European Collaborative Innovation Centres for Broadband Media Service

KARLSTADS UNIVERSITET

Network Emulation for media broadband services
Case Study Final Report

EUROPEAN UNION
European Regional
Development Fund

Investing in the future by working together for a sustainable and competitive region

The Interreg IVB
North Sea Region
Programme

1 Introduction

Network emulation is commonly used to evaluate and examine the behavior and performance of applications and transport layer protocols. The major advantage of emulation, as compared to network simulation, is that real implementations of the applications/protocols under study can be evaluated. Compared to live network experiments, the benefit is the high degree of control over the emulated network characteristics, e.g. delays, packet loss etc. What makes KauNet [4] special, compared to other network emulators, is its ability to apply emulation effects deterministically. That is, using KauNet it is possible to specify exactly which packets to apply a certain emulation effect on. Thus, KauNet is able to provide fully repeatable emulations at a very high level of control.

This report describes the KauNet scenario functionality. KauNet scenarios are essentially files that can describe all aspects of an emulated scenario. For instance, it is possible to create a scenario with packet loss, bit-errors, reordering, bandwidth and delay changes, and evaluate how a network application/protocol would behave under the specific circumstances. Each of the emulated effects can also be applied to specific packets or at specific points in time. As scenarios are stored offline, in files, it is possible to exactly replicate experiments.

In addition to describing KauNet and the scenario functionality, this report provides an example of how to create scenarios from measurements conducted in real Internet access networks. The resulting scenarios can then be reused to evaluate the performance and/or behavior of applications/protocols running in those specific networks. The networks modeled in this report use five access technologies: backbone, wireless LAN (WLAN), wireless mesh network (WMN), asymmetric digital subscriber line (ADSL) and third generation mobile telecommunications (3G). The different networks will be described more closely later in the report.

The contributions of this report can be summarized as follows:

1. a description of the KauNet network emulation system is provided, with focus on KauNet scenarios;
2. measurements of network characteristics in five different Internet access networks are presented;
3. and it is shown how to model different networks with KauNet scenarios, using data from network measurements.

The remainder of this report is structured as follows. Section 2 gives the necessary background on KauNet and KauNet scenarios. Section 3 shortly describes the access networks that are modeled in this report. Section 4 describes the access network measurements and the results from them. Section 5 shows how the results were used to create the different scenarios. Finally, Section 6 concludes this report with a summary and some final remarks.

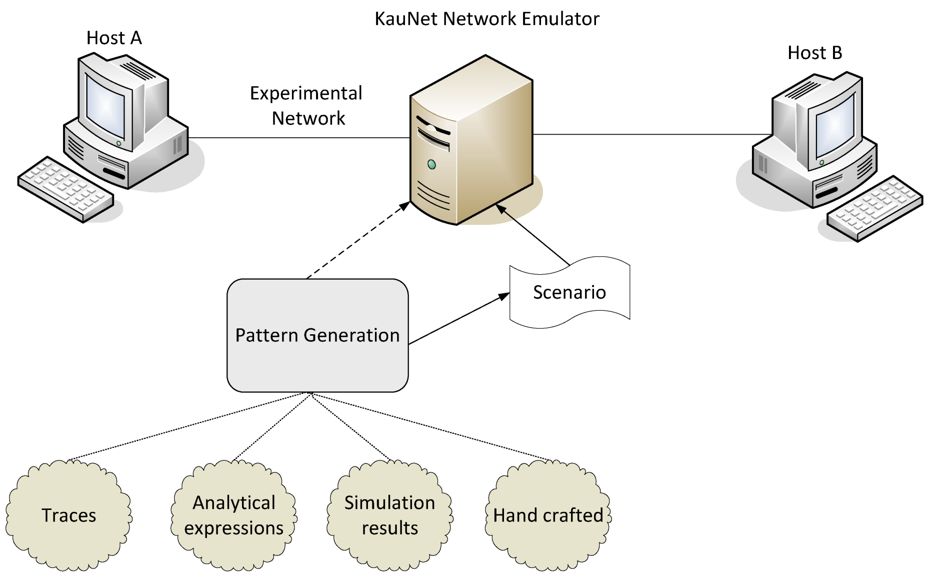

Figure 1: KauNet emulation system.

2 KauNet

This section describes the KauNet emulation system. First, the basics of KauNet are covered. Then, KauNet patterns are described. KauNet patterns are the basic building blocks of KauNet scenarios. Finally, the KauNet scenario functionality is detailed.

