Migrating from a commercial DPI library to nDPI: our journey

Ivan Nardi

ntopConf'23

Sept, 21-22 · Pisa

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

A little bit of history

- 2005: started writing a DPI library in-house, completely from scratch
  - 2009: Gmail login page switched to HTTPS
  - 2010: OpenDPI released
- 2011: switched to a commercial solutions
  - commercial library + custom code for specific protocols (3GPP) and metadata
- 2012: nDPI announced. First test: too limited
- 2019: re-discovered nDPI and started using it for “second opinions”

A little bit of history

- Two major issues:
  - 2017-2020: missing/incomplete QUIC support in the commercial DPI engine
    - 2020: wrote a QUIC dissector from scratch and integrated it into our application
      - A dirty hack causing performance drop
  - 2022: vendor drastically changed license terms (features, availability and fees)

- What other DPI library could we used?

Comparing DPI engines is hard!

- Not so many choices:
  - Sandvine, Rohde and Schwarz, Qosmos
  - nDPI, ..., Libprotoident [2020], Tstat [2016], L7-filter [2011], peafowl [2020]

- Articles/papers:
  - "DPI Solutions in Practice: Benchmark and Comparison" [2021]
  - "Independent Comparison of Popular DPI Tools for Traffic Classification [2015]
  - Blog post by ipoque on commercial vs open-source DPI [2021]

Comparing DPI engines is hard!

- How to uniform results?
  - TLS vs TLS/HTTP vs HTTPS;
  - FB vs FB_VIDEO vs FB_MSG vs FB_VOIP;
  - STUN vs STUN/XXX; SKYPE vs MSTEAMS;
  - DUO vs HANGOUT vs GOOGLECHAT
- How to get the ground truth?
- A DPI engine might provide other information than classification
- Protocols number is a marketing goal

nDPI: the good

- Maintained code (by a real company)
- Permissible license and access to the code
- The code is maintainable and it has been written paying attention to performance
- Good overall performances
- Good overall classification capabilities
- Interesting algorithms

nDPI: the bad

- Missing any configuration knobs: pretty much everything is enabled by default and you can’t disable it
  - Slightly better recently: all features added in the last ~2 years are fully configurable

nDPI: the ugly

- nDPI development lifecycle is typically 6 to 9 months: 2 releases/year at best!
- No API/ABI compatibility at all!

Biggest challenges

- Compatibility with existing deployments:
  - application configurations
  - (new) flow information must be compatible with existing information already stored
    - protocol/category IDs
    - metadata format
- IPv6 support

So they say - Is the root of all evil today

- Licenses of commercial DPI libraries are quite expensive while nDPI is free

- Using (or, worse, integrating) open-source software is not free
  - support/updates/bug-fixes costs

- We are quite confident that moving to nDPI will be the right choice even from a cost perspective (in the medium-long term)

This is the end, my only friend

- Is nDPI perfect? No
- Do commercial libraries provide more features or capabilities? Yes
- Are we happy with nDPI? Yes
- Should we recommend nDPI to anyone? Definitely