Bachelor Technical Project: Network Analysis Tool

Final Evaluation Presentation

Syamantak Das
Anshuman Choudhuri
Roshan John

Abstract

Network simulators are often used for research purposes and the tracedata they produce is invaluable to the work being done. However, this trace data is often verbose and difficult to understand at a glance for humans.

This tool provides the user with a more readable result after parsing the simulations' resultant tracedata. It also calculates end-to-end delay, number of dropped packets and other statistics.

Technologies

- ***Network Simulator 3:*** used for creating a sample network to generate trace data. The network consists of an Ethernet LAN and a Wireless LAN connected by a point-to-point link.
- ***tcpdump:*** used for interpreting PCAP trace files.
- ***Python:*** used for parsing the trace data and for the processing scripts that calculate the statistics.
- ***MySQL:*** used for database management.

Structure

- **Source Layer**
  NS-3 simulation script, produced PCAP trace files
- **Processing Layer**
  Parsing, Cleaning, Calculation and Database related scripts
- **Presentation Layer**
  Queryable database, generated graphs

Approach

- Created sample network in NS-3 consisting of Ethernet and Wireless LANs connected by a point-to-point link.
- Used TCP Bulk Send and Packet Sink applications to simulate packet transfer on the network.
- Used an error model to simulate packet drops.
- Used a mobility model to simulate movement of wireless nodes.
- Generated ASCII trace data from the simulation

Approach

Problem: ASCII trace data had different format depending on link type (point-to-point/wireless/ethernet), and this led to the need for specific parsers for each type of network. However, this was too specific to network type and did not fit with our goal of generalizing parsing of trace data.

Solution: Switched to NS-3's functionality to produce PCAP trace files. These files once analyzed by tcpdump, produce a more uniform output that can be generally parsed without considering the network type.

Approach

- Wrote a Python script for parsing tcpdump outputs into a python data structure.
- Wrote a Python script to resolve each individual IP DATA packet.
- Wrote a Python script to calculate statistics.
- Wrote a Python script to generate database model.
- Created database using model and populated with all trace data.

Approach

Problem: PCAP trace file is generated at each node. Depending on the network-type, each node may or may not contain traces for every packet passing through the network. How to identify which node is the sender/receiver?

Solution: Based on the network being simulated, a list of key-value pairs of [node-name -> IP address] must be provided. Using this list we are able to identify the source and destination nodes for each packet.

Statistics

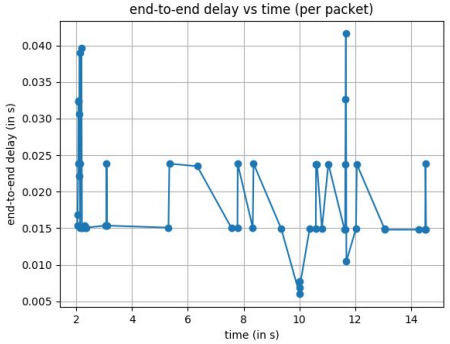
- throughput = *bytes sent / time taken*
- packet loss = *packets sent - packets received*
- jitter = *(|(delay of i+1) - (delay of i)|) / packets received*
- end-to-end delay = *receive time - send time*

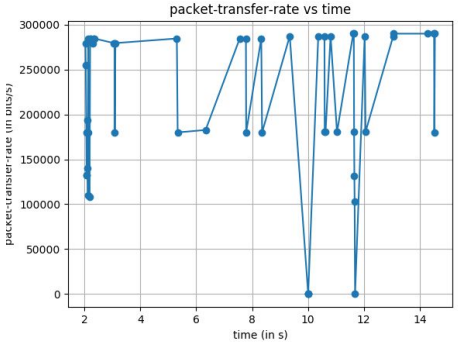
Result

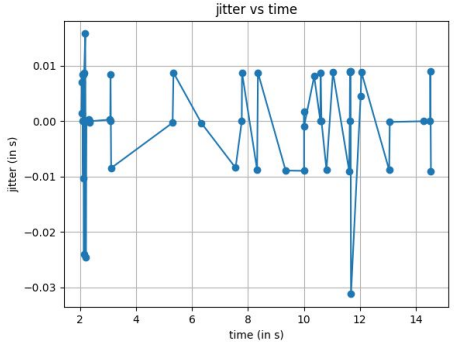
- A queryable MySQL database created for trace files from a NS-3 simulation. Contains data regarding each IP DATA packet, including source, destination, sent time, end-to-end delay, sequence number and whether the packet was dropped.
- Graphs for analysis.

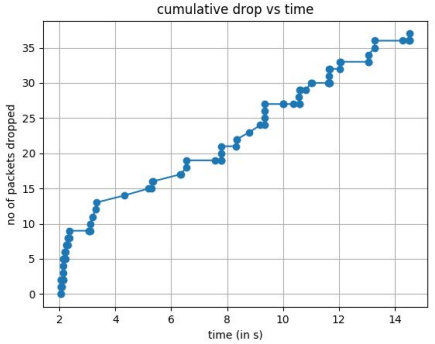
Result

Graphs are generated for the entire network, or on user prompt, for specific source-destination pairs.









Future Scope

- Apart from TCP/IP DATA packets, further, UDP and control packets can be integrated into the analysis tool.
- Handling very large trace data dumps. Will involve data storage using files instead of Python data structures.

THANK YOU

End of Presentation