

# **Preliminary CDPD Performance Testing Results**

*David B. Johnson*

*John Drum*

Information Networking Institute  
Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213

October 3, 1995

## **Abstract**

Cellular Digital Packet Data (CDPD) is a new standard, adopted by a consortium of the major United States cellular telephone companies, for data communication with wireless mobile hosts. CDPD is designed to carry data packets using either IP or CLNP over the current cellular telephone network, utilizing the available idle capacity of the network not used by voice calls. Although much has been written about the promise of CDPD, little experience exists with connecting real mobile applications using CDPD. This report describes the current state of CDPD and discusses its performance based on actual measurements of the system currently deployed in Pittsburgh. The performance measurements presented here represent data collected over a mostly continuous period spanning over 3 weeks.

## 1. Introduction

Cellular Digital Packet Data (CDPD) is a new service for wide-area data communication with wireless mobile hosts. The original version of the CDPD System Specification (Release 1.0) was developed by IBM and released in July 1993 by a consortium of the major United States cellular telephone carriers consisting of Ameritech Mobile Communications, Bell Atlantic Mobile Systems, Contel Cellular, GTE Mobile Communications, McCaw Cellular Communications, NYNEX Mobile Communications, PacTel Cellular, and Southwestern Bell Mobile Systems. The standard is now controlled by an open trade organization known as the CDPD Forum, which released a revision of the CDPD standard (Release 1.1) in January 1995.

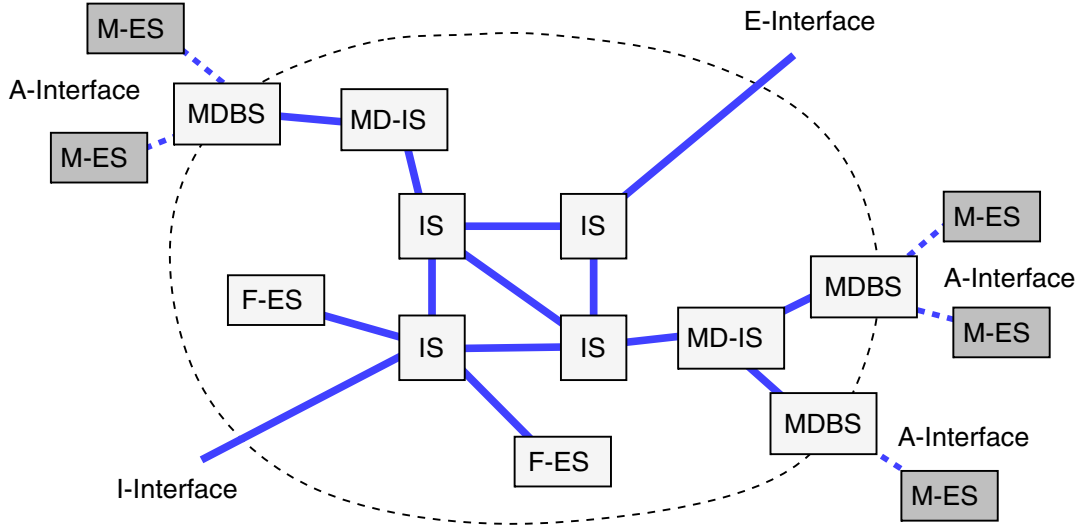
CDPD is designed to carry data packets using either IP or CLNP over the current cellular telephone network, utilizing the available idle capacity of the network not used by voice calls. Although much has been written about the promise of CDPD, little experience exists with connecting real mobile applications using CDPD. This report describes the current state of CDPD and analyzes the performance of CDPD based on actual measurements of the system currently deployed in Pittsburgh by Bell Atlantic NYNEX Mobile. The performance measurements presented here represent data collected over a mostly continuous period spanning over 3 weeks.

Section 2 of this report describes the general architecture of a CDPD system. Section 3 then describes some of the history of our experience with CDPD at Carnegie Mellon University. In Section 4, we describe the network configuration and testing methodology we used in measuring the performance of CDPD in Pittsburgh, and in Section 5, we discuss the results of this testing. Finally, we present graphs showing our measured performance of the CDPD network.

## 2. CDPD Architecture

The CDPD network exists as an overlay on top of the existing Advanced Mobile Phone Service (AMPS) cellular telephone network. It uses the existing radio channel, base station, and antenna infrastructure of the AMPS network, and is designed to exploit the idle capacity of the existing channels. With CDPD, the use of a channel to carry a voice call is given priority, and CDPD “hops” between idle channels within each cell. That is, between voice calls, a channel may be used for CDPD, but when a voice call again needs that channel, CDPD must switch to another channel. CDPD uses only time on a channel *between* voice calls, not silence periods *within* any voice call. A cellular carrier may also dedicate one or more channels in a cell to CDPD usage, but this is not the normal configuration.

Figure 1 shows the configuration of a typical CDPD provider network. A mobile node in CDPD is referred to as a Mobile End System (M-ES), each of which is assigned a unique equipment identifier (EID) by its manufacturer; the M-ES is usually implemented by a dedicated “CDPD modem” that executes all functions of the CDPD protocol on behalf of a host computer to which it is attached. When a customer signs up for CDPD service, the cellular carrier assigns an IP (or CLNP) address to the M-ES (to the CDPD modem), which is used for routing packets to the M-ES. CDPD distinguishes between routing packets to other M-ESs and to “fixed” (non-mobile) nodes, called Fixed End Systems



**Figure 1** Example CDPD Provider Network Configuration

(F-ESs). An F-ES may be any node on a traditional (non-CDPD) network connected to the CDPD system, or may be a fixed node used internally by the CDPD system. In each cell that supports CDPD, a Mobile Data Base Station (MDBS) manages the cellular radio channels of that cell. Packets destined for an M-ES are routed to the correct MDBS by the Mobile Data Intermediate System (MD-IS), a specialized router within the CDPD system. A CDPD system may have one or many MD-ISs, each serving a group of MDBSs.

Internally, a CDPD network may have one or more intermediate systems (ISs, or routers) to connect the CDPD network to other networks. In particular, CDPD defines two special types of interfaces to other networks connected by an IS. The E-Interface is the defined interface which CDPD uses to connect to “external” networks that are not part of any CDPD system. For example, since CDPD can carry IP packets, it is natural to imagine a CDPD network connected directly to the Internet. However, this is not generally possible yet for many CDPD providers due to the restrictions placed on the regional Bell operating companies (RBOCs) during the breakup of the Bell System. This restriction is expected to change at some point in the future, and CDPD customers are currently able to connect their own wired networks into the CDPD system through their CDPD service provider. The I-Interface is the defined interface which CDPD uses to connect one CDPD network to another, such as to support roaming from one provider’s network to another. Roaming protocols and agreements have not yet been worked out, however, and CDPD does not currently support roaming outside the provider’s CDPD system. CDPD also defines an A-Interface (“air” interface) which defines the protocols used over the cellular radio link between the MDBS and each M-ES registered in its cell.

