Firewall & Service Tickets *FAST*

draft-herbert-fast-01

Tom Herbert <tom@quantonium.net>

Goal

A method to allow applications to signal the network for services it wants applied to packets that is secure, expressive, not spoofable, deployable, efficient, dynamic, requires no DPI, doesn’t disclose internals of the network, and works with any transport protocol

Where we are today

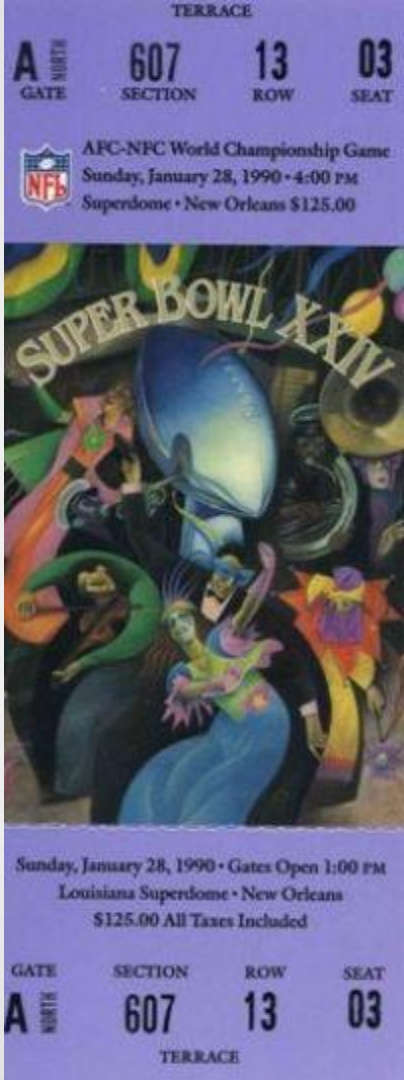
- **Diffserv** allows application to set QoS-- lacks granularity, security, and enforcement
- **DPI** into application layer data-- obsoleted by increased use of TLS, has myriad of issues when stateful
- **SPUD/PLUS** proposals-- requires UDP, flow state, DPI, and global definition of service characteristics
- **Ad hoc mechanisms**-- e.g. proposals to encode network characteristics in TCP option
- **Other in-band signaling** (draft-han-6man-in-band-signaling-for-transport-qos)

Tickets solution

They’re pretty much what you think!

Each packet bears a ticket

Ticket gives right of entry into the network and describes the services granted to the bearer



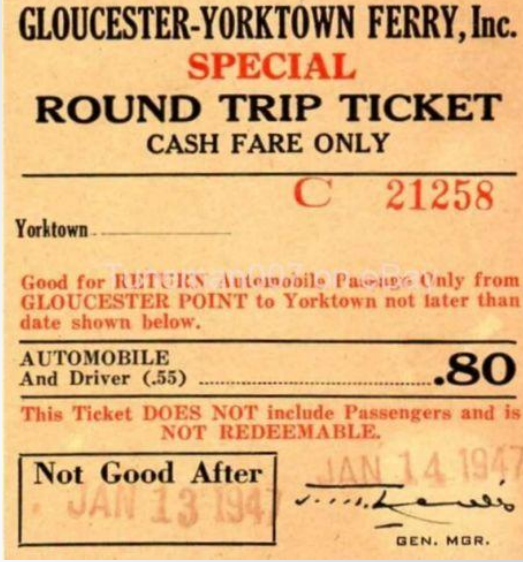
Ticket agents

- Applications requests tickets for flows from a ticket agent in the local network
- A ticket request describes desired services in an expressive and rich language
- Ticket agent issues ticket for host to attach to packets

Using tickets

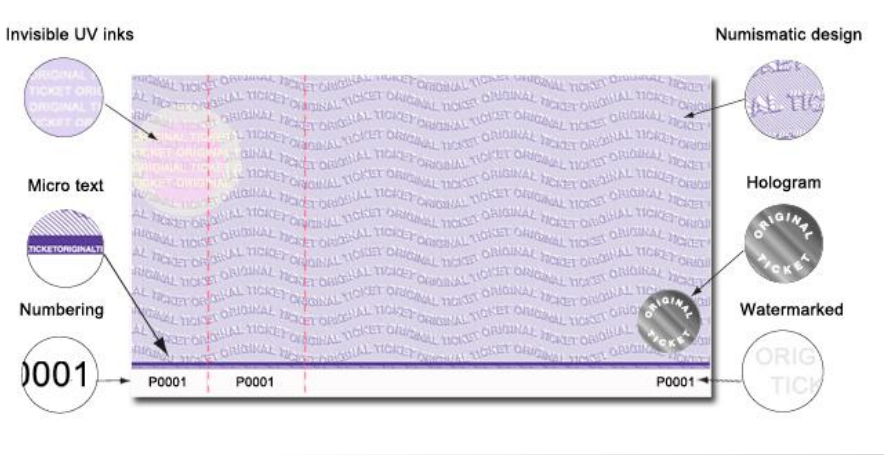
- On ingress into network a node interprets the ticket
- If it’s valid, the packet is allowed to enter the network and specified services are applied
- Services are implemented by methods
  - Diff-serv marking, segment routing encap.
  - Network slices, SFC, NFV, etc.

