Decoding and analysis of high precision nanosecond Aggregation Tap timestamps

CSU22013 Software Engineering Project I
Group 13

Final Presentation & Showcase
The Development Team

3rd Years

- Finn Jaksland
  Project Manager

- Alannah Henry
  Product Owner

- Brendan Jobus
  Backend Manager

2nd Years

- Cillian Fogarty
  Frontend Developer

- Owen Gallagher
  Backend Developer

- Darren Aragones
  Backend Developer
The Client:
Pico Quantitative Trading

Scope:
Deconstruct Arista 7280 packets & handle timestamp errors

- Command-line program that analyses packet capture (.pcap files)
- Find erroneous packet timestamps and output report to .csv
Technologies: Libpcap

Provides implementation-independent access to the underlying packet capture facility provided by the operating system

- Returns pointer to .pcap data in memory
- With knowledge of packet type and offsets can manually decode packet
Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
Ethernet II, Src: 3Com_03:04:05 (00:01:02:03:04:05), Dst: Cisco_20:72:c0 (00:13:5f:20:72:0c)

Arista Vendor Specific Protocol
  Sub Type: timestamp (1)
  Version: Version 1 (0x0010)
  Timestamp (TAI)
    Seconds: 1585552437
    Nanoseconds: 687513958
    Type: 802.1Q Virtual LAN (0x8100)

802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 137
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
User Datagram Protocol, Src Port: 1, Dst Port: 1
Data (18 bytes)
<table>
<thead>
<tr>
<th><strong>Functional Requirements</strong></th>
<th><strong>Non-functional requirements</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Take packet capture files as an input</td>
<td>High-speed performance</td>
</tr>
<tr>
<td>Parse the pcap. files and verify</td>
<td>Robustness of the system</td>
</tr>
<tr>
<td>Detect date format</td>
<td></td>
</tr>
<tr>
<td>Outputting a copy of modified capture files as a CSV file</td>
<td></td>
</tr>
</tbody>
</table>
Challenges

Initial project scope (live capture from network) vs. actual product

Learning curve for Linux-based library Libpcap <pcap.h>

Knowledge about packet structure and timestamping
- UTC format vs. TAI format
- Packet timestamp vs. Aggregation timestamp
Evaluation

What went well

- Met functional and non-functional requirements
- Communicated effectively with the client and demonstrator
- Overcame any development issues such as platform, specs

How some things turned out

- An initial slow start due to project allocation delays
- Uncertainty with the project’s design early on, concerning the client not asking for user interface of any kind
Our program takes in a PCAP file as an input. In this example 'marketData.pcap' which contains multiple packets, is used.

The program then analyses each packet in the file, for a number of different errors, such as the time between the current and previous packet.

The analysis is both printed to console and exported in CSV format, this is to streamline the process of understanding the analysis.
Thanks for listening!

Any questions?