14

Si Panda

Firewall and Service Tickets (FAST)

Tom Herbert

Netdev0x17

Problem

Hosts and the network don’t work together for the benefit of the user

It’s hard for users to get the End-to-End service, like QoS, that they need

Solution: Firewall and Service Tickets (FAST)

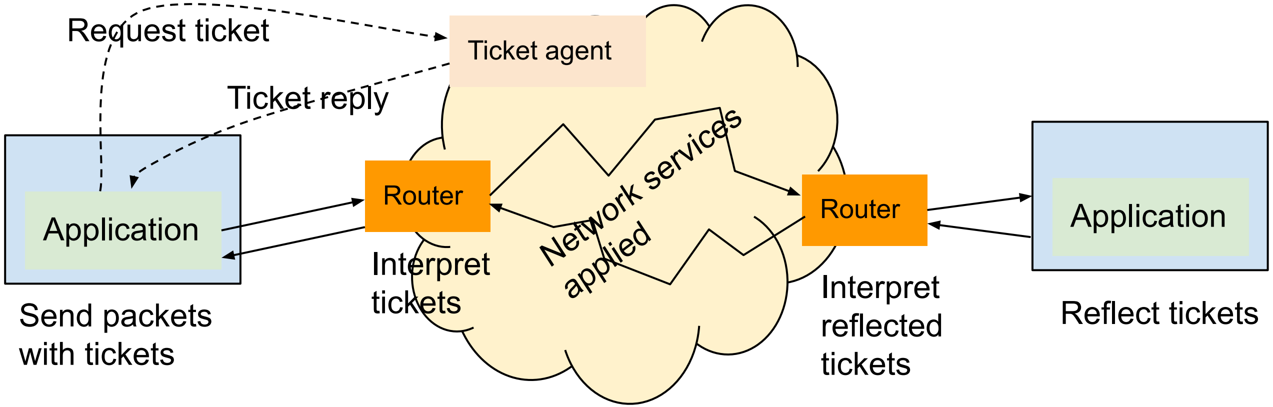
Applications request services from the network, and the network provides “tickets” that are host to network signals attached to packets to provide those services

A win-win-win!

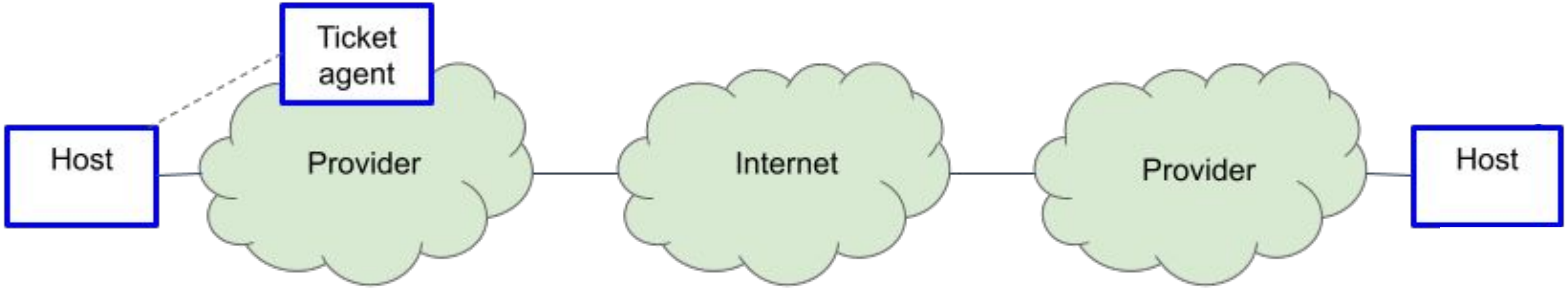
Users benefit from better services, network providers can monetize services, host developers can make ever more interesting applications

FAST basics

1. Applications makes request to a “ticket agent” in their local provider
2. Ticket agent provides a “ticket” that describes the requested services
3. Application attaches ticket to packets it sends
4. Tickets in packets are interpreted by network nodes to provide services
5. Destination reflects tickets for services in the return path



