Internet Draft
Expiration: July 2004
File: draft-ietf-forces-framework-13.txt
Working Group: ForCES

L. Yang
Intel Corp.
R. Dantu
Univ. of North Texas
T. Anderson
Intel Corp.
R. Gopal
Nokia

January 2004

Forwarding and Control Element Separation (ForCES) Framework

draft-ietf-forces-framework-13.txt

Status of this Memo

Copyright Notice

Abstract

This document defines the architectural framework for the ForCES (Forwarding and Control Element Separation) network elements, and identifies the associated entities and the interactions among them.

Table of Contents

1. Definitions

1.1. 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.

1.2. Terminologies

A set of terminology associated with the ForCES requirements is defined in [3] and we only include the definitions that are most relevant to this document here.

Addressable Entity (AE) - An entity that is directly addressable given some interconnect technology. For example, on IP networks, it is a device to which we can communicate using an IP address; on a switch fabric, it is a device to which we can communicate using a switch fabric port number.

Physical Forwarding Element (PFE) - An AE that includes hardware used to provide per-packet processing and handling. This hardware may consist of (but is not limited to) network processors, ASICs (Application-Specific Integrated Circuits), or general purpose processors, installed on line cards, daughter boards, mezzanine cards, or in stand-alone boxes.

PFE Partition - A logical partition of a PFE consisting of some subset of each of the resources (e.g., ports, memory, forwarding table entries) available on the PFE. This concept is analogous to that of the resources assigned to a virtual switching element as described in [8].

Physical Control Element (PCE) - An AE that includes hardware used to provide control functionality. This hardware typically includes a general purpose processor.

PCE Partition - A logical partition of a PCE consisting of some subset of each of the resources available on the PCE.

Forwarding Element (FE) - A logical entity that implements the ForCES Protocol. FEs use the underlying hardware to provide per-packet processing and handling as directed by a CE via the ForCES Protocol. FEs may happen to be a single blade (or PFE), a partition of a PFE or multiple PFEs.

Control Element (CE) - A logical entity that implements the ForCES Protocol and uses it to instruct one or more FEs how to process packets. CEs handle functionality such as the execution of control and signaling protocols. CEs may consist of PCE partitions or whole PCEs.

ForCES Network Element (NE) - An entity composed of one or more CEs and one or more FEs. To entities outside an NE, the NE represents a single point of management. Similarly, an NE usually hides its internal organization from external entities.

Pre-association Phase - The period of time during which an FE Manager (see below) and a CE Manager (see below) are determining whether an FE and a CE should be part of the same network element. It is possible for some elements of the NE to be in pre-association phase while other elements are in the post-association phase.

Post-association Phase - The period of time during which an FE does know which CE is to control it and vice versa, including the time during which the CE and FE are establishing communication with one another.

ForCES Protocol - While there may be multiple protocols used within the overall ForCES architecture, the term "ForCES Protocol" refers only to the ForCES post-association phase protocol (see below).

ForCES Post-Association Phase Protocol - The protocol used for post-association phase communication between CEs and FEs. This protocol does not apply to CE-to-CE communication, FE-to-FE communication, nor to communication between FE and CE managers. The ForCES Protocol is a master-slave protocol in which FEs are slaves and CEs are masters. This protocol includes both the management of the communication channel (e.g., connection establishment, heartbeats) and the control messages themselves. This protocol could be a single protocol or could consist of multiple protocols working together, and may be unicast based or multicast based. A separate protocol document will specify this information.

FE Manager - A logical entity that operates in the pre-association phase and is responsible for determining to which CE(s) an FE should communicate. This process is called CE discovery and may involve the FE manager learning the capabilities of available CEs. An FE manager may use anything from a static configuration to a pre-association phase protocol (see below) to determine which CE(s) to use; however, this is currently out of scope. Being a logical entity, an FE manager might be physically combined with any of the other logical entities mentioned in this section.

CE Manager - A logical entity that operates in the pre-association phase and is responsible for determining to which FE(s) a CE should communicate. This process is called FE discovery and may involve the CE manager learning the capabilities of available FEs. A CE manager may use anything from a static configuration to a pre-association phase protocol (see below) to determine which FE to use, however this is currently out of scope. Being a logical entity, a CE manager might be physically combined with any of the other logical entities mentioned in this section.

Pre-association Phase Protocol - A protocol between FE managers and CE managers that is used to determine which CEs or FEs to use. A pre-association phase protocol may include a CE and/or FE capability discovery mechanism. Note that this capability discovery process is wholly separate from (and does not replace) that used within the ForCES Protocol. However, the two capability discovery mechanisms may utilize the same FE model.

FE Model - A model that describes the logical processing functions of an FE.

ForCES Protocol Element - An FE or CE.

Intra-FE topology - Representation of how a single FE is realized by combining possibly multiple logical functional blocks along multiple data path. This is defined by the FE model.

FE Topology - Representation of how the multiple FEs in a single NE are interconnected. Sometimes it is called inter-FE topology, to be distinguished from intra-FE topology used by the FE model.

Inter-FE topology - see FE Topology.

2. Introduction to Forwarding and Control Element Separation (ForCES)

An IP network element (NE) appears to external entities as a monolithic piece of network equipment, e.g., a router, NAT, firewall, or load balancer. Internally, however, an IP network

element (NE) (such as a router) is composed of numerous logically separated entities that cooperate to provide a given functionality (such as routing). Two types of network element components exist: control element (CE) in control plane and forwarding element (FE) in forwarding plane (or data plane). Forwarding elements typically are ASIC, network-processor, or general-purpose processor-based devices that handle data path operations for each packet. Control elements are typically based on general-purpose processors that provide control functionality like routing and signaling protocols.

ForCES aims to define a framework and associated protocol(s) to standardize information exchange between the control and forwarding plane. Having standard mechanisms allows CEs and FEs to become physically separated standard components. This physical separation accrues several benefits to the ForCES architecture. Separate components would allow component vendors to specialize in one component without having to become experts in all components. Standard protocol also allows the CEs and FEs from different component vendors to interoperate with each other and hence it becomes possible for system vendors to integrate together the CEs and FEs from different component suppliers. This interoperability translates into more design choices and flexibility for the system vendors. Overall, ForCES will enable rapid innovation in both the control and forwarding planes while maintaining interoperability. Scalability is also easily provided by this architecture in that additional forwarding or control capacity can be added to existing network elements without the need for forklift upgrades.

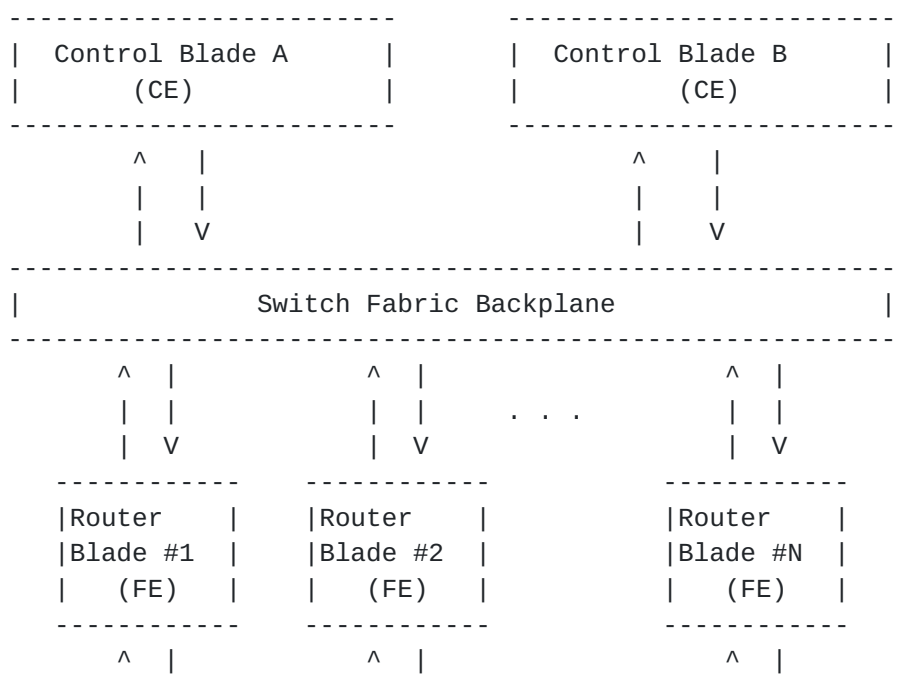


Figure 1. A router configuration example with separate blades.

One example of such physical separation is at the blade level. Figure 1 shows such an example configuration of a router, with two control blades and multiple forwarding blades, all interconnected into a switch fabric backplane. In such a chassis configuration, the control blades are the CEs while the router blades are FEs, and the switch fabric backplane provides the physical interconnect for all the blades. Control blade A may be the primary CE while control blade B may be the backup CE providing redundancy. It is also possible to have a redundant switch fabric for high availability support. Routers today with this kind of configuration use proprietary interfaces for messaging between CEs and FEs. The goal of ForCES is to replace such proprietary interfaces with a standard protocol. With a standard protocol like ForCES implemented on all blades, it becomes possible for control blades from vendor X and forwarding blades from vendor Y to work seamlessly together in one chassis.

