configurations are found [here](#).



These are the three main stages between the point where an interface is Up/Up and the point where the routing table and tag forwarding tables are complete:

1. The Tag Distribution Protocol (TDP) creates adjacency between the two MPLS entities: in this example, between Damme and Capri.
2. The two MPLS entities exchange routing updates.
3. For each route found in the routing table, the router asks the LSR for a specific label.

Adjacency Creation

Assume that the link between Damme and Capri is down for administrative reasons. Enable the debugs as shown in the output:

```
damme#show debug
Tag Switching:
TDP peer state machine debugging is on
TDP transport events debugging is on
TDP transport timers debugging is on
TDP transport connection events debugging is on
```

Bring the interface back up:

```
damme#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
damme(config)#interface atm 1/0
damme(config-if)#no shut
damme(config-if)#
```

Damme immediately starts to send TDP hello packets on both FastEthernet0/0 and ATM1/0.2:

```
*Mar 2 21:34:58.456: tdp: Send hello; FastEthernet0/0, src/dst 125.125.0.1/255.255.255.255
*Mar 2 21:34:59.412: enabling tdp on ATM1/0.2

*Mar 2 21:34:59.412: tdp: tdp_set_intf_id: intf 0x6201F27C, ATM1/0.2, tc-atm, intf_id 1
1d21h: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
1d21h: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
*Mar 2 21:35:03.376: tdp: Send hello; FastEthernet0/0, src/dst 125.125.0.1/255.255.255.255
1d21h: %LANE-6-INFO: ATM1/0: ILMI prefix add event received
```

In the midst of its TDP hello-packet send to its neighbors, Damme receives a call from Capri (128.128.0.2). A TDP session is now established with Capri:

```
*Mar 2 21:35:04.112: tdp: Send hello; ATM1/0.2, src/dst 128.128.0.1/255.255.255.255, inst
*Mar 2 21:35:04.116: tdp: Incoming conn 128.128.0.1:711 <-> 128.128.0.2:11069
*Mar 2 21:35:04.116: tdp: Call from 128.128.0.2 with no adj or dir'd hello cb
*Mar 2 21:35:06.724: tdp: Rcvd hello; ATM1/0.2, from 128.128.0.2 (103.103.0.1:2), intf_id
*Mar 2 21:35:06.724: tdp: Hello from 128.128.0.2 (103.103.0.1:2) to 255.255.255.255, opt
*Mar 2 21:35:06.728: tdp: New adj 0x620156F4 from 128.128.0.2 (103.103.0.1:2), ATM1/0.2
*Mar 2 21:35:08.140: tdp: Send hello; FastEthernet0/0, src/dst 125.125.0.1/255.255.255.255
*Mar 2 21:35:08.948: tdp: Send hello; ATM1/0.2, src/dst 128.128.0.1/255.255.255.255, inst
*Mar 2 21:35:08.952: tdp: Incoming conn 128.128.0.1:711 <-> 128.128.0.2:11070
*Mar 2 21:35:08.952: tdp: Found adj 0x620156F4 for 128.128.0.2 (Hello src addr)
*Mar 2 21:35:08.952: tdp: New temporary adj 0x620D3208 from 128.128.0.2
*Mar 2 21:35:08.964: tdp: Real adj 0x620156F4 bound to 103.103.0.1:2, replacing temp adj0
*Mar 2 21:35:08.964: tdp: Adj 0x620D3208; state set to closed
*Mar 2 21:35:08.988: tagcon: start TDP TCP timers for 103.103.0.1:2 (pp 0x620DA794)
*Mar 2 21:35:08.988: tagcon: adj 103.103.0.1:2-1 (pp 0x620DA794): Event unsol open
      unsol op pdg -> estab
```

Once the session is established, the system uses keepalive packets to ensure that the LDP session is still Up:

```
*Mar 2 21:35:11.036: tdp: Rcvd hello; ATM1/0.2, from 128.128.0.2 (103.103.0.1:2), intf_id
*Mar 2 21:35:11.040: tdp: Start holding timer; adj 0x620156F4, 128.128.0.2
*Mar 2 21:35:12.632: tdp: Send hello; FastEthernet0/0, src/dst 125.125.0.1/255.255.255.255

*Mar 2 21:35:12.732: tdp: Send hello; ATM1/0.2, src/dst 128.128.0.1/255.255.255.255, inst
*Mar 2 21:35:15.164: tdp: Rcvd hello; ATM1/0.2, from 128.128.0.2 (103.103.0.1:2), intf_id
```

Use the **show tag tdp nei detail** command to check that adjacency has been created:

```
damme#show tag tdp nei detail
Peer TDP Ident: 103.103.0.1:2; Local TDP Ident 104.104.0.1:1
      TCP connection: 128.128.0.2.11074 - 128.128.0.1.711
      State: Oper; PIEs sent/rcvd: 25/25; ; Downstream on demand
      UID: 32; Up time: 00:14:09
      TDP discovery sources:
        ATM1/0.2; holdtime: 15000 ms, hello interval: 5000 ms
      Peer holdtime: 180000 ms; KA interval: 60000 ms; Peer state: estab
      Clients: TC ATM
```

Completion of the Routing Table

Make sure that the debugs listed are enabled. The output shows what happens once the sessions are established.