2.1 Overview

The design of KauNet is centered around a number of pattern-handling extensions to Dummynet [2], together with user-space programs for pattern creation and management. This allows KauNet to control all emulation effects on a per-packet or per-millisecond basis. Furthermore, it is also possible to create and use emulation scenarios. As described in the introduction, a KauNet scenario is essentially a file that includes a number of emulation effects (patterns), and can thus be used to emulate multiple effects simultaneously.

KauNet is flexible with regards to the origin of the patterns, which can be created from many different sources including collected traces, analytical expressions, simulations, or be hand-crafted. A schematic overview of KauNet is shown in Figure 1. To apply the emulation effects a user loads one or more pattern(s), or scenarios, into KauNet and selects the traffic that should be affected. For example, Figure 2 shows a user configuring KauNet to drop packets number $2, 4$ and 6 in a data flow going from host `10.0.2.1` to host `10.0.1.1`. The configuration is made possible by the use of a packet loss pattern.

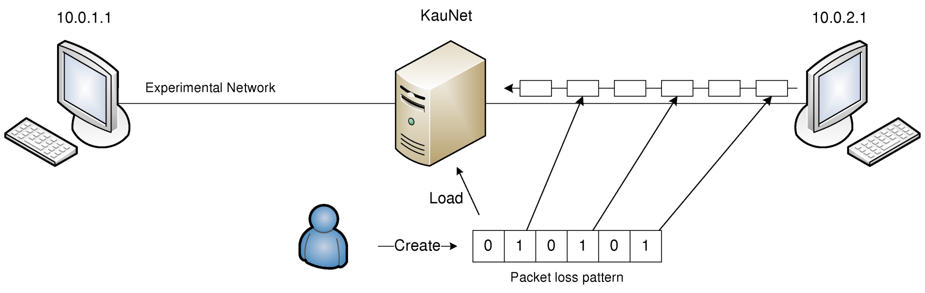

Figure 2: KauNet emulation system.

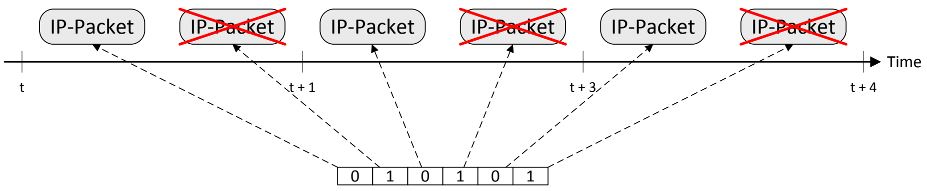

Figure 3: Data-driven packet loss pattern.

2.2 Patterns

As described in the earlier section, emulation effects are stored in patterns. It was shown that emulation effects could be applied deterministically, causing specific packets to be e.g. lost. All supported emulation effects can be applied in this data-driven manner. It is also possible to apply emulation effects in a time-driven fashion, where the pattern specifies during which milliseconds of the emulation a certain effect should be applied.

Thus, patterns can be in one of two modes: data-driven or time-driven. Figure 3 illustrates the data-driven packet loss pattern that was exemplified earlier. The pattern has one value for each incoming packet. If the current value is zero no packet loss occurs for the corresponding packet. If the value is one, on the other hand, the packet is dropped. Figure 4 shows a time-driven packet loss pattern. Using this pattern, all packets arriving at the emulator when the pattern indicates packet loss will be dropped. For this particular pattern, packets arriving at millisecond two of the emulation will be dropped.

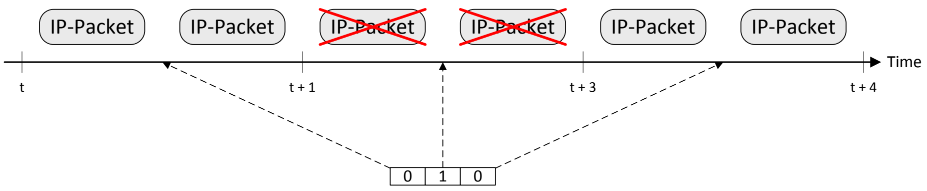

Figure 4: Time-driven packet loss pattern.

Pattern	Binary	Value
Packet loss	X	-
Bit-error	X	-
Bandwidth change	-	X
Delay change	-	X
Packet reordering	-	X
Trigger	-	X

Table 1: KauNet patterns.