“Round trip” tickets


GLOUCESTER-YORKTOWN FERRY, Inc.
SPECIAL
ROUND TRIP TICKET
CASH FARE ONLY
C 21258
Yorktown
Good for RETURN Automobile Passage Only from GLOUCESTER POINT to Yorktown not later than date shown below.
AUTOMOBILE And Driver (.55) .80
This Ticket DOES NOT include Passengers and is NOT REDEEMABLE.
Not Good After
JAN 13 194
JAN 14 194
GEN. MGR.

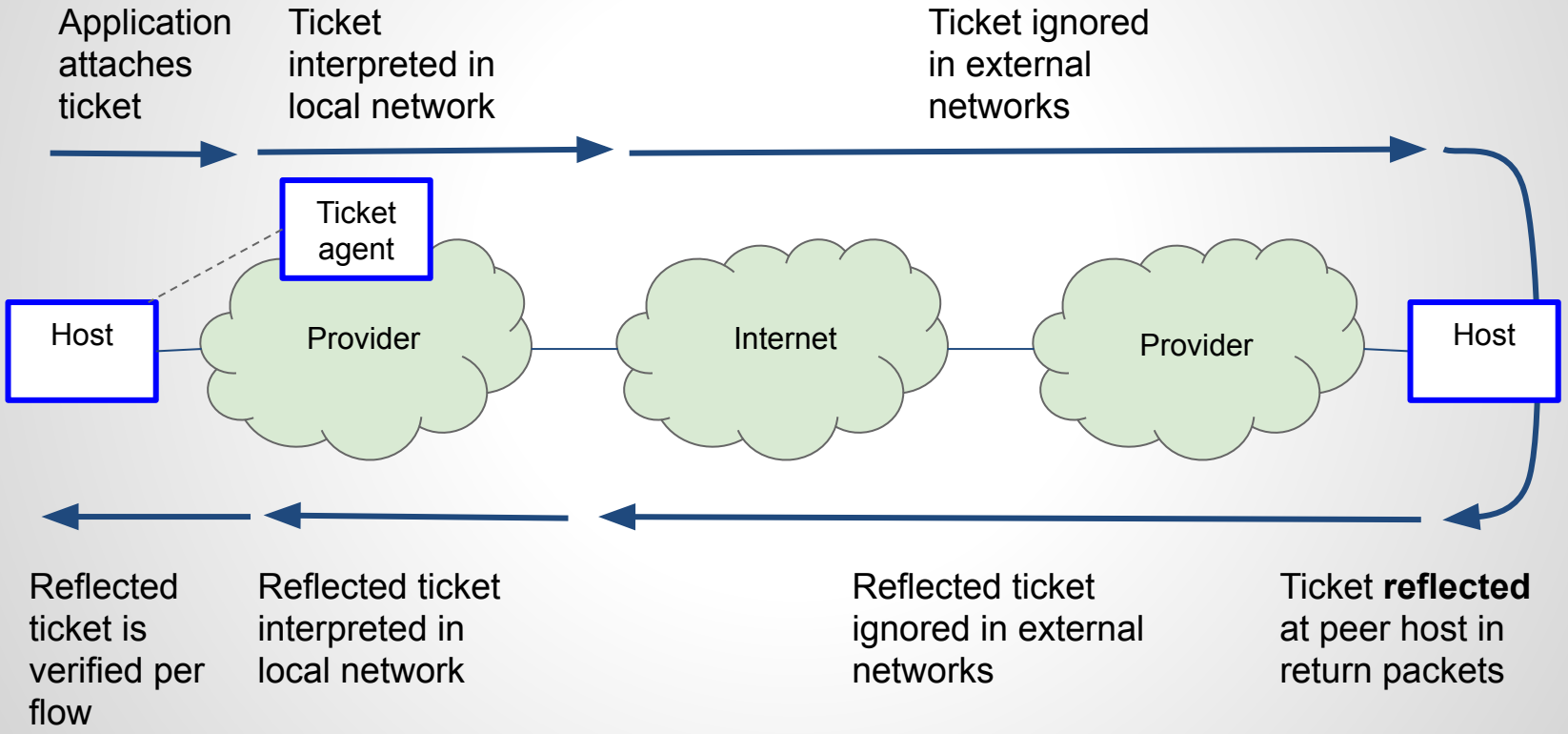
- Tickets must be applied in both directions of a flow
- To be stateless we need the remote peer to attach a return ticket to its packets
- Solution is that peer *reflects* received tickets without trying to interpret them

