Furquan Ansari
Lucent Tech.
Hormuzd Khosravi
Intel Corp.
Jamal Hadi Salim
Znyx Networks
Joel M. Halpern
Megisto Systems
Xiaoyi Guo
Huawei Tech.

Internet Draft
Document: draft-ietf-forces-discovery-02.txt
Expires: September 5, 2006
Working Group: ForCES
March 6, 2006

ForCES Intra-NE Topology Discovery

draft-ietf-forces-discovery-02.txt

Status of this Memo

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119.

Abstract

This document describes a mechanism for discovering inter-FE topology, topology maintenance in Intra-NE. Such a mechanism is essential for all these elements in the set to behave as a single Network Element, as required by the ForCES architecture as well as to perform certain optimizations at the FE by making use of the topology. The mechanism only operates during post-association phase of ForCES protocol.

Table of Contents

1. Definitions

Inter-FE topology discovery: Topology discovery relates to how the FEs are interconnected with each other with respect to packet forwarding. This is the complete view of the intra-NE network as seen by the CE.

Inter-FE topology maintenance: Once the inter-FE topology has been discovered, it has to be continuously monitored to ensure that any changes to the topology are reported to the corresponding CE. This represents the steady state and final phase of the protocol.

Edge FE: edge FE which has a port connected to other NEs or routers outside of this NE, and also connects to intra-NE FE port.

Intra-NE: It describes the connection and status in a NE, these status do not contact with outside NEs or other routers.

2. Introduction

The ForCES framework document [RFC 3746] describes how a set of control elements (CEs) and forwarding elements (FEs) interact with each other to form a single network element (NE). It describes the ForCES post-association phase protocol working across the Fp reference point between CE and FE. This document describes an important aspect of the ForCES operational infrastructure - that of discovering the layout of the different elements and forwarding packet within an NE.

The Inter-FE/Intra-NE topology discovery protocol module may be implemented as some separate LFBs on the FE. The protocol runs in an ongoing discovery and maintenance mode wherein the LFB maintains information about the known adjacencies per interface it is operated on. Each FE simply maintains its own adjacency tables and notifies the CE of any changes to the adjacency table based on the ForCES notification mechanism or if the CE explicitly requests an update. It is up to the CE to construct the full topology based on the information received from individual FEs within the NE and generate the NE routing table. Given that the CE can request and the FEs should report back the topology updates using ForCES protocol.

The proposed discovery mechanism is required to scale to a very large number of forwarding elements in the NE, with minimal impact on the resources. The following list provides some of the features and goals of the discovery mechanism.

- Determine connectivity between elements
- React to changes in link connectivity
- Construct topology information from the collected partial topology information
- Tolerant to protocol message losses
- Applicable to all inter-FE network topologies such as ring, mesh, star etc.
- Cause minimal overhead
- Agnostic of the network interconnect technology

As noted above, it is implicit that all the phases occur in the ForCES post-association phase. In other words, the ForCES protocol

association between the CEs and the FEs should already have taken place.

2.1. Motivation

The ForCES architecture defines a network element (NE) as a single managed entity made up of a collection of FEs and CEs and is indistinguishable from other network elements in the network. This NE model definition leads to three types of links from the network perspective: internal (or intra-NE) links and external (or inter-NE) links and control links. Intra-NE links are purely internal to the NE and are not exposed to the external world; whereas, inter-NE (or external) links are exposed to the external world and over which routing adjacencies (such as OSPF, IS-IS, BGP etc.) can be formed. An NE can contain FEs that have zero or more internal/external links e.g. in Fig. 1, FE3 has two internal links and no external links while FE1 and FE2 have two internal links and one external link each. Control links are those links that are used for communication between the CE and FE. If the CE and FE are a single Layer 3 hop apart as in Fig.1, the control link is typically a physical link e.g. link A of FE1 in the figure. Control links can be logical as well. It is important to note that the type definition for given for a link is only logical, because a given physical link may be a combination of more than one type - e.g. it could simultaneously be a control link and an internal link at the same time. Based on this link definitions, an FE can be considered to be an internal FE if it only contains internal links and an edge FE if it contains external links (and may also contain internal links).