As previously mentioned patterns support all emulation effects, not only packet loss. It is also possible to e.g. specify an emulated delay for each packet. Such emulation effects are not binary, as packet loss (loss/no loss). Two kinds of patterns exist: binary patterns and value patterns. The value patterns do not only indicate that a certain effect should be applied, but also provide the necessary data for that effect. Table 1 provides an overview of the different types of patterns that are supported by KauNet.

2.3 Scenarios

A KauNet scenario is a collection of patterns. As a scenario supports each possible emulation effect, the maximum number of patterns in a scenario is six. Figure 5 illustrates an example scenario including a bit-error pattern, a packet loss pattern, a delay change pattern and a bandwidth change pattern. As indicated by the figure, all patterns are in the data-driven mode. Furthermore, it is possible to see how the delay and bandwidth changes during the emulated scenario, and how bursts of bit-errors and packet losses are present. By issuing only two command line arguments at the KauNet network emulator it is possible to load this scenario and select the traffic to be affected.

When using scenarios it is important to consider the order that KauNet applies emulation effects. If KauNet, for instance, is instructed to both drop and delay a packet, only the packet drop will performed as packet drops happens before packet delaying. Emulation effects are applied in the following order: (i) packet loss; (ii) bandwidth changes; (iii) delay changes; (iv) packet reordering; and (v) bit-errors.

3 Access Networks

The number of ways to connect to the Internet has grown significantly over the last years. Not many years ago a regular user could only access the Internet using very slow dial-up modems. Nowadays, Internet access is almost ubiquitous. A number of technologies are available, allowing users to connect using a variety of devices. As different networking technologies provide different network characteristics, the performance and behavior of network applications/protocols can

