

QUIC LIBRARY IDENTIFICATION ON THE INTERNET

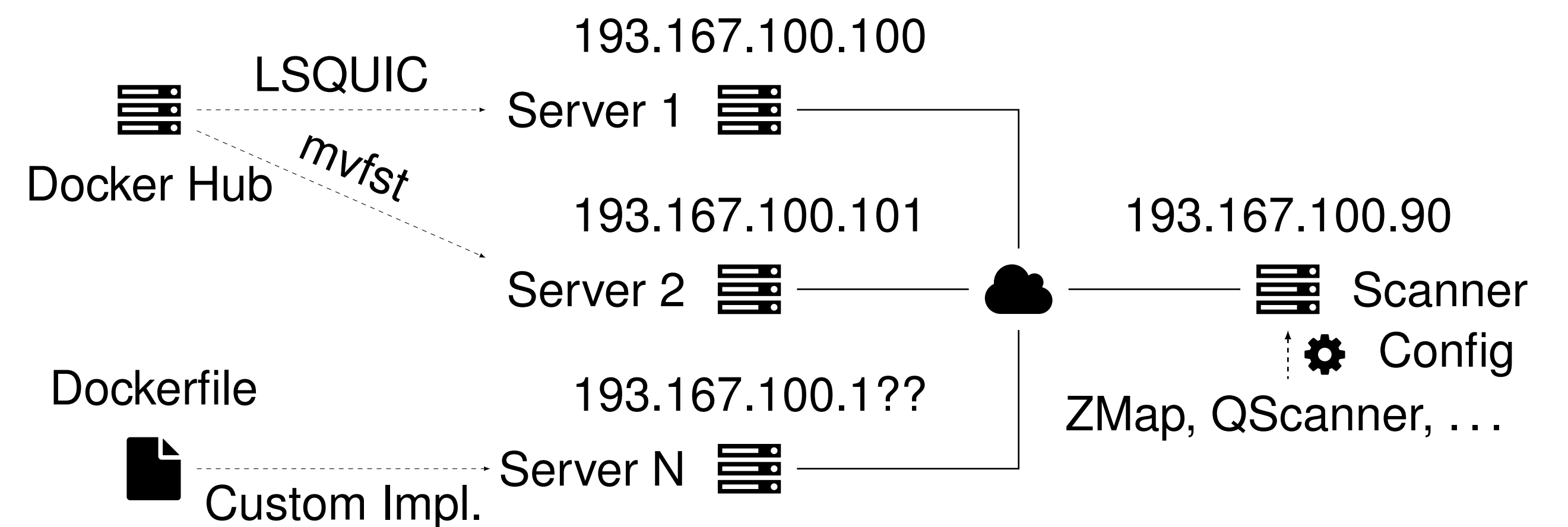
The Diversity of QUIC

The IETF lists 24 libraries [2]

- ▶ QUIC is designed on top of UDP [3]
 - ▶ Can be implemented in user space
 - ▶ All implementations follow the same standard
 - ▶ General interoperability is given
- Differences in performance and functionality are visible

Knowledge about used libraries is helpful to analyze QUIC measurements and traffic

Test Environment



▶ Local environment to test scanners:

- ZMap
- QScanner
- quicreach

Library Identification

- ▶ Based on:
 - Order of transport parameter
 - Error messages

Implementation	Code	Message
LSQUIC	0x178 0x150	no suitable application protocol TLS alert 80
NGINX	0x178	handshake failed
Google Quiche	0x178	(frame type: 0x6): 28:TLS handshake failure (ENCRYPTION_INITIAL) 120: no application protocol
Haskell QUIC	0x178	no supported application protocols
Akamai QUIC	0x150 -----	200:TLS handshake failure (ENCRYPTION_INITIAL) 80: internal error PROTOCOL_VIOLATION: 28:No known ALPN provided by client
mvfst	0x178	(frame type: 0x1c): fizz::FizzException: Unable to negotiate ALPN, as required by policy. policy=AlpnMode::Required

