BBN Technical Memorandum No. 1245

FIRE State Message Protocol Specification

Job No. 11871003

July 12, 2000

Prepared for:
Defense Advanced Research Projects Agency
Information Technology Office
3701 North Fairfax Drive
Arlington, VA 22203-1714

Prepared by:
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
FIRE State Message Protocol Specification

C. Partridge (craig@bbn.com)  I. Castineyra (isidro@bbn.com)  T. Strayer (strayer@bbn.com)  
A. Snoeren (snoeren@bbn.com) B. Schwartz (bschwartz@bbn.com)  

Abstract

This memo specifies FIRE State Message formats and protocol. FIRE, the Flexible Intra-AS Routing Environment, is a flexible routing system for use within an Autonomous System (AS). Through exchange of state advertisements containing properties associated with links and nodes, each routing node in the AS can build an identical Property Repository. Various routing algorithms are applied to this repository to produce forwarding tables. Filters applied to incoming packets determine which forwarding table is appropriate, permitting various criteria to determine the path of a packet through the network. FIRE State Messages are the mechanism for exchanging the properties that are used to build the Property Repository.

1. Introduction

FIRE, the Flexible Intra-AS Routing Environment, is a flexible routing system for use within an Autonomous System (AS) [FIREarch]. All entities in an Autonomous System are either hosts, nodes, or links. Since hosts do not actively participate in the routing, only nodes and links have properties associated with them. These properties, along with connectivity information, are exchanged between all nodes to build a Property Repository. The Property Repository is central to producing the router's forwarding tables.

State Advertisements (carried by the State Messages described in this document) are collected and applied to the Property Repository. The State Advertisements tell the database what nodes are connected via links to which other nodes, as well as the set of properties associated with the node or link. Each link and node is conceptually labeled with one or more properties. These properties can correspond to quantities such as bandwidth, delay, and ownership rights.

A reliable flooding protocol is used to disseminate State Advertisements. The result is that each router of the network will have a description of the AS's FIRE network — including connectivity information and the properties for all the routers and links of the network.

An Operator Configuration File (OCF) [FIREconf] describes the properties to be advertised in an AS. Multiple OCFs can be loading, that is, in the process of being obtained, or advertising, that is, exchanging SAs, but only one will be running, that is, being used to route traffic. By allowing multiple OCFs to be advertised, a FIRE operator can have all properties associated with a new OCF distributed before choosing to route with it.
All participants in a FIRE Autonomous System MUST have a concept of wall clock time, and be set with an accuracy of 5 minutes. Once a FIRE AS is booted, each participant SHOULD use some method for clock synchronization.

1.1 Status of this Memo

Discussion and suggestions for improvement are requested. Please refer to the current edition of the "IANA Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

This document is available in HTML, ASCII and Postscript.

Please send comments to fire@ir.bbn.com.

This document is in draft form and is incomplete.

1.2 Notational Conventions

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 [RFC2119].

This section lists a few conventions used through the text.

1.2.1 Byte Order

All protocol fields in SA Messages must be sent in network standard byte order, where the most significant byte of a 32-bit field is sent first. All header and control information follows this rule.

1.2.2 Names

The following terms refer to data objects within a message: "word" indicates a 4-octet (32-bit) object, "field" refers to an object of any size, and "bitflag" denotes a 1-bit field contained within another field.

The bits in a field are numbered left to right from most significant (high order, number 0) to least significant (low order, number n). Likewise with octets. Bits referred to by name are annotated in capital letters.

The number of bytes in a field is given by a number in parenthesis (e.g., flags(2)). When a field has variable width, a variable such as n is used as the number of octets or a factor of the number of octets (e.g., name(n) or name(8n)). The number of bits in a bit field is given by a number following the field name and a colon (e.g., bit:1).

1.2.3 Inequalities

Throughout the text, comparison of the magnitude of one unsigned integer with another follows this convention: The value of A is said to be "less than" the value of B if

\[(A - B - 1) < 2^{**n-1}\]

where \(n\) is the number of bits in the field. Note that the subtraction is an unsigned integer operation. This rule handles the logic when B rolls over (overflows and returns to 0) before A does, and assumes that any two values, when compared, will never have a distance of more than half of the number space.

2. Concepts
2.1 Nodes and Links

Entities in a FIRE AS are all nodes or links. Routers are router nodes. Broadcast media are network nodes. Hosts are passive nodes. Links connect two nodes, either a point-to-point link between two routers and/or hosts, or a connection between a router or host and broadcast media. Links in a FIRE AS are always unidirectional. Two FIRE links are used to represent a bidirectional link between two nodes.

A link is represented by the nodes on each end of the link. One end is the point-of-reference; the other end is the peer. Traffic flows from the point-of-reference to the peer.

```
| point-of-reference | peer |
```

The topology of the AS can be determined by using the adjacency relationship between nodes and links.

