Analysis of a Billing Trace of a CDPD Network

Kara McNair

91201-1352

CMPT885/ENSC 833 High-Performance Networks

April 14, 2001

Abstract

In this paper, I analyze a trace of billing data from Telus Corporation's CDPD (Cellular Digital Packet Data) network. The trace has a duration of almost three weeks (20 days) between December 22, 2000 and January 11, 2001. Data extraction was performed in a variety of different ways including using some custom software and an SQL database. A total of 60 unique cell IDs were found in the trace. There were 2096 unique Mobile NEIs (user IDs). Of these 2096 users, only 1730 of them actually generated data events. There was a total of 1580236 events in the trace and the number of deregistration events is an order of magnitude smaller than either the numbers of registration or data events. An attempt was made to discover the network topology by identifying 'neighbouring' cells, but the algorithm used was too generous in classifying cells as neighbouring.

Network activity displays periodicity on two scales - one is clearly related to the time of day and the other may be a weekly cycle. The actual trace may not be representative of typical network activity because it spans the holiday season.

It is noticed that a few users dominate the generation of events in the trace. Some of these may be users attempting to gain illicit access to the network. Some users' traffic appears to be unfairly discarded and others unfairly retained.

In order to map the network and to accurately identify patterns of network activity and user behaviour, a longer trace is needed and a more optimized database design required.

1) Introduction

In this paper, I analyze a trace of billing data from Telus Corporation's CDPD (Cellular Digital Packet Data) network. The trace itself is not a CDPD trace, but instead summarizes the network activity in ways which allows Telus to bill its customers for usage. The purpose of this analysis is to characterize the network and the user behaviour. This characterization will focus on four main areas – static network information (questions of 'how many'), dynamic network characteristics (questions of 'when'), user behaviour and cell activity.

2) Description of the trace

The trace consists of three different kinds of events [1]. There are registration events, deregistration events and data events. The registration events occur when a user's mobile end system (generally an IP-enabled cellular phone) attempts to identify itself to the network to gain service. Deregistration events can occur when a user leaves a cell, but may not. They are optional events and may or may not be recorded. Data events describe the actual network traffic generated by users. They include a count of how many packets and octets are transferred as well as how many packets are dropped as a part of that event.

The trace has been generously provided by Telus. In order to protect user privacy, the Mobile NEIs (Mobile Network Entity Identifiers – hereinafter used interchangeably with 'user ID') have been scrambled. However, they are scrambled in a consistent way – all events generated by user X have been assigned to user X' in the provided trace – so it is possible to perform user behaviour analysis.

The structure of the trace is a set of directories containing files. Each directory is named for a sequence number (e.g. 528/, 529/) and contains up to 100 files of data, also named sequentially (e.g. 00074-52800, 00074-52801, 00074-52802). Each file contains a record of approximately 15 minutes network activity. A file consists of a header (which contains timestamp and sequencing information) and a series of 'Traffic Matrix Segment' (TMS) rows. Each row represents a single event of type registration, deregistration or data. A total of 1897 files were provided as part of this trace.

The trace starts at 11:30 am on December 22, 2001 and goes until 6:30 am on January 11, 2001 – a duration of almost three weeks (20 days). There is a hole in the trace on January 2, 2001 from 7:30 PM until 10:30 PM. The directory 536/ contains only 87 files instead of the standard 100.

Data extraction was performed in a variety of different ways. A certain amount of analysis was performed using standard Unix tools (grep, wc, sort) but this was inadequate for any significant work. I then wrote a Java[tm] [2] program to parse the trace files and report data about them (total number of unique users, total number of dropped packets, etc), but for each new statistic I wanted to collect, a new algorithm had to be written. Since I was interested in looking at the data from a variety of different perspectives and didn't want to be constantly writing new code, I modified my program to instead write out the data as a series of SQL INSERT() statements and loaded the trace into a MySQL database.[1]

3) Static Network Characteristics

3.1) Network elements

A total of 60 unique cell IDs were found in the trace. There were 2096 unique Mobile NEIs (user IDs). Of these 2096 users, only 1730 of them actually generated data events. It should be noted that it is not known whether the trace actually covers the entire Telus CDPD network nor whether all network elements were active during the duration of the trace – these numbers represent minimum values.

Element	*Number of unique occurrences*
Cells	60
MobileNEI (all)	2096
MobileNEI (sending data)	1730

Table 1: Some network statistics

3.2) Network events

Table 2 shows the breakdown of the events in the trace by type.

[1] The database table creation script is included in Appendix A of this report. It is not fully optimized or in third-normal form; simplicity was more important than efficiency.

Event Type	*Number of events*	*Proportion of total events*
Registration	619268	39.19%
Deregistration	71741	4.54%
Data	889227	56.27%
Total	1580236	100%

Table 2: Breakdown of events by type

The number of deregistration events is an order of magnitude smaller than either the numbers of registration or data events. This is most likely accountable to two things: first, the deregistration events are optional (as mentioned before), and second, because registration events may fail. If a mobile end-system (phone) attempts to gain access to the system but fails, it will be generating registration events, but no deregistration events.

Another aspect of network behaviour that is interesting to look at is the overall 'problems' in the network – what proportion of registrations are rejected; what proportion of packets are dropped (see Tables 3 and 4). In later sections, it will be shown that the high proportion of rejected registrations does not really represent any failure on the part of the network. The proportion of dropped packets is actually fairly low and would be lower if the control packets were included in the calculation[2].