A packet entering a ForCES NE may travel multiple FEs within the NE before it exits onto the output link. This requires that the packet be correctly forwarded from the ingress FE to the egress FE. This internal forwarding requires knowledge of the physical FE interconnection topology so that the CE can appropriately setup internal LFB tables at each FE to handle packet traversal in a sane manner.

```
                   NE 1
   .....................................
   .         -----------------          .
   .         |      CE       |          .
   .         -----------------          .            ----------
   .        A ^    B ^    C ^           .            | NE 1   |
   .         /       |       \          .            |        |
   .        /      A v        \         .            ----------
   .       /      ------- B    \        .             ^      ^
   .      /    +->| FE3 |<-+    \       . <====>     /        \
```

```
   .     /     |C |     |  |     \      .           /          \
   .  A v      |  -------  |      v A   .          v            v
   .  -------B |           |D -------   .       --------      ---------
   .  | FE1 |<-+           +->| FE2 |   .       | NE 2 |<---->|  NE 3 |
   .  |     |<--------------->|     |   .       |      |      |       |
   .  ------- C             C -------   .       --------      ---------
   .   D^                      ^B      .
   .....|......................|........
        |                       |
        V                       V
    --------                 --------
    | NE 2 |<--------------->| NE 3 |
    |      |                 |      |
    --------                 --------
               (a)                                        (b)
```

Figure 1:(a) illustrates the internal/external links and topology within a NE. (b) Shows the network topology as seen by external

routing protocols

3. Topology Discovery Mechanism

Since the topology discovery protocol described here operates in the ForCES post-association phase, it is independent of whether the CE and the FE is a single or multiple hops (layer 2 or layer 3) apart from each other. It is up to the ForCES association protocol to determine how to setup the ForCES channel between the CE and FE if they are multiple hops away. The topology discovery protocol is expected to work on all types of media and interfaces such as point-to-point as well as multi-access links.

In order to keep the discovery and maintenance mechanism as simple as possible, the FEs only maintain relationships with their respective neighbors to determine the status of the neighbors. No databases are exchanged between the neighbors. This implies that the topology view for each FE is only limited to the adjacent elements. This partial topology information may be reported back to the CE (or queried by the CE) over the ForCES protocol using the ForCES notification mechanism. Since the CE receives such information from all the FEs, it can easily construct the full topology from individual partial topologies reported by each FE. Once the CE constructs the full topology, such information can be passed to the FEs, if needed (depending on policy). The FEs may use such information for dynamic intra-NE route calculation or certain other optimizations.

Topology information is needed by a lot of LFBs and associated services that span multiple FEs within a NE. In the case where the

FE aids the CE in offloading the table updates, then it makes sense for the FE to be topology aware. It is sometimes also helpful to keep full topology information at the FEs for cases such as message snooping optimizations. For example, if an FE is aware of the topology, it could snoop on messages sent to other FEs (e.g., broadcasts, multicasts) and update its own tables dynamically without involving the CE. Another example would be FE-FE primary-backup handover scenario. With each FE being fully aware of the complete topology, the backup FE can take over the responsibilities of the primary without involving the CE for such a handover.

3.1 Minimum requirements

In order for the protocol to work as described, the following assumptions are made.

- Each element has been configured with their respective IDs (CEID, FEID)

- Element bindings process has already taken place. In other words, the CE know all the FEs it wants to control and each FE knows which CE is allowed to control it.

- The ForCES protocol association has already taken place between the CE and the FE in question.

- The protocol is enabled on the required interfaces.

Note that these are configuration requirements and are satisfied by the respective managers (CEM/FEM).