Note: We have separated the debugs in order to prevent the display of the keepalives in the debugs once the session is Up.

```
damme#show debug
IP routing:
  OSPF events debugging is on
Tag Switching:
  TC-ATM state machine debugging is on
  TC-ATM API debugging is on
  TC-ATM routes debugging is on

damme#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
damme(config)#interface atm 1/0
damme(config-if)#no shut
damme(config-if)#
```

Once the session is Up/Open, the Shortest Path First (OSPF) hello packets are exchanged over the control TVC (VPI0 and VCI32). Damme and Capri now exchange routing updates. Refer to the OSPF Design Guide for more information.

```
*Mar  2 21:42:18.364: OSPF: Interface ATM1/0.2 going Up
1d21h: %LINK-3-UPDOWN: Interface ATM1/0, changed state to up
*Mar  2 21:42:20.836: OSPF: Rcv hello from 129.129.0.2 area 0.0.0.0 from ATM1/0.2 128.128.0.2
*Mar  2 21:42:20.836: OSPF: End of hello processing
1d21h: %LINEPROTO-5-UPDOWN: Line protocol on Interface ATM1/0, changed state to up
*Mar  2 21:42:21.232: OSPF: Rcv hello from 101.101.0.1 area 2.2.2.2 from FastEthernet0/0 101.101.0.1
*Mar  2 21:42:21.232: OSPF: End of hello processing
*Mar  2 21:42:22.364: OSPF: Rcv hello from 129.129.0.2 area 0.0.0.0 from ATM1/0.2 128.128.0.2
*Mar  2 21:42:22.364: OSPF: End of hello processing
1d21h: %LANE-6-INFO: ATM1/0: ILMI prefix add event received
*Mar  2 21:42:26.340: OSPF: Rcv DBD from 129.129.0.2 on ATM1/0.2 seq 0x6FC opt 0x2 flag 0x0
mtu 4470 state INIT
*Mar  2 21:42:26.340: OSPF: 2 Way Communication to 129.129.0.2 on ATM1/0.2, state 2WAY
*Mar  2 21:42:26.340: OSPF: Send DBD to 129.129.0.2 on ATM1/0.2 seq 0x1654 opt 0x42 flag 0x0
*Mar  2 21:42:26.340: OSPF: NBR Negotiation Done. We are the SLAVE
*Mar  2 21:42:26.340: OSPF: Send DBD to 129.129.0.2 on ATM1/0.2 seq 0x6FC opt 0x42 flag 0x0
*Mar  2 21:42:26.344: OSPF: Rcv DBD from 129.129.0.2 on ATM1/0.2 seq 0x6FD opt 0x2 flag 0x0
mtu 4470 state EXCHANGE
*Mar  2 21:42:26.344: OSPF: Send DBD to 129.129.0.2 on ATM1/0.2 seq 0x6FD opt 0x42 flag 0x0
*Mar  2 21:42:26.344: OSPF: Database request to 129.129.0.2
*Mar  2 21:42:26.348: OSPF: sent LS REQ packet to 128.128.0.2, length 12
*Mar  2 21:42:26.348: OSPF: Rcv DBD from 129.129.0.2 on ATM1/0.2 seq 0x6FE opt 0x2 flag 0x0
mtu 4470 state EXCHANGE
*Mar  2 21:42:26.348: OSPF: Exchange Done with 129.129.0.2 on ATM1/0.2
*Mar  2 21:42:26.348: OSPF: Send DBD to 129.129.0.2 on ATM1/0.2 seq 0x6FE opt 0x42 flag 0x0
*Mar  2 21:42:26.348: OSPF: Synchronized with 129.129.0.2 on ATM1/0.2, state FULL
```

Use the **show ip route** command to check that the routing table is now complete:

```
damme#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set
```

```

102.0.0.0/32 is subnetted, 1 subnets O
    102.102.0.1 [110/3] via 128.128.0.2, 3d02h, ATM1/0.2
103.0.0.0/32 is subnetted, 1 subnets O          103.103.0.1 [110/2] via 12
100.0.0.0/32 is subnetted, 1 subnets O IA
    100.100.0.1 [110/13] via 128.128.0.2, 3d02h, ATM1/0.2
101.0.0.0/32 is subnetted, 1 subnets O
    101.101.0.1 [110/11] via 125.125.0.2, 3d04h, FastEthernet0/0 C
128.128.0.0/16 is directly connected, ATM1/0.2 O
    129.129.0.0/16 [110/2] via 128.128.0.2, 3d02h, ATM1/0.2
125.0.0.0/16 is subnetted, 1 subnets C
    125.125.0.0 is directly connected, FastEthernet0/0
123.0.0.0/16 is subnetted, 1 subnets O IA
    123.123.0.0 [110/12] via 128.128.0.2, 3d02h, ATM1/0.2
104.0.0.0/16 is subnetted, 1 subnets C
    104.104.0.0 is directly connected, Loopback0

```

TVC VPI/VCI Details for Routes in the Routing Table

For the next stage, make sure that the debugs enabled in the Completion of the Routing Table section are still enabled.