Number of registrations	*Number rejected*	*Proportion rejected*
619268	166525	26.89%

Table 3: Proportion of rejected registrations

Data packets	*Dropped packets*	*Proportion dropped*
1984159	30966	1.56%

Table 4: Proportion of dropped packets

3.3) Network topology

An attempt was made to discover the network topology by identifying 'neighbouring' cells. The algorithm used was as follows:

```
Alg1) If the same MobileNEI (user) generates data events in multiple cells
      within a single file (15 minute Traffic Matrix Segment), then those
      cells are all deemed to be neighbouring. (If a user visits cells X, Y
      & Z, the graph edges (X, Y), (X, Z), (Y, Z) are added to the network
      topology.)
```

This is a very generous algorithm – a user travelling in a car can pass through a lot of cells in 15 minutes. It would be better to limit the timeslices to something smaller, but this is not possible with this algorithm because data events do not carry a timestamp and the order of the events in a single TMS file is not guaranteed to be correct.

In fact, the algorithm generated upwards of 30 neighbours per cell – clearly not useful in determining cell topology. A 'standard' honeycomb arrangement of cells should yield no more than about 6 neighbours per cell. A more miserly algorithm could look at registration and deregistration events (which do carry timestamps) in the following way:

```
Alg2) For all Registration and Deregistration event pairs
      If their timestamps are the same (or within a few seconds) and their
      userIDs are the same and their cellIDs are different, then the cells
      are neighbours.
```

This algorithm would attempt to capture the handoffs where a Mobile NEI deregisters in one cell and registers in the next as the user crosses the boundary between them. It might be hampered by the lack of

[2] The choice to compare data packets vs. dropped packets as opposed to total number of packets (control + data) is somewhat arbitrary but is based on the premise that the control packets belong to the network and don't affect the user's perspective of the network's performance.

deregistration events though. And in fact, all we're trying to do is identify user location and movement – the following algorithm should suffice:

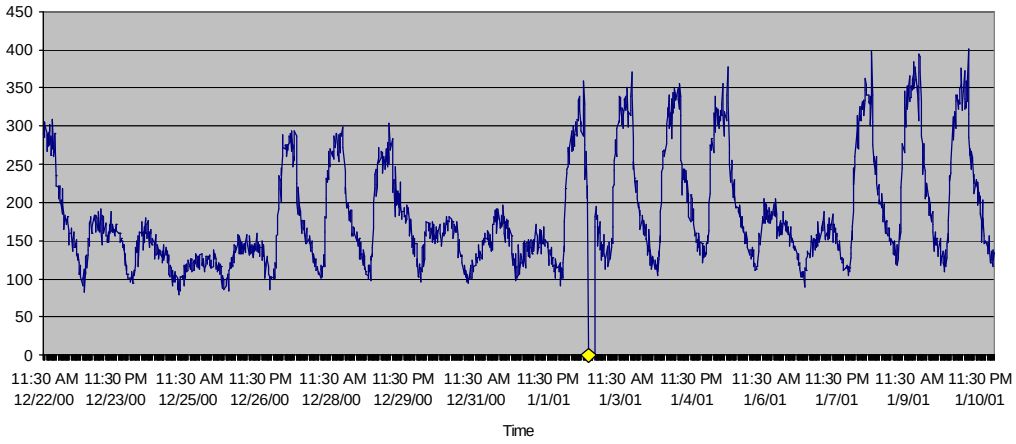
```
Alg3) Sort all events by user and then by timestamp.
      For each list of user events, if the timestamp of event(I) is within N
      seconds of the timestamp of event(I + 1) and the cell IDs differ, then
      the cells are neighbours.
```

This algorithm can be tuned by changing the value of 'N' until the average number of neighbours by cell is a reasonable value. These are suggested as future work.

4) Dynamic Network Characteristics

The first aspect of the network's dynamic behaviour that was investigated was the number of users active over time. Figure 1 shows this series (the hole in the trace is marked with a diamond)[3]. Two levels of periodicity are observable. The first is a cycling between peaks and valleys – these correspond to the 24 hours in a day. In general, the peaks occur at around 5:00 PM and the valleys at 5:30 am. The second cycle is slightly irregular, but clearly visible. There are two ranges of peaks visible. The higher ones tend to fall on weekdays and the smaller ones on weekends. It is possible that the irregularity is due to the

Figure 1: Number of users over time

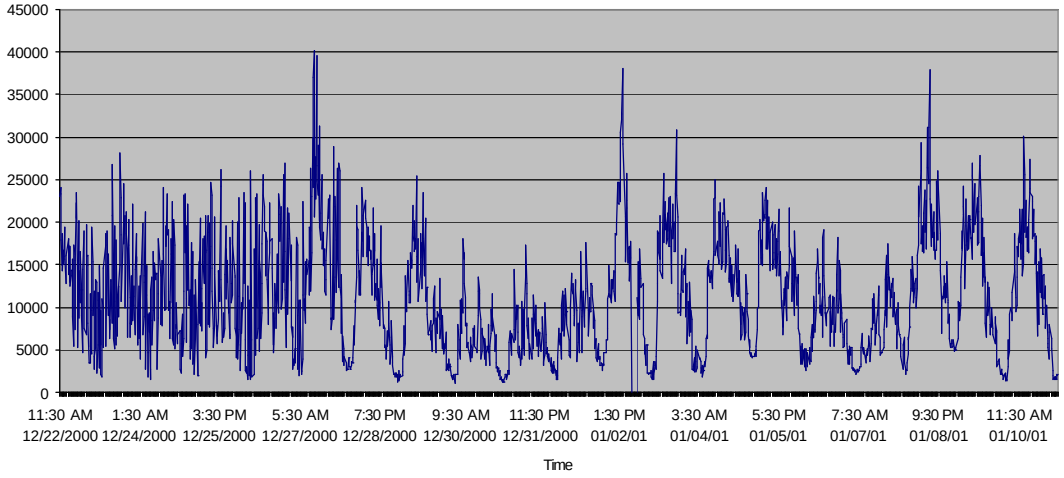


holiday season – if we assume that the high peaks are expected on weekdays and the low peaks on weekends, we find that there are too many low peaks in the first half of the trace. But those extra low peaks occur between December 23 and December 26 and again around December 29 through January 1. The next pair of low peaks appear over January 6 & 7 – the first 'regular work-week' weekends to appear in the trace. Unfortunately, the trace isn't long enough to be sure about the pattern, but this lessening of activity on weekends matches behaviour found in [4]. A longer trace would clarify the existence or non-existence of this pattern.