A link is adjacent to a node if traffic can flow from the node to the link with nothing in between. Thus, the node will be the point-of-reference end of the link.

A node is adjacent to a link if traffic can flow from the link to the node with nothing in between. Thus, the node will be the peer end of the link.

2.2 Identifiers

Each entity in a FIRE AS has a unique identifier by which it can be addressed. An identifier has six parts: the point-of-reference address, the peer address, a subscript, an entity type, a netmask bitflag, and a netmask length field.

There are five entity types: router node, network node, passive node, link and operator.

The operator is a person rather than an entity. The operator is responsible for distributing Operator Configuration Files (OCF) and Operator Configuration Messages (OCM). All fields in the operator identifier MUST be zero.

Router and passive nodes use only the entity type and point-of-reference fields; all other fields are set to 0. The point-of-reference address is a canonical address which the node chooses from its set of IP addresses. The node MUST pick only one address and use it consistently throughout routing. This canonical address will be used to determine the topology, but won't be used for routing. The IP-Addresses property is used for routing. (See below in the section on the Property SA.)

Network nodes use the type, point-of-reference and netmask length field; all other fields are set to 0. The point-of-reference address will contain the broadcast media's subnet broadcast address, and the netmask length field will contain the length of the CIDR netmask.

Links use all of the fields in the FIRE identifier. The point-of-reference address is the canonical address of the node at the point-of-reference end. The peer address is the canonical address of the node at the peer end. If there are multiple links between two nodes, then the subscript is used to differentiate between them. If one of the link's endpoints is a network node, then the netmask length field is used for the network node's CIDR netmask. The netmask bitflag is used to indicate which end of the link the netmask should be applied to; 0 indicates the point-of-reference end, 1 indicates the peer end.
2.3 State Advertisements

State Advertisements (SAs) contain information about the AS in which FIRE is running. There are four types of SAs: configuration, certificate, property, and external routes. Configuration SAs come from the FIRE operator and lists OCFs to load, advertise and run for the AS. Certificate SAs are used to distribute public keys. External routes SAs are used to advertise routes external to the AS.

Property SAs advertise an entity's property values; every router and passive node can create its own Property Repository of information about the AS from the Property SAs. Property SAs can be sent only for router nodes, network nodes, and links. Because passive nodes only listen to traffic but do not participate in routing, they cannot send Property SAs.

2.4 Properties

FIRE distributes values called properties in its Property State Advertisements. Properties are values that describe the link, router or network node being advertised in the SA. Routing programs use the properties to decide how to build forwarding tables.

Properties can be thought of as name-value pairs, where the value is typed and can have multiple parts. For example, routers might support a property named "multicast-core" whose value is a list of multicast groups for which the router is in the core. Properties can also be defined as the results of Java applications: a node runs the Java application to get the value of the property.

A link, router or network node may have hundreds of properties. However, in normal operation, only a limited set of properties is advertised. The properties that are advertised are determined by the Operator, who lists the properties to advertise in the OCF.

2.5 Property Repository

The Property Repository is a collection of all of the Property State Advertisements. This repository holds the information over which the routing algorithms are run to produce forwarding tables. There is an identical Property Repository in each routing node. Synchrony is kept by exchanging State Messages containing Property SAs. Reliable flooding is employed to make sure that an SA generated by one node is eventually seen by all nodes in the AS.

3. FIRE State Messages

Information is exchanged between FIRE nodes using State Messages. A State Message contains either a State Advertisement or an acknowledgement of the receipt of one or more SAs. State Messages are sent using the FIRE Layer InterNetwork Transport protocol (FLINT), using the specified port offset from the configured base port. See [FIREFLINT] for details. Since FLINT does not support fragmentation, FIRE State Messages are limited by the maximum size of an IP datagram, currently 64K, less any overhead incurred by security services such as IPsec utilized by FIRE.

A FIRE State Message consists of a Header segment and a Payload segment. The Header contains data that is not authenticated; a protocol version; a field which indicates whether the message is a State Advertisement or an SA acknowledgement; and a timestamp to be used by TCP’s smoothed round trip time algorithm for calculating Round Trip Time (RTT) and exponential backoff for retries. The Payload contains either a State Advertisement or a list of SA acknowledgements.

If, for any reason, a State Message fails to parse, contains inconsistent information (e.g., the actual length is less or greater than the specified length), or the authentication fails, the message is to be dropped without further processing.
3.1 State Message Header Format

The State Message Header is common to all FIRE State Messages. It contains version information, a special ACK count field used only for acknowledgements, and a time stamp used to compute round trip times. The format is shown below.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------------------------------------+
| Version | Flags | ACK Count |
|---------------------------------------------+
| Time Stamp                                  |
+---------------------------------------------+
:---------------------------------------------+
: Sender                                      |
```

3.1.1 Version (8 bits)

The Version field is used to ensure that an incoming FIRE Message can be parsed by the protocol engine. The high-order nibble of the Version field is the major version number. The low-order nibble of the Version field is the minor version number.