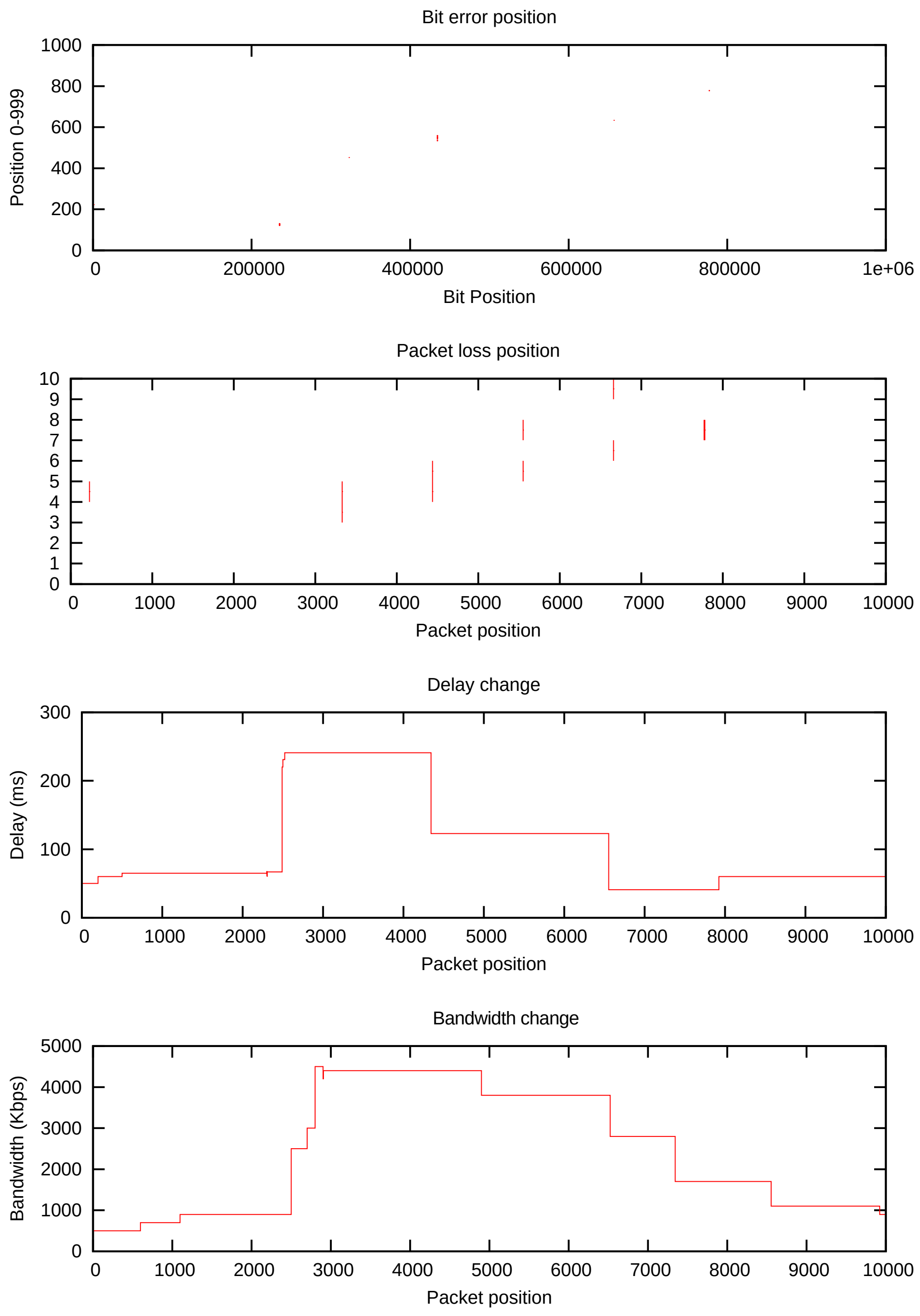

Figure 5: KauNet example scenario.

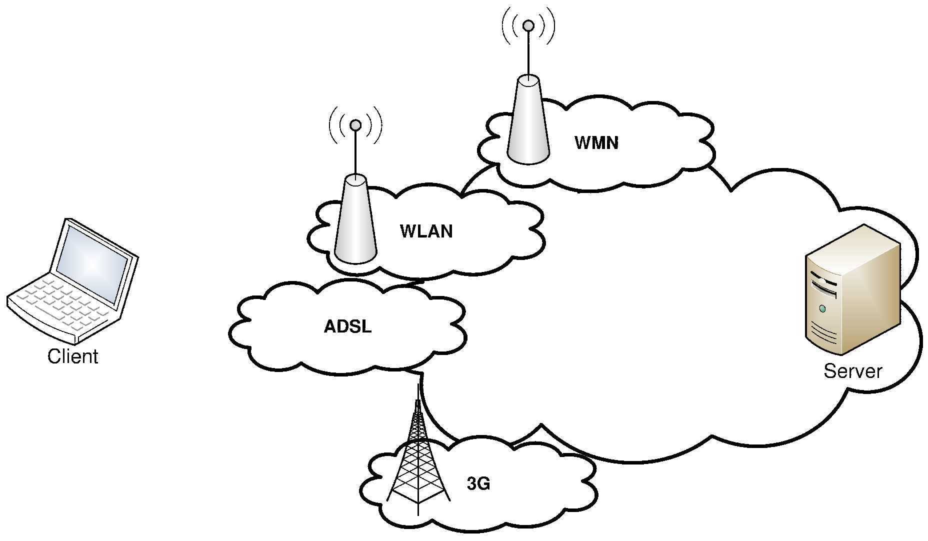

Figure 6: Access networks.

vary significantly. For instance, a streaming video application that works well using a computer with an ADSL connection might not perform well, or function at all, using a mobile device equipped with a 3G interface. A 3G interface usually provides significantly higher latencies and lower bandwidth than ADSL modems. It is therefore important for developers and researchers to be able to evaluate applications and protocols in different environments.

As previously mentioned, the aim of this report is to show how to create and use KauNet emulation scenarios to enable the evaluation of applications over different types of networks. To do this, we have created five example scenarios modeling the following access technologies: backbone, WLAN, WMN, ADSL and 3G. The scenario creation is based on real network characteristics that we have measured in these access networks. The measurements were conducted between a measurement server located in the Swedish university network (SUNET) and a measurement client that used the different access technologies to communicate with the server. Figure 6 depicts the environment.

To create the backbone scenario, we conducted measurements where both the client and the server were located in the SUNET network. For creation of the WLAN scenario we conducted measurements in various WLANs across Sweden, both public and private ones. The WMN measurements were conducted in a 2-hop static configuration inside the KAUMesh network [3]. For readers unfamiliar with WMN technology we refer to [1]. The ADSL scenario was constructed using characteristics from measurements in three Swedish Internet service providers' ADSL networks. Finally, the 3G scenario was created from measurements in various locations in Sweden.