Some other ways of deciding how busy the network is are – number of packets over time, number of octets (bytes) over time, and number of dropped packets over time. Figure 2 shows the graph for data packets over time. Again, there is a short but in this case, it doesn't quite correspond to the 24 hour clock. In this graph, we see more activity throughout both the day and night around Christmas. Since Figure 1 shows that there weren't more users active during that time, we can conclude that some specific users were using the CDPD around the clock more just before Christmas (perhaps contacting merchants to see if they were still open for last minute gift-buying?)

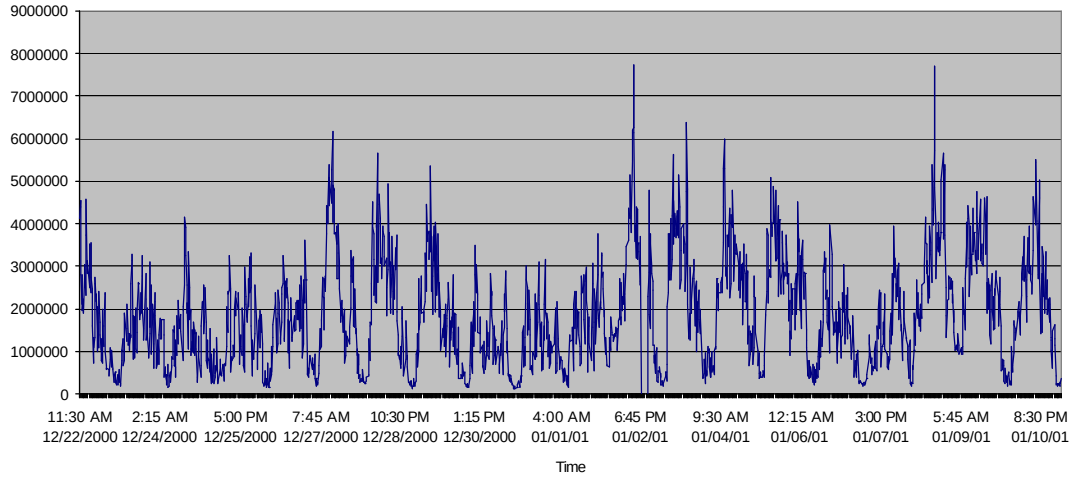
After New Year's Day, more of a cycle begins to appear across the days. Still, a longer trace is needed to detect the pattern.

[3] This series was extracted using my cdpdParser Java[tm] program with the 'userTotalsOn_' flag set to true.



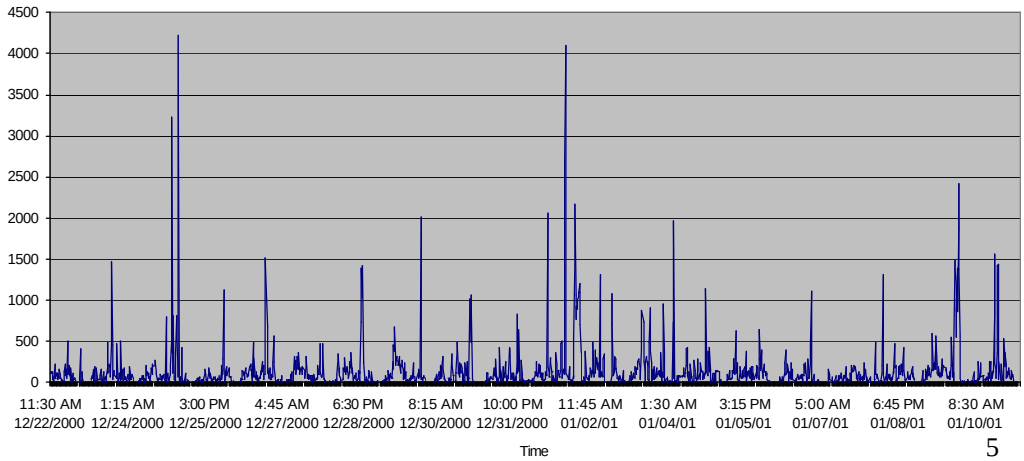
Interestingly, when data octets (as opposed to packets) are plotted over time, the same 'bunching' at the beginning of the trace is not observed (see Figure 3).

Figure 3: Data octets over time



And when discarded packets are plotted over time, the graph looks nothing like the previous ones (see Figure 4), although again, it looks like there are some cycles present; most of the spikes occur around the hours of 12:00 am – 1:00 am.

Figure 4: Discarded packets over time



5) User Behaviour

For this section, the events in the trace were grouped by userID (Mobile NEI). The following graphs were generated to look at the distribution of events by user. Figure 5 shows the breakdown of total events by user. It is immediately obvious that a few users (approximately 10) account for the majority of events in the trace. Figures 6, 7 and 8 show the breakdown of events per user by event type.