```
0 1 2 3 4 5 6 7
+-------------+
| major | minor |
+-------------+
```

This version of the protocol, FSP 0.1, is represented as 0x01.

3.1.2 Flags (8 bits)

The Flags field is used to modify the processing of State Message payload. It has the following format:

```
  flag
   value | SA Type
   ----  |-------
   0x080 | IPv6  
```

The following bitflags are defined for the flags field:

- **IPv6**
  - The IPv6 flag indicates whether the sender, identifier, and creator fields are using IPv4 or IPv6 addresses. If this bitflag is clear, the identifier fields are expressed in terms of IPv4 addresses; if set, the fields are expressed in terms of IPv6 addresses.

3.1.3 ACK Count (12 bits)

For all State Advertisements, this field MUST be zero. For SA Acknowledgements, it serves as a 16-bit unsigned integer value specifying the number of SAs acknowledged by this Message, and MUST NOT be zero.

3.1.4 Time Stamp (32 bits)

All State Messages have a Time Stamp that is used by TCP smoothed round trip time algorithm to compute the Round Trip Time (RTT). For all State Advertisements, this is set to the current time as defined in the next paragraph. For SA acknowledgement messages, the Time Stamp depends on the type of messages being acknowledged. If the ACK is being generated as part of a State Dump (meaning the SAs have not actually been received from this neighbor), the time stamp is set to all zeros. Otherwise, the time stamp is copied from the most recently received State Message from that neighbor.
Since the timestamp is used only by the entity which created the timestamp, it is the amount of
time in hundredths of a second since some epoch. So as not to worry about rollover, the epoch
SHOULD be the time the FIRE daemon was started.

3.1.5 Sender

The *Sender* is the identifier of the entity that sent this State Message. This is used during reliable
flooding (see State Message Processing section below).

3.2 SA Acknowledgements

An SA Acknowledgement Message may contain acknowledgements for multiple State Advertis-
ements. The number of acknowledgments MUST be specified in the State Message header ACK
Count field, and MUST NOT be zero. The State Message header is then followed directly by indi-
vidual ACKs for each State Advertisement. The format of these ACKs is shown below:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Version | Flags | Type | Reserved |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Timestamp |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Sequence Number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| OCF Number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Identifier |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Expiration Time |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

The Version, Flags, Type, Timestamp, Sequence Number, OCF Number, Identifier and Expiration
Time fields are all copied directly from the State Advertisement being acknowledged. (See next
section.)

3.3 State Advertisement Header

The State Advertisement Header is common to all FIRE State Advertisements. It contains a
sequence number that gives relative order of the message, an identifier to map the message con-
tents onto a routing entity, an expiration time of the contents, an Operator Configuration File
number to map this message to a particular OCF, the creator of the message, and a digital signa-
ture. The format is shown below.
3.3.1 Version (8 bits)

is the same as the version in the State Message Header. By duplicating it here, the State Advertisement can be parsed without a State Message Header.

3.3.2 Flags (8 bits)

is the same as the flags in the State Message Header. By duplicating it here, the State Advertisement can be parsed without a State Message Header.

3.3.3 Type (8 bits)

The Type field contains the type of State Advertisement being transported in this message.

There are four types of State Advertisements, listed below:

<table>
<thead>
<tr>
<th>value</th>
<th>SA Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Configuration SA (Operator Configuration Message OCM)</td>
</tr>
<tr>
<td>1</td>
<td>Certificate SA</td>
</tr>
<tr>
<td>2</td>
<td>Property SA</td>
</tr>
<tr>
<td>3</td>
<td>External Routes SA</td>
</tr>
</tbody>
</table>

3.3.4 Timestamp and Sequence Number (32 bits and 32 bits)

The Timestamp and Sequence Number fields are unsigned 32-bit quantities representing a unique serial number for the outgoing State Message. The timestamp is monotonically increasing modulo 32 with granularity of one second and set to the time when the SA is created (seconds since January 1, 1970, in Zulu Time). The sequence number is set to 0. If multiple SAs must be issued at a rate faster than the timestamp granularity, the new sequence number is one greater than the last. Consequently, for each SA with the same timestamp, the sequence numbers are monotonically increasing, providing a strict ordering on all SAs.

A State Advertisement is uniquely identified by its type, identifier and OCF number fields. A new timestamp sequence number pair is assigned to each SA as described above. There is no relationship between the timestamp and sequence numbers spaces of two different types of State Advertisement messages, or for two Property State Advertisements with different OCF numbers.
3.3.5 OCF Number (32 bits)

The OCF Number field contains the version of the Operator Configuration File this State Advertisement corresponds to. This field MUST always be zero for Certificate, Configuration and External Route SAs.

3.3.6 Identifier

The Identifier field specifies the entity described by the State Advertisement. In a State Advertisement, the information is used to build the Property Repository by mapping the SA contents to the Identifier. In an acknowledgement, the Identifier is used along with the Type, OCF Number and Sequence Number values to identify what State Advertisement is being acknowledged.

Note that in some cases this Identifier may not be the same as the identifier of the entity that created the message (that is, the Identifier value is different than the Creator Identifier described below). This happens when the entity being described is a link or network node, since the creator of these entities must be a router node.

Identifiers have six parts: ID-Point-Of-Reference-Address, ID-Peer, ID-Subscript, ID-Type, ID-Netmask-Flag, ID-Netmask-Length.

