INTERNET-DRAFT
Intended Status: Informational
Expires: April 2019

T. Herbert
Quantonium

October 10, 2018

Encoding Routing in Firewall and Service Tickets
draft-herbert-route-fast-00

Abstract

This document describes a method to encode routing information in Firewall and Service Tickets (FAST). Encoded routing information provides the local routing for packets sent in either the forward or return paths of a flow. FAST ticket reflection at peer hosts ensures that the routing information is attached to packets being sent in the return path. When a packet with a FAST ticket containing routing information enters the network in which the ticket was issued, the ticket is parsed to extract the routing information and is forwarded per the information. Routing in Firewall and Service Tickets has a number of use cases. It can be used as a type of source routing, used with identifier-locator protocols to provide a locator in the return path, and can be used to specify a backend instance in Layer 4 load balancing for processing connections to a virtual IP address.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

Copyright and License Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1 Introduction

This document specifies a method to encode routing information in Firewall and Service Tickets (FAST). FAST allows hosts to signal the network for services to be applied to packets in the form of tickets that are issued by the network and attached to packets. Tickets may encode information as to how the packet is to be routed in the local network. Tickets can be reflected by peer nodes for a connection, so that routing information can be applied in the reverse path of a flow. When a packet with an attached ticket containing routing information enters a network, the ticket is parsed and the encoded routing is applied to the packet. The routing information is arbitrary per the network's needs. For instance, it may indicate a tunnel to send the packets on, a segment routing list, a locator in identifier-locator protocols, or the IP address for a backend instance of a layer 4 load balancer.

2 Background and motivation

This section provides background and motivation.

2.1 Problem statement

A network may implement arbitrary complex routing of packets, mobile routing functions, or routing for service function chains. Often such functions require routing packets on a per flow basis or perhaps based on some awareness of application content.

Typically, this advanced routing is done by an ingress router into a network. This router may need to perform Deep Packet Inspection (DPI) into the transport layer, may maintain state for every flow traversing it, or may maintain a mapping table of mobile or virtual addresses to their locators or physical nodes. The net effect is that intermediate nodes perform complex transport protocol specific functions that are difficult to to scale. In turn, the complexity in such nodes contributes to protocol ossification.

2.2 Firewall and Service Tickets

Firewall and Service Tickets (FAST) is a technique to allow an application to signal to the network requests for admission and services for a flow. A ticket is data that is attached to a packet by the source that is inspected and validated by intermediate nodes in a network. Tickets express a grant or right for packets to traverse a network or have services applied to them. Tickets may also be modified by intermediate nodes under certain circumstances.

In order to apply services to inbound packets for a communication,

remote peers reflect received tickets in packets they send without interpreting them. Tickets are stateless within the network, however they can be used to attain per flow semantics. Note that in lieu of creating flow state in the network, state of interest to the network can be cached at the endpoints and provided in data packets. Tickets are encrypted and opaque to the end points for security and privacy. FAST is IPv6 specific and uses Hop-by-Hop options.

This draft proposes to encode routing information into FAST tickets. The necessary information for routing packets is contained within the network layer headers of each packet. The encoded data obviates the need for DPI, flow state, or complex route lookups at ingress routers.

2.3 Similar efforts

This section highlights similar efforts that encode routing information in packets with the intent that intermediate nodes will consume the information.

2.3.1 Segment routing

Segment routing [SPRING] is a type of source routing that employs a routing header. A segment routing header contains a list of segment identifiers (SIDs) that indicate the nodes to visit by their IPv6 addresses. Segment routing is often deployed with IP encapsulation. When a packet enters a segment routing domain, a lookup is performed on the packet to retrieve the segment list. The packet is then encapsulated in IPv6 and a segment routing header is attached to the packet. The lookup may be stateless in simple cases, or may require a stateful lookup with full blown connection tracking.

Segment routing is intended for use in a limited domain and not on the Internet. Use over the Internet would expose internal addresses in the route list, offers no security, and would have considerable overhead. Additionally, segment routing only describes routing in the the forward path to a destination, not the return path.