Figure 5: All events per user

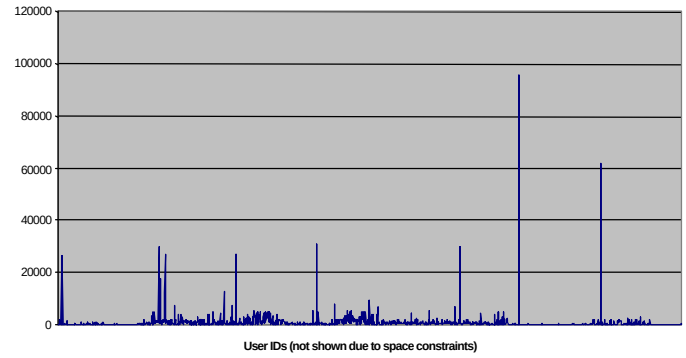


Figure 6: Registration events per user

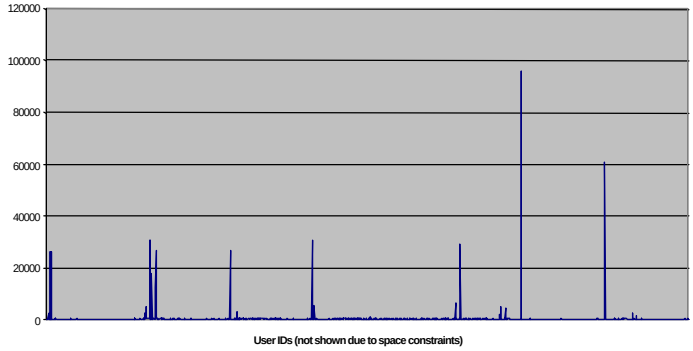


Figure 7: Deregistration events per user

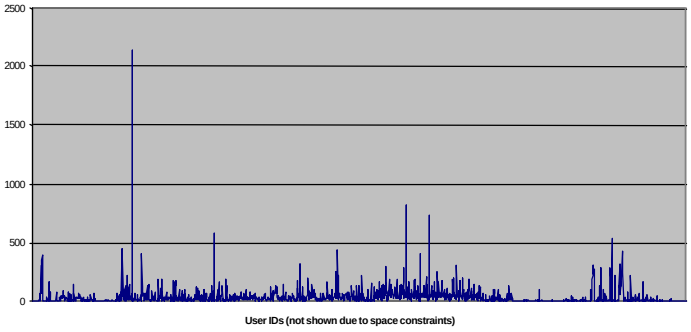
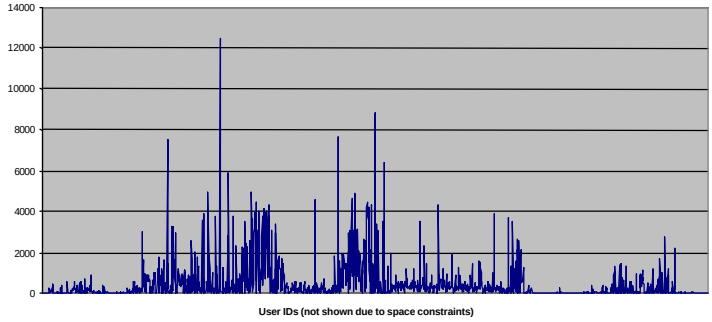


Figure 8: Data events per user



Note that just a few users make up the majority of events in registration, deregistration and total events. However, the effect total is just due to the vastly disproportionate number of registration events for a few users. Detail for these users is shown in Table 5.

Mobile NEI (UserID)	*Registration Events*	*Number of Cells Visited*	*Event Result*
102.133.238.133	2190	30	3
91.171.227.151	2611	6	4
59.207.111.139	2895	9	3
59.207.206.47	5225	7	1
59.206.118.178	12328	2	1
59.206.118.183	12395	12	1
59.206.118.151	17614	28	1
130.43.254.31	26234	20	3
247.33.244.20	26421	32	3
59.206.118.180	26901	32	1
59.206.118.147	30299	35	1
59.207.206.36	31042	7	1
61.131.154.9	95838	2	3

Table 5: 'Misbehaving' registration users

For all of these users, all their events are rejected registration events. None of these users ever gain access to the system. The various results are translated in Table 6.

Result Value	*Result Meaning (from [1])*
1	Registration rejected – no particular reason
3	Registration rejected – Mobile End System not authorized
4	Registration rejected – Mobile End System gave insufficient credentials.

Table 6: Registration result values.

These users account for 47.15% of the registration events and user 61.131.154.9 accounts for 6.06% of the total number of events in the entire trace. None of these users have any successful registrations (and therefore never send data)

It is not obvious whether these users represent someone trying to fraudulently gain access to the network or not, but it is certainly worth considering. Even if the activity is benign, by accounting for so much network activity, these misbehaving users are wasting network bandwidth that other, legitimate, users are entitled to.

The user (59.206.117.22) with the most deregistration events (see Figure 7) actually only has one less registration event than deregistration events (2136 vs 2137) .

The data events are far more evenly distributed (see Figure 8). There are still a few users which appear as outliers in the number of data events which they generate, but the graph is far more even. (The reader should be reminded that this graph represents the number of events generated and does not necessarily correlate with the amount of data sent.)

Figures 9, 10 and 11 show the relative proportion of registration, deregistration and data events per user.