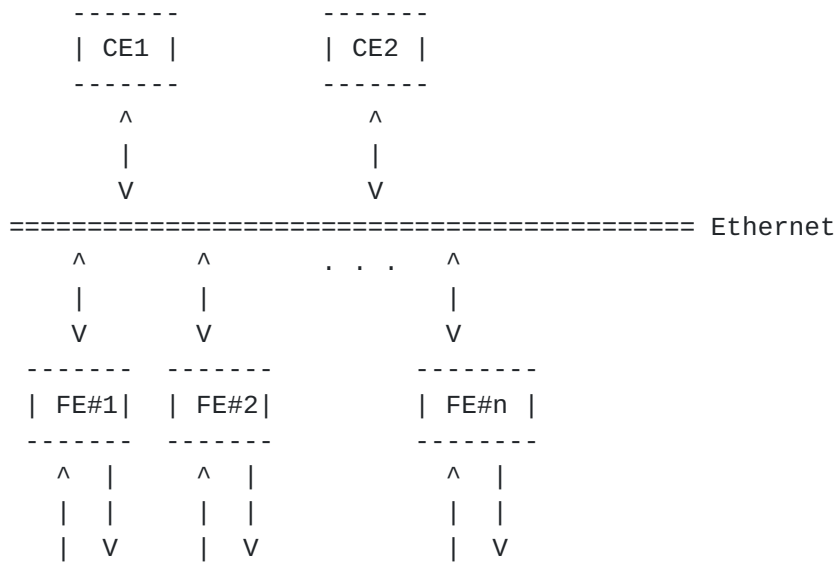


Figure 2. A router configuration example with separate boxes.

Another level of physical separation between the CEs and FEs can be at the box level. In such configuration, all the CEs and FEs are physically separated boxes, interconnected with some kind of high speed LAN connection (like Gigabit Ethernet). These separated CEs and FEs are only one hop away from each other within a local area

network. The CEs and FEs communicate to each other by running ForCES, and the collection of these CEs and FEs together become one routing unit to the external world. Figure 2 shows such an example.

In both examples shown here, the same physical interconnect is used for both CE-to-FE and FE-to-FE communication. However, that does not have to be the case. One reason to use different interconnects is that CE-to-FE interconnect does not have to be as fast as the FE-to-FE interconnect, so the more expensive fast connections can be saved for FE-to-FE. The separate interconnects may also provide reliability and redundancy benefits for the NE.

Some examples of control functions that can be implemented in the CE include routing protocols like RIP, OSPF and BGP, control and signaling protocols like RSVP (Resource Reservation Protocol), LDP (Label Distribution Protocol) for MPLS, etc. Examples of forwarding functions in the FE include LPM (longest prefix match) forwarder, classifiers, traffic shaper, meter, NAT (Network Address Translators), etc. Figure 3 provides example functions in both CE and FE. Any given NE may contain one or many of these CE and FE functions in it. The diagram also shows that ForCES Protocol is used to transport both the control messages for ForCES itself and the data packets that are originated/destined from/to the control functions in CE (e.g., routing packets). Section 4.2.4 provides more detail on this.

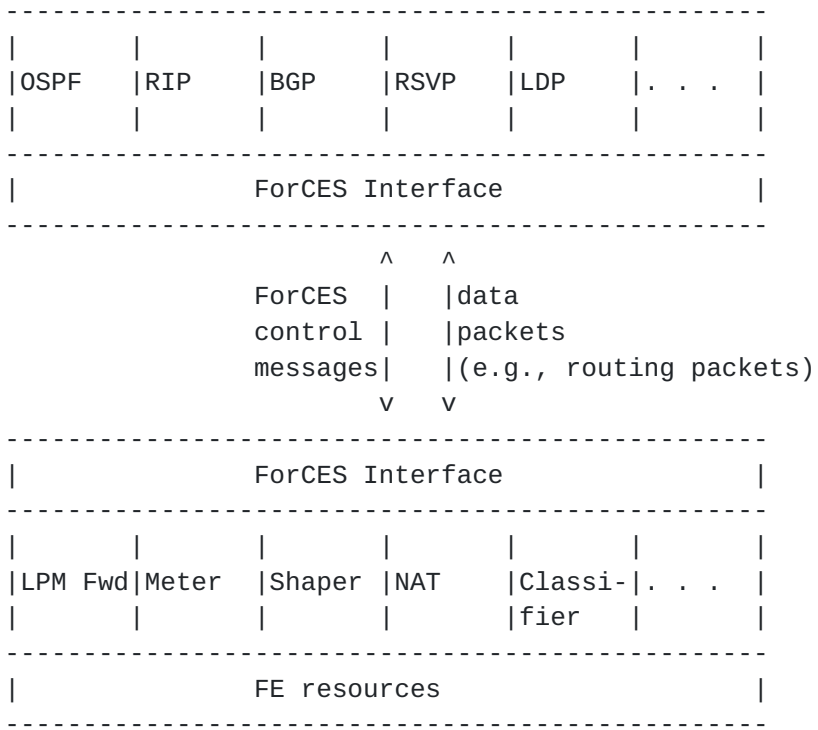
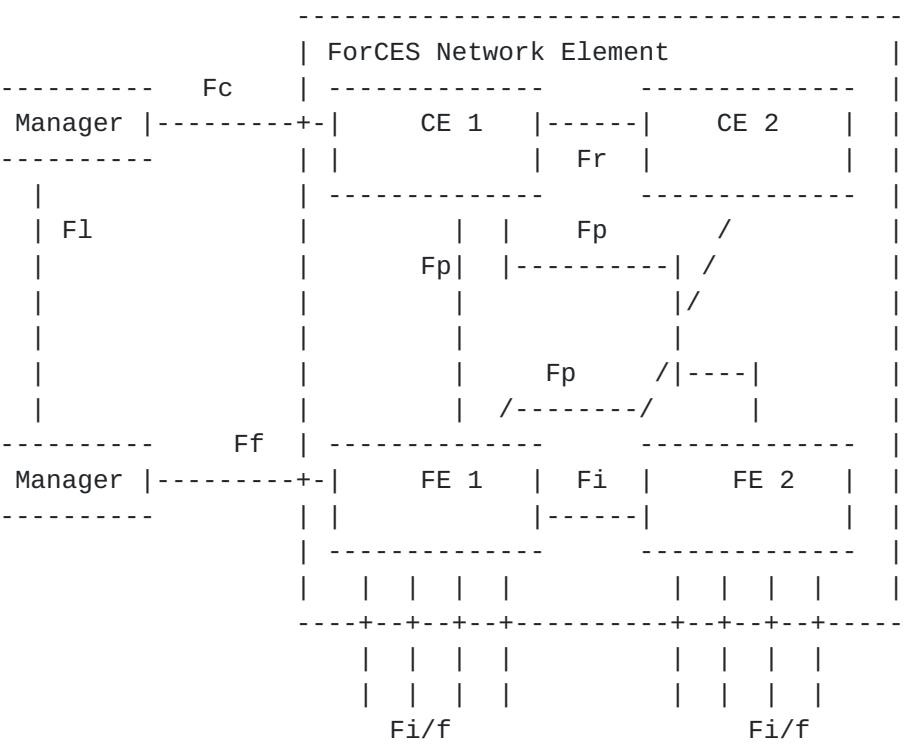


Figure 3. Examples of CE and FE functions

A set of requirements for control and forwarding separation is identified in [3]. This document describes a ForCES architecture that satisfies the architectural requirements of that document and defines a framework for ForCES network elements and the associated entities to facilitate protocol definition. Whenever necessary, this document uses many examples to illustrate the issues and/or possible solutions in ForCES. These examples are intended to be just examples, and should not be taken as the only or definite ways of doing certain things. It is expected that separate document will be produced by the ForCES working group to specify the ForCES Protocol.

3. Architecture

This section defines the ForCES architectural framework and the associated logical components. This ForCES framework defines components of ForCES NEs including several ancillary components. These components may be connected in different kinds of topologies for flexible packet processing.



Fp: CE-FE interface

Fi: FE-FE interface
Fr: CE-CE interface
Fc: Interface between the CE Manager and a CE
Ff: Interface between the FE Manager and an FE
Fl: Interface between the CE Manager and the FE Manager
Fi/f: FE external interface

Figure 4. ForCES Architectural Diagram

The diagram in Figure 4 shows the logical components of the ForCES architecture and their relationships. There are two kinds of components inside a ForCES network element: control element (CE) and forwarding element (FE). The framework allows multiple instances of CE and FE inside one NE. Each FE contains one or more physical media interfaces for receiving and transmitting packets from/to the external world. The aggregation of these FE interfaces becomes the NE's external interfaces. In addition to the external interfaces, there must also exist some kind of interconnect within the NE so that the CE and FE can communicate with each other, and one FE can forward packets to another FE. The diagram also shows two entities outside of the ForCES NE: CE Manager and FE Manager. These two ancillary entities provide configuration to the corresponding CE or FE in the pre-association phase (see Section 4.1).

For convenience, the logical interactions between these components are labeled by reference points Fp, Fc, Ff, Fr, Fl, and Fi, as shown in Figure 4. The FE external interfaces are labeled as Fi/f. More detail is provided in Section 3 and 4 for each of these reference points. All these reference points are important in understanding the ForCES architecture, however, the ForCES Protocol is only defined over one reference point -- Fp.

The interface between two ForCES NEs is identical to the interface between two conventional routers and these two NEs exchange the protocol packets through the external interfaces at Fi/f. ForCES NEs connect to existing routers transparently.

3.1. Control Elements and Fr Reference Point

It is not necessary to define any protocols across the Fr reference point to enable control and forwarding separation for simple configurations like single CE and multiple FEs. However, this architecture permits multiple CEs to be present in a network element. In cases where an implementation uses multiple CEs, the invariant that the CEs and FEs together appear as a single NE must be maintained.