2.3.2 hICN mobility proposal

hICN with anchorless mobility [HICN-AMM] is an alternative proposal to using a distributed mapping system with identifier locator protocols.

[HICN-AMM] highlights some drawbacks of traditional mapping systems:

- They entangle forwarding operations and changes to network location

- Mapping systems at large scale are difficult to manage
- Convergence time after a location change (handover in mobile network terminology) may be excessive.

[HICN-AMM] and [HICN] describe a solution that eliminates the need for a global mapping system by sending locator information in-line with the data path. The locator information is embedded in a new transport layer header ("Path Label" in the transport header shown in [HICN]). The transport layer header is parsed by routers in the path and they use the information to create state for forwarding packets in the return path.

Similar to FAST, the hICN proposal adopts an in-line approach to convey routing information. However, the protocol in the FAST method is strictly confined to the network layer and there is no requirement for transport state to be maintained by the network.

2.4 Use cases

This section highlights some use cases of FAST to carry routing information.

2.4.1 Identifier-locator protocols and virtualization

Identifier-locator protocols, such as ILA and LISP, rely on a distributed mapping system that maps identifiers to locators for transit across a network overlay. Typically, the set of mappings are maintained in a distributed database or mapping cache. Border routers of an identifier-locator domain access the database or cache to map ingress packets to their appropriate locator. The router can then use a network overlay method to deliver packets to their mobile destination; this could be done by address transformation (like in ILA) or encapsulation (as done by LISP).

This proposal suggests an alternative which is to encode locators in Firewall and Service Tickets. This method allows fast path anchorless mobility with identifier-locator protocols that eschews accessing a mapping database or mapping cache for every packet in the datapath.

2.4.2 Segment routing

A FAST ticket could encode bits that network nodes interpret and expand into a segment routing list. When a ticket with such an encoding enters a network, or segment routing domain in this case, it is parsed and the information is expanded to the full segment list. The packet is encapsulated, the segment list is attached in a routing header, and the packet is forwarded towards the destination. The

encoded information might have some compressed form such as an index into a table or a bit map of nodes or services that are needed. FAST tickets are encrypted or obfuscated so that they are opaque to the Internet, they are also reflected so that segment routing can be applied to the reverse path of flow.

2.4.3 Layer 4 load balancing

Layer 4 load balancers route packets addressed to virtual IP addresses (VIPs). A virtual IP address is shared amongst some number of backend servers. Routing to backend servers is done on a per flow basis so that the targeted backend is the one that terminates the flow. Routing must be consistent for the lifetime of the flow lest packets are received on the wrong backend (in which case packets are dropped and the connection might be disconnected.

Consistent routing is accomplished in two ways: 1) a consistent hash of the 5-tuple is used to load balance flows across the backend servers, and 2) connection tracking is used to map a flow to its assigned backend server. The number of backend servers may be dynamic and connections may be migrated between servers. The goal of routing by a load balancer is to minimize black holes and connection drops. The Maglev load balancer [MAGLEV] implements an efficient algorithm employing a hybrid of these two techniques, however it does not entirely eliminate the possibility of connections being dropped.

Both tuple hashing and connection tracking have some disadvantages. Consistent hashing in the presence of dynamic number servers is a hard problem. Connection tracking entails network state which is difficult to scale and can be susceptible to Denial of Service (DOS) attack.

FAST can be applied to provide a stateless and robust solution for load balancing. When a server backend server receives a connection request it creates a ticket that encodes its instance identity (for example the backend IP address). The ticket is attached to packets for the connection that are sent back the client. The client in turn will reflect the tickets in subsequent packets it sends. When the load balancer receives a packet with a reflected ticket, it decodes the ticket and extracts the identity of the assigned backend server. The load balancer then forwards the packet to the address for the indicated backend. If received packets don't have a ticket attached (the first packet on the connection or the peer doesn't reflect tickets) then tuple hashing can be used as a fallback.

Using FAST to encode the backend server instance is impervious to changes to the number of backend servers and requires no flow state in the network. If a connection migrates, the server simply creates a

new ticket that indicates the new backend server for the flow. Once the server sends packets with the updated ticket, the client will reflect it and the load balancer will properly route packets for the flow to the correct backend.