3.2 Protocol Details

Once the ForCES protocol association has been established between a CE and a given FE i.e. it is in post-association phase, the CE starts sending/advertising Hello/Probe messages to the FE neighbors such that the messages go through the given FE. In other words, it looks like the given FE is generating probe messages to the neighbor (except that these messages are coming from the CE over the ForCES protocol first). However, this functionality of generating probe messages by the CE can be offloaded to the FE itself (to be more precise, to an FE LFB) so that the FE can originate and terminate the probe messages. This provides better scalability of the CE and it resources. The CE can now simply query each FE neighbor relationship database and register for any events related to topology changes.

All Hello/Probe messages travel a single PE hop and are not routed to other elements beyond the first hop. An on-link IP multicast address is used for sending all Hello packets. The packets should be sent with a TTL of 255 and ignored on receipt if the TTL is not 254 (based on some of the recommendations from the generalized TTL security mechanism to use TTL 255 rather than TTL 1). Hello packets are only sent on interfaces configured for topology discovery protocol operation. Further, the Hello messages will be multicast on multicast capable links. Each FE topology LFB component maintains the neighbor relationships as long as the Hello messages are received from the neighbor. If it does not receive Hello messages after a given (configured) period of time (called FE Neighbor dead interval), it deletes the entry from the database and reports this change to the CE in the form of an event-notification message over the ForCES protocol. This ensures that the CE has the complete and up-to-date information of the underlying topology of the Inter-FE network.

The Hello message contains information necessary for discovering and maintaining neighbor relationships. It contains the PE ID, type of protocol element (i.e. CE or FE), interval between any two messages, interval for deeming a neighbor inactive, capability information etc. This is, in some ways, similar to the capabilities of the OSPF Hello protocol.

On receiving the Hello messages from a neighbor, the FE responds back with its own Hello message in a packet format similar to the one received from the neighbor. Essentially, both sides are independently sending Hello messages to each other and listing their neighbor table. Also, each neighbor will see itself listed on its neighbors Hello message. This ensures bi-directionality of the link between any two neighbors.

The operation is concisely described by the following steps:

. CE activates the topology LFB/component on the FE to initialize on specific ports

. FE topology LFB/component sends neighbor probes/hellos

. CE queries FE for its neighbors

. FE continues to send these probes afterwards (maintenance) and updates asynchronously any new updates

Note: We would like to point out here that the Hello messaging mechanism can very well be replaced by the BFD (Bi-Directional

Forwarding Detection) protocol in the future since it performs similar task of detecting bi-directional faults between two forwarding engines. Further, BFD protocol has the ability to be bootstrapped by any other protocol that automatically forms peer, neighbor or adjacency relationships to seed BFD endpoint discovery.

3.2.1. Neighbor Finite State Machine

In order to obtain bi-directionality verification of the links, and to make the protocol more robust, a neighbor finite state machine is needed. It consists of the following three states:

Neighbor-down: This is the initial state of the neighbor conversation. It indicates that there has been no recent information received from the neighbor

Neighbor-heard: In this state, a Hello packet was recently seen from the neighbor. However, bi-directional communication has not been fully established with the neighbor (i.e. the PE itself was not listed in the neighbor Hello packet which is the check for bi-directionality). All neighbors in this state (or higher) are listed in the Hello packets sent from the associated interface.

Neighbor-adjacent: In this state, the communication between the two neighbors is bi-directional. This has been assured by the Hello protocol operation. This state corresponds to the final steady state of the protocol.