Multiple CEs may be used for redundancy, load sharing, distributed control, or other purposes. Redundancy is the case where one or more CEs are prepared to take over should an active CE fail. Load sharing is the case where two or more CEs are concurrently active and any request that can be serviced by one of the CEs can also be serviced by any of the other CEs. For both redundancy and load sharing, the CEs involved are equivalently capable. The only difference between these two cases is in terms of how many active CEs there are. Distributed control is the case where two or more CEs are concurrently active but certain requests can only be serviced by certain CEs.

When multiple CEs are employed in a ForCES NE, their internal organization is considered an implementation issue that is beyond the scope of ForCES. CEs are wholly responsible for coordinating amongst themselves via the Fr reference point to provide consistency and synchronization. However, ForCES does not define the implementation or protocols used between CEs, nor does it define how to distribute functionality among CEs. Nevertheless, ForCES will support mechanisms for CE redundancy or fail over, and it is expected that vendors will provide redundancy or fail over solutions within this framework.

3.2. Forwarding Elements and Fi reference point

An FE is a logical entity that implements the ForCES Protocol and uses the underlying hardware to provide per-packet processing and handling as directed by a CE. It is possible to partition one physical FE into multiple logical FEs. It is also possible for one FE to use multiple physical FEs. The mapping between physical FE(s) and the logical FE(s) is beyond the scope of ForCES. For example, a logical partition of a physical FE can be created by assigning some portion of each of the resources (e.g., ports, memory, forwarding table entries) available on the ForCES physical FE to each of the logical FEs. Such concept of FE virtualization is analogous to a virtual switching element as described in [8]. If FE virtualization occurs only in the pre-association phase, it has no impact on ForCES. However, if FE virtualization results in resource change taken from an existing FE (already participating in ForCES post-association phase), the ForCES Protocol needs to be able to inform the CE of such change via asynchronous messages (see [3], Section 5, requirement #6).

FEs perform all packet processing functions as directed by CEs.

FEs have no initiative of their own. Instead, FEs are slaves and only do as they are told. FEs may communicate with one or more CEs concurrently across reference point Fp. FEs have no notion of CE redundancy, load sharing, or distributed control. Instead, FEs accept commands from any CE authorized to control them, and it is up to the CEs to coordinate among themselves to achieve redundancy, load sharing or distributed control. The idea is to keep FEs as simple and dumb as possible so that FEs can focus their resource on the packet processing functions. Unless otherwise configured or determined by a ForCEs Protocol exchange, each FE will process authorized incoming commands directed at it as it receives them on a first come first serve basis.

For example, in Figure 5, FE1 and FE2 can be configured to accept commands from both the primary CE (CE1) and the backup CE (CE2). Upon detection of CE1 failure, perhaps across the Fr or Fp reference point, CE2 is configured to take over activities of CE1. This is beyond the scope of ForCES and is not discussed further.

Distributed control can be achieved in a similar fashion, without much intelligence on the part of FEs. For example, FEs can be configured to detect RSVP and BGP protocol packets, and forward RSVP packets to one CE and BGP packets to another CE. Hence, FEs may need to do packet filtering for forwarding packets to specific CEs.

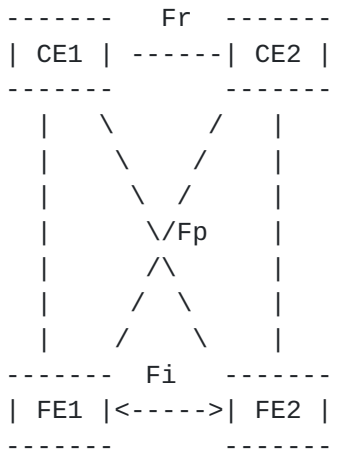


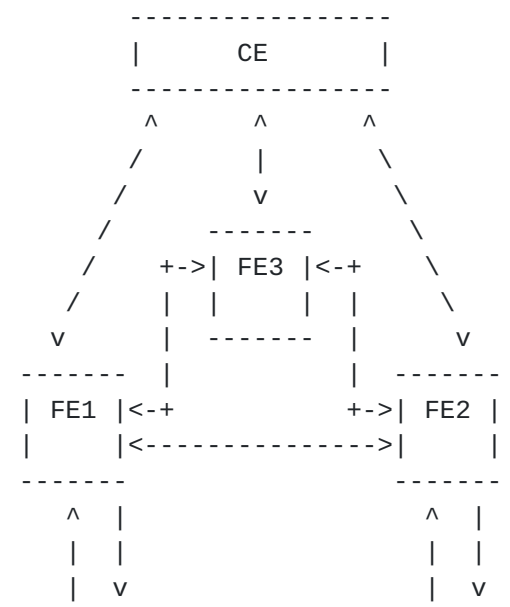
Figure 5. CE redundancy example.

This architecture permits multiple FEs to be present in an NE. [3] dictates that the ForCES Protocol must be able to scale to at least hundreds of FEs (see [3] Section 5, requirement #11). Each of these FEs may potentially have a different set of packet processing functions, with different media interfaces. FEs are responsible

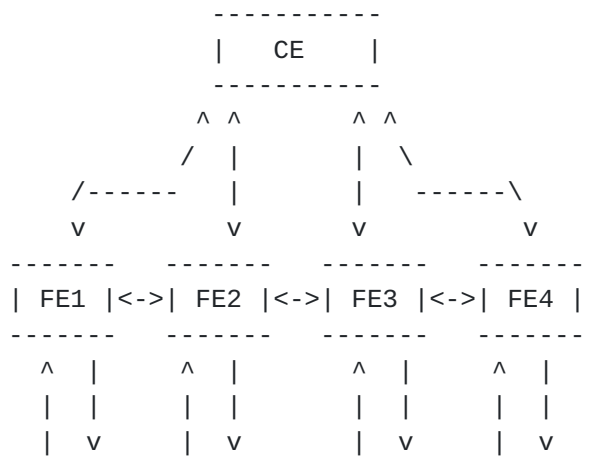
for basic maintenance of layer-2 connectivity with other FEs and with external entities. Many layer-2 media include sophisticated control protocols. The FORCES Protocol (over the Fp reference point) will be able to carry messages for such protocols so that, in keeping with the dumb FE model, the CE can provide appropriate intelligence and control over these media.

When multiple FEs are present, ForCES requires that packets must be able to arrive at the NE by one FE and leave the NE via a different FE (See [3], Section 5, Requirement #3). Packets that enter the NE via one FE and leave the NE via a different FE are transferred between FEs across the Fi reference point. Fi reference point could be used by FEs to discovery their (inter-FE) topology, perhaps during pre-association phase. The Fi reference point is a separate protocol from the Fp reference point and is not currently defined by the ForCES architecture.

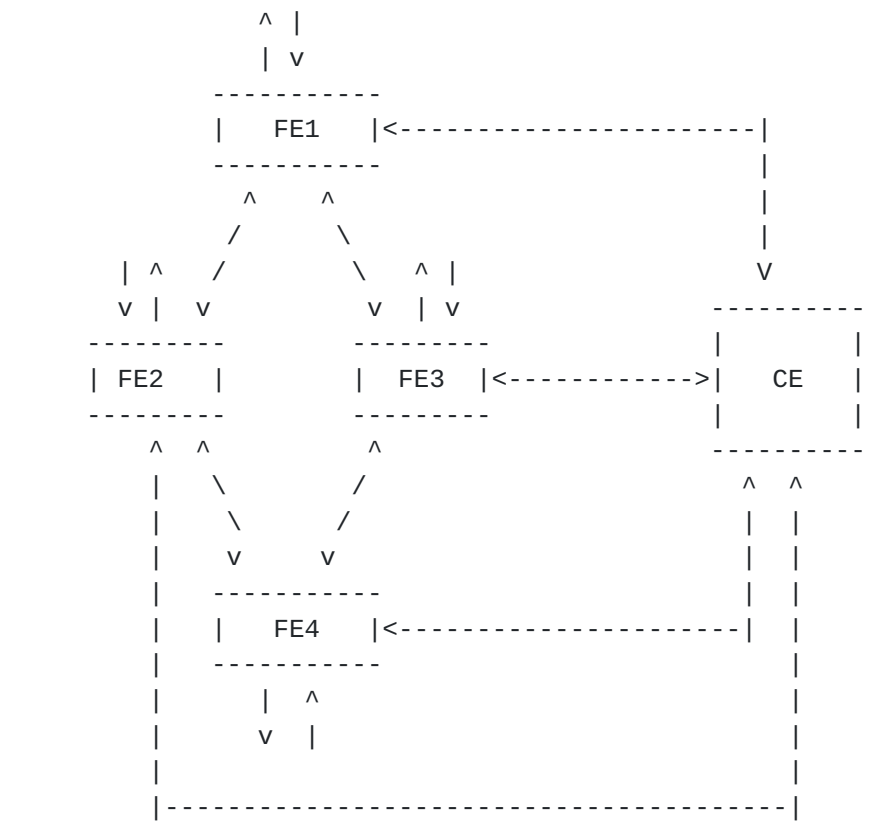
FEs could be connected in different kinds of topologies and packet processing may spread across several FEs in the topology. Hence, logical packet flow may be different from physical FE topology. Figure 6 provides some topology examples. When it is necessary to forward packets between FEs, the CE needs to understand the FE topology. The FE topology may be queried from the FEs by the CEs via ForCES Protocol, but the FEs are not required to provide that information to the CEs. So, the FE topology information may also be gathered by other means outside of the ForCES Protocol (like inter-FE topology discovery protocol).



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



(b) Multiple FEs in a daisy chain



(c) Multiple FEs connected by a ring

Figure 6. Some examples of FE topology.