3 Solution Overview

The central idea of this design is that arbitrary routing information is attached to packets and is interpreted by network nodes to achieve expedited forwarding. In FAST, routing information is encoded in tickets and can be encrypted or otherwise obfuscated. Only the nodes in the origin network of the information are able to parse and interpret it. Since the information is secured it can safely be contained in packets sent on the open Internet.

Since packets contain the information needed to route them, route lookups and per flow state maintained in the network to hold routing information is obviated. Packets can effectively be source routed in both the forward and return paths. FAST operates at the network layer and doesn't require network nodes to perform Deep Packet Inspection (DPI) into the transport layer. Any signaling between applications and the network, or between transport protocols and the network, is done via Hop-by-Hop options. This solution is specific to IPv6.

3.1 Requirements

This section lists some requirements for encoding routing in FAST.

Requirements are:

- Solution works with any transport protocol or application

- Intermediate devices only process, inspect, or change network layer headers as specified by RFC8200

- Communication for a flow is bidirectional

- Solution is an optimization for fast path routing. The system must allow a slow path fallback (for instance, if extension headers are dropped for some path)

- Client side supports FAST. Necessary support is described in [FAST]. This should entail no kernel or protocol stack changes since FAST is encoded in extension headers than can be set for flows using existing APIs

- Server side supports ticket reflection as described in [FAST]. This is a generic change in the protocol stack that should be

transparent to applications.

- FAST routers need to be deployed. These are typically border routers of a domain as described in [FAST].

- Other intermediate nodes need to properly forward packets with FAST tickets (Hop-by-Hop options). Note that RFC8200 relaxes the requirement that all nodes in a path process Hop-by-Hop options

- Does not rely an transport state being maintained in the network

- Makes no assumptions that packets for a flow always follow the same path, or that any part of the network path must be symmetric for both directions of a flow

- Maintain security and privacy. In particular, locators are only visible in the network that implement an overlay that uses them

3.2 Reference Topology

The diagram below provides a reference topology for a simple mobile network.

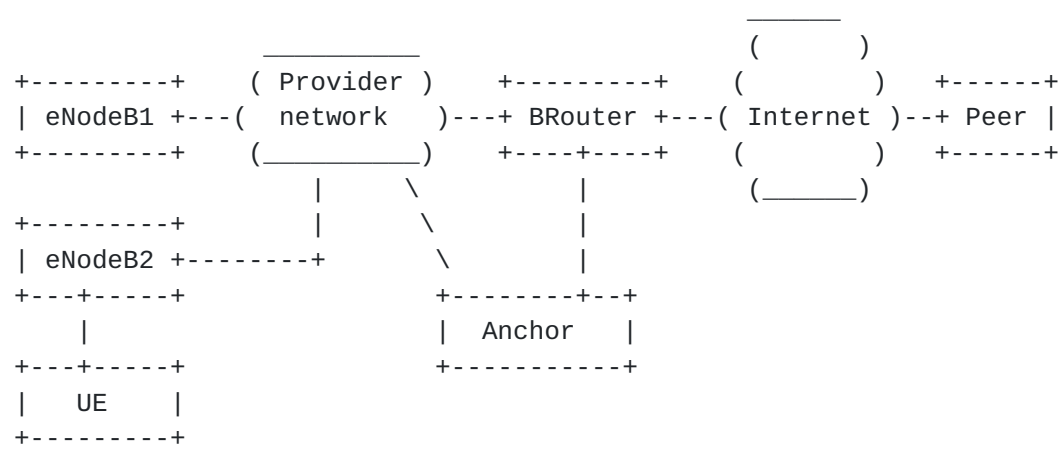


Figure 1

The diagram shows a mobile network that contains two eNodeB's (points of attachment) and one UE (end host device) that is attached to eNodeB2. The UE is mobile so it can move from one eNodeB to another (for instance it may attach to eNodeB1 at some point). The externally visible address of the UE is an identifier that does not change when the UE moves in the network. In the diagram, UE may be communicating with the "Peer" host in the Internet. The Peer only knows the UE by its externally visible address. To achieve communications, packets from the Peer to the UE are sent over some type of network overlay when transiting the provider network.

