Performance Analysis of the ANGEL System for Automated Control of Game Traffic Prioritisation

Jason But, Thuy Nguyen, Lawrence Stewart, Nigel Williams, Grenville Armitage

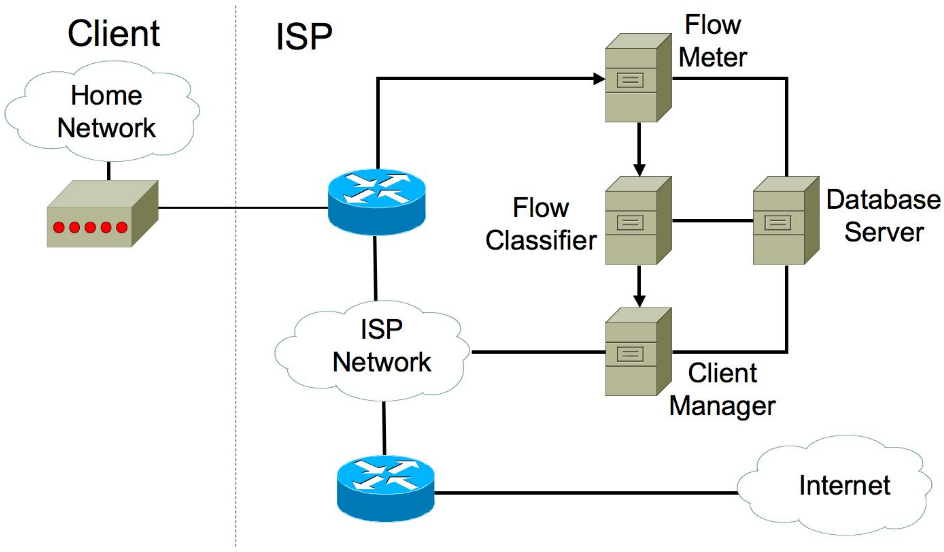
jbut@swin.edu.au

Centre for Advanced Internet Architectures (CAIA)
Swinburne University of Technology

Outline

- System Architecture
- System Implementation
- Classification Techniques
- System Performance
  - Accuracy of Classification
  - Timeliness of Classification
  - Processing Capability
- Demonstration Video
- Conclusions

ANGEL Architecture



Performance Issues

- Processing Performance
- Scalability
- Accuracy and Stability
- Timeliness

Scalability

- Built into the design
- Difficult to test without deployment

Classification Techniques

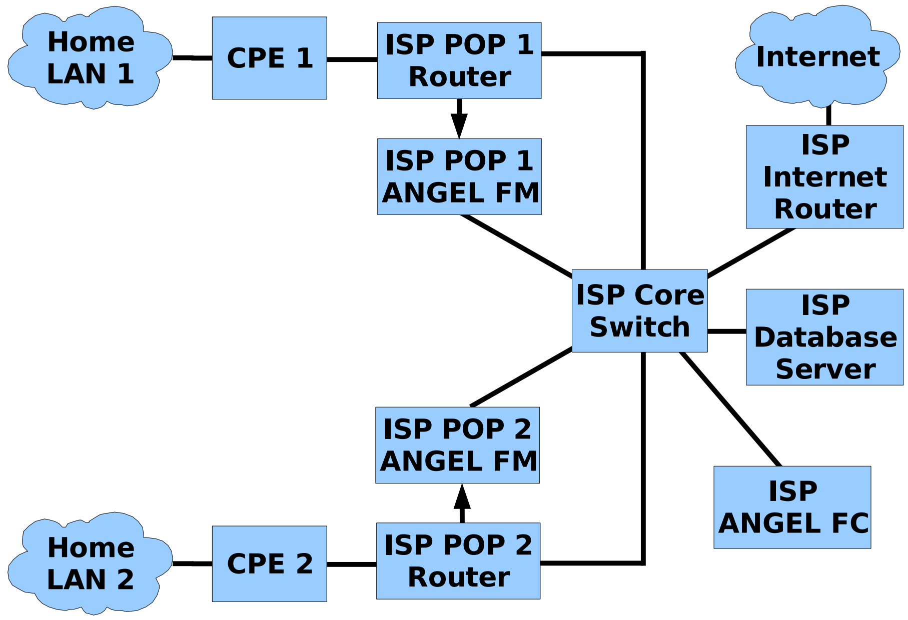
- Possible Approaches
  - Port based
  - Stateful reconstruction
  - Machine Learning algorithms
- ANGEL deliberately separates flow classification from prioritisation
- Extensible to support different traffic types