```
 0 1 2 3
+---+---+---+---+
|   |   |   |   |
+---+---+---+---+
```

* ID-Netmask-Flag

3.3.6.1 ID-Point-Of-Reference (32 or 128 bits)

ID-Point-Of-Reference is an IP address (IPv4 or IPv6). For router and passive nodes, the ID-Point-Of-Reference is simply their designated IP address. For network nodes, the ID-Point-Of-Reference is their subnet broadcast address. For links, the ID-Point-Of-Reference is the ID-Point-Of-Reference for the point of reference node. The Operator's ID-Point-Of-Reference is the all zeros address.

3.3.6.2 ID-Peer (32 or 128 bits)

ID-Peer is also an IP address (IPv4 or IPv6). For nodes and the Operator, the value of ID-Peer is zero. For links, the ID-Peer is the ID-Point-Of-Reference for the peer node.

3.3.6.3 ID-Subscript (16 bits)

ID-Subscript is a value used to disambiguate cases where the ID-Point-Of-Reference and ID-Peer together do not uniquely identify an association. Such an example is the case where there are multiple parallel links between two nodes, or a router has multiple interfaces on the same broadcast network. Where no disambiguation is required, the ID-Subscript is set to zero.

3.3.6.4 ID-Type (7 bits)

ID-Type indicates the type of the entity. Currently five values are defined: Operator (0), Router (1), Network (2), Link (3), and Passive Participant (4).
3.3.6.5 ID-Netmask-Flag (1 bit)

If the netmask length field is non-zero, the ID-Netmask-Flag indicates which address (ID-Point-Of-Reference or ID-Peer) to apply a CIDR-style netmask to. If set, the mask is applied to the ID-Peer value, if clear the mask is applied to the ID-Point-Of-Reference value.

3.3.6.6 ID-Netmask-Length (7 bits)

ID-Netmask-Length indicates the length of the CIDR-style netmask to be applied to either the ID-Address or the ID-Peer field, as indicated by the ID-Netmask-Flag. A netmask is used only for network nodes or for links in which one end refers to a network node.

3.3.7 Expiration Time (32 bits)

All State Advertisements have an Expiration Time — that is, a time when the information is no longer valid. For State Advertisements, this may be several minutes to several days. The expiration time field contains the time in seconds since January 1, 1970, in Zulu Time.

3.3.8 Creator Identifier

The Creator Identifier is the identifier of the entity that created this State Advertisement. Receivers of the State Advertisement should confirm that the creator identifier is entitled to advertise information for the identifier (see the Authentication section below).

3.3.9 Signature

The Signature field holds the digital signature of the creator identifier. It is computed by prefixing a Full Header to the message, zeroing out the signature field, and then computing the signature.

3.4 State Advertisement Types

This section defines the format of individual SAs. Each State Advertisement begins with a common State Advertisement header followed by type-specific information.

3.4.1 Properties SA

The Properties SA (Advertisement Type 2) is a list of properties describing the entity identified by the SA message Identifier. The particular properties listed are determined by the type of the Identifier (links may have different properties from routers). There is no property listing in the payload, as the information about the properties and types is contained in the OCF. The payload of the property SA is an S-expression, a NULL-terminated string.

The payload of the property SA has the following format: (( name value ) ( name value ) ( name value ) ...)

where, name is the name of a property, and value is the entity’s value for that property. Valid values [FIRE100c] are: (a) nonparticipant, (b) unsupported, (c) true or false (for type boolean), (d) a hex integer (for type byte), (e) a decimal integer (for type integer), (f) a floating point number (for
type float), (g) a double quote delimited string (for type string) or (h) an S-expression (for arrays of any of the above types).

All entities MUST flood a Property SA for each advertising OCF (see configuration message, below), the running OCF, and for OCF 0. Algorithms specified for different OCFs MUST have access to the property values flooded by OCF 0 Property SAs. OCF 0 contains exactly four properties. These four properties are:

Fire-Up
This indicates whether the FIRE entity is up or down. It is type boolean. True means up; false means down.

Fire-Metric
The hop count for this SA when computing the routing table for FIRE management traffic. It is type integer. This value determines how all multi-hop FIRE messages are routed. It defaults to one (1) for each router node and zero (0) for each link and network node.

IP-Addresses
The set of IP addresses associated with this entity. For links, this is the set of IP addresses associated with the source node, both canonical and aliases. For routers, this includes any stub hosts that are reachable through this node. Networks do not participate in this property. IP addresses MUST be expressed as an array of bytes.

OCFs-Loaded
A list of the OCF numbers for which this entity has the OCF and all support files associated with the OCF loaded. This is expressed as a single dimension array of type integer. Since OCF 0 is required, 0 SHOULD NOT be put in the OCFs-Loaded list. Only router nodes participate in this property.

All Property SAs MUST be digitally signed.

3.4.2 Certificate SA
The Certificate SA (Advertisement Type 1) is a set of certificates containing (and confirming) the public keys of the SA Identifier.

3.4.3 Configuration SA
The Configuration SA (Advertisement Type 0) is a list of configuration rules for the current FIRE system. It may only be sent by the Operator (both SA Identifier and SA Creator Identifier must contain the Operator's identifier) and MUST be digitally signed by the Operator. The Expiration Time field SHOULD have value 0.