The packets of interest with respect to mobility are those destined to the UE. The Peer might itself be mobile, but that case should be symmetric and transparent. In this model, it is the local network of a mobile node that deals with mobility of devices in the network.

It is common for mobile communications to go through an anchor point that has access to the complete mapping database and provides the entry point to the network overlay. Anchors may be heavyweight and force packets into a "long" path with respect to latency. There is motivation to allow a fast path for critical latency sensitive communications that bypass anchor points. In figure 1, this fast path would be implemented in BRouter. That is, instead of forwarding packets through the Anchor, it performs identifier locator-mapping and network overlay functions itself.

The typically proposed method to eliminate the anchor point is to implement a mapping cache (in the diagram this would be to place a cache at BRouter). The proposal described here is an alternative that doesn't require a cache.

3.3 Basic packet flow

To use routing with FAST, an application running on a host gets a ticket from the network. The ticket encodes the routing information of interest to the network. When the application sends a packet, the ticket is attached (in an extension header). Packets with the ticket are the forwarded to the peer destination. At the peer, the ticket is saved in a flow context. Subsequently, when an application sends response packets back to its peer the ticket is attached to packets as a reflected ticket.

When a packet with a ticket is received at a border router of the domain that issued the ticket, the router parses the reflected ticket and verifies it. If it is valid, the border router applies the encoded routing information to forward the packet on to its destination.

For example, if ILA is in use in the network topology show above, the flow might be:

1) Application in UE requests a ticket. A ticket agent in the provider network provides a ticket with locator information indicating that UE's location is at eNode2.

2) Application sends packets. The ticket containing locator information for eNodeB2 is attached to each packet.

3) Packets traverse network and are received at Peer. The peer

node saves the receive ticket in a flow context for the application.

4) Peer sends responses. The saved ticket is attached to packets as reflected tickets.

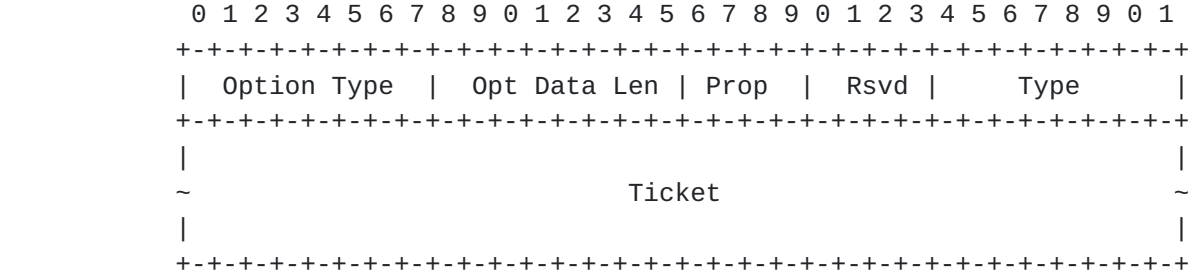
5) When a packet is received from the peer at BRouter, the attached ticket is parsed. The locator information is extracted that indicates the locator for eNodeB2. BRouter transforms the destination address to an ILA locator address and forwards the packet.

6) eNodeB receives ILA packets and performs the reverse transformation to restore the original destination address (typical ILA-N processing).

7) Packets are forward to UE. They still contain a ticket which can be validated by the UE.

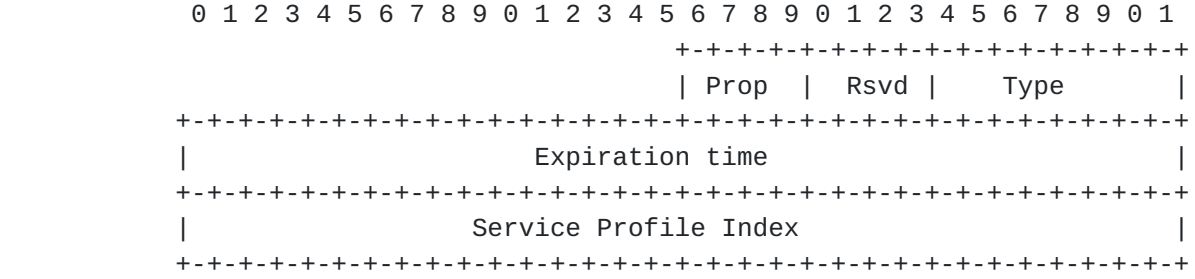
4 Firewall and Service Tickets encoding

FAST tickets are encoded in Hop-by-Hop options. The format of a FAST ticket in a Hop-by-Hop option is:



Tickets are not intended to be parsed outside of the origin network domain that issues them. Therefore, the format is arbitrary per the discretion of the domain in which they are issued.

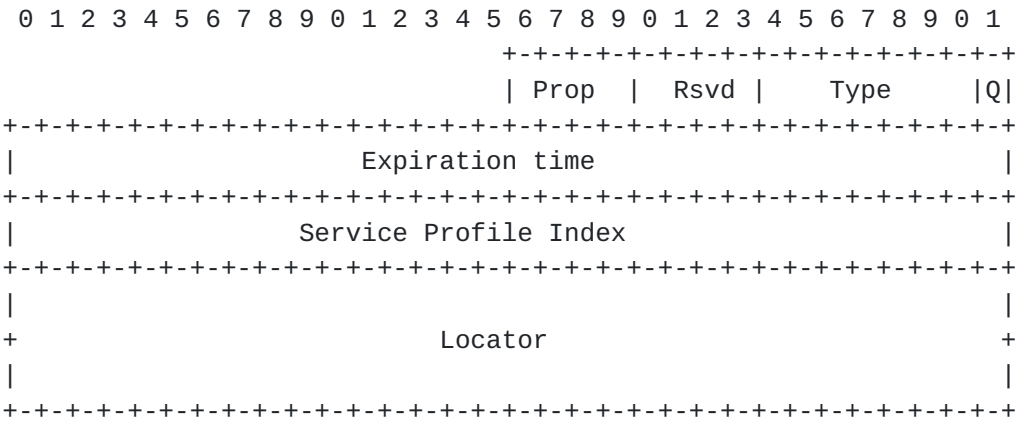
[FAST] suggests a simple and efficient encoding of a Service Profile Index:



This format can be amended to include routing information. Below are some example encodings. Note that tickets are potentially set in all datapath packets so minimizing protocol overhead is a consideration.

4.1 ILA locator encoding

ILA locators are sixty-four bits, and they can be encoded in a FAST ticket in a straightforward fashion. A ticket with expiration time, service profile and an ILA locator may have format:



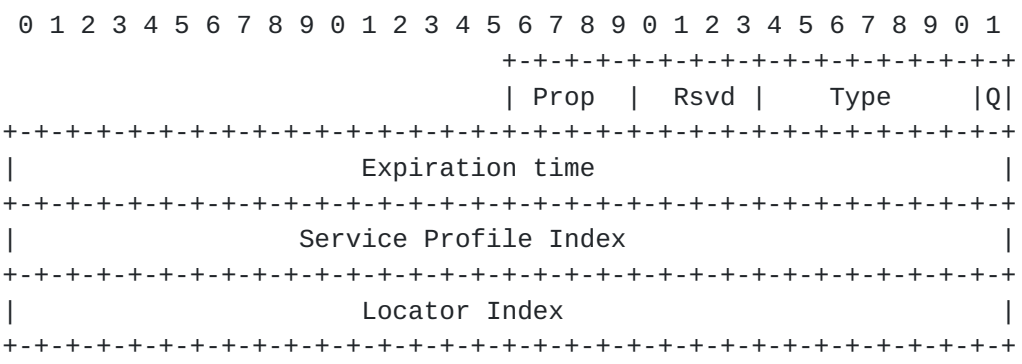
Where the 'Q' bit indicates that an unqualified locator was overwritten. See Section 5.2.

A full 128 bit address could similarly be encoded for use with LISP or other encapsulation protocols.

4.2 Locator index encoding

A network may have a comparatively small number of locators. For instance, a mobile provider might associate each eNodeB with a locator and there may only be a few million of these. In this case, the border routers might maintain a static table of locator addresses that can simply be indexed by number in a small range. Similarly, the backend server in the layer 4 load balancing case might also be indicated by an index into a table of backend servers.