3.3.CE Managers

CE managers are responsible for determining which FEs a CE should control. It is legitimate for CE managers to be hard-coded with the knowledge of with which FEs its CEs should communicate with. A CE manager may also be physically embedded into a CE and be implemented as a simple keypad or other direct configuration mechanism on the CE. Finally, CE managers may be physically and logically separate entities that configure the CE with FE information via such mechanisms as COPS-PR [6] or SNMP [4].

3.4. FE Managers

FE managers are responsible for determining with which CE any particular FE should initially communicate. Like CE managers, no restrictions are placed on how an FE manager decides with which CE its FEs should communicate, nor are restrictions placed on how FE managers are implemented. Each FE should have one and only one FE manager, while different FEs may have the same or different FE manager(s). Each manager can choose to exist and operate independently of other manager.

4. Operational Phases

Both FEs and CEs require some configuration in place before they can start information exchange and function as a coherent network element. Two operational phases are identified in this framework: pre-association and post-association.

4.1.Pre-association Phase

Pre-association phase is the period of time during which an FE Manager and a CE Manager are determining whether an FE and a CE should be part of the same network element. The protocols used during this phase may include all or some of the message exchange over Fl, Ff and Fc reference points. However, all these may be optional and none of this is within the scope of ForCES Protocol.

4.1.1. Fl Reference Point

CE managers and FE managers may communicate across the Fl reference point in the pre-association phase in order to determine whether an individual CE and FE, or a set of CEs and FEs should be associated. Communication across the Fl reference point is optional in this architecture. No requirements are placed on this reference point.

CE managers and FE managers may be operated by different entities. The operator of the CE manager may not want to divulge, except to specified FE managers, any characteristics of the CEs it manages. Similarly, the operator of the FE manager may not want to divulge FE characteristics, except to authorized entities. As such, CE managers and FE managers may need to authenticate one another. Subsequent communication between CE managers and FE managers may require other security functions such as privacy, non-repudiation, freshness, and integrity.

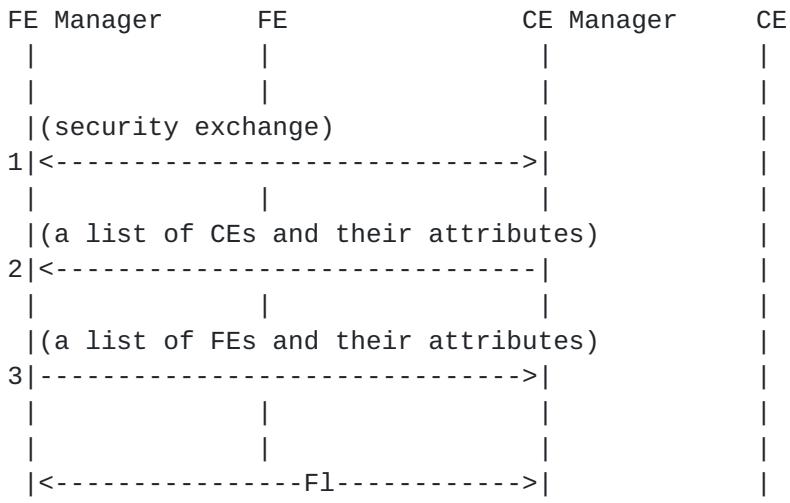


Figure 7. An example of message exchange over Fl reference point

Once the necessary security functions have been performed, the CE and FE managers communicate to determine which CEs and FEs should communicate with each other. At the very minimum, the CE and FE managers need to learn of the existence of available FEs and CEs respectively. This discovery process may entail one or both managers learning the capabilities of the discovered ForCES protocol elements. Figure 7 shows an example of possible message exchange between CE manager and FE manager over Fl reference point.

4.1.2. Ff Reference Point

The Ff reference point is used to inform forwarding elements of the association decisions made by the FE manager in pre-association phase. Only authorized entities may instruct an FE with respect to which CE should control it. Therefore, privacy, integrity, freshness, and authentication are necessary between the FE manager and FEs when the FE manager is remote to the FE. Once the appropriate security has been established, the FE manager instructs the FEs across this reference point to join a new NE or to

disconnect from an existing NE. The FE Manager could also assign unique FE identifiers to the FEs using this reference point. The FE identifiers are useful in post association phase to express FE topology. Figure 8 shows example of message exchange over Ff reference point.

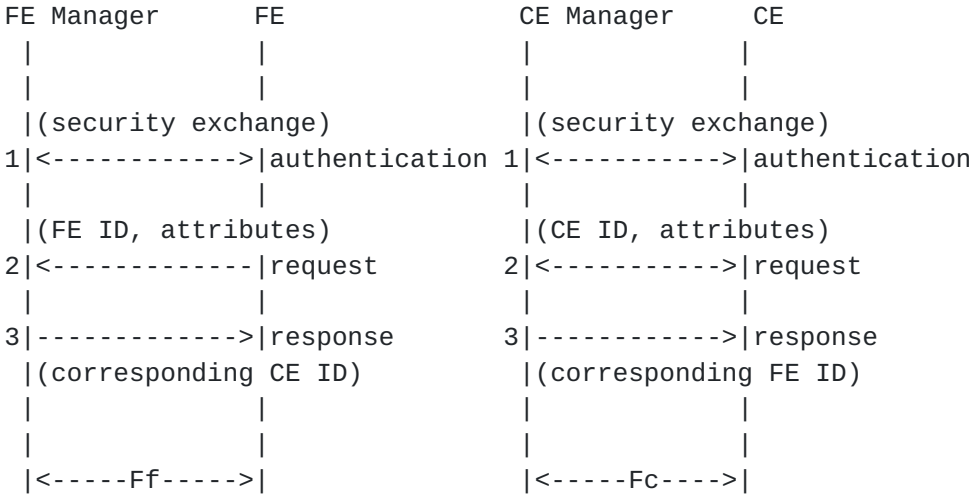


Figure 8. Examples of message exchange over Ff and Fc reference points.

Note that the FE manager function may be co-located with the FE (such as by manual keypad entry of the CE IP address), in which case this reference point is reduced to a built-in function.

4.1.3. Fc Reference Point

The Fc reference point is used to inform control elements of the association decisions made by CE managers in pre-association phase. When the CE manager is remote, only authorized entities may instruct a CE to control certain FEs. Privacy, integrity, freshness and authentication are also required across this reference point in such a configuration. Once appropriate security has been established, the CE manager instructs CEs as to which FEs they should control and how they should control them. Figure 8 shows example of message exchange over Fc reference point.

As with the FE manager and FEs, configurations are possible where the CE manager and CE are co-located and no protocol is used for this function.

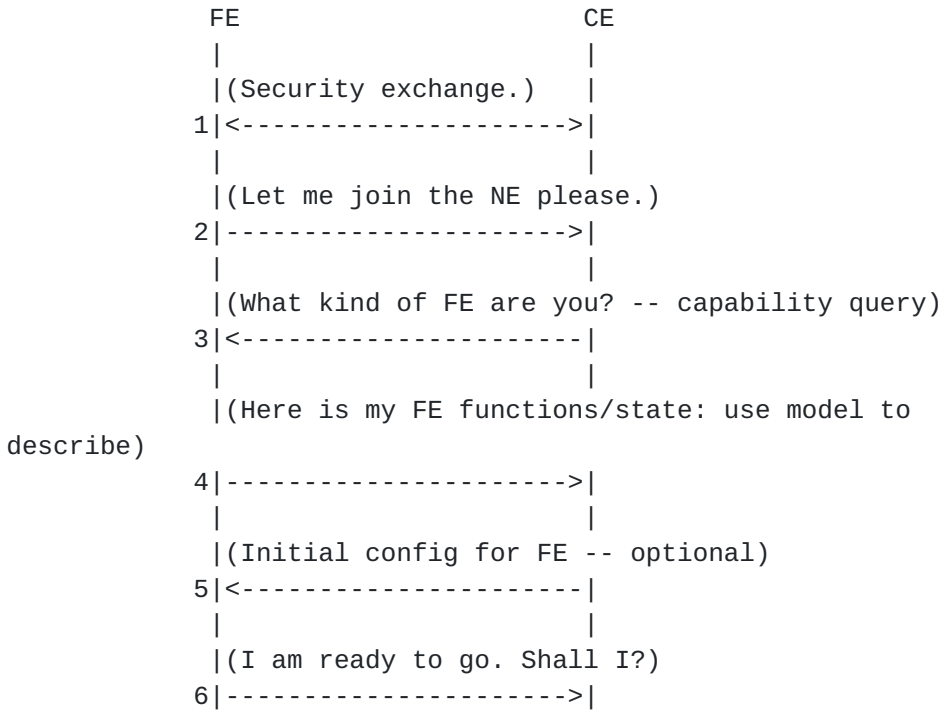
4.2. Post-association Phase and Fp reference point

Post-association phase is the period of time during which an FE and CE have been configured with information necessary to contact each other and includes both association establishment and steady-state communication. The communication between CE and FE is performed across the Fp ("p" meaning protocol) reference point. ForCES Protocol is exclusively used for all communication across the Fp reference point.

4.2.1. Proximity and Interconnect between CEs and FEs

The ForCES Working Group has made a conscious decision that the first version of ForCES will be focused on "very close" CE/FE localities in IP networks. Very Close localities consist of control and forwarding elements that either are components in the same physical box, or are separated at most by one local network hop ([7]). CEs and FEs can be connected by a variety of interconnect technologies, including Ethernet connections, backplanes, ATM (cell) fabrics, etc. ForCES should be able to support each of these interconnects (see [3] Section 5, requirement #1). When the CEs and FEs are separated beyond a single L3 routing hop, the ForCES Protocol will make use of an existing RFC2914 compliant L4 protocol with adequate reliability, security and congestion control (e.g. TCP, SCTP) for transport purposes.