Figure 9: % of reg events for all user's events per user

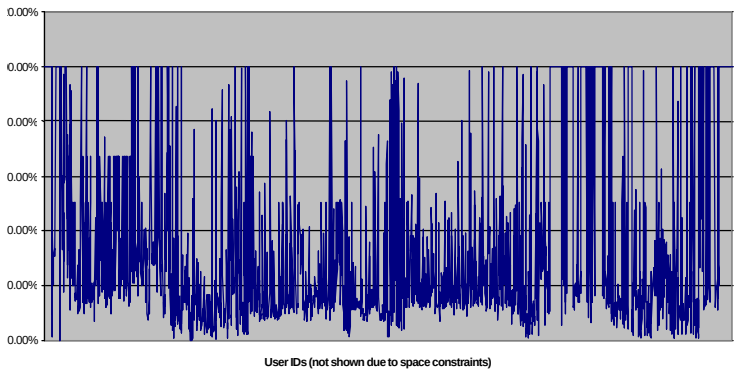


Figure 10: % of dereg events for all user's events per

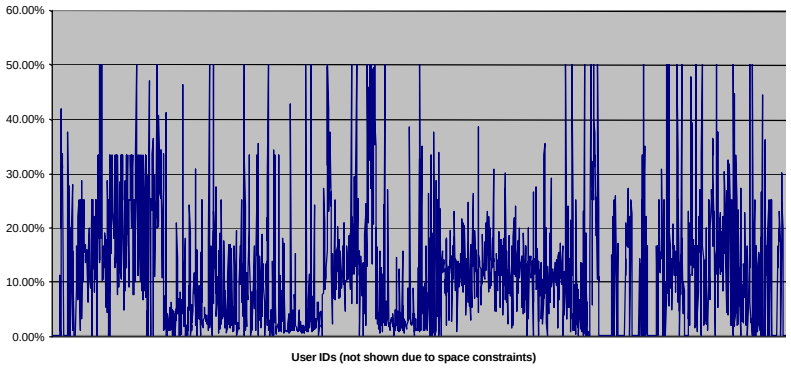
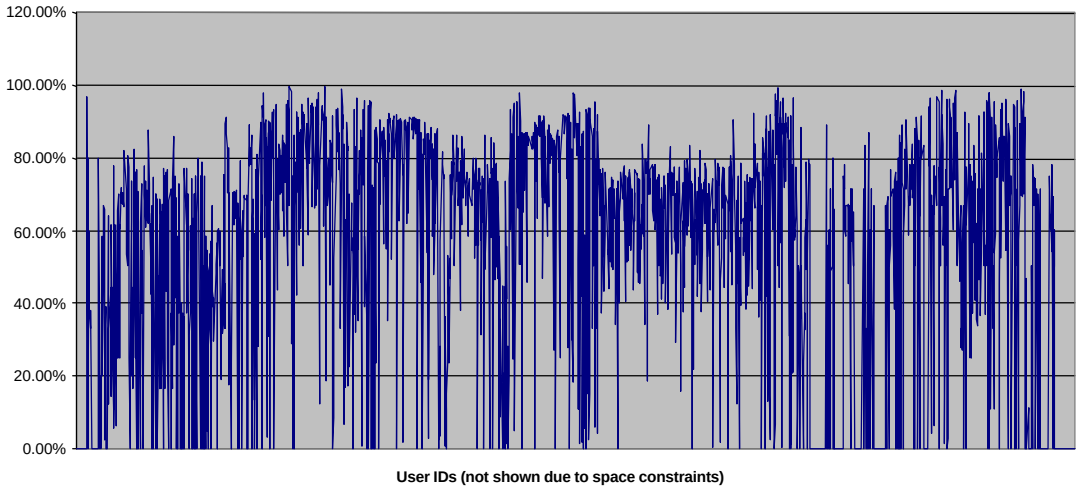


Figure 11: % of data events for all user's events per



It is easy see that for most users, the percentage of registration and deregistration events they generate is low, but there are a lot of users for whom registration events account for 100% of their traffic.

The relationship between data events & packets is shown in Figure 12. The outliers 59.207.209.164 and 59.207.209.151 prevent the scale of the graph from being useful, so those users are removed from the plot for Figure 13.

Figure 12: Data events vs. packets by user

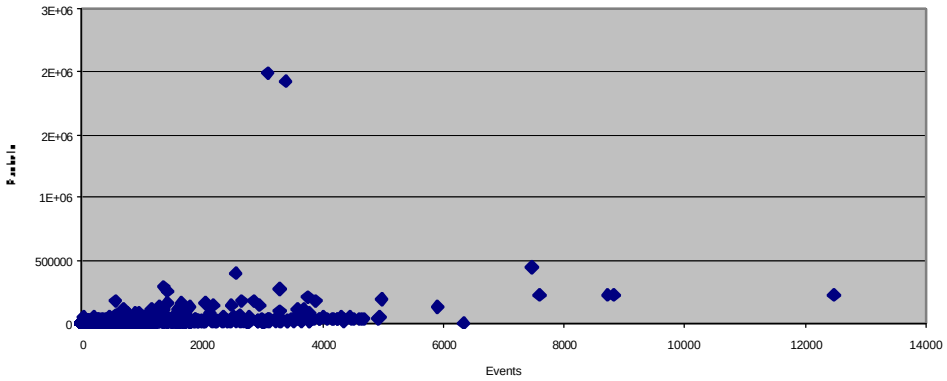
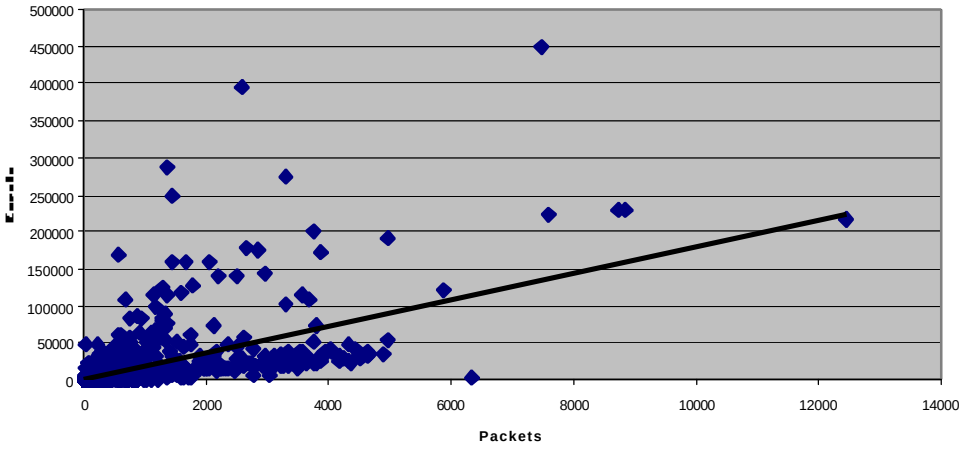


Figure 13: Data events vs packets (w/o outliers)



6) Cell Activity

There are 60 unique cell IDs which appear in the billing trace. Figure 14 shows the total number of events per cell (cell IDs are ordered by when they first appear in the trace) and Figure 15 shows the breakdown by event type. A few of the cells are significantly busier than others, but without a network diagram, it's hard to say much about why.