A locator index encoded in a ticket might look like this:



Where the 'Q' bit indicates that an unqualified locator was overwritten. See section 5.2.

5 Operation

This section describes the operation of encoding routing information in FAST tickets.

5.1 Ticket requests

Applications request FAST tickets from a ticket agent in the network local to the application. The ticket agent can return a ticket for the application to use in its data packets. The ticket includes information that is parsed by elements in the issuing network. The ticket information may include routing information. For example, if the application is on a mobile device, the network may provide a ticket that has a locator indicating the current location of the device.

[FAST] describes the process of an application requesting tickets and setting them in packets. An application will not normally need to make any special requests for routing information and the use of routing information is expected to be transparent to the application.

5.2 Qualified locators

There are two possibilities for locator information in an issued ticket:

1) The locator is fully qualified.

2) The locator is not qualified.

5.2.1 Fully qualified locators

If the locator is qualified then the issued ticket contains the locator for the end node. If the locator changes, that is the node moves, then a new ticket will need to be issued to the application.

5.2.2 Unqualified locators

If the locator is not qualified, then the locator information in the issued ticket contains a "not set" value. For instance, in the case the locator is expressed by a Locator Index as in section 4.2, the "not set" value may be -1 (all ones). A upstream router of an end node may write a locator value into the locator information to make it qualified; most often this would just be its own locator value in cases where it is the first upstream hop of end devices that provides location in the network. For instance, an eNodeB router acting as the first hop for a UE may write its locator value in tickets of packets it is forwarding from UEs. The implication is that this will be the locator used in the network overlay on the return path to reach the end node. Note that to write a locator into to a ticket requires that the ticket is in a modifiable Hop-by-Hop option.

5.3 First hop router processing

Once an application has been issued a ticket with routing information it will set the ticket in all packets sent to the peer node. The first hop upstream router in the FAST domain parses the ticket; this may typically be the first hop router in a provider network closest to end user nodes.

If the ticket contains a qualified locator, the first hop node may validate it (as part of FAST ticket validation). If the ticket has unqualified locator information, the first hop node may set it to a qualified locator value in the packet. As described above, the locator information written is likely to be that corresponding to the locator of the first hop device.

5.4 Transit to the peer

Beyond the first hop router to the ultimate peer destination, no processing of routing information in a ticket should be needed. Intervening networks and routers should deliver the ticket to the destination host unchanged.

At the peer host, the procedures described in [FAST] are followed to save the received ticket in a flow context and to reflect it in subsequent packets. As with other reflected tickets, one containing routing information is treated as an opaque value that is not parsed

or modified by the peer or any network outside of the origin network.

Packets sent by a peer will include reflected tickets for a flow. No processing of reflected routing information in a ticket should be needed until the packet reaches the origin network of the ticket. Intervening networks and routers should deliver the ticket to the destination origin network unchanged.

5.5 Ingress into the origin network

At the border router for the origin network, tickets are parsed and the encoded services are applied. If a ticket contains routing information then that can be used to forward the packet over an overlay network.

The specific operation depends on the protocol used to provide the overlay network functionality. For instance, if ILA is in use and the locator information is just a sixty-four bit locator as described in Section 4.1, then the given locator is written to the destination of the packet and the packet is forwarded to the locator address following the procedures of ILA. Similarly, if a locator index is contained in the ticket as described in Section 4.2, then that value is used to index the locator table to get a sixty-four bit locator that can be written into the destination address. Procedures for use of the locator information to do encapsulation, such as that done by LISP, are similar.

Note that the service parameters contained in the ticket may provide additional information about how the packet is to be sent over the network overlay. For instance, the service parameters might indicate the packet is encrypted or to use some extensions of an encapsulation protocol.

5.6 Network overlay termination

When a packet is received at the terminating point of an overlay it is restored to the original packet. If the packet was encpasulated then it's unencpasulated, or if the destination address was transformed then the reverse transformation is done to restore the original address.

