QUIC-LB Update

Martin Duke

IETF 110, Virtual (10 Mar 2021)

What's in it?

QUIC is opaque!

Encode stuff in CIDs

- CID length, mainly for hardware accelerators
- "Server ID", a routing instruction for load balancers

Encode stuff in Retry Tokens

- Original Destination CID
- Retry Source CID
- Validation Information (sometimes)

Retry Services

- Non-shared-state
  - Very simple
  - Only the service can issue/authenticate Retry tokens
  - NEW_TOKEN tokens admitted when not under attack
- Shared state
  - Both service and server can generate tokens and authenticate each other's tokens
  - Must share key and IV
  - Format substantially revised to improve security properties (thanks Christian)
- Servers can control handling of unsupported versions via allow- or deny-list

CID Encoding

- Length self-encoding is stable
- Management of key rotation/rolling config changes is stable – first two bits indicate which config a CID should be decoded with
- Three algorithms:
  - Plaintext CID: no protection of server ID, CIDs ≥ 3 Bytes
  - Stream Cipher CID: FFX-like encryption, CIDs ≥ 10 Bytes (thanks Christian)
  - Block Cipher CID: “more” encrypted, CIDs ≥ 17 Bytes
- Two Server ID Allocation Mechanisms (#64)
  - Static: Configuration Agent defines server ID mapping
  - Dynamic: Servers learn server IDs through passive observation of entropy provided by client-generated CIDs (thanks Ian)

Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state

Table (config 0):

Server	IDs
a.b.c.d	
e.f.g.h	

a.b.c.d

Server IDs (config 0): none

e.f.g.h

Server IDs (config 0): none

Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state

Table (config 0):