4.2.2. Association Establishment



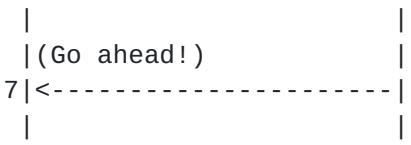


Figure 9. Example of message exchange between CE and FE over Fp to establish NE association

As an example, figure 9 shows some of the message exchange that may happen before the association between the CE and FE is fully established. Either the CE or FE can initiate the connection.

Security handshake is necessary to authenticate the two communication endpoints to each other before any further message exchange can happen. The security handshake should include mutual authentication and authorization between the CE and FE, but the exact details depend on the security solution chosen by ForCES Protocol. Authorization can be as simple as checking against the list of authorized end points provided by the FE or CE manager during the pre-association phase. Both authentication and authorization must be successful before the association can be established. If either authentication or authorization fails, the end point must not be allowed to join the NE. After the successful security handshake, message authentication and confidentiality are still necessary for the on-going information exchange between the CE and FE, unless some form of physical security exists. Whenever a packet fails authentication, it must be dropped and a notification may be sent to alert the sender of the potential attack. Section 8 provides more details on the security considerations for ForCES.

After the successful security handshake, the FE needs to inform the CE of its own capability and optionally its topology in relation to other FEs. The capability of the FE is represented by the FE model, described in a separate document. The model would allow an FE to describe what kind of packet processing functions it contains, in what order the processing happens, what kinds of configurable parameters it allows, what statistics it collects and what events it might throw, etc. Once such information is available to the CE, the CE may choose to send some initial or default configuration to the FE so that the FE can start receiving and processing packets correctly. Such initialization may not be necessary if the FE already obtains the information from its own bootstrap process. Once the necessary initial information is exchanged, the process of

association is completed. Packet processing and forwarding at the FE cannot begin until association is established. After the association is established, the CE and FE enter steady-state communication.

4.2.3. Steady-state Communication

Once an association is established between the CE and FE, the ForCES Protocol is used by the CE and FE over Fp reference point to exchange information to facilitate packet processing.

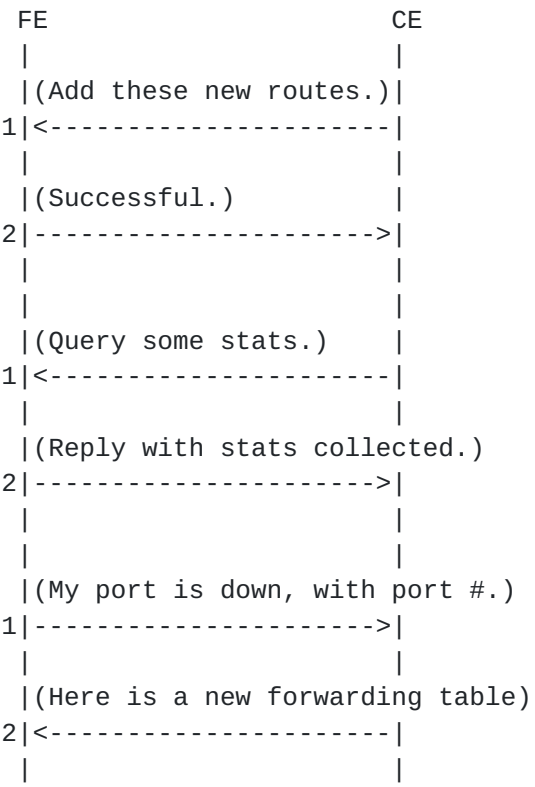


Figure 10. Examples of message exchange between CE and FE over Fp during steady-state communication

Based on the information acquired through CEs' control processing, CEs will frequently need to manipulate the packet-forwarding behaviors of their FE(s) by sending instructions to FEs. For example, Figure 10 shows message exchange examples in which the CE sends new routes to the FE so that the FE can add them to its forwarding table. The CE may query the FE for statistics collected by the FE and the FE may notify the CE of important events such as port failure.

4.2.4. Data Packets across Fp reference point

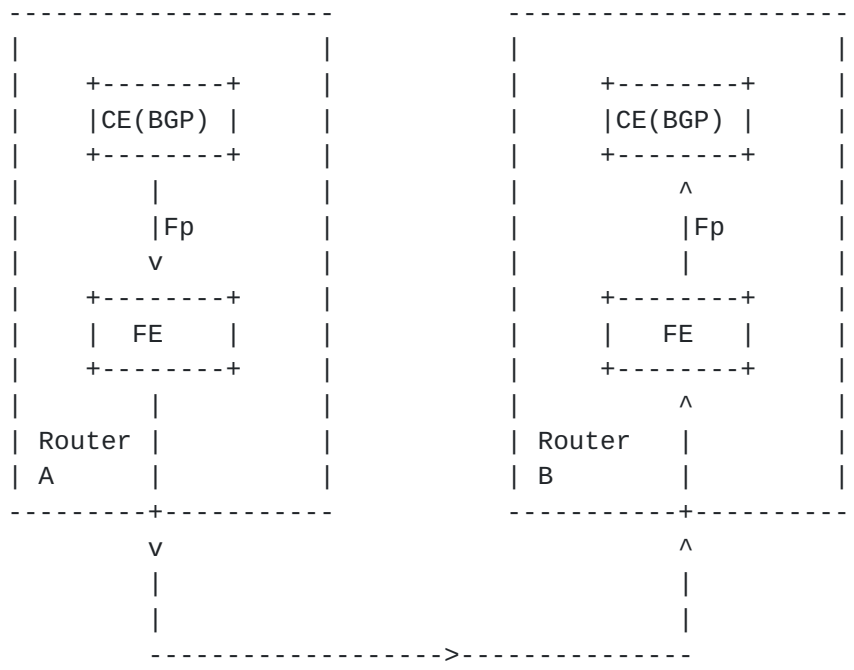


Figure 11. Example to show data packet flow between two NEs.

Control plane protocol packets (such as RIP, OSPF messages) addressed to any of NE's interfaces are typically redirected by the receiving FE to its CE, and CE may originate packets and have its FE deliver them to other NEs. Therefore, ForCES Protocol over Fp not only transports the ForCES Protocol messages between CEs and FEs, but also encapsulates the data packets from control plane protocols. Moreover, one FE may be controlled by multiple CEs for distributed control. In this configuration, the control protocols supported by the FORCES NEs may spread across multiple CEs. For example, one CE may support routing protocols like OSPF and BGP, while a signaling and admission control protocol like RSVP is supported in another CE. FEs are configured to recognize and filter these protocol packets and forward them to the corresponding CE.

Figure 11 shows one example of how the BGP packets originated by router A are passed to router B. In this example, the ForCES Protocol is used to transport the packets from the CE to the FE inside router A, and then from the FE to the CE inside router B. In light of the fact that the ForCES Protocol is responsible for transporting both the control messages and the data packets between the CE and FE over Fp reference point, it is possible to use either a single protocol or multiple protocols to achieve this.

4.2.5. Proxy FE

In the case where a physical FE cannot implement (e.g., due to the lack of a general purpose CPU) the ForCES Protocol directly, a proxy FE can be used to terminate the Fp reference point instead of the physical FE. This allows the CE communicate to the physical FE via the proxy by using ForCES, while the proxy manipulates the physical FE using some intermediary form of communication (e.g., a non-ForCES protocol or DMA). In such an implementation, the combination of the proxy and the physical FE becomes one logical FE entity. It is also possible that one proxy act on behalf of multiple physical FEs.

One needs to be aware of the security implication introduced by the proxy FE. Since the physical FE is not capable of implementing ForCES itself, the security mechanism of ForCES can only secure the communication channel between the CE and the proxy FE, but not all the way to the physical FE. It is recommended that other security mechanisms (including physical security property) be employed to ensure the security between the CE and the physical FE.

4.3. Association Re-establishment

FEs and CEs may join and leave NEs dynamically (see [3] Section 5, requirements #12). When an FE or CE leaves the NE, the association with the NE is broken. If the leaving party rejoins an NE later, to re-establish the association, it may need to re-enter the pre-association phase. Loss of association can also happen unexpectedly due to loss of connection between the CE and the FE. Therefore, the framework allows the bi-directional transition between these two phases, but the ForCES Protocol is only applicable for the post-association phase. However, the protocol should provide mechanisms to support association re-establishment. This includes the ability for CEs and FEs to determine when there is a loss of association between them, ability to restore association and efficient state (re)synchronization mechanisms (see [3] Section 5, requirement #7). Note that security association and state must be also re-established to guarantee the same level of security (including both authentication and authorization) exists before and after the association re-establishment.

When an FE leaves or joins an existing NE that is already in operation, the CE needs to be aware of the impact on FE topology and deals with the change accordingly.

4.3.1. CE graceful restart

The failure and restart of the CE in a router can potentially cause much stress and disruption on the control plane throughout a network. Because when a CE has to restart for any reason, the router loses routing adjacencies or sessions with its routing neighbors. Neighbors who detect the lost adjacency normally re-compute new routes and then send routing updates to their own neighbors to communicate the lost adjacency. Their neighbors do the same thing to propagate throughout the network. In the meantime, the restarting router cannot receive traffic from other routers because the neighbors have stopped using the router's previously advertised routes. When the restarting router restores adjacencies, neighbors must once again re-compute new routes and send out additional routing updates. The restarting router is unable to forward packets until it has re-established routing adjacencies with neighbors, received route updates through these adjacencies, and computed new routes. Until convergence takes place throughout the network, packets may be lost in transient black holes or forwarding loops.