3.2.2. Topology Discovery and Maintenance

Since the CE needs to maintain consistent and up-to-date view of the inter-FE topology, it needs to obtain real-time information of the status of the internal links connecting the FEs. Since the topology discovery and maintenance occurs during the post-association phase, we make use of the event-notification and query/response messages [ForCESP] of the ForCES protocol to provide this information to the CE. It is important to note that each FE only maintains partial topology information obtained through neighbor relationship maintenance through Hello messages. The partial topology view seen by each FE is only the neighbor connectivity information. The CE has to derive the complete topology view of the interconnected FEs based on the partial topology information reported by each FE (or queried by the CE). This ensures that that only the CE maintains all the intelligence and the protocol operation on the FEs is very simple and has minimal overhead. However, as mentioned above, if optimizations can be performed by having the complete topology information available at the FEs, the CE can push such information to any FE interested in it (interest on the FE may be shown in the

form of policy configuration). This is an optional feature available on each FE, which can be turned on or off through configuration or during capability exchange negotiation at setup time. Each FE vendor may decide to make use of this feature in different ways, so the capability to obtain such topology information should exist.

The periodic Hello messages maintain PE neighbor relationships. Any change in the link or neighbor status causes the FE to generate an asynchronous/event-driven message to the CE indicating this change. The mechanism defined in [ForCESP] is used for delivering event-driven messages from the FE to the CE. This involves the CE subscribing to such event-driven messages from the FE.

3.2.3. Full topology computation at the CE from partial topologies

The CE receives neighbor relationships information from each FE that it uses to construct the full topology of the internal network. Each FE neighbor relationship table contains information regarding the local element ID, local port connecting the neighbor, the neighbor ID, the neighbor port and any optional additional information. Note that the fact that the FE already knows the neighbor port information implies that it received the probe/hello messages from the neighbor on that port in response to the hello sent and was, therefore, able to establish bi-directionality of the link. If all the links in the internal network are point-to-point links, the CE simply has to aggregate all the neighbor relationship tables obtained from all the FEs to generate the full topology. If we assume the topology to be a graph, each edge of the graph will be present twice essentially providing the same information from the two endpoints of the graph. After deleting all the duplicate entries (and thus reducing the table size by half), the CE now has accurate view of the full topology. Please refer section 3.4 [Fig. 3(b)] for more details.

[TBD: Sub-section on generating full topology from partial topology information for broadcast/multi-access, point-to-multipoint etc. type of links]

3.2.4. Update and Maintenance

The periodic Hello messages maintain PE neighbor relationships. Any change in the link or neighbor status causes the FE to generate an asynchronous/event-driven message to the CE indicating this change. The mechanism defined in [ForCESP] is used for delivering event-driven messages from the FE to the CE. This involves the CE subscribing to such event-driven messages from the FE.

The CE aggregates the partial topology information received from each FE and generates the inter-FE topology. With this complete knowledge of the inter-FE topology, it can now make appropriate updates to the LFB tables and routing table on each FE to move packets inside the NE from ingress FE to egress FE, assuming that the destination of the packet is not the current NE itself. Any changes in the internal link states (and hence the topology) requires that the CE reconfigure the LFB tables on the FEs based on the most up-to-date information to ensure that the packets do not end up in a black hole or enter a loop.

3.3. Protocol and Message Headers

The protocol hello message consists of a fixed length header (16 bytes) followed by one or more optional TLVs. The format of the message is as follows.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Version    |    Flags      |         Packet Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Checksum            |         Port ID               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         PE ID                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        TLV-Type              |         TLV-Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         TLV-Value    ...                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            ...                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure (2)

Version (8 bit):
Version number of this protocol. Currently acceptable value is 0x01

Flags (8 bit):
These indicate whether the message is sent by a FE, (0x01) or CE (0x02). More options may be defined in the future.

Packet Length (16 bit):
The length of the protocol message in bytes, including the header and the following TLVs.

Checksum (16 bit):
Checksum for the protocol message. The checksum calculation does not include the IP header.

Port ID (16 bit):
This indicates the port on which this packet was sent out by the sender useful for topology construction.

PE ID (32 bit):
32-bit identifier of the sender. It could either be CE ID or FE ID, depending on the sender. See more details in ForCES protocol section 6.1.

TLV Type (16 bit):
The TLV type field is two octets, and indicates the type of data encapsulated within the TLV.