CDPD networks charge for usage based on a price per packet sent and per kilobyte of data sent. This differs significantly from the charging system used for voice circuit switched cellular calls, which charge per connect time minute. There are no connect time charges on CDPD, as there is

no connection. To use CDPD, the CDPD modem does not dial a telephone number and connect to a remote system. Instead, individual data packets (either IP or CLNP) are sent over the air to the MDBS, for forwarding through the MD-IS to other M-ESs registered with CDPD, or to any F-ESs on any wired network connected to the CDPD system.

Packets are sent digitally over the cellular radio channel at a raw bit rate of 19.2 Kbits/second, although a number of factors significantly reduce the effective usable data rate. Subtracting only the deterministic overhead bits defined by the CDPD air interface protocol, the usable data rate falls to only 11.8 Kbits/second for packets transmitted from the MDBS to an M-ES, and 13.3 Kbits/second for packets transmitted from an M-ES to the MDBS. A packet being transmitted is divided into *segments*, and a segment header is added to each. Each segment is then encrypted using the proprietary RC4 encryption algorithm, and a *frame* header is added to each segment. The resulting frames are then concatenated together, with flag bits added to delimit the individual frames, and the resulting bit sequence is divided into a series of fixed-sized *blocks*. Each block then has Reed-Solomon Forward Error Correction (FEC) codes added, allowing errors in transmission to be corrected by the receiver. The usable data rate that can be transmitted is thus reduced by each additional level of header overhead, and the Reed-Solomon FEC code adds substantial additional overhead bits to the transmission.

In each cell, the MDBS is responsible for channel usage within that cell, and “hops” from one channel to another if the channel it is using is taken for a voice cellular connection. The M-ESs using CDPD in the cell follow the MDBS, either by receiving an explicit channel hop command from the MDBS, or by scanning all channels to find the one now in use by CDPD. The forward link channel (from the MDBS to the M-ESs in the cell) is a broadcast channel with no contention. The MDBS simply transmits packets on this channel as they arrive for any of the M-ESs registered in the cell. Interspersed in this data stream are “channel idle” and “collision” flag bits used to approximate the CSMA/CD (carrier sense multiple access with collision detect) multiple access protocol used for example on ethernet. An M-ES wishing to transmit a packet on the reverse link channel (from the M-ESs in the cell to the MDBS) listens to the forward link channel until a flag bit from the MDBS indicates that the reverse link channel is currently idle. It then transmits its packet on the reverse link channel and checks the collision flag bit on the forward link channel. If the collision bit indicates that the packet was not successfully received by the MDBS, the M-ES waits a short period and retransmits the packet. The average retransmission interval for a single packet doubles with each attempted transmission, with the actual delay chosen randomly from this interval. The lower theoretical data rate in the channel from the MDBS to an M-ES (11.8 versus 13.3 Kbits/second) is caused by the additional overhead from the “idle” and “collision” flag bits sent on this channel.

The current CDPD standard includes two types of compression that may improve the usable transmission data rate. The original CDPD standard (Release 1.0) provided for use of “header compression” for packets carrying transport and network level protocols that support header compression, although currently, only TCP header compression is defined. TCP headed compression removes the fields from the combined TCP and IP headers that do not change between sequential packets on the same TCP connection; the 40 bytes of TCP and IP headers can typically be reduced to 3 bytes. The headers for other types of packets, such as UDP, are not compressed, and TCP header compression is not supported by all CDPD implementations. In particular, the Bell Atlantic NYNEX Mobile system

in Pittsburgh which we are using does not support TCP header compression. In addition to header compression, Release 1.1 of the standard added support for V.42bis data compression, as is used on recent circuit switched telephone modems. V.42bis can compress some data by as much as a factor of 3 or 4, but few CDPD systems or products, including the Bell Atlantic NYNEX Mobile system in Pittsburgh, have yet been upgraded to support Release 1.1 of the standard.

### **3. Carnegie Mellon CDPD Experience**

We are currently using the Ubiquity 1000 CDPD modem made by PCSI, a division of Cirrus Logic. This CDPD modem is designed specifically to be mounted inside an IBM ThinkPad model 75x laptop computer, and was co-designed by IBM. Our CDPD service is provided through a service grant from Bell Atlantic NYNEX Mobile. The Bell Atlantic CDPD system currently only implements Release 1.0 of the CDPD protocol.

During the past year, several issues arose that hindered the adoption of CDPD service on the CMU campus. Last fall, the introduction of CDPD service to CMU hosts required the establishment of a 56 Kbps Frame Relay connection from the Bell Atlantic Mobile network to the CMU campus. The necessity of purchasing new equipment on the CMU side, debugging the Frame Relay link between routers from different vendors, and developing Frame Relay experience among both Bell Atlantic Mobile and CMU personnel conspired to push back the initiation of service for some time. Frame Relay problems also contributed to several other CDPD service outages in the past year.

In the early summer of the year, it was found that the Bell Atlantic Mobile Oakland cell site, which is the site closest to the CMU campus, frequently dropped off the air in its CDPD service. Since Bell Atlantic at the time had few other customers using CDPD (and we believe none in Oakland) other than CMU, this problem went unnoticed by Bell Atlantic, and it took some time to work out a trouble reporting mechanism with Bell Atlantic whereby the Oakland cell site CDPD service could be restarted. A daily monitoring regimen at CMU provided the feedback that substantially remedied this problem.

The fragility of the PC/TCP ftp server which we are using was a continual frustration during our testing. The server never remained up for more than about 20 hours and frequently failed after only several hours of operation. It is not clear whether the fault with this lies with the ftp server itself, with the TCP/IP software that is part of PC/TCP, with the Ubiquity 1000 modem, or with the Bell Atlantic system. We are exploring options for adding CDPD to UNIX platforms to take advantage of their stability and more mature networking code.

Late last fall, CDPD coverage no longer extended to the INI cluster in the basement of Hamburg Hall on the CMU campus where testing was to be performed, we believe due to a change in the cellular antenna configuration. This reduced coverage hampered testing by requiring placement of the CDPD-equipped laptop in Wean Hall rather than in the INI cluster in Hamburg Hall.

Finally, the CDPD system, either in the TCP/IP software on the ThinkPad, in the CDPD modem, or in the Bell Atlantic CDPD system itself, occasionally hangs completely when stressed with heavy usage. Unlike the ftp server problem mentioned above, this problem caused a complete failure of all subsequent CDPD communication, rather than only a failure of future ftp data transfers. For example,

in our tests reported here, we transferred only 60 Kbytes in each ftp file transfer test, and we allowed each ping test to run for only 5 ping packets. If, instead, ftp is allowed to run for much longer, such as to transfer a megabyte or so of data, or if ping is allowed to run for much longer, such as a couple of hours of continuous pings (1 packet per second), the connection hangs and can only be restarted by rebooting the ThinkPad and reinitializing the CDPD modem. Our current hypothesis is that this may be a problem in the PCSI Ubiquity CDPD modem itself, but we are still attempting to verify this as the source of the problem and to work out a solution to it.