Life of a packet with a FAST ticket attached

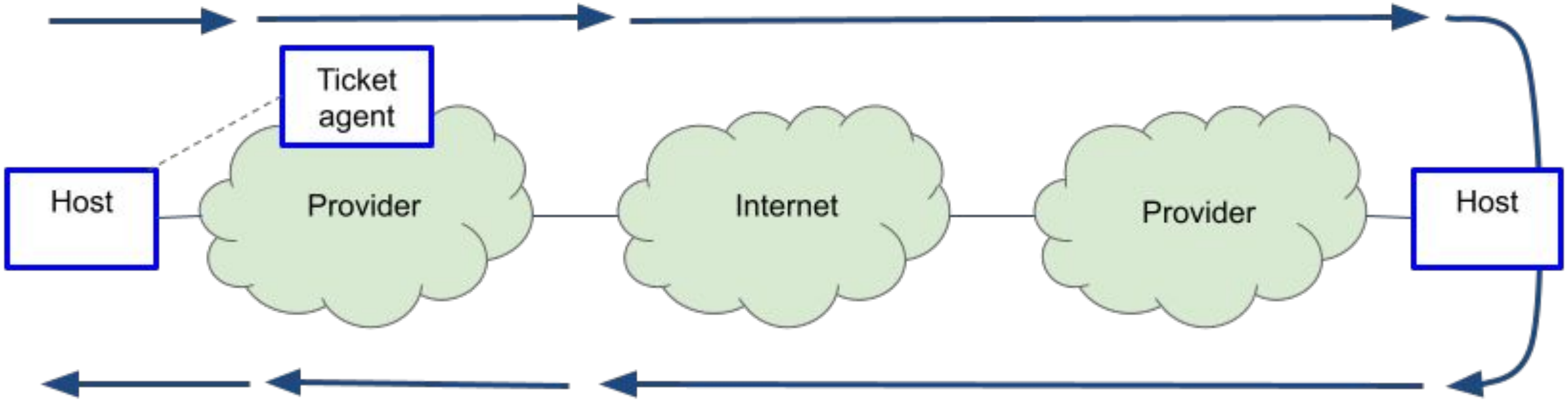


Life of a packet with a FAST ticket attached

Application attaches ticket

Ticket interpreted in local network

Ticket ignored in external networks



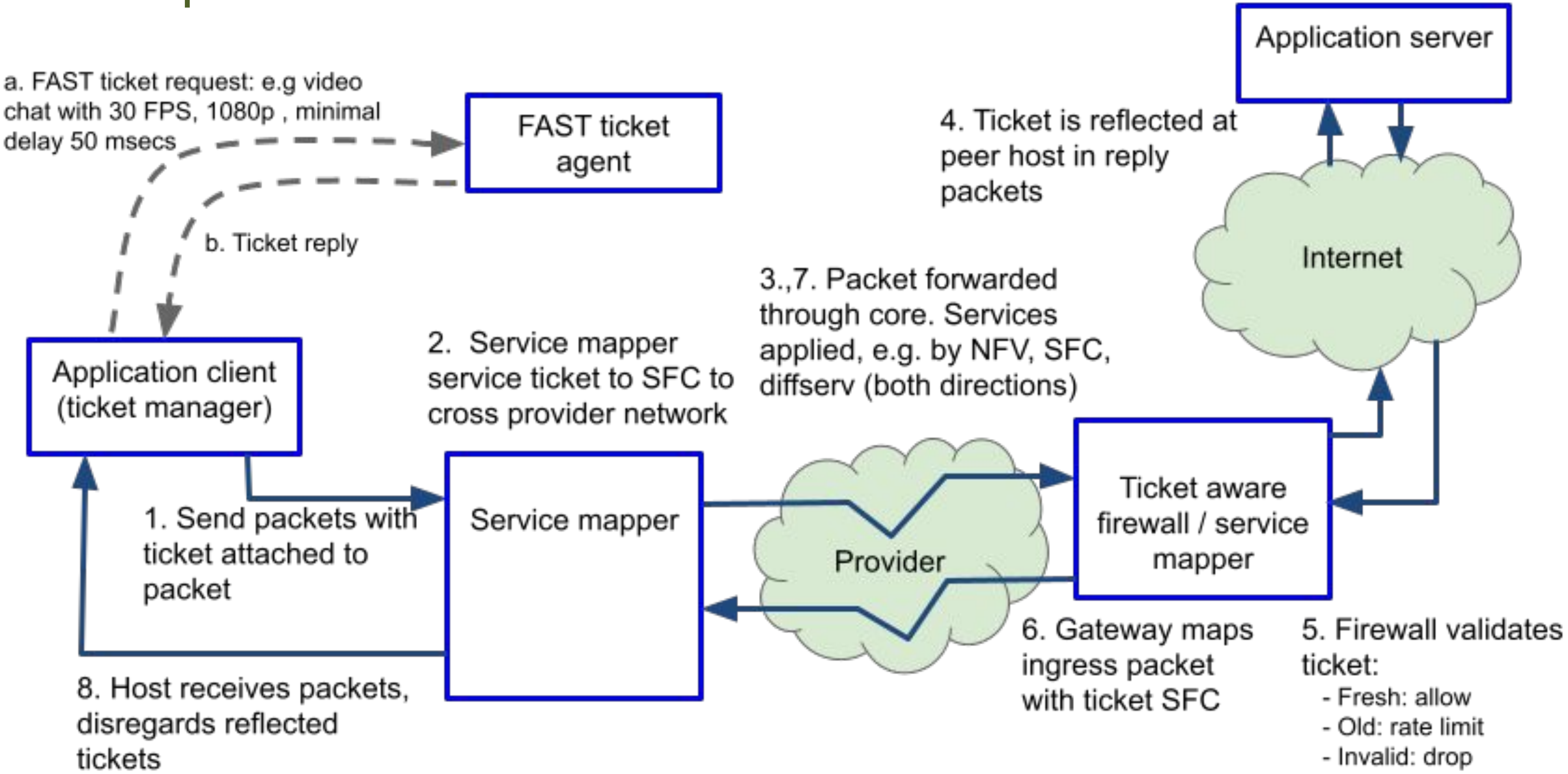
Reflected ticket verified

Reflected ticket interpreted in local network

Reflected ticket ignored in external networks

Ticket **reflected** at peer host in return packets

FAST operation in 5G



a. FAST ticket request: e.g video chat with 30 FPS, 1080p , minimal delay 50 msecs