Figure 14: Total events per cell

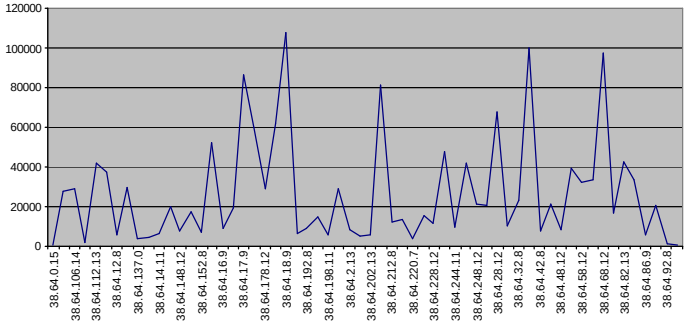
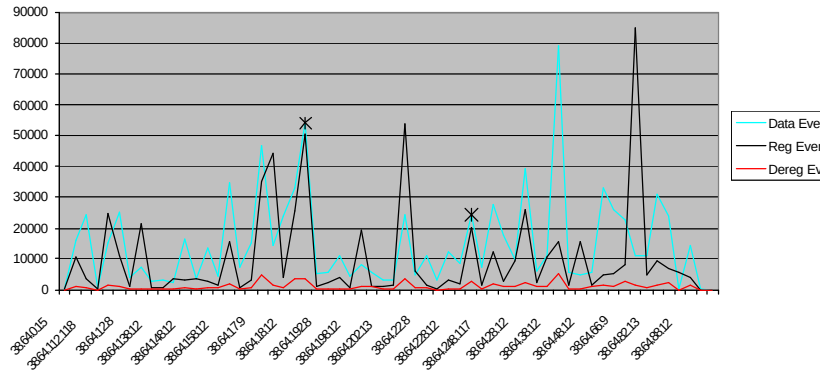


Figure 15: All events per cell



It's fairly obvious that there's not much of a correlation between the different kinds of events. There are two cells in which the number of registration events matches the number of data events fairly well (these are marked with stars in Figure 15). Figures 16, 17 and 18 show the proportion of registration, deregistration and data events per cell.

Figure 16: % registration events per cell

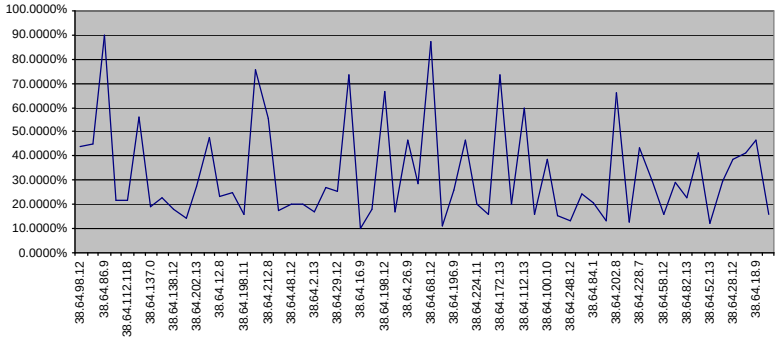


Figure 17: % Deregistration events per cell

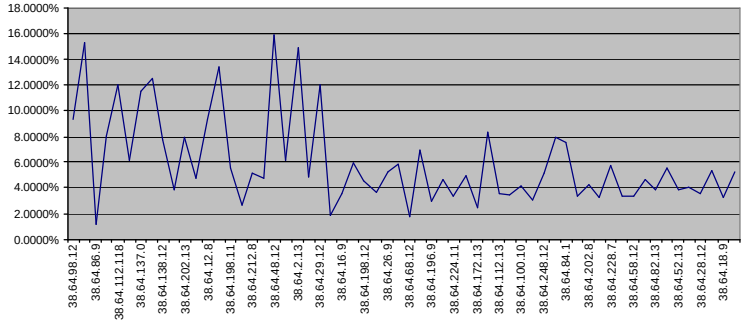
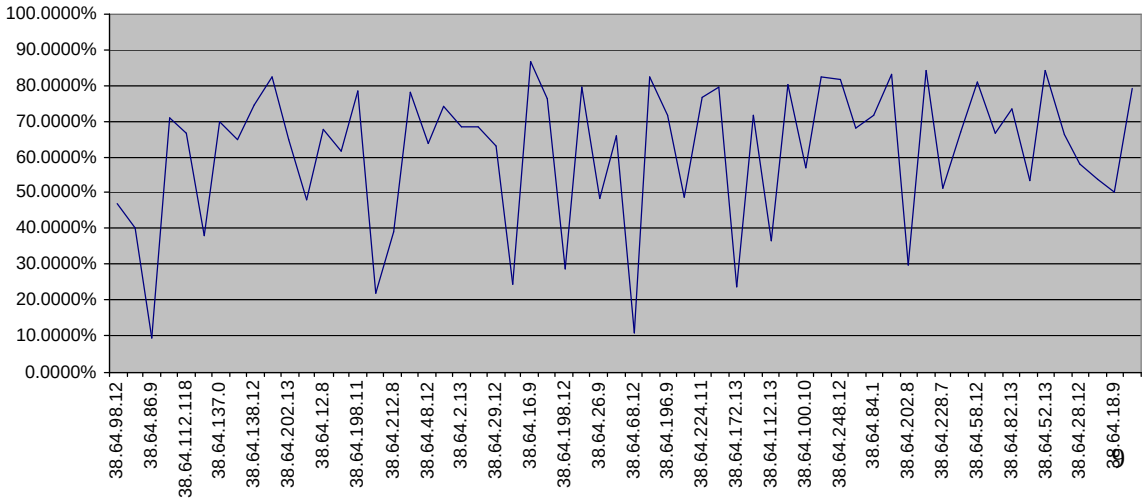
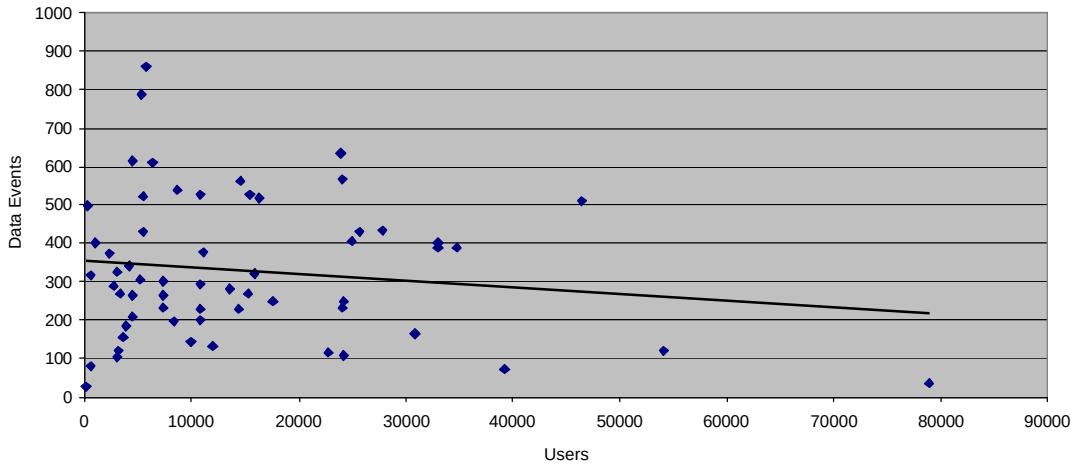