Identified Libraries

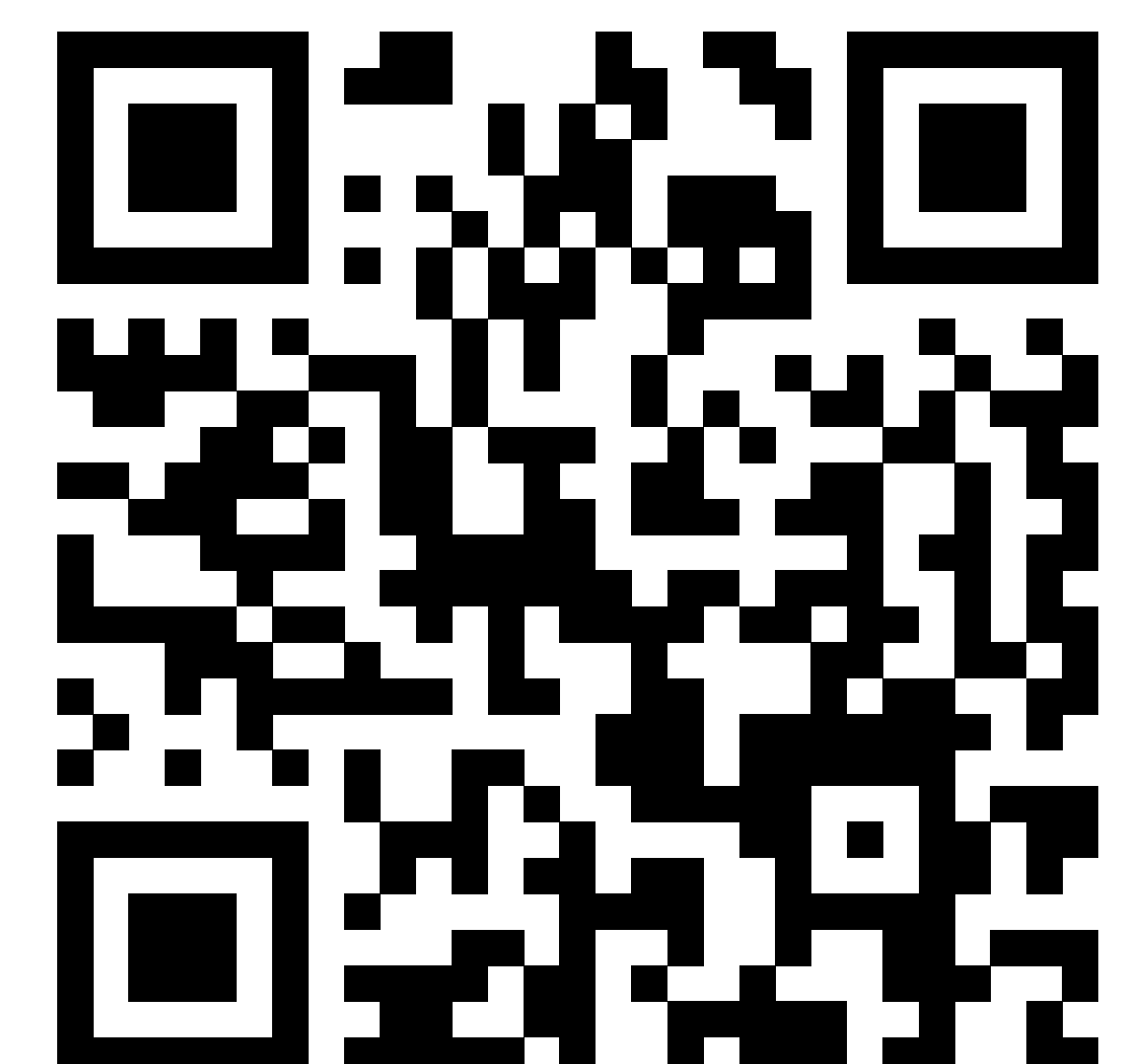
Libraries found during Internet-wide cans:

Implementation	IPv4		IPv6	
	Addresses	ASes	Addresses	ASes
Akamai QUIC	7 791 313	910	141 921	257
LSQUIC	388 650	2339	1 781 219	332
Google Quiche	262 673	4771	83 811	1356
quiche	119 988	303	99 387	46
quicly	103 503	23	485	13
mvfst	69 653	2555	14 953	1457
NGINX	50 743	558	2206	175
quic-go	23 076	1246	1443	300
s2n-quic	7910	2	404 015	1
MsQuic	662	169	222	30
Quinn	361	66	28	19
HAProxy	229	62	0	0
XQUIC	25	2	2	1
ngtcp2	4	4	1	1
aioquic	3	2	1	1
Neqo	2	1	0	0
Quant	1	1	4	1
Haskell QUIC	1	1	1	1
None	3 110 411	10 275	5 841 819	679

Conclusion & Future Work

The approach is effective to identify server libraries.

- ▶ We find at least one deployment for 18 libraries
- ▶ The approach based on error messages does not require a successful handshake but not all libraries send a specific string
- ▶ The order of transport parameters requires a handshake but can be used for all libraries
- ▶ Some libraries randomize the order, but the set is still unique, e.g., Google QUICHE [1]



[1] Google. Randomization of QUIC Transport Parameter Order, 2023.
[2] IETF QUIC Working Group. Implementations, 2023.
[3] J. Iyengar and M. Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, May 2021.