## 4. Test Configuration and Methodology

We conducted a series of experiments to measure the performance of CDPD service as currently deployed in the Pittsburgh metropolitan area by Bell Atlantic NYNEX Mobile. Throughput and latency were the chosen performance metrics. Between two machines connected with a CDPD connection, throughput was measured as the time required to transmit a block of randomly chosen characters from one machine to the other; latency was defined as the time required to transmit a small packet from one machine to the other and back again to the first machine. All experiments involved computers on the CMU campus using Bell Atlantic Mobile's Oakland cell site for CDPD service.

CDPD performance was measured between a stationary machine (F-ES) and a CDPD-equipped mobile machine (M-ES). The stationary machine was a Sun Microsystems SPARCstation 20 workstation running SunOS 4.1.3 Unix, located in the INI cluster at Carnegie Mellon. The mobile machine was an IBM ThinkPad 755Cs laptop with a 75 MHz Intel i486 CPU, running Microsoft Windows 3.1 and IBM PC-DOS 6.3. The TCP/IP software on the ThinkPad was FTP Software's PC/TCP version 2.3. The PCSI Ubiquity 1000 CDPD modem we used provides a SLIP interface to the host machine, and we used the built-in SLIP implementation of PC/TCP for this. Packets are thus transmitted between the host machine and the CDPD modem using SLIP; the CDPD system itself then uses the CDPD air interface protocol rather than SLIP for sending packets between the modem and the CDPD MDS.

Once the laptop established a CDPD connection, its packets traveled via cellular radio to the Bell Atlantic Mobile Oakland cell site, from which they traveled by wire within the Bell Atlantic Mobile system to the Bell Atlantic Mobile switching office in the Pittsburgh Strip District. Each packet was then forwarded by the Bell Atlantic Mobile MD-IS to a router connecting the CDPD network to the Carnegie Mellon Campus via a 56 Kbps Frame Relay link. At the CMU end of this link, a router located in CMU's Cyert Hall forwarded each packet to CMU's campus ethernet network, consisting of a large collection of bridged ethernet segments.

The ThinkPad was booted with PC-DOS 6.3. After a CDPD connection was achieved, the ftp server was started to enable the transmission of data between the two machines. From this point onward, the ThinkPad played a passive role in the testing, as the SPARCstation generated the network requests and logged the results.

On the SPARCstation, special versions of the UNIX network utilities "ping" and "ftp" were coded to log network performance. Ping is a widely-available network diagnostic tool that determines reachability and measures round-trip times between Internet hosts. Ftp is a popular program/protocol

that uses TCP to transfer files or other arbitrary data between a pair of machines. These two utilities were modified to record the time and performance of each operation in a log file. A control program was then created to invoke the modified ping and ftp programs periodically over a longer span of time to generate a performance database.

For transferring data from the SPARCstation to the ThinkPad, a further modification to the ftp program allowed us to send random data generated in memory rather than read from the local disk of the SPARCstation, thus avoiding the effects of disk performance in the performance results. Similarly, for transferring data from the ThinkPad to the SPARCstation, the data received via ftp was discarded by the SPARCstation rather than being written to the local disk in order to avoid the overhead of this disk transfer in the network performance measurements. However, in both cases, the local disk of the ThinkPad was involved in the performance measurement, since we were unable to modify the PC/TCP ftp server. Thus, for data transferred to the ThinkPad, the data was written to the ThinkPad's local disk, and for data transferred from the ThinkPad, the data was read from the ThinkPad's local disk. This disk overhead did not affect our ping performance measurements, however, since the ping protocol only transfers a short packet in each direction and does not operate as a file transfer protocol.

In the experiments that have been conducted to date, the periodic network operations consisted of repetitions of the sequence: ping mobile host, file transfer to mobile host, file transfer from mobile host. The ping operation involved the exchange of 5 ping (ICMP Echo Request/Reply) packets between the machines. The ftp operations involved the transmission of 60 Kbytes of randomly-generated data between the two hosts. One network operation was initiated every 2 minutes. That is, the ping program was first run for 5 ping packets, then the ftp program was run to transfer 60 Kbytes to the mobile host, and finally the ftp program was run to transfer 60 Kbytes from the mobile host. This entire cycle repeated every 6 minutes. Operations that timed out were killed by the control program.

## 5. Results

The following pages show the measured performance were able to obtain from the CDPD system. Separate plots are shown for the ping latency, ftp throughput to the mobile machine, and ftp throughput from the mobile machine. Each plot shows one day's results from these measurements, with the time of day along the x-axis and the performance measure along the y-axis. For the ping latency tests, the performance measure is the packet round-trip time in milliseconds. For the two ftp throughput tests, the performance measure is the ftp data throughput in kilobytes per second.

In general, we found that CDPD latency (defined as the time required to exchange packets sequentially in both directions) usually fell in a narrow band around 0.5 seconds. However, latencies of up to 12 seconds occurred sporadically. CDPD throughput demonstrated considerable variation with an average value of almost 1.4 Kbytes/second (11 Kbits/second) going to the mobile node and almost 1.2 Kbytes/second (10 Kbits/second) coming from the mobile node.

Table 1 compares our measured performance to the maximum theoretical throughput of a CDPD channel, accounting only for the deterministic overhead bits added by the CDPD air interface protocol. The difference between theoretical and measured throughput in the "to M-ES" direction can be

**Table 1**

Comparison of theoretical to measured throughput (Kbits/second)