4 Access Network Measurements

The access networks, which were shortly described in the previous section, represent a mix of widely used networking technologies (e.g. ADSL) and emerging technologies (e.g. WMN). The coming subsection shortly describes measurement considerations. That is, which characteristics can later be used in the KauNet scenario creation and how to measure them? The following subsection then describes how the actual measurements were conducted. Finally, the results from the measurements are reported. The actual scenario creation, based on the results, is described in Section 5.

4.1 Measurement Considerations

The overall requirement of the scenarios is that they become representative for the corresponding access network, not that they are exact models. It is unfeasible, if possible at all, to exactly model a network. Thus, the aim is to capture a representative behavior for each access technology, in terms of network characteristics.

When conducting the measurements it is important to consider the restrictions that follows from using the KauNet framework. First, KauNet is an IP-level emulator. Thus, it is impossible to emulate effects other than those at the IP-layer. For instance, possible link-layer retransmissions in a WLAN network can not be modeled directly. However, such link-layer retransmissions are likely to result in delay jitter, which is possible to model. As previously mentioned, we are restricted to five different emulation effects: (i) bit-errors; (ii) packet loss; (iii) bandwidth changes; (iv) delay changes; and (v) packet reordering; Bit-errors are unlikely to appear at IP-level, as most link-layer schemes will retransmit corrupted frames. We therefore only considered effects (ii)-(v) in our measurements.

As previously mentioned, KauNet is able to emulate effects in a data-driven or a time-driven manner. When exactly evaluating a protocol or an application, it is sometimes useful to do it using data-driven emulation. However, the data-driven mode will produce results that are highly dependent on the underlying traffic pattern, as the traffic effectively drives the emulation. Thus, if various applications and protocols should be evaluated using the same scenario, it is better to use the time-driven mode. In time-driven mode, KauNet is able to emulate effects with a granularity of 1 ms. Although variations might appear on a millisecond basis, this has little effect on most applications/protocols. Furthermore, to be able to measure characteristics like bandwidth it is often necessary to have a granularity that is coarser than the delay of the underlying network path. This is at least true for active network measurements, which we have employed. To make sure that we are able to correctly estimate e.g. the bandwidth we chose a granularity of 1 s.

To be able to capture both start up and steady state effects, for applications using e.g. TCP, it is necessary that the scenarios does not become too short. We chose measurement periods of 5 minutes. This amount of time is likely

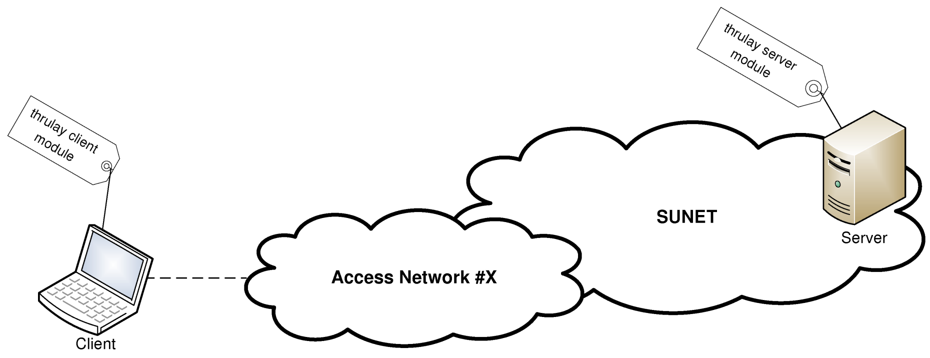

Figure 7: Measurement setup.

to meet our requirements, while it made it possible for us to conduct a lot of experiments in a relatively short period of time.

4.2 Access Network Measurements

There are different ways of measuring network characteristics. The two most common techniques are active measurements and passive measurements. Naturally, there exist a lot of different techniques in between, but essentially these two forms the basis for network measurements. Passive measurements are attractive as they do not affect the network traffic during the measurements. Unfortunately, it is often hard to conduct passive measurements, as access to the network infrastructure might be needed.

Active measurements are often more easy to conduct, but can be misleading, as the test traffic itself affects the results of the measurement. For instance, TCP senders try to find the maximum available bandwidth by flooding intermediate routers. It is therefore not a good idea to use a TCP flow to determine the average packet loss rate, as losses will occur due to TCP's mode of operation. However, TCP flows can be useful in measuring other characteristics. For example, as TCP tries to estimate the maximum available bandwidth with its slow-start phase, TCP can be used to measure the available bandwidth.