Ticket security

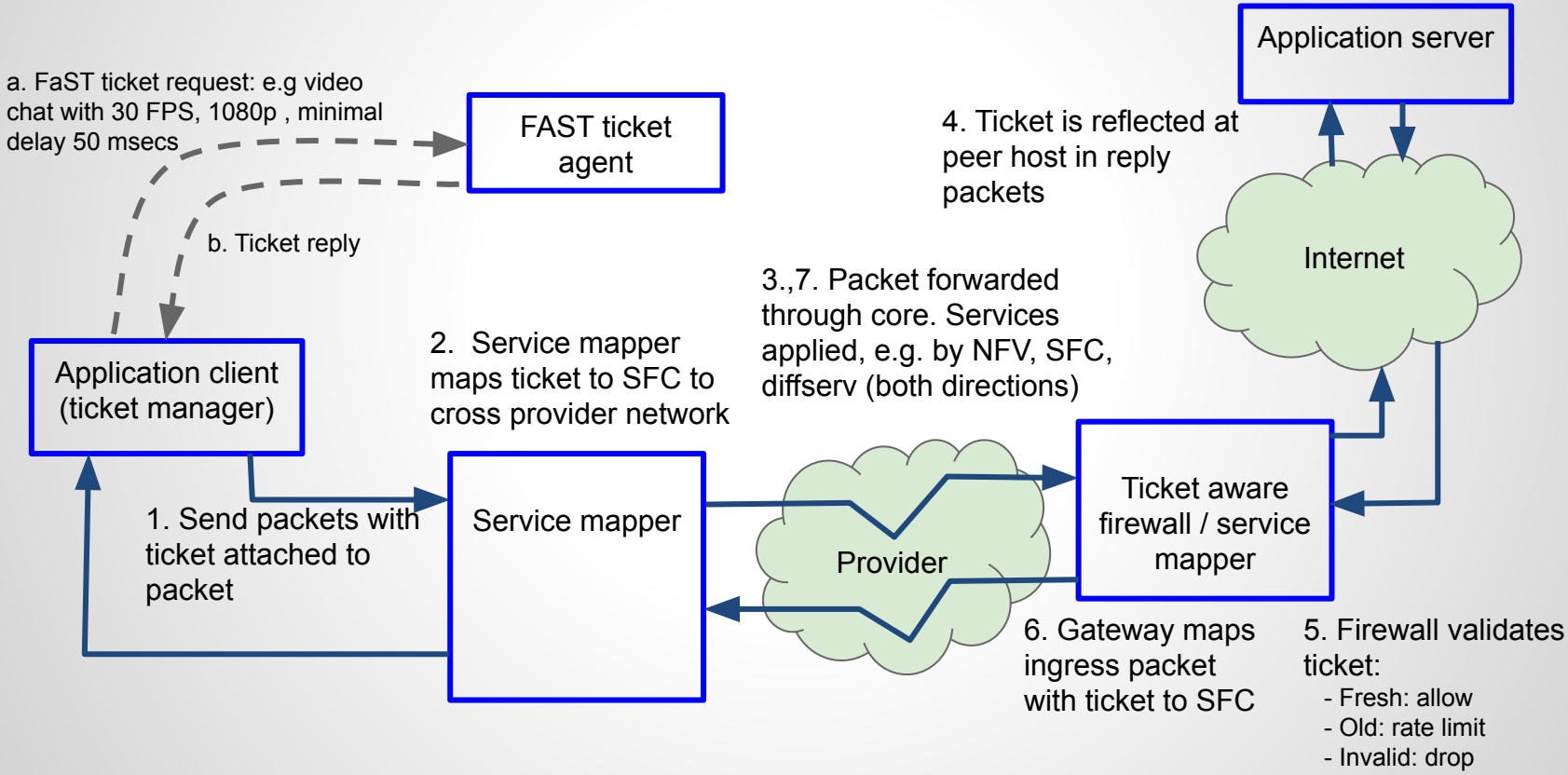


- Non-transferable
- Cannot be forged
- Have an expiration time
- Can’t be interpreted by anyone except by the network in which they were issued (basically they are encrypted)