Classification – ANGEL

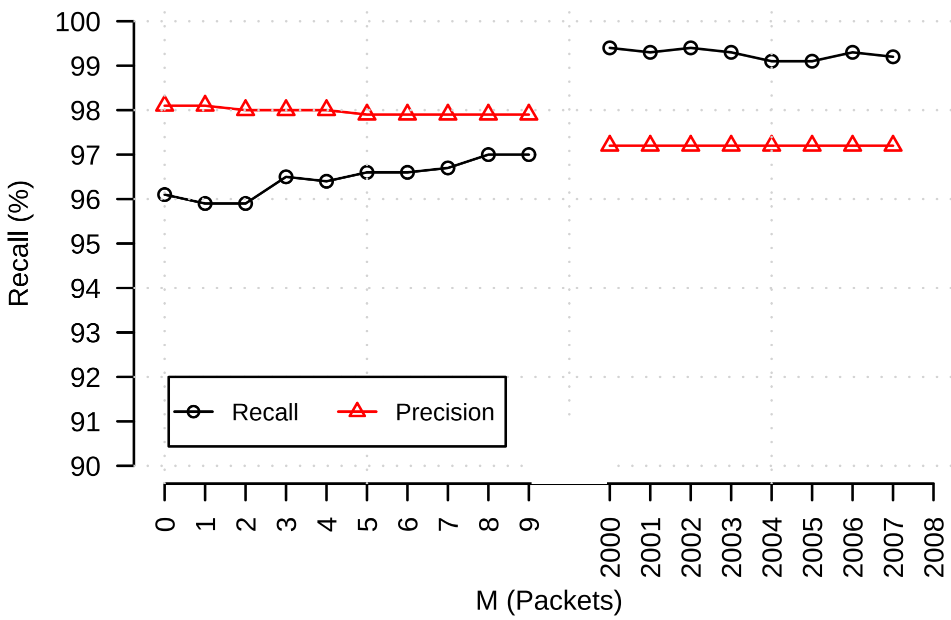
- Machine Learning based
- Naive Bayes Algorithm
  - Classification based on probabalistic knowledge
- Real-time classification
  - Must classify using a small portion of flow
  - Should continuously classify
  - We use a sliding window of 25 packets per unique flow
- Classification Model
  - Constructed as stated here[1]
  - Game Traffic – Wolfenstein Enemy Territory (ET) traffic of a month-long trace collected at a public server in Australia
  - Non Game Traffic – From a 24-hour trace collected by the University of Twente, Germany, at an aggregated 1Gbps link

[1] T. Nguyen and G. Armitage. *"Training on multiple sub-flows to optimise the use of machine learning classifiers in real-world IP networks"*. Proceedings of the IEEE 31st Conference on Local Computer Networks, Florida, USA, 2006

ANGEL Testbed



Performance – Accuracy



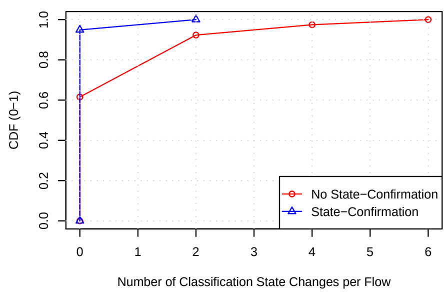
- Where *M* is the number of packets missed from the beginning of a flow