Every Configuration SA contains a number of distribution sets and a number of OCF descriptions.

Each distribution set contains an identifier for the set (dist set number), the number of nodes in the set (Num of Distrib nodes in set), and a list of IP addresses where mirror copies of OCF files are stored.

Each OCF description contains an OCF number, the distribution set number where the OCF can be obtained, a directive as to what each recipient of the OCM should do with the OCF, and a filename for the OCF.

If the number of
The Distribution Node Set Details structure is defined as:

<table>
<thead>
<tr>
<th>Num Dist Sets</th>
<th>Num OCFs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Latest Expiration Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Earliest Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Latest Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Distribution Node Set Details (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCF Details(n)</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The OCF details are defined as:

<table>
<thead>
<tr>
<th>OCF Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>dist set number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>directive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hash Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hash Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OCF Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>padding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OCF filename</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>padding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

### 3.4.3.1 Num Dist Sets (16 bits)

is the number of distribution sets in the Configuration SA.

### 3.4.3.2 Num OCFs (16 bits)

is the number of OCFs described in the Configuration SA.

### 3.4.3.3 Latest Expiration Time (32 bits)

During State Message Processing, the State Advertisement expiration time is checked to see if it is too late in time. If it is, the SA is silently dropped. The operator can set this value (in seconds), and it is added to the current time and checked against the expiration time.

Valid values are 1 minute (60 seconds) to 1 non-leap year (31536000 seconds). If this field has an invalid value (less than 60 or greater than 31536000), this value reverts to the default, 3 days (259200 seconds).

### 3.4.3.4 Earliest Timestamp (32 bits)

During State Message Processing, the sequence number / timestamp is checked for validity, either too early or too late in time. This field allows an operator to set "too early" in seconds. It is subtracted from the current time and checked against the sequence number / timestamp.

Valid values are 1 hour (3600 seconds) to 10 non-leap years (315360000 seconds). If this field has an invalid value (less than 3600 or greater than 315360000), this value reverts to the default, 1 non-leap year (31536000 seconds).
3.4.3.5 Latest Timestamp (32 bits)

During State Message Processing, the sequence number / timestamp is checked for validity, either too early or too late in time. This field allows an operator to set "too late" in seconds. It is added to the current time and checked against the sequence number / timestamp.

Valid values are 1 minute (60 seconds) to 1 non-leap years (31536000 seconds). If this field has an invalid value (less than 60 or greater than 31536000), this value reverts to the default, 10 minutes (600 seconds).

3.4.3.6 Num of Distrib nodes in set (16 bits)

is the number of distribution nodes in the distribution set. Each distribution node is a host which has an LDTP server [FIREldtp] and a file repository.

3.4.3.7 dist set number (16 bits)

is a unique identifier for the distribution set. This is defined in a distribution set, and used by the OCF details to indicate where the OCF can be found.

3.4.3.8 IPv4 or IPv6 addresses (32 or 128 bits)

are the addresses of the distribution nodes.

3.4.3.9 OCF Number (32 bits)

is the OCF's unique id.

3.4.3.10 directive (8 bits)

is a directive from the operator to the participating node as to what to do with the OCF. There MUST NOT be more than one running OCF. If no OCFs are listed as running, then OCF 0 is the running OCF.

<table>
<thead>
<tr>
<th>directive</th>
<th>directive action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LOAD: get the OCF and the support files listed in the OCF from the distribution nodes</td>
</tr>
<tr>
<td>1</td>
<td>ADVERTISE: load the OCF if not already done, and flood SAs for this OCF</td>
</tr>
<tr>
<td>2</td>
<td>RUN: load the OCF if not already done, flood SAs for this OCF, and route traffic using the SAs for this OCF and OCF 0</td>
</tr>
</tbody>
</table>

3.4.3.11 OCF hash

is a hash of the OCF file which must match a hash of the obtained OCF. If it does not, the OCF is thrown away and not supported.

3.4.3.12 Filename of the OCF

is the name of OCF at the distribution nodes. This field MUST be NULL terminated, and it MUST be padded so OCF details fall on a four byte boundary.

3.4.4 External Routes SA

Type 3 [advertising routes here, how do we do that -- what metrics do we use -- need to able to pass in arbitrary properties per route...]

4. Reliable Flooding

Reliable flooding is the method by which a State Advertisement is delivered to all nodes in the AS. Each node has a set of neighbor nodes connected to the node by a link. This forms a star topology. Call the node in the center the central node, and the nodes connected to the central node by one hop along a link are the central node’s neighbors. A neighbor that sends State Messages to the central node is called a source. There can be multiple sources. The figure below shows this star topology.

During the reliable flooding procedure, each node becomes both the receiver and the sender - the node receives State Advertisements from one or more of its neighbors, then the node sends these advertisements to each neighbor that does not already have a copy.

