nDPI performance and QUIC

Ivan Nardi
Agenda

• nDPI performance:
  • testing nDPI with existing probes with REAL traffic

• QUIC: let's demystify this new protocol
Who am I?

• Ivan Nardi, @ AI2M:
  • lawful interception, investigation analysis, big data retention
  • voice/IP metadata collection, processing and reporting
  • network probes and DPI

• ivan@ai2m.eu
nDPI: integration on existing probes

- Software:
  - nDPI (dev branch, 2560260a) with default configuration
  - all ~300 protocols enabled + ~20 other protocols
- Full metadata extraction. Exceptions:
  - no DNS sub-classification
  - no parsing of HTTP replies
  - no JA3/JA3S calculation
- Some private patches: integration, performance, statistics, ipv6
nDPI: single thread performance

- Environment (single-thread)
  - Intel Xeon E5-2690 @ 2.90GHz (2012!)
  - Intel X710 4x10Gb
  - 4 * 10Gb links
- Traffic: residential (fiber & ADSL), mobile, enterprise
Input traffic and packet loss
Classification: top protocols

![Classification: top protocols](image)
Packets/flow in DPI data path
Profiling via perf
nDPI performance: multiple threads

- Environment (multi-threads)
  - 2 x Intel Xeon E5-2697A v4 @ 2.60GHz, 16 core (2016)
  - Intel X710 4x10Gb
  - 24 * 10Gb links

- Results:
  - no packet loss; same classifications as ST; no sharing data
nDPI: performance

- Conclusions:
  - nDPI might be extremely cheap (from a resources POV)
  - nDPI has optimal scaling performance
QUIC: what?

• First things first: thanks to @programmingart for allowing to use all these nice images

• "QUIC is a secure general-purpose transport protocol [and it] is secured using TLS" [RFC8999-9002][05/2021]

• Oversimplifying: QUIC = TCP + TLS over UDP
QUIC: who and since when?

- HTTP/3 over QUIC [RFC9114][06/2022]: HTTP traffic from browsers and mobile apps
  - All major browsers
  - All major CDNs: Fastly, Cloudflare, Akamai...
  - Biggest internet company: Google, FB, Snapchat
QUIC: who and since when?

• DNS over QUIC [RFC9250, 05/2022]
  • DoH-DoT privacy + UDP latency
  • AdGuard deployed it on 12/2020[1]

• SMB over QUIC
  • Present in Windows 11 and Windows Server 2022[2]

QUIC: who and since when?

- **ICloud Private Relay [12/2021]**
  - Dual-hop architecture: no single party has access to both the user’s IP address and SNI\(^[[1][2]]\)

- **QUIC Proxy (MASQUE WG)\(^[[3]]\)**

---

\(^[[2]]\) [https://blog.cloudflare.com/icloud-private-relay/](https://blog.cloudflare.com/icloud-private-relay/)
QUIC: who and since when?

- RTP/RTCP/WEBRTC over QUIC
  - MoQ (Working group?)[1]
  - RUSH: Facebook Live Video Ingest [07/2021][2]
  - QUIC demultiplexing (like STUN/RTP/RTCP over UDP)[3]
  - Snapchat (video)calls [07/2020, at least]

QUIC: who and since when?

- Fortigate Url filter [05/2022]
  - Inspecting and blocking HTTP3 traffic depending on keyword match[^1][^2]

- BGP over QUIC[^3]
- SSH over QUIC[^4]

[^1]: https://docs.fortinet.com/document/fortigate/7.2.0/new-features/440398/inspecting-http3-traffic
[^2]: https://www.youtube.com/watch?v=SI4OXspDuNI
QUIC: what?

• Oversimplifying: QUIC = TCP + TLS over UDP
  • all TCP features: reliability, acknowledgements/retransmissions, a highly complex handshake, flow-control and congestion-control
  • all TLS features: encryption always on; no such thing like "plaintext QUIC"
  • it is built on top of UDP
QUIC: differences compared to TLS/TCP/UDP

- Connection set-up is faster

Session resumption and 0-RTT data are TLS features which can be used with both TCP and QUIC to send an HTTP request during the handshake.
QUIC: differences compared to TLS/TCP/UDP

• Better performance when data packets are lost
  • Supports for multiple independent byte streams (like SCTP)

• Stable connections when networks change
  • Connection IDs (like GTP TEID or SCTP Verification Tag)
    • In TCP, connections are identified by the 5-tuple. So, if just one of those five parameters changes, the connection becomes invalid and needs to be re-established
    • In QUIC, a number is assigned to each connection and it uniquely identifies the connection between two endpoints.
QUIC: differences compared to TLS/TCP/UDP

1. TCP handshake from WiFi IP starts connection

2. TCP data packet from unknown IP... no idea what to do with this

Let's use this QUIC Connection ID (CID) for now:

The network is different, but the CID is the same, so it's the same connection
QUIC: differences compared to TLS/TCP/UDP

• Deeply integration with TLS: user data and L4 fields are always encrypted
QUIC: advanced features

- QUIC is easier to improve and develop
  - Rapid deployment of QUIC modifications updating only the endpoints
  - Goal: avoid protocol ossification
- Connection migration: connection ID allows connections to survive changes to endpoint addresses (IP and/or port)
  - Nat rebinding or switching networks
- Multi-path: using multiple path at the same time [2019][1]
- Integrated logging facilities[2]

QUIC: conclusion

- Take a look at what’s happening on your networks at UDP/443
- We will see a lot of changes in network protocols in the next months/years

Thanks for your time. Questions?