Direction	Theoretical	Measured
To M-ES	11.8	11
From M-ES	13.3	10

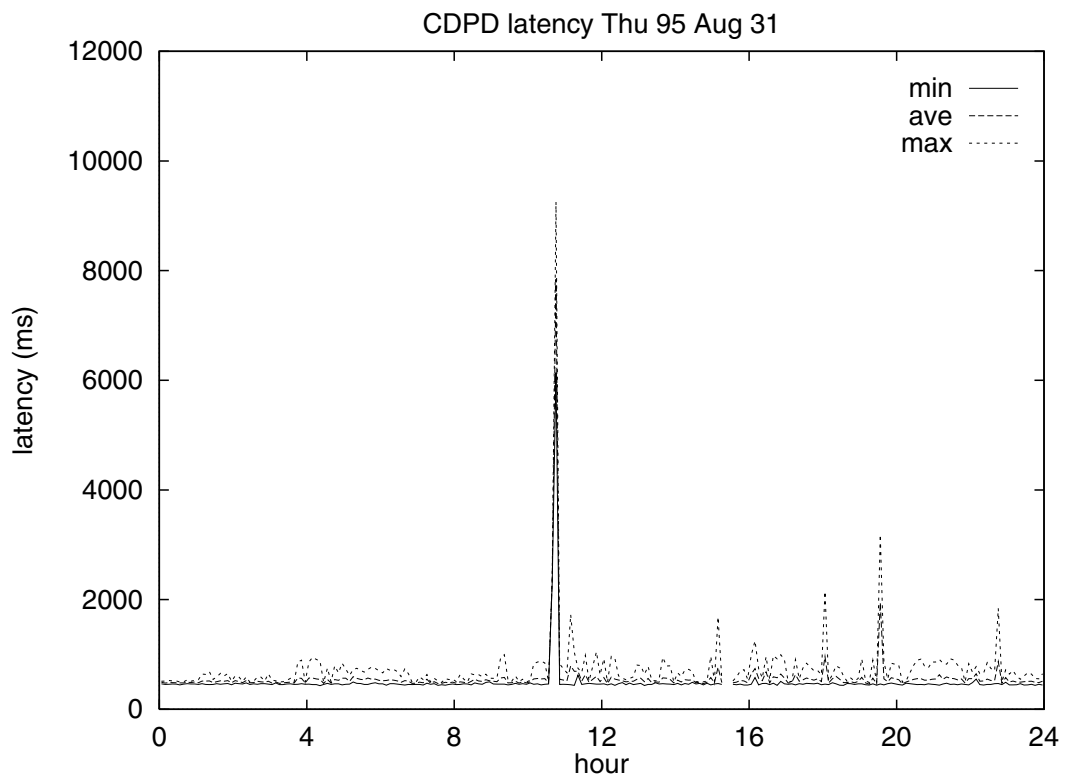
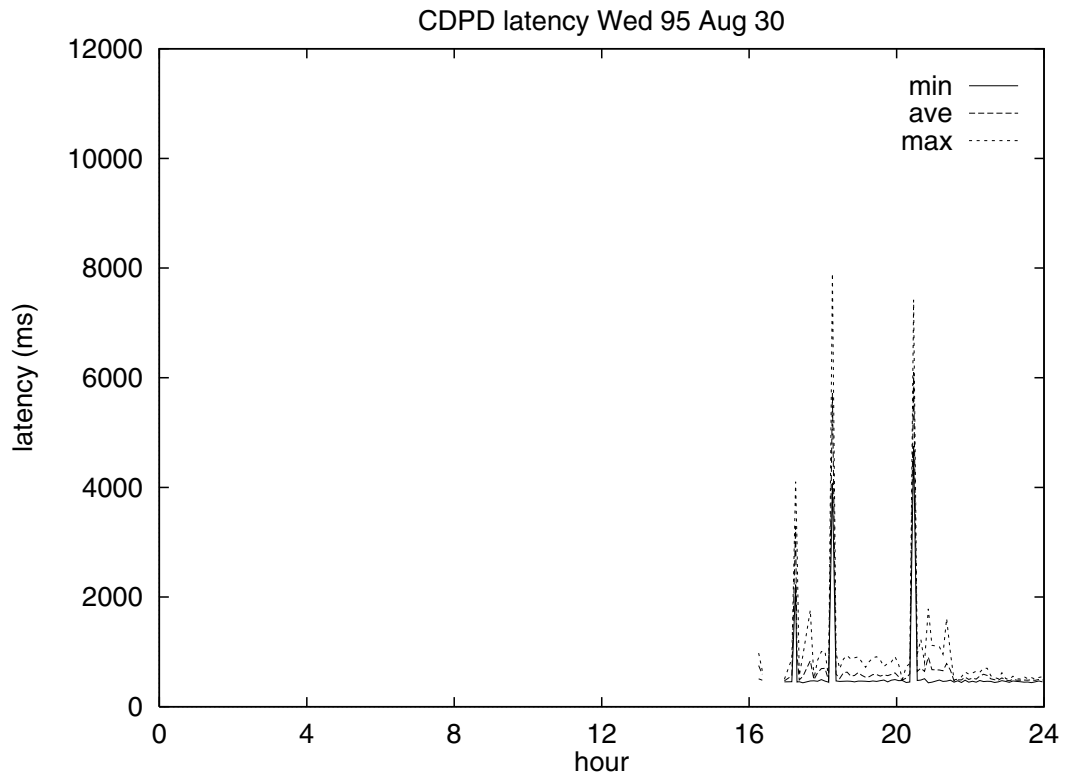
accounted for by added the size of the TCP/IP headers, which are not considered in the theoretical throughput calculations. The difference between theoretical and measured throughput in the “from M-ES” direction can be accounted for by the size of the TCP/IP headers, plus the overhead time spent in contention for access to the channel when needing to transmit. For the channel to the M-ES from the MDDBS, only the MDDBS transmits on that channel and so there is no contention. For the channel from the M-ES to the MDDBS, the M-ES must contend for time against all other M-ESs in the cell also wanting to send. Even if there are no other M-ESs in the cell, the CDPD protocol still requires the sender to wait a short while to decide if the channel is currently idle, and to also wait a short while after sending to decide if that transmission was successful. This extra contention overhead accounts for the larger drop in the “from M-ES” measurements relative to theoretical.