In this report we use active measurements. Mainly because it was impossible to gain access to the network infrastructure of all the modeled access networks. The network capacity tester `thrulay` [6] was used to measure the required metrics for the KauNet scenario creation. `thrulay` uses both TCP and UDP traffic for network measurements. TCP is used to calculate the maximum available bandwidth and the delay, while UDP is used to measure packet loss rates and packet reordering. Furthermore, `thrulay` consists of two modules: one server module and one client module. It is possible to configure these modules to run for a specified period of time, reporting statistics in given intervals. Thus, it is straightforward to collect the relevant metrics for the required granularity and time-periods.

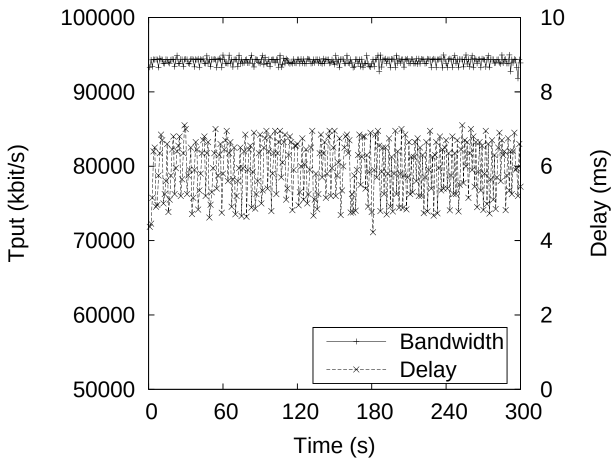

Figure 8: SUNET Measurement.

Figure 7 shows the simple measurement setup. Basically, the client connects to the access network under study, using the corresponding networking technique, and runs the `thrulay` client module to connect to the server. The server, which is located in SUNET, runs the `thrulay` server module which collects metrics and reports them back to the client every second for five minutes. The metrics that are reported back to the client are those that we previously mentioned: bandwidth, delay, packet loss and reordering.

The measurements were conducted during one week in the spring of 2010 and also in the beginning of 2011. To account for possible variations due temporal effects like e.g. traffic load, measurements were conducted at random points in time. In total, 84 measurements were conducted. In the coming subsection, we present one representative measurement for each access technology.

4.3 Measurement Results

Before detailing each access technology, a couple of general remarks should be done. First of all, the amount of packet loss present in our measurements were very low and did not seem to appear in any particular patterns. This was also the case for packet reordering, which was virtually not detected at all. We will therefore only report the measured bandwidths and delays.

The y-axis of Figure 8 shows the bandwidth (left y-axis) and the delay (right y-axis) of a representative backbone measurement. Both the bandwidths and the delays are plotted as a function of time in seconds. The graph shows the entire five minute measurement. As indicated by the graph, the results for the backbone measurements were very stable. That is, both the bandwidth and the delay varied only little. This was expected, and is due to the large amount of capacity in the network and the fact that it is a wired connection. The

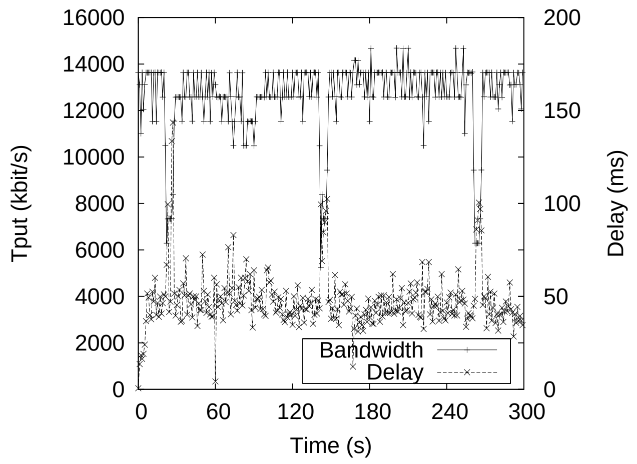

Figure 9: WLAN Measurement.

bandwidth was stable between 90 to 100 Mbit/s, the latter being the maximum supported bandwidth of the measurement machine. The delay was also very stable, and varied only a little around 6 ms.

Figure 9 shows the results from a WLAN measurement. The layout of the graph is the same as in the previous graph. That is, bandwidths and delays are plotted over the entire five minute interval of a measurement. In this graph we can easily note that the bandwidth varies slightly. It is also noticable that a number of significant bandwidth decreases coincide with delay spikes.