A high availability mechanism known as the "graceful restart" has been used by the IP routing protocols (OSPF [10], BGP [11]) and MPLS label distribution protocol (LDP [9]) to help minimize the negative effects on routing throughout an entire network caused by a restarting router. Route flap on neighboring routers is avoided, and a restarting router can continue to forward packets that would otherwise be dropped.

While the details differ from protocol to protocol, the general idea behind the graceful restart mechanism remains the same. With the graceful restart, a restarting router can inform its neighbors when it restarts. The neighbors may detect the lost adjacency but do not recompute new routes or send routing updates to their neighbors. The neighbors also hold on to the routes received from the restarting router before restart and assume they are still valid for a limited time. By doing so, the restarting router's FEs can also continue to receive and forward traffic from other neighbors for a limited time by using the routes they already have. The restarting router then re-establishes routing adjacencies, downloads updated routes from all its neighbors, recomputes new routes and uses them to replace the older routes it was using. It then sends these updated routes to its neighbors and signals the completion of the graceful restart process.

Non-stop forwarding is a requirement for graceful restart. It is necessary so a router can continue to forward packets while it is downloading routing information and recomputing new routes. This ensures that packets will not be dropped. As one can see, one of

the benefits afforded by the separation of CE and FE is exactly the ability of non-stop forwarding in the face of the CE failure and restart. The support of dynamic changes to CE/FE association in ForCES also makes it compatible with high availability mechanisms such as graceful restart.

ForCES should be able to support CE graceful restart easily. When the association is established the first time, the CE must inform the FEs what to do in the case of CE failure. If graceful restart is not supported, the FEs may be told to stop packet processing all together if its CE fails. If graceful restart is supported, the FEs should be told to cache and hold on to its FE state including the forwarding tables across the restarts. A timer must be included so that the timeout causes such cached state to expire eventually. Those timers should be settable by the CE.

4.3.2. FE restart

In the same example in Figure 5, assuming CE1 is the working CE for the moment, what would happen if one of the FEs, say FE1, leaves the NE temporarily? FE1 may voluntarily decide to leave the association. Alternatively, FE1 may stop functioning simply due to unexpected failure. In the former case, CE1 receives a "leave-association request" from FE1. In the latter, CE1 detects the failure of FE1 by some other means. In both cases, CE1 must inform the routing protocols of such an event, most likely prompting a reachability and SPF (Shortest Path First) recalculation and associated downloading of new FIBs from CE1 to the other remaining FEs (only FE2 in this example). Such recalculation and FIB update will also be propagated from CE1 to the NE's neighbors that are affected by the connectivity of FE1.

When FE1 decides to rejoin again, or when it restarts again from the failure, FE1 needs to re-discover its master (CE). This can be achieved by several means. It may re-enter the pre-association phase and get that information from its FE manager. It may retrieve the previous CE information from its cache, if it can validate the information freshness. Once it discovers its CE, it starts message exchange with the CE to re-establish the association just as outlined in Figure 9, with the possible exception that it might be able to bypass the transport of the complete initial configuration. Suppose that FE1 still has its routing table and other state information from the last association. Instead of sending all the information again from scratch, it may be able to use more efficient mechanism to re-sync up the state with its CE if such mechanism is supported by the ForCES Protocol. For example, CRC-32 of the state might give a quick indication of whether or not

the state is in-sync with its CE. By comparing its state with the CE first, it sends an information update only if it is needed. ForCES Protocol may choose to implement similar optimization mechanisms, but it may also choose not to, as this is not a requirement.

5. Applicability to RFC1812

[3] Section 5, requirement #9 dictates "Any proposed ForCES architecture must explain how that architecture supports all of the router functions as defined in RFC1812." RFC1812 discusses many important requirements for IPv4 routers from the link layer to the application layer. This section addresses the relevant requirements in RFC1812 for implementing IPv4 routers based on ForCES architecture and explains how ForCES satisfies these requirements by providing guidelines on how to separate the functionalities required into forwarding plane and control plane.

In general, the forwarding plane carries out the bulk of the per-packet processing that is required at line speed, while the control plane carries most of the computationally complex operations that are typical of the control and signaling protocols. However, it is impossible to draw a rigid line to divide the processing into CEs and FEs cleanly. Nor should the ForCES architecture limit the innovative approaches in control and forwarding plane separation. As more and more processing power is available in the FEs, some of the control functions that traditionally are performed by CEs may now be moved to FEs for better performance and scalability. Such offloaded functions may include part of ICMP or TCP processing, or part of routing protocols. Once off-loaded onto the forwarding plane, such CE functions, even though logically belonging to the control plane, now become part of the FE functions. Just like the other logical functions performed by FEs, such off-loaded functions must be expressed as part of the FE model so that the CEs can decide how to best take advantage of these off-loaded functions when present on the FEs.

5.1. General Router Requirements

Routers have at least two or more logical interfaces. When CEs and FEs are separated by ForCES within a single NE, some additional interfaces are needed for intra-NE communications. Figure 12 shows an example to illustrate that. This NE contains one CE and two FEs. Each FE has four interfaces; two of them are used for receiving and transmitting packets to the external world, while the other two are for intra-NE connections. CE has two logical

interfaces #9 and #10, connected to interfaces #3 and #6 from FE1 and FE2, respectively. Interface #4 and #5 are connected for FE1-FE2 communication. Therefore, this router NE provides four external interfaces (#1, 2, 7 and 8).

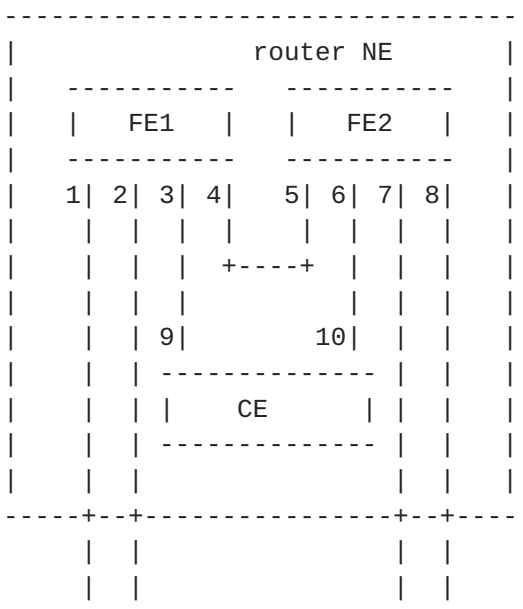


Figure 12. A router NE example with four interfaces.

IPv4 routers must implement IP to support its packet forwarding function, which is driven by its FIB (Forwarding Information Base). This Internet layer forwarding (see RFC1812 [1] Section 5) functionality naturally belongs to FEs in the ForCES architecture.

A router may implement transport layer protocols (like TCP and UDP) that are required to support application layer protocols (see RFC1812 [1] Section 6). One important class of application protocols is routing protocols (see RFC1812 [1] Section 7). In ForCES architecture, routing protocols are naturally implemented by CEs. Routing protocols require routers communicate with each other. This communication between CEs in different routers is supported in ForCES by FEs' ability to redirect data packets addressed to routers (i.e., NEs) and CEs' ability to originate packets and have them delivered by their FEs. This communication occurs across Fp reference point inside each router and between neighboring routers' external interfaces, as illustrated in Figure 11.

5.2.Link Layer

Since FEs own all the external interfaces for the router, FEs need to conform to the link layer requirements in RFC1812. Arguably, ARP support may be implemented in either CEs or FEs. As we will see later, a number of behaviors that RFC1812 mandates fall into this category -- they may be performed by the FE and may be performed by the CE. A general guideline is needed to ensure interoperability between separated control and forwarding planes. The guideline we offer here is that CEs MUST be capable of these kind of operations while FEs MAY choose to implement them. FE model should indicate its capabilities in this regard so that CEs can decide where these functions are implemented.

Interface parameters, including MTU, IP address, etc., must be configurable by CEs via ForCES. CEs must be able to determine whether a physical interface in an FE is available to send packets or not. FEs must also inform CEs the status change of the interfaces (like link up/down) via ForCES.

5.3.Internet Layer Protocols

Both FEs and CEs must implement IP protocol and all mandatory extensions as RFC1812 specified. CEs should implement IP options like source route and record route while FEs may choose to implement those as well. The timestamp option should be implemented by FEs to insert the timestamp most accurately. The FE must interpret the IP options that it understands and preserve the rest unchanged for use by CEs. Both FEs and CEs might choose to silently discard packets without sending ICMP errors, but such events should be logged and counted. FEs may report statistics for such events to CEs via ForCES.

When multiple FEs are involved to process packets, the appearance of single NE must be strictly maintained. For example, Time-To-Live (TTL) must be decremented only once within a single NE. For example, it can be always decremented by the last FE with egress function.

FEs must receive and process normally any packets with a broadcast destination address or a multicast destination address that the router has asked to receive. When IP multicast is supported in routers, IGMP is implemented in CEs. CEs are also required of ICMP support, while it is optional for FEs to support ICMP. Such an option can be communicated to CEs as part of the FE model. Therefore, FEs can always rely upon CEs to send out ICMP error messages, but FEs also have the option to generate ICMP error messages themselves.

5.4.Internet Layer Forwarding

IP forwarding is implemented by FEs. When the routing table is updated at CEs, ForCES is used to send the new route entries from CEs to FEs. Each FE has its own forwarding table and uses this table to direct packets to the next hop interface.