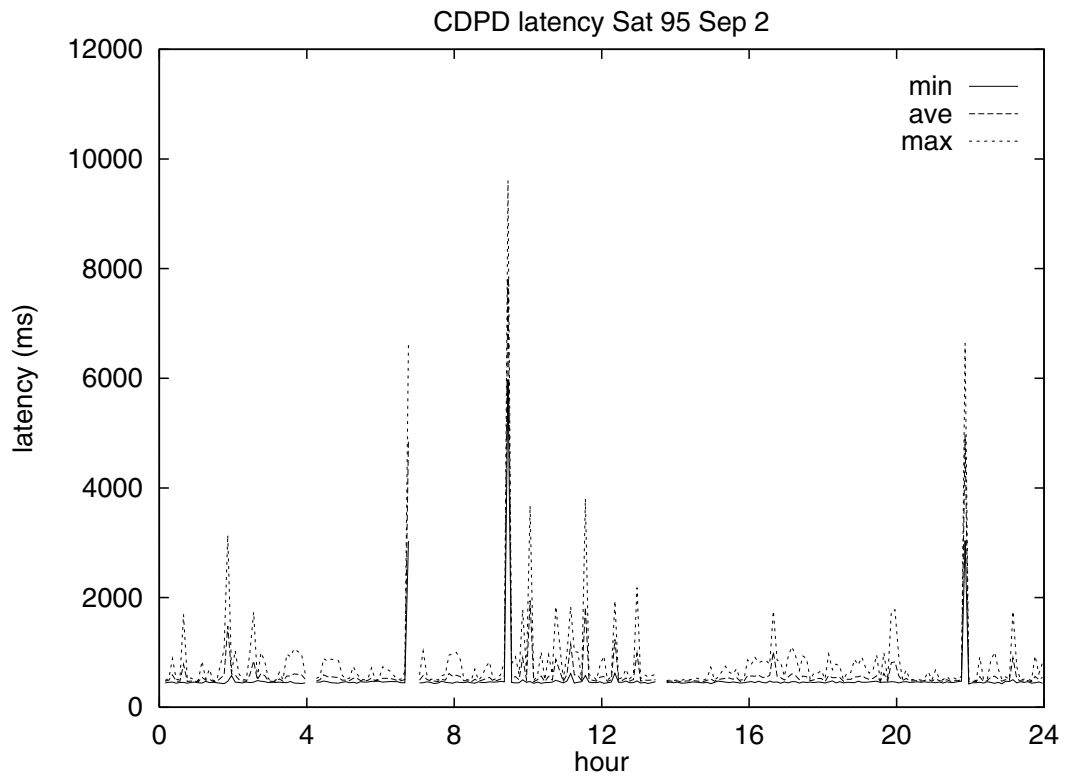
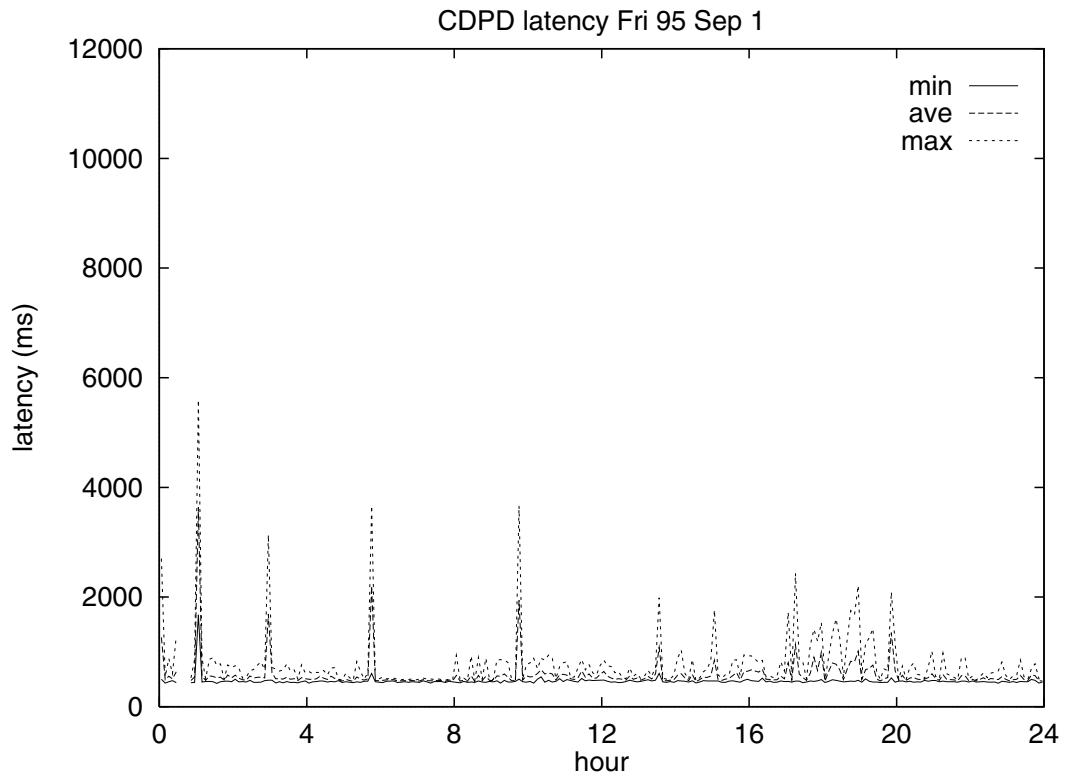
One thing of note in the performance graphs reported here is the large number of gaps in the measurements. These gaps in the plotted results generally indicate periods during which CDPD had hung and we had not yet been able to reboot the ThinkPad and restart the ftp server running on it. Each time CDPD hung, we restarted the ThinkPad as soon as possible; what is perhaps more interesting than the length of the gaps in the measurements, though, is the number of such gaps. Most such gaps represent a separate time that CDPD hung after being restarted. Some gaps in the measurements, however, do represent periods during which, for example, the SPARCstation on which we were running the performance data program was not running (for example, we lost some time due to a hardware upgrade of the machine), or during which we were having an unrelated problem with the local ethernet network at CMU.