In order to prevent State Advertisements from being unnecessarily flooded to neighbors who have already indicated they have a copy, either by pre-emptively acknowledging, or by actually sending the SA itself, each router SHOULD maintain neighbors’ state associated with each SA. This history must contain a record for each neighbor, and hence MUST be of configurable size, such that it can be enlarged or decreased as needed. Modification of the maximum number of neighbors MAY require restarting the router. This neighbor state will be maintained by the flooding process as described below.

For broadcast media, routers SHOULD flood only to the Designated Router (DR), determined by the FIRE Peering Protocol (FIREpeer). The DR is responsible for flooding to all neighbors on the broadcast media. The DR uses multicast for the initial transmission of an SA to its broadcast net, and reverts to unicast for retransmissions.

4.1 State Message Transmission

State Messages are flow controlled using a simple windowing protocol. The transmission window is specified in terms of the number of State Advertisements that may be transmitted without acknowledgement. Each FIRE implementation will have a built-in window size, which SHOULD be the same for all FIRE routers in a domain. This window size MUST be greater than 0, and SHOULD be at least eight.

A router will maintain separate transmission queues for each neighbor. SA Acknowledgement Messages are never queued, but instead are sent as soon as possible. The remaining State Advertisements are queued and sent in a priority order. They are transmitted as soon as window space is available. When possible, messages being flooded to multiple neighbors reachable via the same interface are multicast.

If the transmission window closes, that is the configured window size minus the number of unacknowledged, transmitted packets is zero, further transmission is halted until an acknowledgement is received. Packets in the transmission window are retransmitted, however, according to the procedure described below, in order to insure reliable transmission. If an SA of higher priority than an SA in the transmission window is received, the lower priority SA is removed from the
transmission window and the higher priority SA is sent.

There are three priority levels for SAs. The highest priority is Configuration and Certificate SAs. The middle priority is SAs needed for routing: property SAs for OCF 0, property SAs for the running OCF as set by the Configuration SA, and external SAs. The lowest priority are property SAs not for OCF 0 or the running OCF.

Multicast is used by the Designated Router (DR) whenever possible. When a DR has an SA to send to its neighbors, it checks the transmission window for each of its neighbors on the subnet. If a percentage of those neighbors has space in their windows, then the packet is multicast. For those neighbors with room in their transmission windows, the packet is marked for retransmission, and a space is taken in the transmission window.

For those neighbors that don't have sufficient space, the packet is queued for transmission when a space opens in the neighbor's window. If an ACK is received, the SA is marked as having been received by the neighbor, and the packet is removed from the transmission queue.

If the number of neighbors that has room in their transmission windows is less than the percentage mentioned above, then the SA is held for some period of time. After that time, if the percentage of neighbors rises to the acceptable number, then the SA is multicast as described above. If not, the SA is unicast to those neighbors that have space in their windows and queued for transmission for the rest.

The percentage is implementation dependent and is RECOMMENDED to be 50%. The amount of time to wait before trying to multicast again is also implementation dependent, and is RECOMMENDED to be the average of the smoothed round trip times of the neighbors on the subnet (see below).

### 4.1.1 Retransmission and Round Trip Measurements

In order to provide a reliable flooding mechanism, FIRE retransmits unacknowledged State Messages in the transmission window at intervals based upon the smoothed round trip time (RTT) between neighbors, as measured using RTT calculations from the state message timestamp and the smoothed RTT algorithm from TCP. Successive retry attempts are backed off exponentially until either an acknowledgement is received or a maximum retry count is exceeded. Unlike initial floods, however, retransmissions are unicast directly to specific neighbors rather than multicast.

### 4.2 State Message Processing

When a State Message is received by the central node, the central node looks at the ACK count in the State Message header to determine if the packet contains a State Advertisement or SA Acknowledgements. It then processes the contents.

#### 4.2.1 State Advertisement Processing

1. The message's Type, OCF Number, Sequence Number, Timestamp and Identifier fields are checked against existing State Advertisements to see if this message has already been received. If so, an ACK is generated by the central node and sent to the sender. No further processing is done.

2. The message's Timestamp and Sequence Number is checked for validity. If they are not valid, an ACK is generated by the central node and sent to the sender. No further processing is done. The Timestamp is not valid if it is too early or too late. Too early is determined by subtracting the Earliest Timestamp from the Configuration SA (or 1 non-leap year if there is no Configuration SA) from the current time. Too late is determined by adding the Latest Timestamp from the Configuration SA (or 10 minutes if there is no Configuration SA) to the current time. The Timestamp / Sequence Number pair is not valid if