Upon receiving IP packets, the FE verifies the IP header and processes most of the IP options. Some options cannot be processed until the routing decision has been made. The routing decision is made after examining the destination IP address. If the destination address belongs to the router itself, the packets are filtered and either processed locally or forwarded to CE, depending upon the instructions set-up by CE. Otherwise, the FE determines the next hop IP address by looking up in its forwarding table. The FE also determines the network interface it uses to send the packets. Sometimes an FE may need to forward the packets to another FE before packets can be forwarded out to the next hop. Right before packets are forwarded out to the next hop, the FE decrements TTL by 1 and processes any IP options that cannot be processed before. The FE performs any IP fragmentation if necessary, determines link layer address (e.g., by ARP), and encapsulates the IP datagram (or each of the fragments thereof) in an appropriate link layer frame and queues it for output on the interface selected.

Other options mentioned in RFC1812 for IP forwarding may also be implemented at FEs, for example, packet filtering.

FEs typically forward packets destined locally to CEs. FEs may also forward exceptional packets (packets that FEs do not know how to handle) to CEs. CEs are required to handle packets forwarded by FEs for whatever different reasons. It might be necessary for ForCES to attach some meta-data with the packets to indicate the reasons of forwarding from FEs to CEs. Upon receiving packets with meta-data from FEs, CEs can decide to either process the packets themselves, or pass the packets to the upper layer protocols including routing and management protocols. If CEs are to process the packets by themselves, CEs may choose to discard the packets, or modify and re-send the packets. CEs may also originate new packets and deliver them to FEs for further forwarding.

Any state change during router operation must also be handled correctly according to RFC1812. For example, when an FE ceases forwarding, the entire NE may continue forwarding packets, but it needs to stop advertising routes that are affected by the failed FE.

5.5. Transport Layer

Transport layer is typically implemented at CEs to support higher layer application protocols like routing protocols. In practice, this means that most CEs implement both the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

Both CEs and FEs need to implement ForCES Protocol. If some layer-4 transport is used to support ForCES, then both CEs and FEs need to implement the L4 transport and ForCES Protocols.

5.6. Application Layer -- Routing Protocols

Interior and exterior routing protocols are implemented on CEs. The routing packets originated by CEs are forwarded to FEs for delivery. The results of such protocols (like forwarding table updates) are communicated to FEs via ForCES.

For performance or scalability reasons, portions of the control plane functions that need faster response may be moved from the CEs and off-loaded onto the FEs. For example in OSPF, the Hello protocol packets are generated and processed periodically. When done at CEs, the inbound Hello packets have to traverse from the external interfaces at the FEs to the CEs via the internal CE-FE channel. Similarly, the outbound Hello packets have to go from the CEs to the FEs and to the external interfaces. Frequent Hello updates place heavy processing overhead on the CEs and can overwhelm the CE-FE channel as well. Since typically there are far more FEs than CEs in a router, the off-loaded Hello packets are processed in a much more distributed and scalable fashion. By expressing such off-loaded functions in the FE model, we can ensure interoperability. However, the exact description of the off-loaded functionality corresponding to the off-loaded functions expressed in the FE model are not part of the model itself and will need to be worked out as a separate specification.

5.7. Application Layer -- Network Management Protocol

RFC1812 also dictates "Routers MUST be manageable by SNMP." In general, for post-association phase, most external management tasks (including SNMP) should be done through interaction with the CE in order to support the appearance of a single functional device. Therefore, it is recommended that SNMP agent be implemented by CEs and the SNMP messages received by FEs be redirected to their CEs. AgentX framework defined in RFC2741 ([5]) may be applied here such that CEs act in the role of master agent to process SNMP protocol

messages while FEs act in the role of subagent to provide access to the MIB objects residing on FEs. AgentX protocol messages between the master agent (CE) and the subagent (FE) are encapsulated and transported via ForCES, just like data packets from any other application layer protocols.

6. Summary

This document defines an architectural framework for ForCES. It identifies the relevant components for a ForCES network element, including (one or more) FEs, (one or more) CEs, one optional FE manager, and one optional CE manager. It also identifies the interaction among these components and discusses all the major reference points. It is important to point out that, among all the reference points, only the Fp interface between CEs and FEs is within the scope of ForCES. ForCES alone may not be enough to support all desirable NE configurations. However, we believe that ForCES over Fp interface is the most important element in realizing physical separation and interoperability of CEs and FEs, and hence the first interface that ought to be standardized. Simple and useful configurations can still be implemented with only CE-FE interface being standardized, e.g., single CE with full-meshed FEs.

7. Acknowledgements

Joel M. Halpern gave us many insightful comments and suggestions and pointed out several major issues. T. Sridhar suggested that the AgentX protocol could be used with SNMP to manage the ForCES network elements. Susan Hares pointed out the issue of graceful restart with ForCES. Russ Housley, Avri Doria, Jamal Hadi Salim and many others in the ForCES mailing list also provided valuable feedback.

8. Security Considerations

The NE administrator has the freedom to determine the exact security configuration that is needed for the specific deployment. For example, ForCES may be deployed between CEs and FEs connected to each other inside a box over a backplane. In such scenario, physical security of the box ensures that most of the attacks such as man-in-the-middle, snooping, and impersonation are not possible, and hence ForCES architecture may rely on the physical security of the box to defend against these attacks and protocol mechanisms may be turned off. However, it is also shown that denial of service attack via external interface as described below in Section 8.1.8 is still a potential threat even for such "all-in-one-box" deployment scenario and hence the rate limiting mechanism is still

necessary. This is just one example to show that it is important to assess the security needs of the ForCES-enabled network elements under different deployment scenarios. It should be possible for the administrator to configure the level of security needed for the ForCES Protocol.

In general, the physical separation of two entities usually results in a potentially insecure link between the two entities and hence much stricter security measurements are required. For example, we pointed out in Section 4.1 that authentication becomes necessary between CE manager and FE manager, between CE and CE manager, between FE and FE manager in some configurations. The physical separation of CE and FE also imposes serious security requirement for ForCES Protocol over Fp interface. This section first attempts to describe the security threats that may be introduced by the physical separation of the FEs and the CEs, and then it provides recommendation and guidelines for secure operation and management of ForCES Protocol over Fp interface based on existing standard security solutions.

8.1. Analysis of Potential Threats Introduced by ForCES

This section provides the threat analysis for ForCES, with a focus on Fp interface. Each threat is described in details with the effects on the ForCES Protocol entities or/and the NE as a whole, and the required functionalities that need to be in place to defend the threat.

8.1.1. "Join" or "Remove" Message Flooding on CEs

Threats: A malicious node could send a stream of false "join NE" or "remove from NE" requests on behalf of non-existent or unauthorized FE to legitimate CEs at a very rapid rate and thereby create unnecessary state in the CEs.

Effects: If by maintaining state for non-existent or unauthorized FEs, a CE may become unavailable for other processing and hence suffer from denial of service (DoS) attack similar to the TCP SYN DoS. If multiple CEs are used, the unnecessary state information may also be conveyed to multiple CEs via Fr interface (e.g., from the active CE to the stand-by CE) and hence subject multiple CEs to DoS attack.

Requirement: A CE that receives a "join" or "remove" request should not create any state information until it has authenticated the FE endpoint.

8.1.2. Impersonation Attack

Threats: A malicious node can impersonate a CE or FE and send out false messages.

Effects: The whole NE could be compromised.

Requirement: The CE or FE must authenticate the message as having come from an FE or CE on the list of the authorized ForCES elements (provided by the CE or FE Manager in the pre-association phase) before accepting and processing it.

8.1.3. Replay Attack

Threat: A malicious node could replay the entire message previously sent by an FE or CE entity to get around authentication.

Effect: The NE could be compromised.

Requirement: Replay protection mechanism needs to be part of the security solution to defend against this attack.

8.1.4. Attack during Fail Over

Threat: A malicious node may exploit the CE fail-over mechanism to take over the control of NE. For example, suppose two CEs, say CE-A and CE-B, are controlling several FEs. CE-A is active and CE-B is stand-by. When CE-A fails, CE-B is taking over the active CE position. The FEs already had a trusted relationship with CE-A, but the FEs may not have the same trusted relationship established with CE-B prior to the fail-over. A malicious node can take over as CE-B if such trusted relationship has not been established prior or during the fail-over.

Effect: The NE may be compromised after such insecure fail-over.

Requirement: The level of trust relationship between the stand-by CE and the FEs must be as strong as the one between the active CE and the FEs. The security association between the FEs and the stand-by CE may be established prior to fail-over. If not already in place, such security association must be re-established before the stand-by CE takes over.

8.1.5. Data Integrity

Threats: A malicious node may inject false messages to legitimate CE or FE.

Effect: An FE or CE receives the fabricated packet and performs incorrect or catastrophic operation.

Requirement: Protocol messages require integrity protection.

8.1.6. Data Confidentiality

Threat: When FE and CE are physically separated, a malicious node may eavesdrop the messages in transit. Some of the messages are critical to the functioning of the whole network, while others may contain confidential business data. Leaking of such information may result in compromise even beyond the immediate CE or FE.

Effect: Sensitive information might be exposed between CE and FE.

Requirement: Data confidentiality between FE and CE must be available for sensitive information.

8.1.7. Sharing security parameters

Threat: Consider a scenario where several FEs communicating to the same CE share the same authentication keys for the Fp interface. If any FE or the CE is compromised, all other entities are compromised.

Effect: The whole NE is compromised.