Performance – Stability

- Classification accuracy is relatively high
- Repeated classification on a 25 packet sliding window leads to fluctuating classifications for the duration of the flow
- This leads to:
  - Extra processing load as the client needs to re-deploy prioritisation rules
  - Extra network load as classification changes are communicated to ANGEL devices
  - Poor performance as game traffic may lose prioritisation for short periods of time
- To improve classification statibility we developed the *"Confirmed Classification"* algorithm
  - Essentially deploys a low-pass filter to the output of the classifier
  - Classification changed when two consecutive, non-overlapping windows of packets generate the new classification

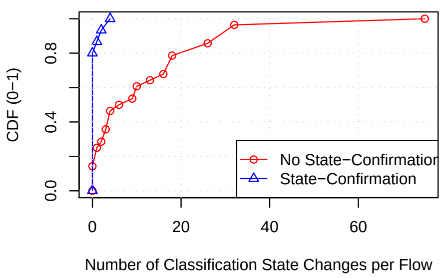
Performance – Stability

Stability for ET flows



- Flows exhibiting classification changes dropped from 15 to 1
- This flow only changed state once

Non-game (Kazaa) flows



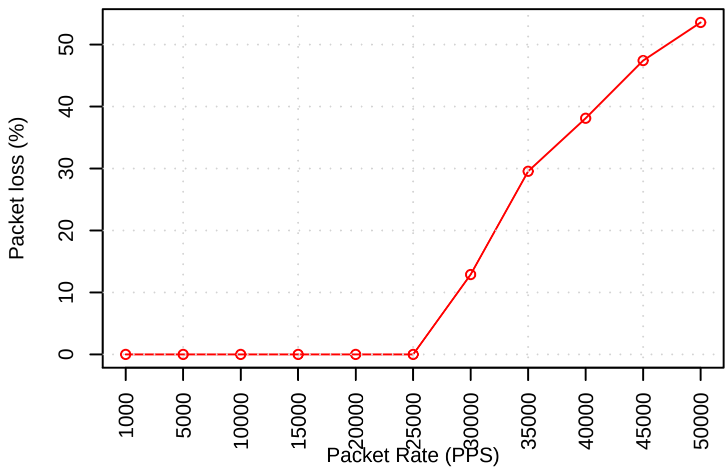
- Other traffic types tested - similar results
- Significant improvement in both the number of flows and the number of classification changes

Performance – Timeliness

- ANGEL – Initial classification for a new flow is non-game
- With the *"Confirmed Classification"* algorithm we need to capture two windows (50 packets) of a flow before it can be classified as game traffic
- Classification timeliness is dependent on (bi-directional) packet rate generated by the game
- Observations for ET show classification typically occurs between 0.5 and 1 second(s) after flow begins

Processing Performance – Flow Meter

- Captures packets and forwards statistics to Classifier
- Need to capture and process traffic with negligible loss



- Compares with performance of underlying capture library
  - Supported by memory (5MB) and CPU (30%) usage rates for all input packet rates

Processing Performance – Flow Classifier

- Tested under a worst case scenario - Single process classifying all flows
- Generated trace file consisting of multiple flows by duplicating and combining a source trace file
- Replayed trace file to a Flow Meter and then onto Classifier
  - Packet rate limited to 25,000pps - Flow Meter limit
  - Equivalent of 500 concurrent flows
- Classifier able to correctly classify all flows
- Memory footprint (< 5MB)
- CPU usage (< 0.2%)
- Suggests the bottleneck is the Flow Meter rather than the classifier

Demonstration

Conclusions

- We have built a working ANGEL System
  - Separate modules
  - Scalable - multiple Metering points
- Game traffic classified with >96% accuracy
- *"Confirmed Classification"* technique improves classification stability
- System bottleneck is the Flow Meter - limited by performance of underlying packet capture facility
- Machine Learning approach can scale to large numbers of flows
- User-perceived performance
  - Game flows typically classified and prioritisation rules established within 1 second
  - Successful classification when traffic captured after flow has started
- Modular system - can grow to support other traffic flow types