In Figure 10 a WMN measurement is shown. Contrary to the previous measurements, we had to increase the reporting intervals of our measurement tool for these experiments. Instead of a granularity of 1 s we were forced to use a 5 s granularity. This was required in order to make correct estimations of the bandwidth. As shown in the graph the bandwidth varied between, approximately, 2 and 4.5 Mbit/s. The delay was rather stable, around 40 ms. There were no competing traffic, and we did only make two hops into the WMN.

Figure 11 shows an ADSL measurement. The ADSL measurements seem to have two distinctive bandwidth modes. In this particular measurement the modes were located at approximately 1.6 Mbit/s and 2.1 Mbit/s. It is not entirely clear why the modes are so distinct. This could be due to an unnecessarily large buffer in the ADSL modem. The delay in this measurement was rather stable, around $10 - 15$ ms.

Figure 12 shows a 3G measurement. Similar to the WMN measurement, we had to increase the reporting interval of the measurement tool to get correct estimates of the bandwidth. As for the WMN, we increased the interval to 5 s here as well. For these experiments, the bandwidth was rather stable around 350 Kbit/s, while the delay varied between approximately 200 ms and 1.3 s.

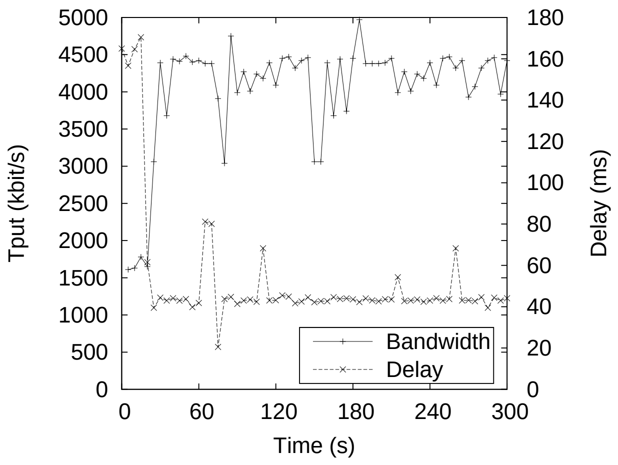

Figure 10: WMN Measurement.

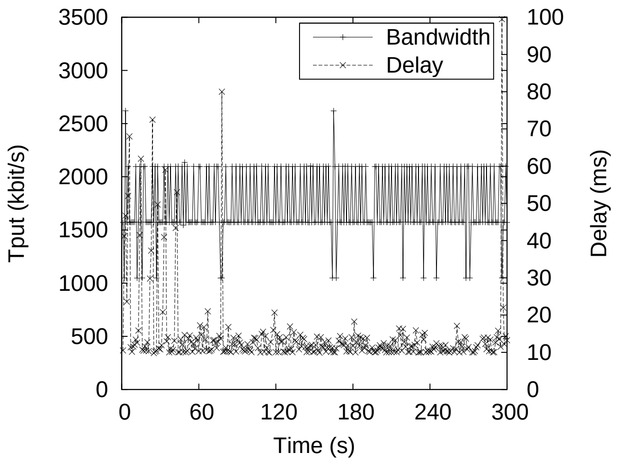

Figure 11: ADSL Measurement.

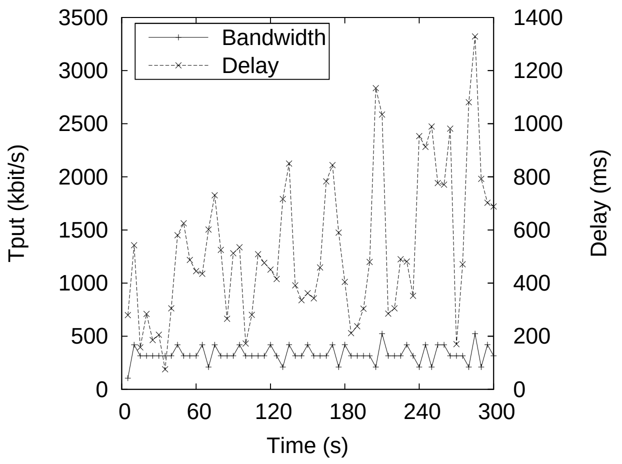

Figure 12: 3G Measurement.