Recommendation: To avoid this side effect, it's better to configure different security parameters for each FE-CE communication over Fp interface.

8.1.8. Denial of Service Attack via External Interface

Threat: When an FE receives a packet that is destined for its CE, the FE forwards the packet over the Fp interface. Malicious node can generate huge message storm like routing protocol packets etc. through the external Fi/f interface so that the FE has to process and forward all packets to CE through Fp interface.

Effect: CE encounters resource exhaustion and bandwidth starvation on Fp interface due to an overwhelming number of packets from FEs.

Requirement: Some sort of rate limiting mechanism MUST to be in place at both the FE and CE. Rate Limiter SHOULD be configured at

the FE for each message type that are being received through Fi/F interface.

8.2. Security Recommendations for ForCES

The requirements document [3] suggested that ForCES Protocol should support reliability over Fp interface, but no particular transport protocol is yet specified for ForCES. This framework document does not intend to specify the particular transport either, and so we only provide recommendations and guidelines based on the existing standard security protocols [18] that can work with the common transport candidates suitable for ForCES.

We review two existing security protocol solutions, namely IPsec (IP Security) [14] or TLS (Transport Layer Security) [13]. TLS works with reliable transports such as TCP or SCTP for unicast, while IPsec can be used with any transport (UDP, TCP, SCTP) and supports both unicast and multicast. Both TLS and IPsec can be used potentially to satisfy all of the security requirements for ForCES Protocol. In addition, other approaches may be used as well but are not documented here, including using L2 security mechanisms for a given L2 interconnect technology, as long as the requirements can be satisfied.

When ForCES is deployed between CEs and FEs inside a box or a physically secured room, authentication, confidentiality and integrity may be provided by the physical security of the box and so the security mechanisms may be turned off, depending on the networking topology and its administration policy. However, it is important to realize that even if the NE is in a single-box, the DoS attacks as described in Section 8.1.8 can still be launched through Fi/f interfaces. Therefore, it is important to have the corresponding counter-measurement in place even for single-box deployment.

8.2.1. Using TLS with ForCES

TLS [13] can be used if a reliable unicast transport such as TCP or SCTP is used for ForCES over the Fp interface. The TLS handshake protocol is used during association establishment or re-establishment phase to negotiate a TLS session between the CE and FE. Once the session is in place, the TLS record protocol is used to secure ForCES communication messages between the CE and FE.

A basic outline of how TLS can be used with ForCES is described below. Steps 1) till 7) complete the security handshake as illustrated in Figure 9 while step 8) is for all the further

communication between the CE and FE, including the rest of messages after the security handshake shown in Figure 9 and the steady-state communication shown in Figure 10.

1) During Pre-association phase all FEs are configured with the CEs (including both the active CE and the standby CE).
2) The FE establishes a TLS connection with the CE (master) and negotiates a cipher suite.
3) The FE (slave) gets the CE certificate, validates the signature, checks the expiration date, checks if the certificate has been revoked.
4) The CE (master) gets the FE certificate and performs the same validation as the FE in step 3).
5) If any of the check fails in step 3) or step 4), endpoint must generate an error message and abort.
6) After successful mutual authentication, a TLS session is established between CE and FE.
7) The FE sends a "join NE" message to the CE.
8) The FE and CE use TLS session for further communication.

Note that there are different ways for the CE and FE to validate a received certificate. One way is to configure the FE Manager or CE Manager or other central component as CA, so that the CE or FE can query this pre-configured CA to validate that the certificate has not been revoked. Another way is to have the CE and the FE configured directly a list of valid certificates in the pre-association phase.

In the case of fail-over, it is the responsibility of the active CE and the standby CE to synchronize ForCES states including the TLS states to minimize the state reestablishment during fail-over. Care must be taken to ensure that the standby CE is also authenticated in the same way as the active CE, either before or during the fail-over.

8.2.2. Using IPsec with ForCES

IPsec [14] can be used with any transport protocol, such as UDP, SCTP and TCP over Fp interface for ForCES. When using IPsec, we recommend using ESP in transport mode for ForCES because message confidentiality is required for ForCES.

IPsec can be used with both manual and automated SA and cryptographic key management. But IPsec's replay protection mechanisms are not available if manual key management is used. Hence, automatic key management is recommended if replay protection is deemed important. Otherwise, manual key management might be

sufficient for some deployment scenarios, esp. when the number of CEs and FEs is relatively small. It is recommended that the keys be changed periodically even for manual key management.

IPsec can support both unicast and multicast transport. At the time this document was published, MSEC working group is actively working on standardizing protocols to provide multicast security [17]. Multicast-based solutions relying on IPsec should specify how to meet the security requirements in [3].

Unlike TLS, IPsec provides security services between the CE and FE at IP level, and so the security handshake as illustrated in Figure 9 amounts to a "no-op" when manual key management is used. The following outline the steps taken for ForCES in such a case.

1) During Pre-association phase all FEs are configured with the CEs (including active CE and standby CE) and SA parameters manually.
2) The FE sends a "join NE" message to the CE. This message and all others that follow are afforded security service according to the manually configured IPsec SA parameters, but replay protection is not available.

It is up to the administrator to decide whether to share the same key across multiple FE-CE communication, but it is recommended that different keys be used. Similarly, it is recommended that different keys be used for inbound and outbound traffic.

If automatic key management is needed, IKE [15] can be used for that purpose. Other automatic key distribution techniques such as Kerberos may be used as well. The key exchange process constitutes the security handshake as illustrated in Figure 9. The following shows the steps involved in using IKE with IPsec for ForCES. Steps 1) to 6) constitute the security handshake in Figure 9.

1) During Pre-association phase all FEs are configured with the CEs (including active CE and standby CE), IPsec policy etc.
2) The FE kicks off IKE process and tries to establish an IPsec SA with the CE (master). The FE (Slave) gets the CE certificate as part of the IKE negotiation. The FE validates signature, checks the expiration date, checks if the certificate has been revoked.
3) The CE (master) gets the FE certificate and performs the same check as the FE in step 2).

4) If any of the check fails in step 2) or step 3), the endpoint must generate an error message and abort.
5) After successful mutual authentication, IPsec session is established between the CE and FE.
6) The FE sends a "join NE" message to CE. No SADB entry is created in FE yet.
7) The FE and CE use the IPsec session for further communication.

FE Manager or CE Manager or other central component can be used as CA for validating CE and FE certificates during the IKE process. Alternatively, during the pre-association phase, the CE and FE can be configured directly with the required information such as certificates or passwords etc depending upon the type of authentication that administrator wants to configure.

In the case of fail-over, it is the responsibility of active CE and standby CE to synchronize ForCES states and IPsec states to minimize the state reestablishment during fail-over. Alternatively, the FE needs to establish different IPsec SA during the startup operation itself with each CE. This will minimize the periodic state transfer across IPsec layer though Fr (CE-CE) Interface.

9. Normative References

[1] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[2] Floyd, S., "Congestion Control Principles", RFC 2914, September 2000.

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

10. Informative References

[4] Case, J., et al., "Introduction and Applicability Statements for Internet Standard Management Framework", RFC 3410, December 2002.

[5] Daniele, M. et al., "Agent Extensibility (AgentX) Protocol Version 1", RFC 2741, January 2000.

[6] Chan, K. et al., "COPS Usage for Policy Provisioning (COPS-PR)", RFC 3084, March 2001.

[7] Crouch, A. et al., "ForCES Applicability Statement", work in progress, June 2003, <draft-ietf-forces-applicability-02.txt>.

[8] Anderson, T. and J. Buerkle, "Requirements for the Dynamic Partitioning of Switching Elements", RFC 3532, May 2003.

[9] Leelanivas, M. et al., "Graceful Restart Mechanism for Label Distribution Protocol", RFC 3478, February 2003.

[10] Moy, J. et al., "Graceful OSPF Restart", RFC 3623, November 2003.

[11] Sangli, S. et al., "Graceful Restart Mechanism for BGP", work in progress, September 2003, < draft-ietf-idr-restart-08.txt>.

[12] Shand, M. and L. Ginsberg, "Restart Signaling for IS-IS", work in progress, July 2003, <draft-ietf-isis-restart-04.txt>.

[13] Dierks, T. and C. Allen, "The TLS Protocol, version 1.0", RFC 2246, January 1999.

[14] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[15] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE) ", RFC 2409, November 1998.

[16] Bellovin, S., "Guidelines for Mandating the Use of Ipsec", work in progress, October 2003, <draft-bellovin-useipsec-02.txt>.

[17] Hardjono, T. and Weis, B. "The Multicast Security Architecture", work in progress, November 2003, <draft-ietf-msec-arch-04.txt>.

[18] S. Bellovin, J. Schiller, and C. Kaufman, "Security Mechanisms for the Internet", RFC 3631, December, 2003.

11. Authors' Addresses

L. Lily Yang
Intel Corp., MS JF3-206,
2111 NE 25th Avenue
Hillsboro, OR 97124, USA
Phone: +1 503 264 8813
Email: lily.l.yang@intel.com

Ram Dantu

Department of Computer Science,
University of North Texas,
Denton, TX 76203, USA
Phone: +1 940 565 2822
Email: rdantu@unt.edu

Todd A. Anderson
Intel Corp.
2111 NE 25th Avenue
Hillsboro, OR 97124, USA
Phone: +1 503 712 1760
Email: todd.a.anderson@intel.com

Ram Gopal
Nokia Research Center
5, Wayside Road,
Burlington, MA 01803, USA
Phone: +1 781 993 3685
Email: ram.gopal@nokia.com

12. Intellectual Property Right

13. Full Copyright Statement