FAST ticket agent

b. Ticket reply

Application client (ticket manager)

1. Send packets with ticket attached to packet

2. Service mapper service ticket to SFC to cross provider network

Service mapper

3.,7. Packet forwarded through core. Services applied, e.g. by NFV, SFC, diffserv (both directions)

Provider

4. Ticket is reflected at peer host in reply packets

Application server

Internet

Ticket aware firewall / service mapper

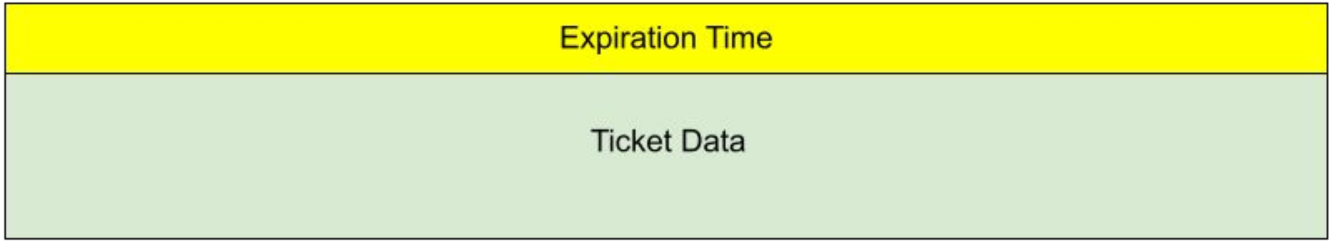
5. Firewall validates ticket:
- Fresh: allow
- Old: rate limit
- Invalid: drop