5 Scenario Construction

Creating scenarios from the measurements that we presented in the previous section is actually a rather easy thing to do. The `patt_gen` tool that ships with KauNet is very helpful in this process. The first step when creating a scenario is to create patterns of the emulation effects that are going to be included in the scenario. There are a number of ways to create patterns using `patt_gen`, including using built in distributions. However, as we have conducted measurements we need to "load" the results into patterns. The easiest way to accomplish that is to use `patt_gen`'s file reading feature. This feature enables an experimenter to transform measurement data directly into patterns.

Let us exemplify by showing how the WLAN bandwidth results were transformed into a time-drive bandwidth pattern. Shown below is a subset of the recorded values for one of the WLAN measurements. These values were filtered directly from the `thrulay` output, and thus only shows a subset of what this tool reports.

```
1.00 1.048
2.00 1.049
3.00 1.048
4.00 1.049
5.00 1.049
...
```

The left column shows the corresponding second of the experiment, and the right column shows the bandwidth measured during that second (in Mbit/s). The `patt_gen` tool is actually able to parse this output almost as is, and create a bandwidth pattern of it. Some formatting is, however, required. Firstly, as KauNet supports a granularity of 1 ms, times must be specified in milliseconds. Secondly, KauNet currently specifies all bandwidths in kbit/s. Thus by formatting the output to:

```
1000 1048
2000 1049
3000 1048
4000 1049
5000 1049
...
```

and saving it to a file (`input_file.bw`), it is possible to create a time-driven bandwidth pattern by issuing the following command:

```
user@computer:~$ ./patt_gen -bw -pos wlan_bw.pattern time 300000
                 -f input_file.bw
```

The `-bw` flag tells `patt_gen` that a bandwidth pattern should be created. Furthermore, the `-pos` flag indicates that the bandwidth changes should occur at specific positions, in this case in time. Next, the filename of the resulting pattern is given. Then, `time` is used to create a time-driven pattern. The `time` directive is then followed by the desired length of the pattern, which in this case is 300000 ms which is equal to 5 minutes. Finally, the `-f` flag tells `patt_gen` to take its input from the file `input_file.bw`. That is all the required syntax to create a time-driven bandwidth pattern based on existing results.

The syntax for creating a delay pattern is similar. However, instead of using the `-bw` flag, the `-del` flag should be used. That is actually the only difference. It is worth remembering that delays should be specified in milliseconds. Thus, the input to the delay pattern should be formatted as:

```
1000 18
2000 19
3000 17
4000 19
5000 16
...
```

The last step of the process is to combine these patterns into a scenario. This is done using `patt_gen`'s `-mkscn` flag. The below syntax will create a scenario containing both the bandwidth pattern and the delay pattern constructed earlier, given that the delay pattern was saved in a file called `wlan_del.pattern`:

```
user@computer:~$ ./patt_gen -mkscn wlan.scenario -t WLAN
                 wlan_bw.patt wlan_del.patt
```

The `-t` flag makes it possible to give the scenario a name, and this is simply followed by the patterns that should be grouped into the scenario.

The few steps above was used to create all of our scenarios. If packet loss and reordering effects are desired it is easy to add such effects to the existing scenarios. The scenarios that were created during the work with this report, corresponding to the results in the previous section, can be found at the KauNet site [5]. This site also hosts the KauNet emulator together with its documentation.

6 Summary

This report has described KauNet's scenario functionality. Furthermore, it has presented network characteristics measured for five different Internet access network technologies, and described how to create emulation scenarios from these measurements. By following the simple methodology in this report it is possible for experimenters to create their own scenarios.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.

[2] M. Carbone and L. Rizzo. Dummynet revisited. *SIGCOMM Comput. Commun. Rev.*, 40(2):12–20, 2010.

[3] P. Dely. Kaumesh. `http://www.kau.se/en/kaumesh`.

[4] J. Garcia, E. Conchon, T. Pérennou, and A. Brunstrom. KauNet: improving reproducibility for wireless and mobile research. In *MobiEval '07: Proceedings of the 1st international workshop on System evaluation for mobile platforms*, pages 21–26, San Juan, Puerto Rico, 2007.

[5] P. Hurtig. Kaunet. `http://www.kau.se/kaunet`.

[6] S. Shalunov. Thrulay, network capacity tester. `http://sourceforge.net/projects/thrulay/`.