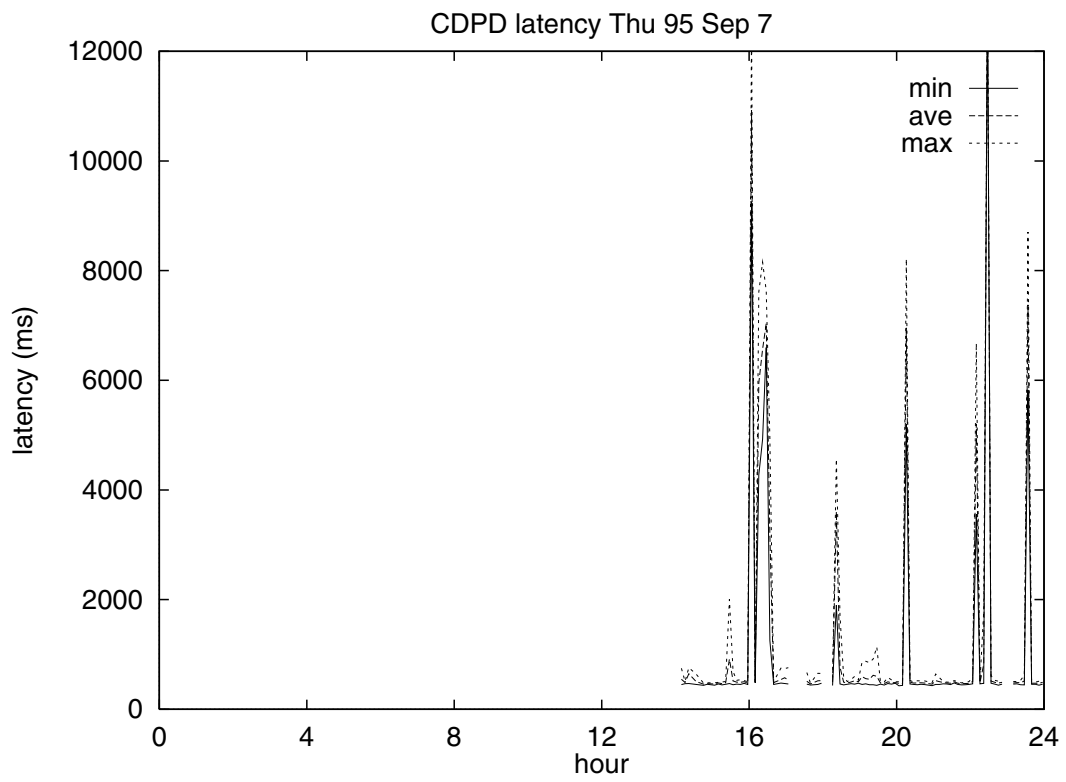
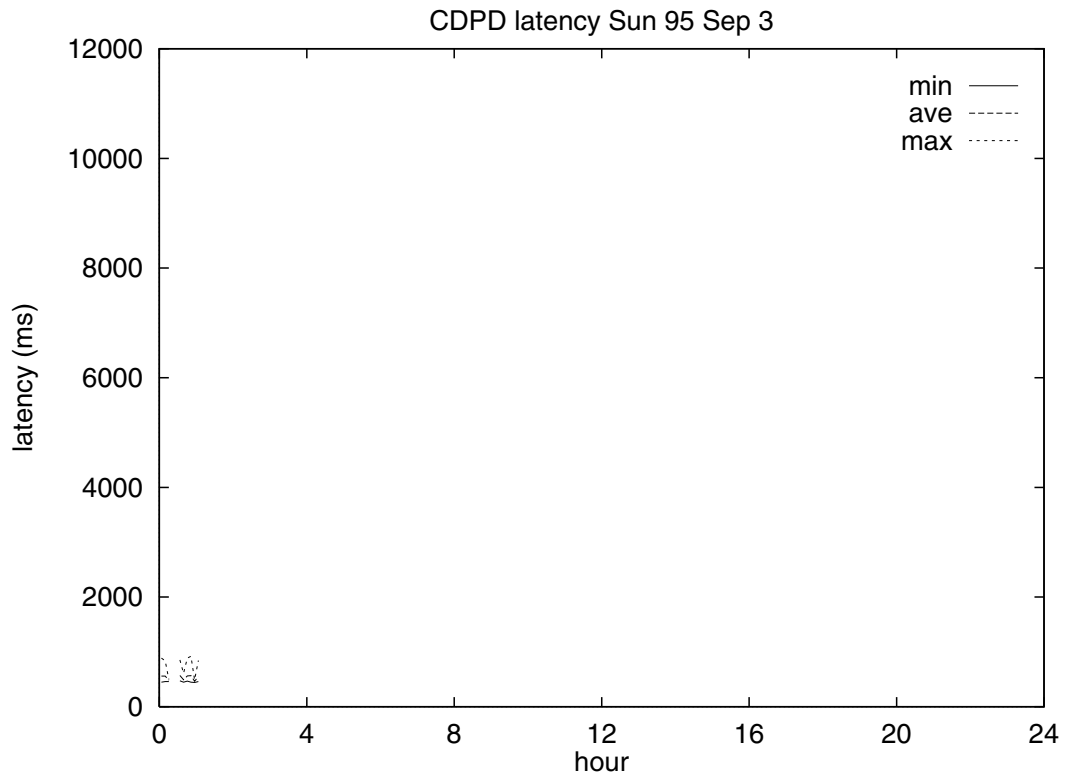
We are continuing to perform additional measurements on the CDPD system and hope in particular to isolate the cause of the occasional connection hanging we have been experiencing. We have recently received two IBM PCMCIA Wireless Modem for Cellular/CDPD cards, which we are currently configuring for use. These cards will allow us to determine if the cause of the problem is specific to our current PCSI Ubiquity 1000 CDPD modems. Also, we are currently working on developing support for CDPD under the NetBSD Unix operating system running on the ThinkPad; the networking software in Unix is in general much more stable and robust than DOS or Windows implementations of the protocols, and we thus hope to be able to determine if the problem lies instead in the DOS/Windows PC/TCP networking software we are currently using.

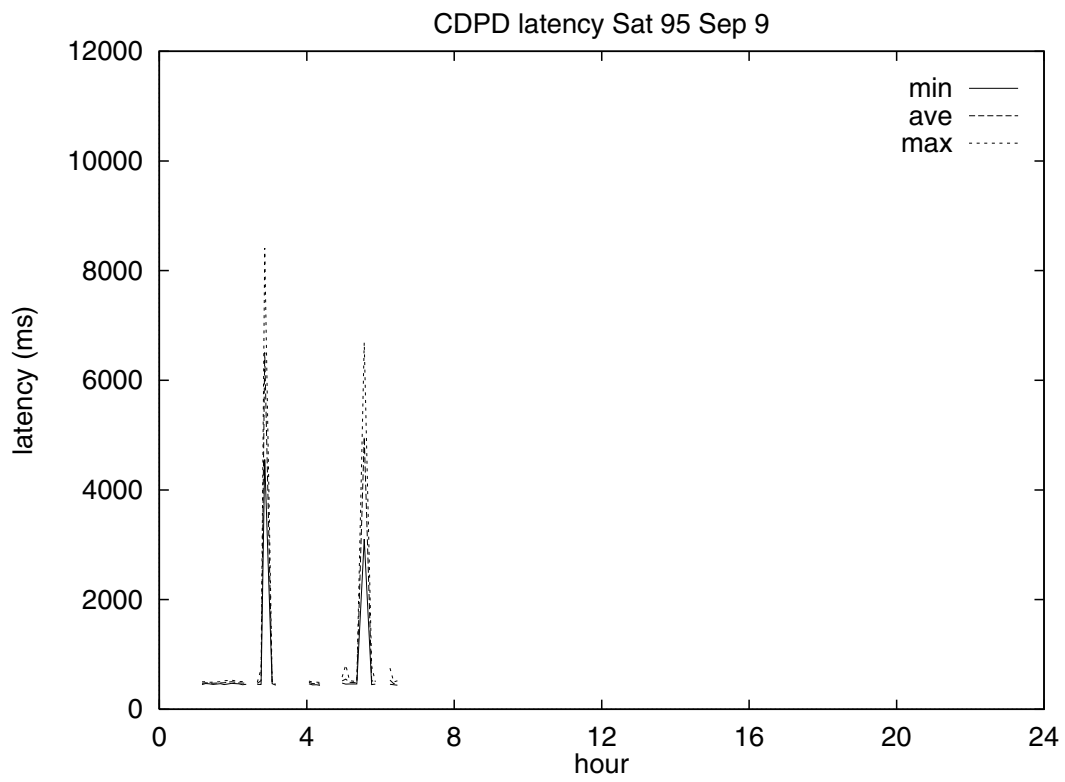
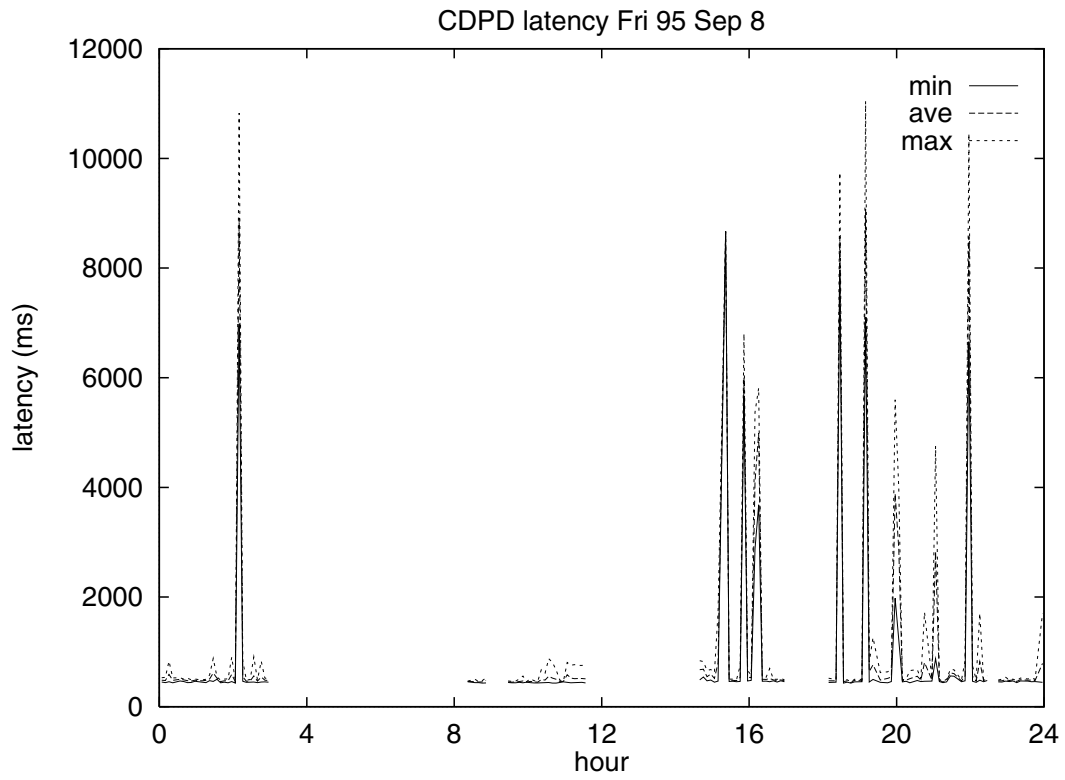