6. Gateway maps ingress packet with ticket SFC

8. Host receives packets, disregards reflected tickets

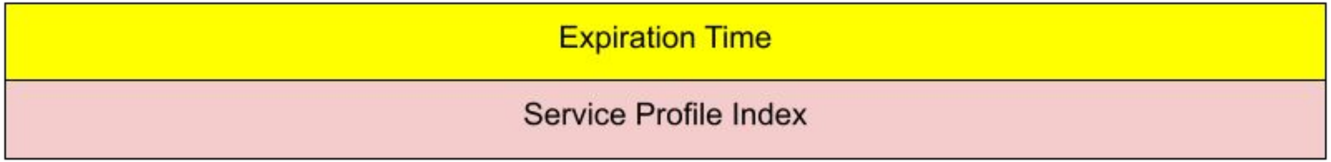
Ticket properties

- Tickets are a type of Host to Network signal
- Encrypted and authenticated to prevent abuse
- Expiration time to limit use, revocable by a “revoked list”
- Reflection properties (to be reflected, don’t reflect, reflected)
- A ticket format that’s concise is nice



Expiration Time

Ticket Data



Expiration Time

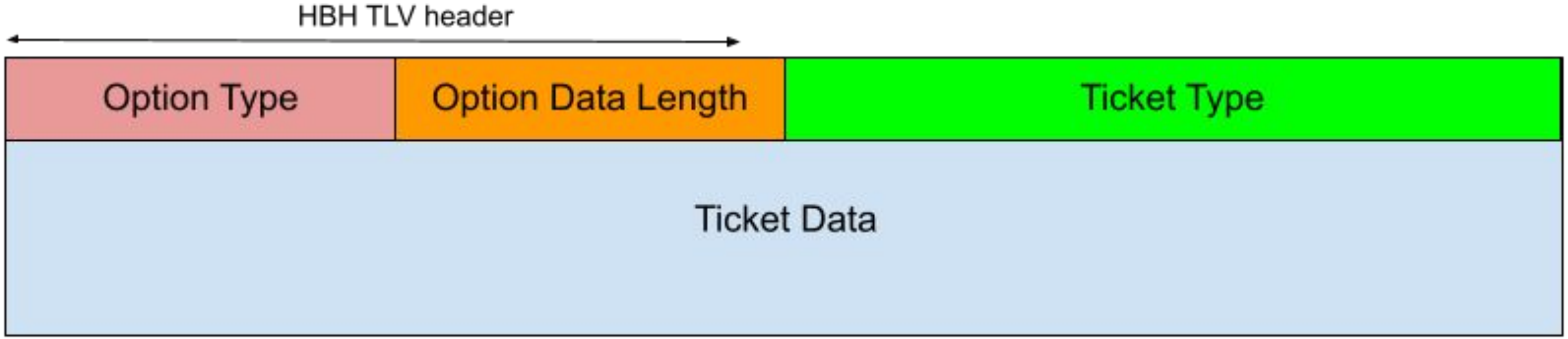
Service Profile Index

Carrier (first the alternatives)

- Stateful firewalls
  - Break E2E model (break multi path, multi-homing)
  - Limited protocol support
  - Limited visibility into application
- DPI
  - Coarse information
  - Narrow protocol support
- SPUD
  - Routers read UDP payload
  - Only works with UDP
  - Still relied on flow tracking
- Segment routing
  - Only in limited domains
  - Verbose protocol
  - No reverse path information
- Overload flow label, v6 addresses
  - Very limited bits
  - Bits are already used
- Diffserv
  - Few bits
  - No authentication

Propose carrier: IPv6 Hop-by-Hop Options

- They were designed for this sort of thing
- Work with any IP protocol
- Fixed position in a packet
- As many bits as needed