Server	IDs
a.b.c.d	
e.f.g.h	

CID: config 0, SID 0x3fa1

a.b.c.d

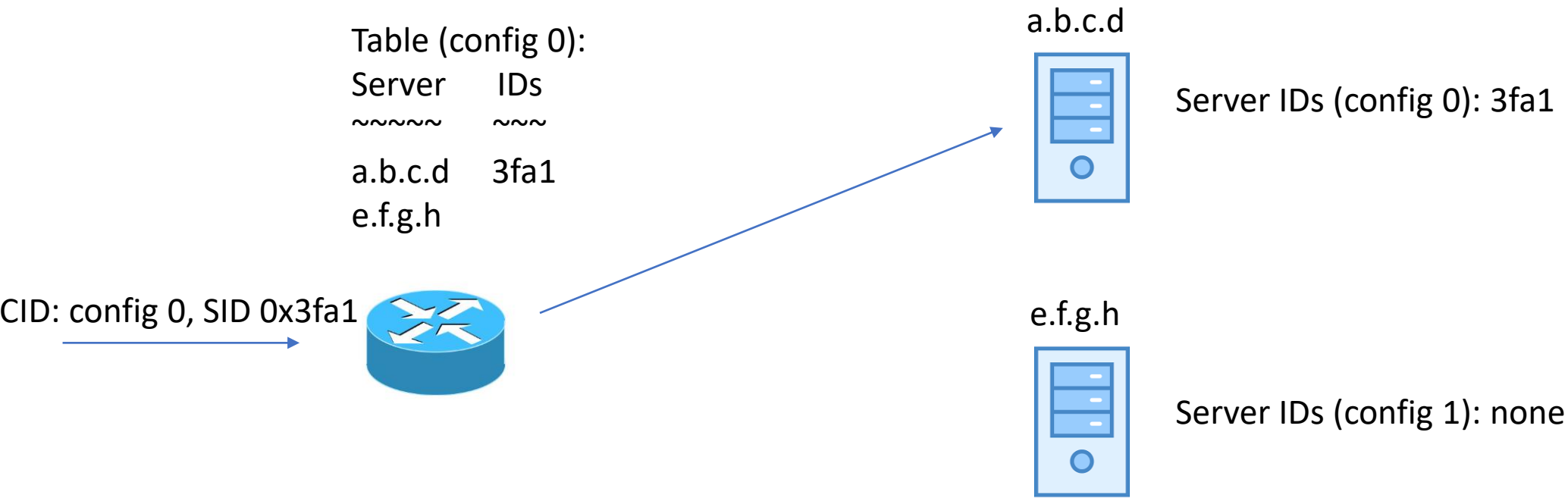
Server IDs (config 0): none

e.f.g.h

Server IDs (config 0): none

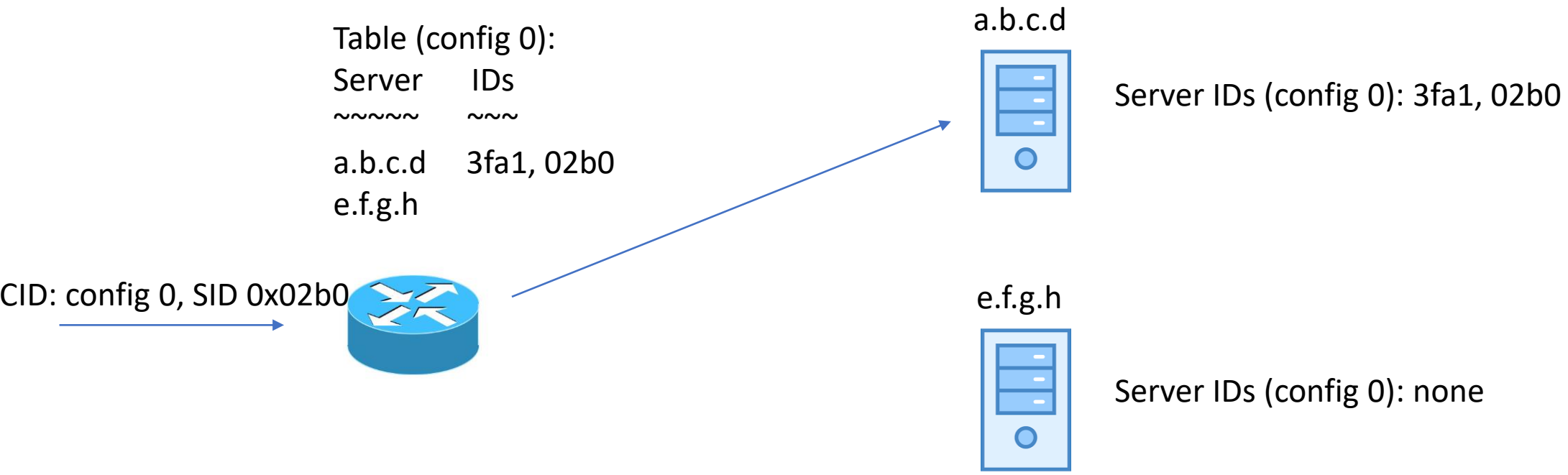
Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state



Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state



Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state

Table (config 0):

Server	IDs
a.b.c.d	3fa1, 02b0
e.f.g.h	

a.b.c.d

Server IDs (config 0): 3fa1, 02b0

CID: config 1, SID ?

e.f.g.h

Server IDs (config 0): none

Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state

Table (config 0):

Server	IDs
a.b.c.d	3fa1, 02b0
e.f.g.h	

a.b.c.d

Server IDs (config 0): 3fa1, 02b0

CID: config 1, SID ?

e.f.g.h

Server IDs (config 0): none

4-tuple routing, for now

Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state

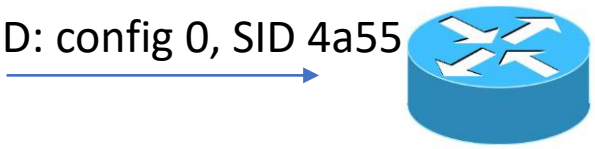
Table (config 0):

Server	IDs
a.b.c.d	3fa1, 02b0
e.f.g.h	

a.b.c.d

Server IDs (config 0): 3fa1, 02b0

CID: config 0, SID 4a55



e.f.g.h

Server IDs (config 0): none

4-tuple routing, for now

Server ID tradeoffs

- Static config is a pain
- Dynamic has unfortunate corner cases, more state

Table (config 0):

Server	IDs
a.b.c.d	3fa1, 02b0
e.f.g.h	4a55

a.b.c.d

Server IDs (config 0): 3fa1, 02b0

CID: config 0, SID 4a55

e.f.g.h

Server IDs (config 0): 4a55

Replace all CIDs

See Issue #84 (closed)

Issue #80

- Server ID Lengths currently expressed in octets
- Express in bits instead?
  - Might save a byte of CID length
  - Yet more complexity

Discussion

Implementation Status

- CID encoding/decoding library is open-source (static SID allocation only)
  - https://github.com/f5networks/quic-lb
- NGINX based load balancer
  - https://github.com/martinduke/nginx-quic-lb
- Plaintext LB + (obsolete) shared-state Retry service
  - https://github.com/alipay/quic-lb

- Gaps!
  - Servers that support mobility – ready to interop!
  - Dynamic SID allocation (Google, sorta)
  - Retry services: both ends

WGLC?