we already have a State Advertisement with the same Identifier, Type and OCF Number which has a greater Timestamp / Sequence Number pair. (Timestamp / Sequence Number pair A is greater the Timestamp / Sequence Number pair B if A's Timestamp is greater than B's Timestamp, or A's and B's Timestamps are equivalent and A's Sequence Number is greater than B's Sequence Number.)

3. The message's Expiration Time is checked for validity. If it is not valid, an ACK is generated by the central node and sent to the sender. No further processing is done. The Expiration Time is not valid if it is either too early or too late. Too early is anything less than the current time. Too late is determined by adding the Latest Expiration from the Configuration SA (or 3 days if there is no Configuration SA) to the current time. An Expiration Time of 0 is invalid for all SAs except the Configuration SA.

4. The OCF number is checked for validity. If it is not valid, an ACK is generated by the central node and sent to the sender. No further processing is done. The OCF number must be 0 for all SAs except the property SA.

5. If the SA is a Configuration SA, then the Identifier is checked to make sure it is the operator. If it is not, an ACK is generated by the central node and sent to the sender. No further processing is done.

6. If the SA is a Configuration SA, then the Creator is checked to make sure it is the operator. If it is not, no ACK is sent, and no further processing is done.

7. Do the right thing for Certificate SAs.

8. The Signature is validated. If the signature is invalid, the occurrence is logged, and the message MUST NOT be processed further.

9. An ACK is generated by the central node and sent to the sender.

10. The SA is stored in the SA List, replacing any previous SAs with the same ID, OCF and type. It SHOULD be noted in the history record for the SA that none of the neighbors have this SA, except the neighbor which sent the SA.

11. The message is forwarded to all neighbors who have not previously indicated they have received it, either through an ACK, or by sending the SA itself. This is precisely the set of neighbors which are indicated as not having the SA in the SA's history record.

4.2.2 Acknowledgement Processing

Upon receipt of an ACK from a neighbor, the Timestamp, Sequence Number and Expiration Time in the ACK MUST be compared with the corresponding fields in the acknowledged SA. If any differ, the ACK is dropped.

The acknowledged SA(s) MUST be removed from the send queue if it has not yet been transmitted. If an acknowledged message was in the transmission window and the time stamp in the ACK message is anything other than the all-zeros time stamp, the window size MUST be increased accordingly, and the TCP smoothed round trip time algorithm run to compute the smoothed RTT. Acknowledged packets removed from the send queue, but not yet transmitted MAY NOT affect the transmission window size.

In addition, it should be noted in the history record for the SA that the ACKing neighbor has received the SA. If the router has has not yet received the SA, it SHOULD silently drop the ACK.

Duplicate ACKS (ACKS for SAs not in the send queue and whose corresponding history record indicates the neighbor has already received the SA) MUST be silently ignored.
5. State Changes

Each FIRE router is responsible for advertising the current state of itself and any links for which it serves as the terminus. In the event that the router serves as a Designated Router (DR) for one or more networks, it must also issue State Advertisements for the network and its adjoining links. These messages are generated as each object comes up.

5.1 Router Initialization

Upon boot-up a router begins the Peering Protocol [FIREpeer] in an attempt to acquire neighbors. Whenever a neighbor is located, the State Dump process is immediately initiated. This includes any time a neighbor is discovered, whether as a result of a link coming up, the neighbor coming up, or the router itself restarting.

The purpose of the dump procedure is to synchronize two neighbors which have discovered each other while generating the least amount of traffic.

5.2 State Dump

In order to rapidly synchronize databases between newly peered routers, each router follows the following procedure:

1. For each SA in the router’s database, it is noted in the SA’s history record that the neighbor does not have the SA. If the router didn’t previously know about the neighbor, it allocates space in the history record for the neighbor.

2. The router waits either one SILENT_PERIOD as defined in the peering protocol [FIREpeer], or until it receives a state message from the neighbor, whichever comes first.

3. The router sends an ACK for every SA in its database to the neighbor. For efficiency, they should be sent together, as opposed to individual State Messages. Since the number of SAs to be acknowledged may be large, it may be necessary to spread them across multiple State Messages. These ACKs will let the router’s neighbor know which SAs it has.

4. SAs are then transmitted in a priority order: Configuration and Certificate SAs first; property SAs for OCF 0 and the running OCF and external SA come next; property SAs that are neither OCF 0 or the running OCF are last. As it receives SAs and ACKs from its neighbor, it can remove the corresponding SAs from its send queue.

5.3 Change of Designated Router (DR)

When the DR changes, a modified state dump is done between the DR and each router on the network. The modified state dump is steps 2, 3 and 4, above; waiting one SILENT_PERIOD, sending ACKS for all of the SAs in the router’s database, followed by all of the SAs which it doesn’t know the neighbor has. This is done by the DR to each router on the network, and by each router on the network to the DR,

5.4 Link Establishment