The router requests a TVC for each route of type *headend* to be reached through the ATM cloud. Use 102.102.0.1/32, the loopback interface of Guilder, as an example:

```

*Mar  2 21:42:33.864: TCATM: tcatmFindRouteTags,102.102.0.1/32,idb=ATM1/0.2,nh=128.128.0.2
*Mar  2 21:42:33.864: TCATM: AddNewRoute,102.102.0.1/32,idb=ATM1/0.2
*Mar  2 21:42:33.868: TCATM: PrepareForHeadRequest 102.102.0.1/32 ATM1/0.2 send request
*Mar  2 21:42:33.868: TCATM: Headend Router 102.102.0.1 ATM1/0.2 NonExistent -> XmitRequest
*Mar  2 21:42:33.868: TCATM: Headend Router 102.102.0.1 ATM1/0.2 XmitRequest -> BindWait T
*Mar  2 21:42:33.872: TCATM: CleanupRoutes,102.102.0.1/32
*Mar  2 21:42:33.872: TCATM: CleanupRoutes,not deleting route of idb ATM1/0.2,rdbIndex 0
*Mar  2 21:42:33.872: TCATM: ProcessRouteAddition,102.102.0.1/32

*Mar  2 21:42:33.956: TCATM: Headend Router 102.102.0.1 ATM1/0.2 2/33 BindWait -> ApiWaitS
*Mar  2 21:42:33.956: Headend Router Allocate Req for 102.102.0.1 on ATM1/0.2 VPI/VCI 2/3
*Mar  2 21:42:33.956: TAGATM_API: received tag allocation request
    interface: ATM1/0.2 dir: out vpi: 2 vci: 33
*Mar  2 21:42:33.956: TAGATM_API: completed tag allocation
    interface: ATM1/0.2 vpi: 2 vci: 33
    result: TAGATM_OK
*Mar  2 21:42:33.960: Allocate response for 102.102.0.1 on ATM1/0.2: success VPI/VCI 2/33
*Mar  2 21:42:33.960: TCATM: Headend Router 102.102.0.1 ATM1/0.2 2/33 ApiWaitSetup -> Acti

```

The router now requests a TVC for each route of type *tailend* to be reached through this router. Use 101.101.0.1/32, the loopback interface of Lira, as an example:

```

*Mar  2 21:42:36.380: TCATM: ProcessRouteAddition,101.101.0.1/32
*Mar  2 21:42:36.380: TCATM: AddNewRoute,101.101.0.1/32,idb=FastEthernet0/0
*Mar  2 21:42:36.380: TCATM: Tailend Router 101.101.0.1 ATM1/0.2 NonExistent -> VCAAllocate
*Mar  2 21:42:36.380: Tailend Router Allocate Req for 101.101.0.1 on ATM1/0.2
*Mar  2 21:42:36.380: TAGATM_API: received tag allocation request
    interface: ATM1/0.2 dir: in vpi: 0 vci: 0
*Mar  2 21:42:36.380: TAGATM_API: completed tag allocation
    interface: ATM1/0.2 vpi: 2 vci: 34
    result: TAGATM_OK
*Mar  2 21:42:36.380: Allocate response for 101.101.0.1 on ATM1/0.2: success VPI/VCI 2/34
*Mar  2 21:42:36.384: TCATM: Tailend Router 101.101.0.1 ATM1/0.2 2/34 VCAAllocate -> ApiWai
*Mar  2 21:42:36.384: TCATM: Tailend Router 101.101.0.1 ATM1/0.2 2/34 ApiWaitSetup -> Xmit
*Mar  2 21:42:36.384: TCATM: Tailend Router 101.101.0.1 ATM1/0.2 2/34 XmitBind -> Active T
*Mar  2 21:42:36.384: TCATM: tcatmTrBindActivate for a specific 101.101.0.1/32

```

The tag forwarding table is now complete, and IP packets can be sent to the TVCs. With the same examples, check this tag forwarding table:

```
damme#show tag forwarding-table 102.102.0.1 32 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
27     2/33      102.102.0.1/32  0          AT1/0.2   point2point
      MAC/Encaps=4/8, MTU=4470, Tag Stack{2/33(vcd=465)}
      01D10900 001D1000
```

```
damme#show tag-switching forwarding-table 101.101.0.1 32 detail
Local  Outgoing  Prefix          Bytes tag  Outgoing  Next Hop
tag    tag or VC  or Tunnel Id    switched   interface
32     Untagged  101.101.0.1/32  1660      Fa0/0     125.125.0.2
      MAC/Encaps=0/0, MTU=1504, Tag Stack{}
```

Related Information

- [ATM Technology Support Pages](#)
- [Technical Support & Documentation – Cisco Systems](#)

All contents are Copyright © 1992–2006 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Jan 02, 2007

Document ID: 10471