At the end host, received reflected tickets are validated for acceptance as described in [FAST]. This is done by comparing the received ticket to that which was sent on the corresponding flow.

5.7 Fallback

The proposal described here is considered an optimization. Routing information in FAST tickets is not intended to completely replace a routing infrastructure. In particular, this solution relies on several parties to implement protocols correctly. For instance, the use of extension headers requires that they can be successfully sent through a network. As reported in [RFC7872], Internet support for forwarding packets with extension headers is not yet ubiquitous.

Therefore, a fallback is required when encoding routing information in FAST is not viable for a flow. In the mobile case, the fallback might be to forward through anchor points. In the layer 4 load balancer case, the fallback may be to use the tuple hash.

5.8 Mobile events

When a mobile node moves and its locator changes, it is desirable to converge to using the new locator as a quickly as possible. With tickets that contain locator information, a modified ticket needs to be sent to a peer host.

If an application was issued a ticket with qualified locator information then a new ticket needs to be issued. This can be done by the application receiving a signal that a mobile event has occurred causing it to make new ticket requests for established flows.

If an application has a ticket with an unqualified locator then the network should start writing the new locator information into packets that are sent by the application after the mobile event. This should be transparent to the application.

Note that in either case, in order to update the tickets that a peer is reflecting, the application needs to send packets to the peer that includes an updated ticket. There is no guarantee when an application may send packets, so there is the possibility of a window where the peer node is sending reflected tickets with outdated locator information. The window should be limited by the expiration time of a ticket (see below), however it is recommended to implement mechanisms to avoid communication blackholes. For instance, a "care of address" mapping entry could be installed at the old locator device to forward to the new one. Such solutions are also used to mitigate database convergence time or cache synchronization time.

5.9 Interaction with expired tickets

FAST typically expects ticket to have an expiration time. If a ticket is received before the expiration time and it is otherwise valid,

then the packet is forwarded per the services indicated by the ticket. If a packet is received with an expired ticket, it might still be accepted subject to rate limiting. Accepting expired tickets is useful in the case that a connection goes idle and after some time the remote peer starts to send.

For tickets that are expired and contain routing information, an implementation should ignore the routing information and take the fallback path. When an application sends new packets, it can include a fresh ticket so that the fast path is taken on subsequent packets. Ignoring the routing information in expired tickets puts an upper bound on the window that outdated information can be used.

6 Security Considerations

Routing information can be highly sensitive data especially if it could reveal the physical location or identity of individuals. Effort must be made to safeguard this information. In particular, locators in plain text should only be exposed within the origin network providing mobility. This includes hiding locators from end hosts. [FAST] discusses security of tickets including recommendations to encrypt them.

Tickets may be reused in packets for some period, and it is desirable to prevent third parties from making inferences based on the fact that a ticket for a flow changed. For instance, it should not be deducible that a node has moved on the basis of a new ticket being used on a flow. To avoid this possibility tickets for a flow should be changed randomly to avoid correlating ticket changes to events.

Per [FAST], tickets are expected to be encrypted or obfuscated to prevent nodes outside the origin network from interpreting them. The only information an external node should be able to deduce is that a packet contains a ticket. Accordingly, routing information encoded in tickets is hidden from external nodes. This allows securely sending packets with encoded local routing information over the Internet.

7 IANA Considerations

There are no IANA considerations in this document. [FAST] requests numbers for Hop-by-Hop options that would be used by this proposal.

8 Acknowledgments

9 References

9.1 Normative References

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[FAST] Herbert, T., "Firewall and Service Tickets", draft-herbert-fast-01, June 2018.

9.2 Informative References

[HICN] L. Muscariello, G. Carofiglio, J. Auge, and M. Papalini, "Hybrid Information-Centric Networking", draft-muscariello-intarea-hicn-00, June 2018

[HICN-AMM] Auge, J., Carofiglio, G., Muscariello, L., and M. Papalini, "Anchorless mobility management through hICN (hICN-AMM): Deployment options", draft-auge-dmm-hicn-mobility-deployment-options-00 (work in progress), June 2018.

Author's Address

Tom Herbert
Quantonium
Santa Clara, CA
USA

Email: tom@herbertland.com