HBH TLV header

Option Type

Option Data Length

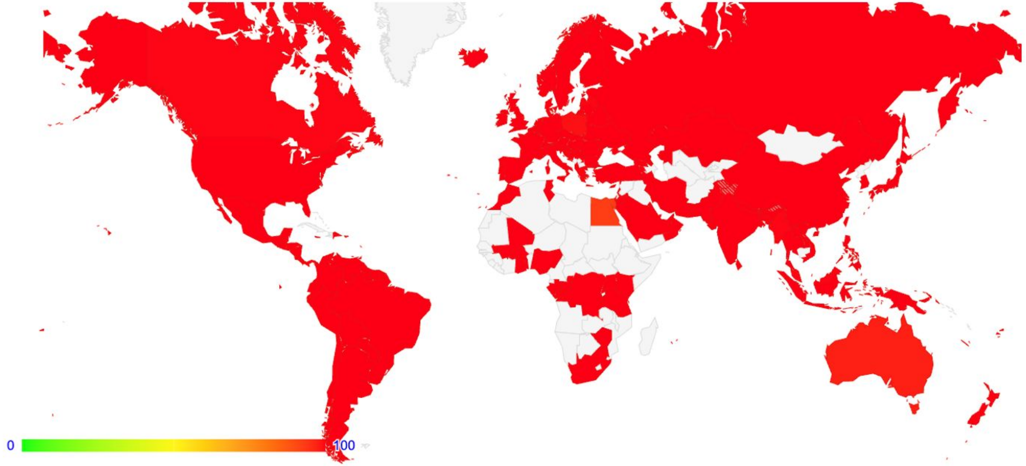
Ticket Type

Ticket Data

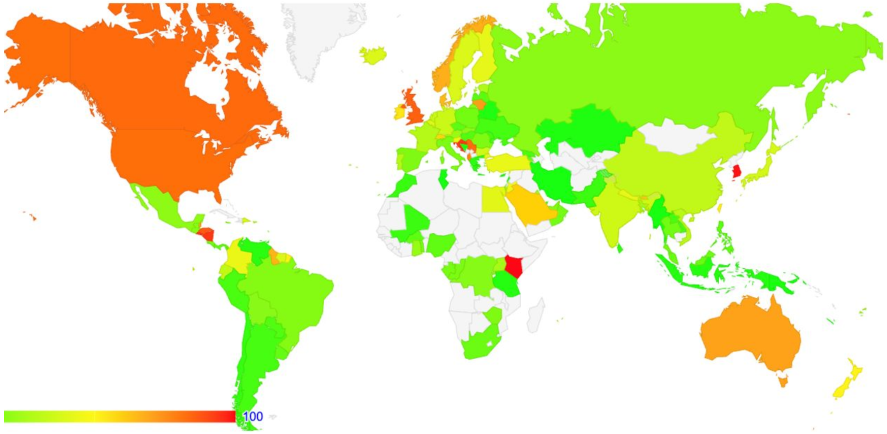
Drawbacks of Hop-by-Hop Options

- IPv6 only (draft-herbert-ipv4-eh-01 to use EH in IPv4)
- Experience high drop rate on the Internet (**>99%** according to APNIC memo)

Hop-by-Hop Options drops



Destination Option Drops



Source https://blog.apnic.net/2022/10/13/ipv6-extension-headers-revisited/

Dealing with HBH Options drops

- RFC8200 lets routers ignore HBH Options completely
- RFC8883 defines ICMP messages that may be sent with a router drops a packet because a processing limit is exceeded
- I-D.ietf-6man-hbh-processing clarifies HBH processing (don't process in slow path)
- draft-ietf-6man-limits specifies limits on HBH processing that may be applied
- draft-herbert-eh-inflight-removal describes protocol to remove HBH Options header (and Routing header) in-flight

Linux kernel support to make extension headers useful

[illegible]