Figure 18: % data events per cell



In general, most of the events occurring in most cells are data events. This indicates less than 50% overhead in the network (in terms of events, not actual bits). Figure 20 shows the interaction between the number of data events in a cell and the number of users in that cell.

Figure 19: Users vs data events/cell



7) Mobility and cell user density

These two statistics are the number of cells visited by each user and the number of users occurring in each cell. They are orthogonal measures of the same data plot. Figure 20 shows the number of unique cells visited by each user & Figure 21 shows the total number of unique users to visit each cell during the trace.

Figure 20: Cells visited in trace per user (mobility)

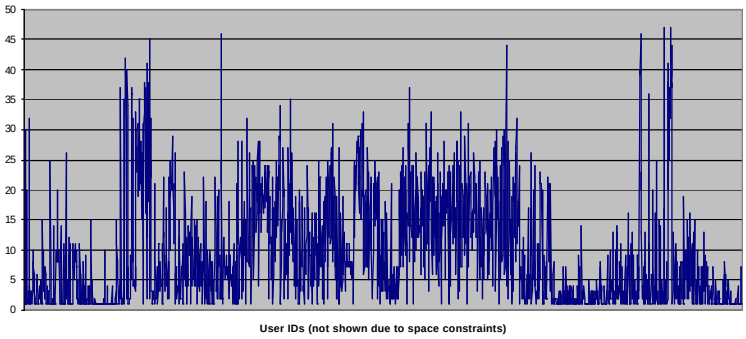
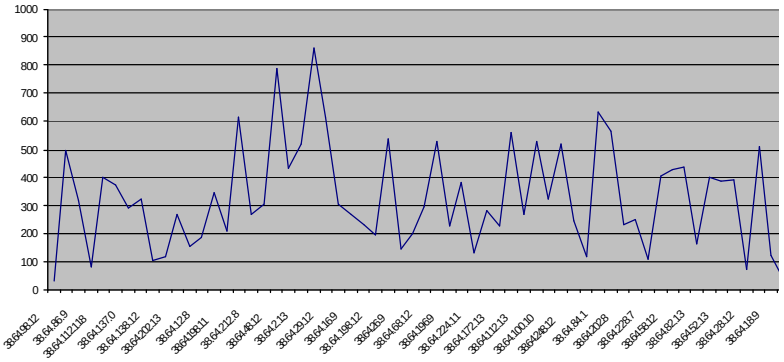


Figure 21: Number of users per cell



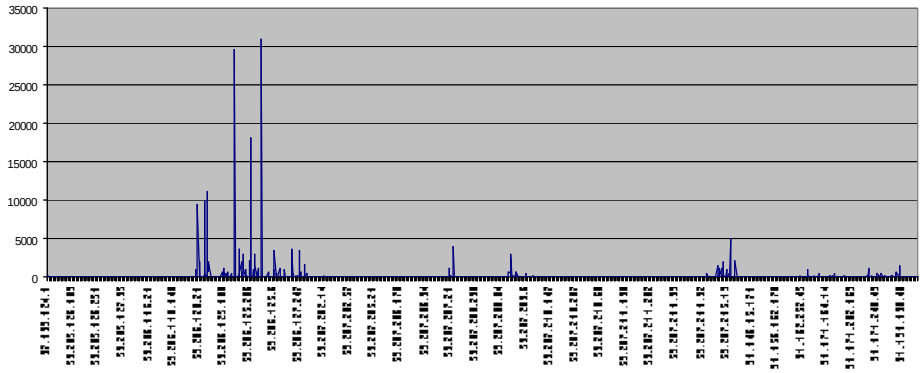
Both graphs show a fairly distributed population. Most users move around at least a bit; the average number of cells visited is 9.4, standard deviation 9.0, maximum 47 and minimum 1. The distribution of users among cells is a bit more varied. The average number of users in a cell is 329.05, with standard deviation of 183.03, maximum 860 and minimum 28.

8) Data traffic

The amount of data packets and octets, as well as discarded packets are plotted over time in Figures 2, 3 and 4. The question here is – are there users whose packets are unfairly discarded? Figure 22 shows

the number of discarded packets per user. While there are some users who have significantly more packets dropped than others, this in itself tells us nothing; these users may also be sending many more packets of data than the other users.

Figure 22: Discarded packets per user



However, Figures 23 and 24 show us the number of discarded packets vs data events and data packets respectively.