What ticket flow might look like



Example for video chat service



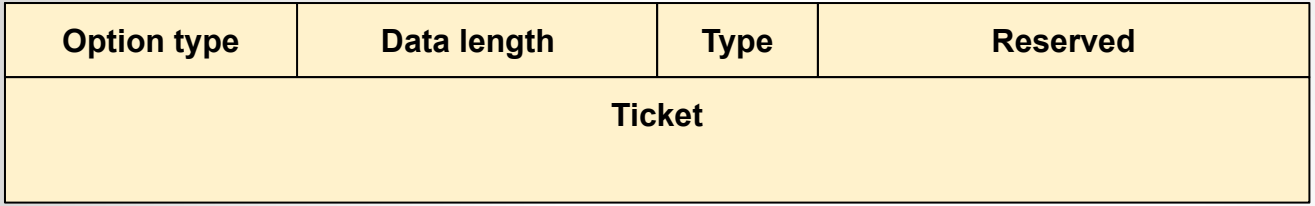
Bidirectional use of tickets

- Each side can use tickets
- Core network is oblivious to tickets
- Up to two tickets/packet (one for each direc.)
  - One not reflected
  - One reflected
- Ticket is for local network if:
  - Destination address is local and was ticket reflected
  - Source address is local and ticket not reflected

Some specifics

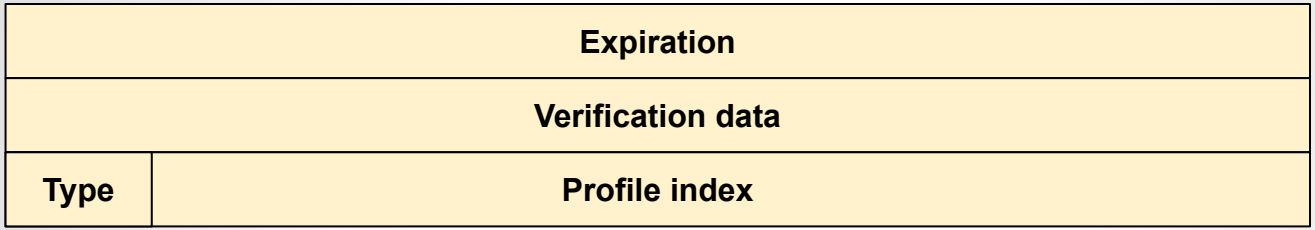
- Tickets encoded in IPv6 Destination options
- They have network specific encoding
- Likely encrypted, where only issuing network has the key
- Have an expiration time (network clock)
  - Valid tickets: allow to pass
  - Old tickets: allow but maybe rate limit
  - Expired or invalid ticket: drop packet

Option format



- Option type: 0x4F* for unmodifiable, 0x6F* for modifiable option
- Data length: Normal option length
- Type: 0 for don’t reflect, 1 for reflect, 2 for reflected ticket
- Ticket: ticket data