TLV Length (16 bit):
The TLV Length field is two octets, and indicates the length of this TLV including the TLV Type, TLV Length, and the TLV data in octets.

TLV Value (variable):
The TLV Value field carries the data. For extensibility, the TLV value may in fact be a TLV. TLVs must be 32 bit aligned. Padding used for the alignment must be zero on transmission and must be ignored upon reception.

3.3.1. TLV definitions

The protocol header is followed by one or many TLVs. The following TLVs types are defined:

Hello TLV: Indicates the Hello message as exchanged by the neighbors. The TLV defines the common hello parameters such as the Hello Interval, Hold time, Unidirectional targeted Hellos, Sequence space number, if needed etc.

Data TLV: Indicates the neighbor or the topology information from CE or neighbor FE.

Capabilities TLV: Provides the capabilities information - TBD

Vendor specific TLV: TBD

Other TLV: TBD

3.4. Inter-FE Topology Discovery Examples

The following examples illustrate the topology discovery mechanism. For sake of simplicity, we assume that there is only one CE per NE. The FEIDs of the FEs in the topologies below are FE1, FE2, FE3, and FE4. Each FE has port IDs labeled alphabetically. This is also the case with the CE.

```
           -----------------
           |      CE       |
           -----------------
           A ^    B ^    C ^
           /       |       \
          /      A v        \
         /      ------- B    \
        /    +->| FE3 |<-+    \
       /     |C |     |  |     \
    A v     2|  -------  |1     v A
    -------B |           |E -------
    | FE1 |<-+           +->| FE2 |
    |     |<--------------->|     |
    ------- C     5       D -------
     E ^ D|                 C ^  | B
       |  |                   |  |
       |  v                   |  v
```

FE3 Control Element reachability Table

<Dest Addr>	<local intf>
CE	A

FE3 NEIGHBOR ASSOCIATION TABLE

<local intf>	<neighbor_FEID>	<neighbor_portID>
B	FE2	E
C	FE1	B

Figure 4. (a) Full mesh among FE1, FE2, and FE3

During the element-binding phase, each FE sends out hello messages with its FEID and Port ID (as outlined earlier) to all of its neighbors. Since each neighboring FE also listens to such messages, it receives the hello message and adds it to the neighbor association table, which may look like that shown in Fig.4(a). In the topology discovery phase, which is post ForCES association stage, the CE queries each FE about its neighbor table. The FE responds back with the partial topology information available through its

neighbor relationships. Both the query and the response are carried by the ForCES protocol. The CE collects the partial topology information from all the FEs in the NE and aggregates this information to fully construct the inter-FE topology. Any changes to the FE neighbor table, e.g. when a link state changes, generates a trigger/update message to the CE. The new information is used to recalculate the new topology and subsequently the CE takes appropriate actions based on the new topology such as updating the packet forwarding tables on the FEs.

The following examples show the neighbor association tables.

3.4.1. Forwarding Elements connected in a daisy chain

```
                  --------------
                  |     CE     |
                  --------------
               A  ^ ^ B    ^    ^ D
                 /  |       \    \
          /------   |        --\  -------\
       A v        A v           v A       v A
     -------B    -------B    -------B    -------
     | FE1 |<--->| FE2 |<--->| FE3 |<--->| FE4 |
     -------    E-------    E-------    D-------
      D ^  |C     D ^  |C      D^  |C      C^  |B
        |  |        |  |        |  |        |  |
        |  v        |  v        |  v        |  v
```

FE1 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
B	FE2	E

FE2 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
E	FE1	B
B	FE3	E

FE3 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
B	FE4	D
E	FE2	B

FE4 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
D	FE3	B

```
    CE Topology (Aggregate)View            CE Topology View
   --------------------------------    ------------------------------
   <Node>  <Port>  <Node>  <Port>     <Node>  <Port>  <Node>  <Port>
    FE1     B       FE2     E\         FE1      B      FE2     E
    FE2     E       FE1     B/   =>    FE2      B      FE3     E
    FE2     B       FE3     E\         FE3      B      FE4     D
    FE3     E       FE2     B/       ------------------------------
    FE3     B       FE4     D\
    FE4     D       FE3     B/
   --------------------------------
```