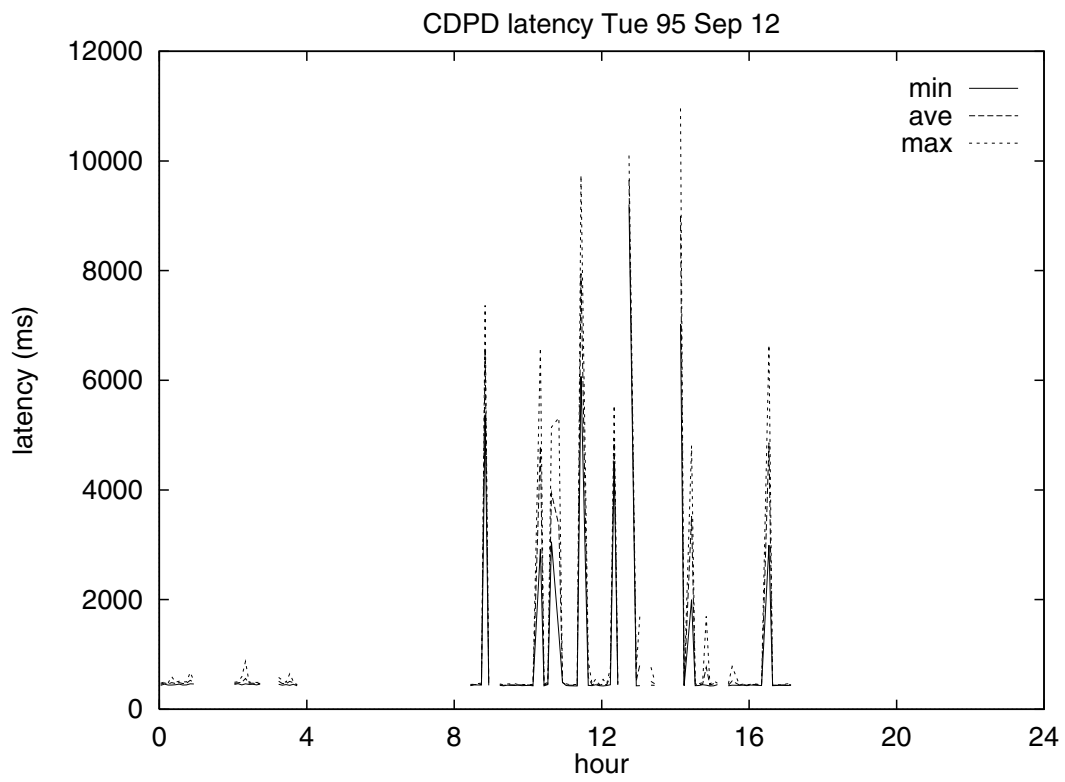
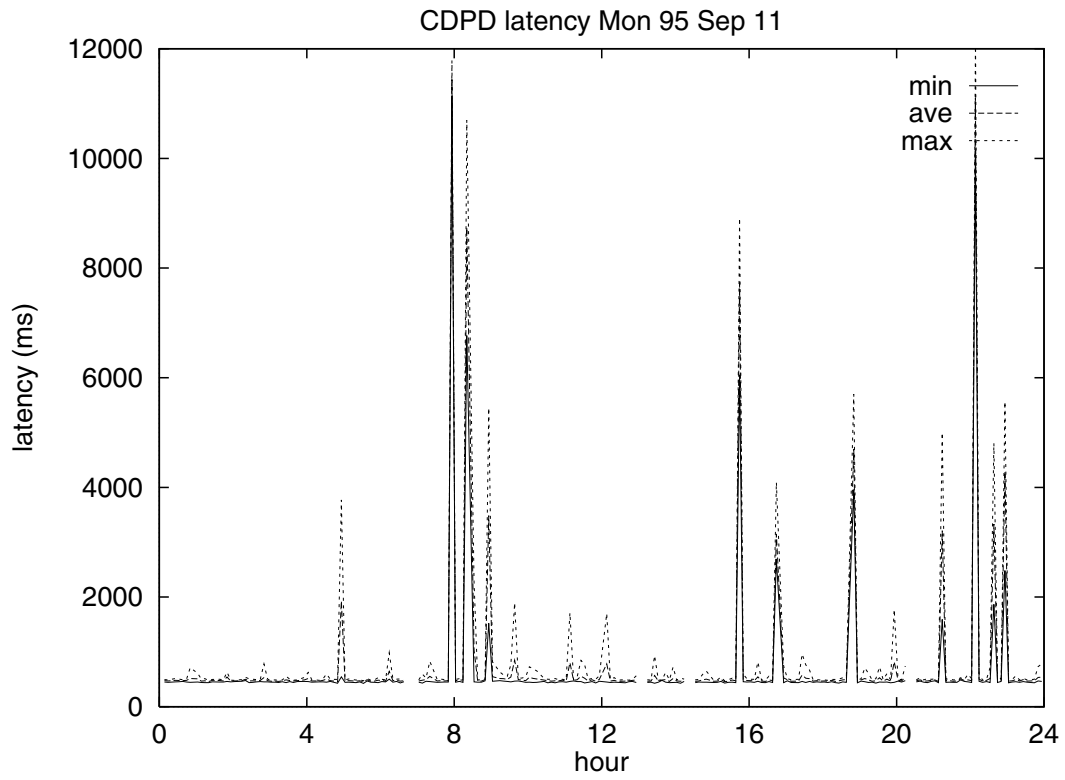
## **CDPD Latency Measurements**