Whenever an attached interface comes up, the router must advertise its existence by issuing SAs. In order to prevent flapping, however, the router should track the last three times the interface was declared up. If the earliest of those three occurrences was less than some configurable time from now, the router must suppress sending the new SA to its neighbors. In either event, it should generate a new SA for this link and place it in the SA List for each currently advertising OCF. The router MUST then flood the newest SA whenever the difference between the earliest of the last three transmissions and the current time becomes greater than the configured value.
5.5 Link Removal

If a router determines an associated link has gone down, it must immediately generate a new SA for that link for OCF 0 with the Fire-Up property set to false. The full procedure for issuing an updated SA is described in the following section.

6. SA Refreshment

Routers should renew any State Advertisements they have generated before the previous version expires. Allowing SAs to expire without replacement causes instability in the network since routers will be unable to generate tables that depend on properties advertised in expired SAs. Routers should periodically examine the expiration date of all SAs generated by itself and reissue any whose expiration date is within some configurable threshold from the current time.

To reissue an SA, the router first deletes the superseded SA from its own SA list, replaces it with the newly generated version, and then floods the new SA to each of its neighbors.

7. SA Expiration

Occasionally Routers may need to expire SAs contained in their SA list. If the expiration date of any SA is ever reached, it should be silently discarded. If, due to some strange situation, a router expires the SA advertising its own existence, it must generate an error and halt, since it clearly is not operating properly.

8. Security Considerations

Security considerations for the State Messaging Protocol and all other aspects of FIRE are discussed in the FIRE Security Document [FIREsec].

9. References


10. Authors’ Addresses

Craig Partridge, Isidro Castineyra, Tim Strayer, Alex Snoeren, and Beverly Schwartz

BBN Technologies
10 Moulton Street
Cambridge, MA 02138

{craig, isidro, strayer, snoeren, bschwart}@bbn.com
CONTENTS

1. Introduction ................................................................................................................. 1
   1.1 Status of this Memo ................................................................................................. 2
   1.2 Notational Conventions ......................................................................................... 2

2. Concepts ....................................................................................................................... 2
   2.1 Nodes and Links ...................................................................................................... 3
   2.2 Identifiers ............................................................................................................... 3
   2.3 State Advertisements .............................................................................................. 4
   2.4 Properties ............................................................................................................... 4
   2.5 Property Repository ............................................................................................... 4

3. FIRE State Messages .................................................................................................... 4
   3.1 State Message Header Format .............................................................................. 5
   3.2 SA Acknowledgements ......................................................................................... 6
   3.3 State Advertisement Header ................................................................................. 6
   3.4 State Advertisement Types ................................................................................... 9

4. Reliable Flooding .......................................................................................................... 13
   4.1 State Message Transmission .............................................................................. 13
   4.2 State Message Processing .................................................................................... 14

5. State Changes .............................................................................................................. 16
   5.1 Router Initialization ............................................................................................. 16
   5.2 State Dump ........................................................................................................... 16
   5.3 Change of Designated Router (DR) ..................................................................... 16
   5.4 Link Establishment .............................................................................................. 16
   5.5 Link Removal ..................................................................................................... 17

6. SA Refreshment .......................................................................................................... 17

7. SA Expiration .............................................................................................................. 17

8. Security Considerations ............................................................................................ 17

9. References .................................................................................................................. 17

10. Authors’ Addresses ................................................................................................... 18