Fig.4 (b) Multiple FEs in a daisy chain

3.4.2. Forwarding Elements connected in a ring

```
                     ^ |
                     D| v E
                   ----------- A
                   |   FE1   |<-----------------------|
                   -----------                        |
                   C ^    ^ B                         |
                    /      \                          |
             | ^   /        \    ^ |                   V A
           B v |C v D      C v  D| v E             ----------
           ---------        ---------             D|        |
           | FE2   |        |  FE3  |<------------>|   CE   |
           ---------        ---------  A           |        |
             A ^  ^ E        ^ B                   ----------
               |   \        /                      C ^  ^ B
               |    \      /                         |  |
               |   D v   E v                         |  |
               |   ----------- A                     |  |
               |   |   FE4   |<----------------------|  |
               |   -----------                          |
               |    C |  ^ B                            |
               |      v  |                              |
               |                                        |
               |----------------------------------------|
```

FE1 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
B	FE3	C
C	FE2	D

FE2 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
E	FE4	D
D	FE1	C

FE3 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
B	FE4	E
C	FE1	B

FE4 NBR ASSOCIATION TABLE

<locl intf>	<nbr_FEID>	<nbr_port>
D	FE2	E
E	FE3	B

Fig. 4(c) Multiple FEs connected in a ring

4. Security Considerations

The ForCES protocol should ensure the communication security between CEs and FEs. FEs should ensure that only properly authenticated ForCES protocol participants are performing such manipulations.

5. References

5.1 Normative

[RFC3746] Yang, L., Dantu, R., Anderson, T. and R. Gopal, "Forwarding and Control Element Separation (ForCES) Framework", RFC 3746, April 2004.

[RFC3654] Khosravi, H. and T. Anderson, "Requirements for Separation of IP Control and Forwarding", RFC 3654, November 2003.

[ForCESP] F P Team, "ForCES protocol specification", draft-ietf-forces-protocol-05.txt, Nov 2006.

5.2 Informative

[OSPF] J. Moy, OSPF Version 2, 1998, RFC 2328.

[BGP] Y. Rekhter, T. Li, Border Gateway Protocol 4 (BGP-4) 1995, RFC 1771.

[IS-IS] R. Collela et al., guidelines for OSI NSAP Allocation in the Internet 1994, RFC 1629.

6. Authors' Addresses

Furquan Ansari
Bell Labs Research, Lucent Tech.
101 Crawfords Corner Road
Holmdel, NJ 07733
USA
Phone: +1 732-949-5249
Email: furquan@lucent.com

Hormuzd Khosravi
Intel
2111 N.E. 25th Avenue JF3-206
Hillsboro, OR 97124-5961
USA
Phone: +1 503 264 0334
Email: hormuzd.m.khosravi@intel.com

Jamal Hadi Salim
ZNYX Networks
Ottawa, Ontario, Canada
Email: hadi@znyx.com

Joel M. Halpern
Megisto systems, Inc.
0251 Century Blvd.
Germantown, MD, 20874-1162, USA
Phone: +1 301 444 17
Email: jhalpern@megisto.com

Xiaoyi Guo
Huawei Tech.
No.3 Xinxi Rd., Shang-Di,
Hai-Dian District Beijing P.R. China

7. IANA Considerations

There are no IANA considerations at this point since the protocol completely operates within an NE.

8. Full Copyright Notice

9. Acknowledgments

We would like to thank Thyaga Nandagopal of Lucent Technologies for his thoughts and contributions to the initial draft.

Funding for the RFC Editor function is currently provided by the Internet Society.