Example ticket data

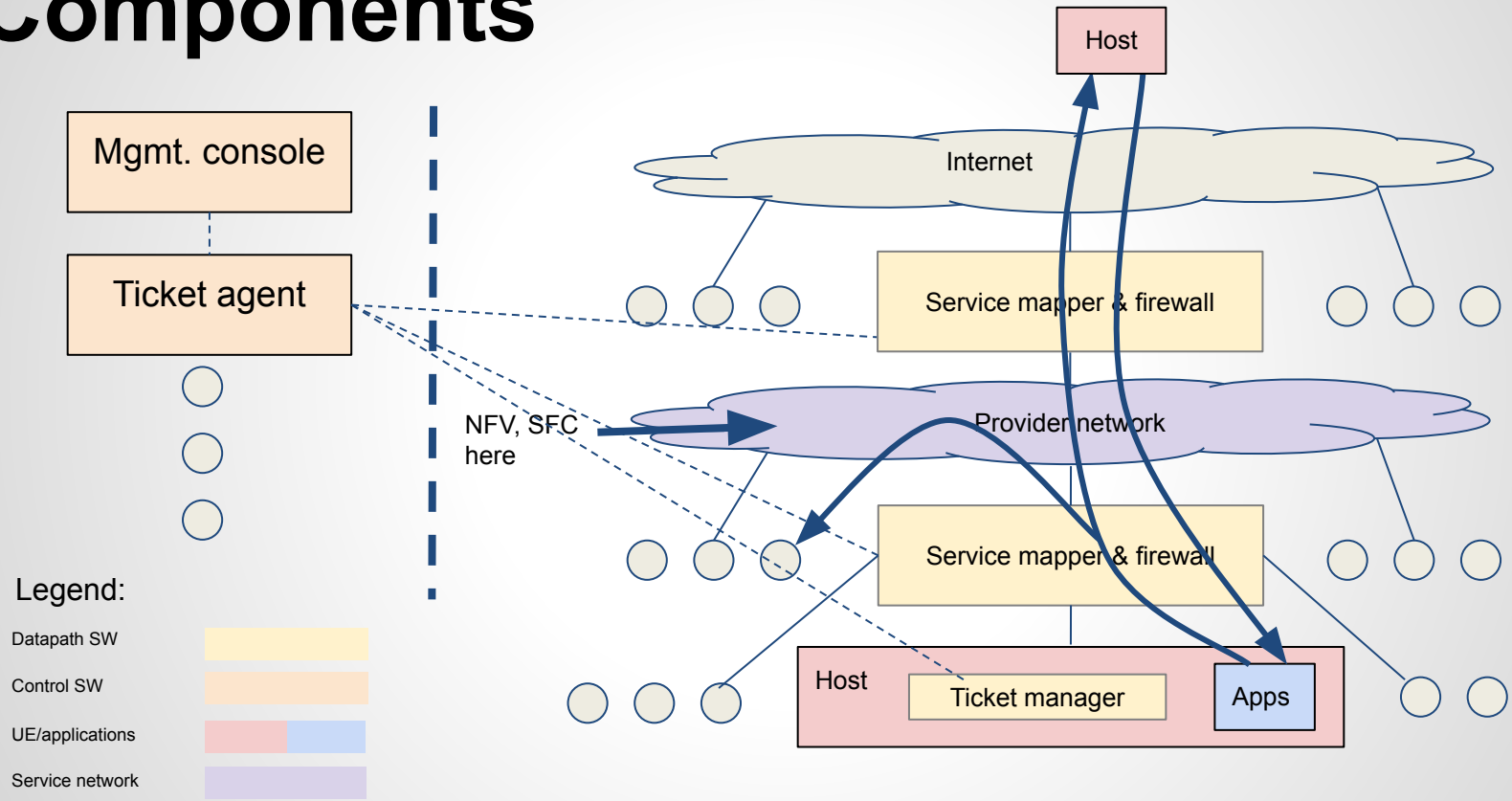


- Expiration: time ticket expires per the clock of the local network
- Verification data (e.g. HMAC)
- Type of ticket (e.g. flags, profile index, fields)
- Profile index- index into a table of service profiles

Control plane

- Applications and ticket agent speak a protocol
- Ticket requests
  - Source and destination of communication
  - Desired services (low latency, real-time video, peak throughput, etc.)
  - High level protocols (XML, REST, etc.)
- Tickets reply
  - Expiration time
  - Opaque ticket to attach to packets

Components


Host
Mgmt. console
Internet
Ticket agent
Service mapper & firewall
NFV, SFC here
Provider network
Service mapper & firewall
Host
Ticket manager
Apps
Legend:
Datapath SW
Control SW
UE/applications
Service network

Host implementation

- OS
  - Enable application setting of tickets. Use sockets APIs to extension headers
  - Library to for applications to talk to ticket manager
- Applications
  - Use library API to request tickets
  - Use API to attach tickets to packets
- Server reflection
  - Automatic reflection in kernel for stateful protocols
  - Connectionless protocols use API to get/set EH

Network support

- Ticket agents can be standalone network processes
- Firewalls and service mappers
  - Modified to speak to tickets agent for configuration
  - Implement ticket validation and service mapping
  - Likely want a ticket cache so don’t have to go through full validation for every packet

Fallbacks for EH drop

- Destination reachable, but tickets not reflected
  - Tickets usable in one direction
  - Set “ticket state” on ingress/egress router
- Packets with EH dropped to destination
  - Happy Eyeballs for extension headers
  - Proposed ICMP errors may help
  - Application stops sending tickets
  - Set quasi “ticket state” on edge devices for tickets seen

Thank you!