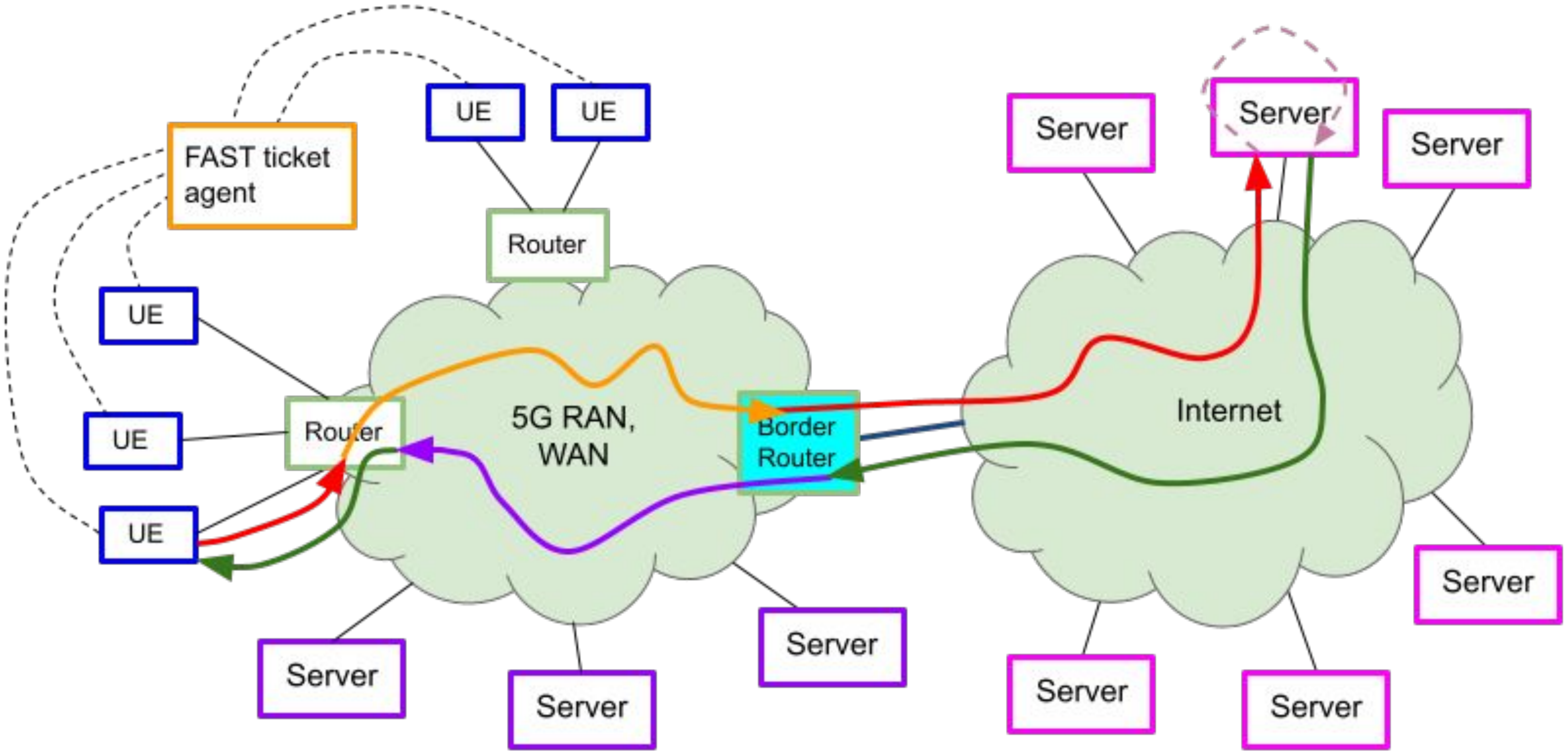
Status and future work

- FAST draft generating interest in IETF
- More generally Host2Net signaling is drawing initial interested
- Work need on EH kernel patches

- EH limits and EH support in Linux (also RFC8883)
- PoC development
    - Host/application
    - Network nodes
    - Ticket agent

Thankyou!

Use case: 5G


FAST ticket agent
UE
UE
Router
UE
UE
Router
UE
5G RAN, WAN
Border Router
Server
Server
Server
Internet
Server
Server
Server
Server
Server
Server
Server
Server