Figure 23: Data Events vs. Discarded Packets (per user)

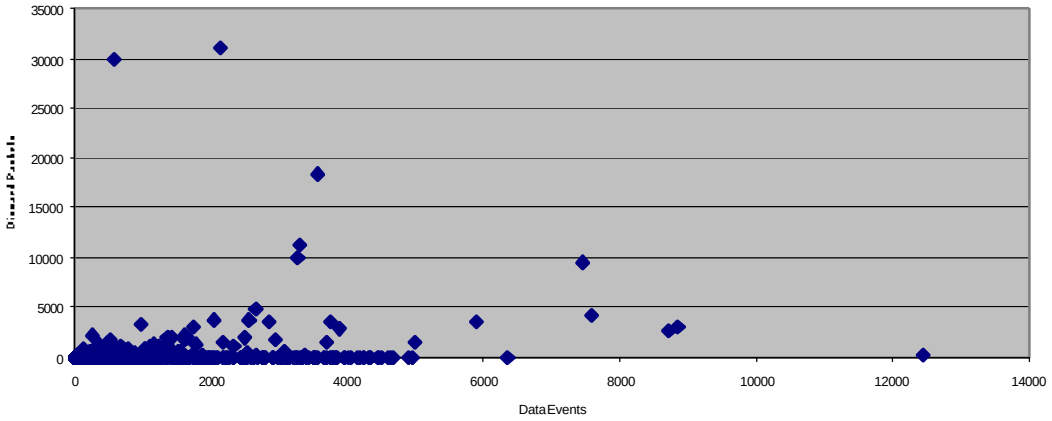
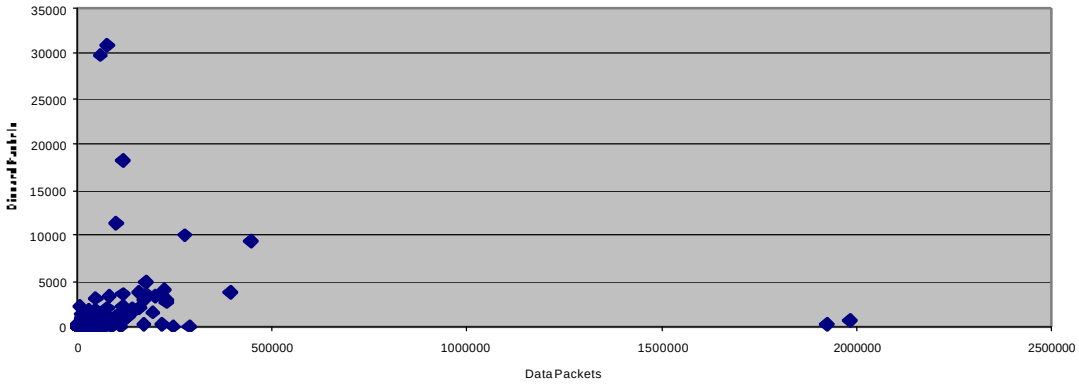


Figure 24: Data Packets vs. Discarded Packets (per user)



Any of the dots which occur very high up & close to the Y axis are ones which have been 'unfairly' dropped and any dots far to the right and close to the X axis are ones which have been very lucky with regard to packet loss . Some specific examples inequity of packet discards are shown in Table 6 below.

MobileNEI (UserID)	*Data Packets*	*Control Packets*	*Discarded Packets*	*Data/discard*
59.207.209.71	285223	98	0	0%
59.206.127.246	202068	184067	3441	1.7%
59.206.120.20	447523	5237	9474	2.12%
59.206.125.155	60342	2586	29871	49.50%
59.206.125.245	74767	2181	30966	41.42%

Table 6: Inequity of packet discards

These users listed represent the extremes of packet loss. On the one hand, user 59.207.209.71 sends almost 5 times the number of packets (with no loss) that user 59.206.125.155 sends (with almost 50% loss). More investigation into these two users' personal traces to try to find a reason for this discrepancy would probably be worthwhile.

9) Conclusion

This trace has yielded some very interesting data despite the fact that it contains only a high-level description of the network traffic. The fact that the trace spanned the Christmas holiday season means that the data may not be completely representative of typical network activity. A longer trace with which to compare this one would allow detection of normal patterns of user behaviour and characterization of any holiday-specific behaviour.

A small subset of users account for a large number of rejected registration events. It could be prudent for the network provider to track and investigate such users to prevent fraudulent network use as well as excessive network overhead.

Some users seem to be discriminated against in terms of packet loss. More work needs to be done to determine what conditions cause this inequity.

The entering of the trace data into an SQL database greatly helped in retrieving interesting statistics and traces. In future, an optimized database should be used to allow faster queries.

References:

[1] “Part 630 Accounting Service and Protocol” “CDPD System Specification” Release 1.1. January 19, 1995. CDPD Forum

[2] The Java™ Programming Language website. http://java.sun.com

[3] The MySQL Home Page. http://www.mysql.com

[4] Diane Tang and Mary Baker, "Analysis of a Local-Area Wireless Network," Proceedings of Mobicom 2000, Boston, August 2000. http://mosquitonet.stanford.edu/publications.html

[5] Diane Tang and Mary Baker, "Analysis of a Metropolitan-Area Wireless Network," Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1999), Seattle, Washington, August 1999. http://mosquitonet.stanford.edu/publications.html

[6] Experiences with a Mobile Testbed (1998) Kevin Lai, Mema Roussopoulos, Diane Tang, Xinhua Zhao, Mary Baker http://citeseer.nj.nec.com/101593.html

[7] Anselm Linhnau, Oswald Drobnik "User Data Management for Mobile Communications An object oriented approach" Johann Wolfang Göthe-Universitaet Frankfurt http://mercan.cmpe.boun.edu.tr/~onure/paper_index.html

[8] http://www.nais.com/business/cdpd.asp