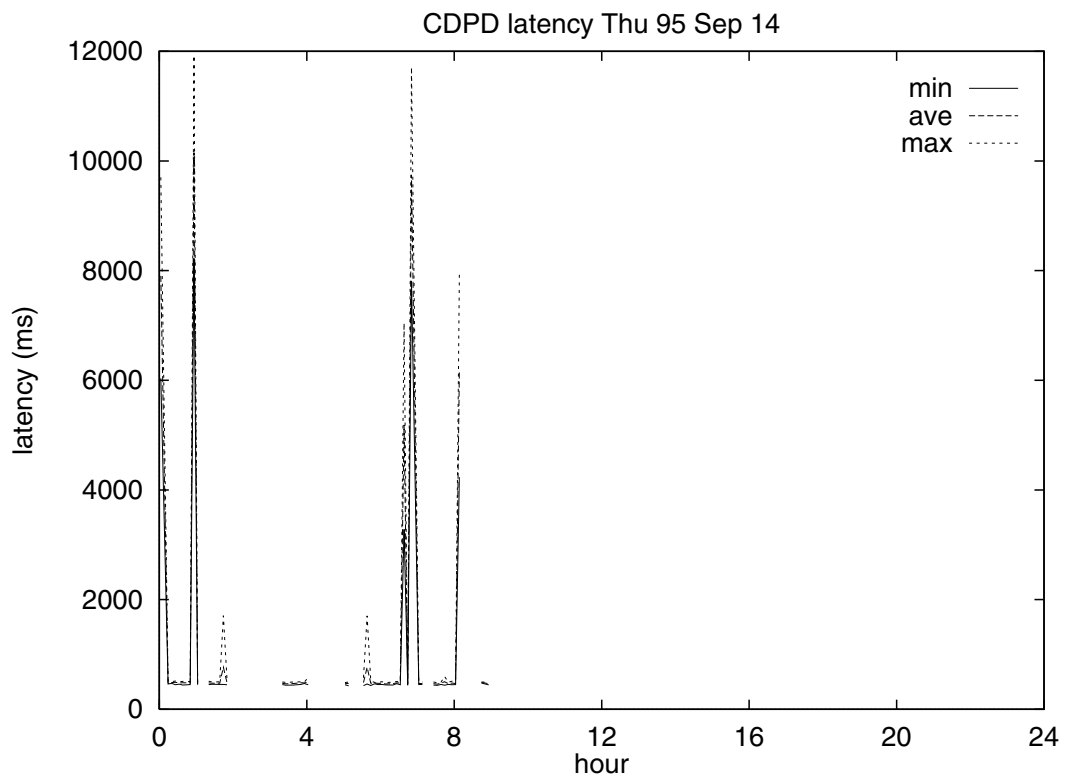
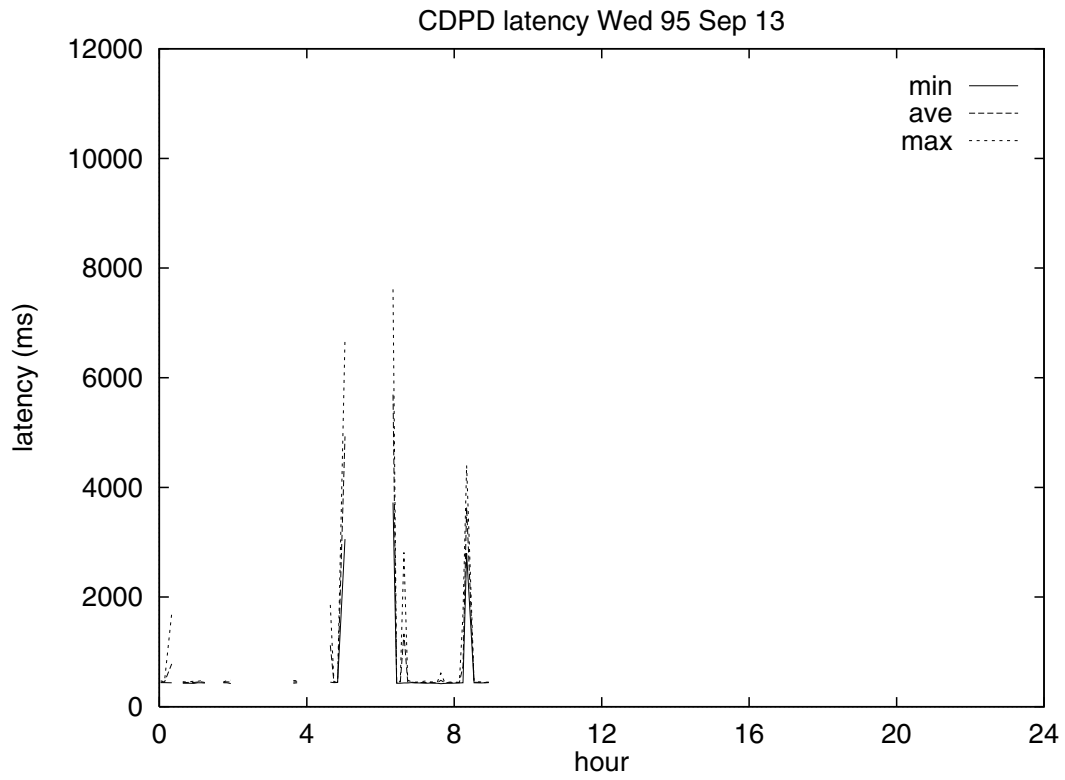


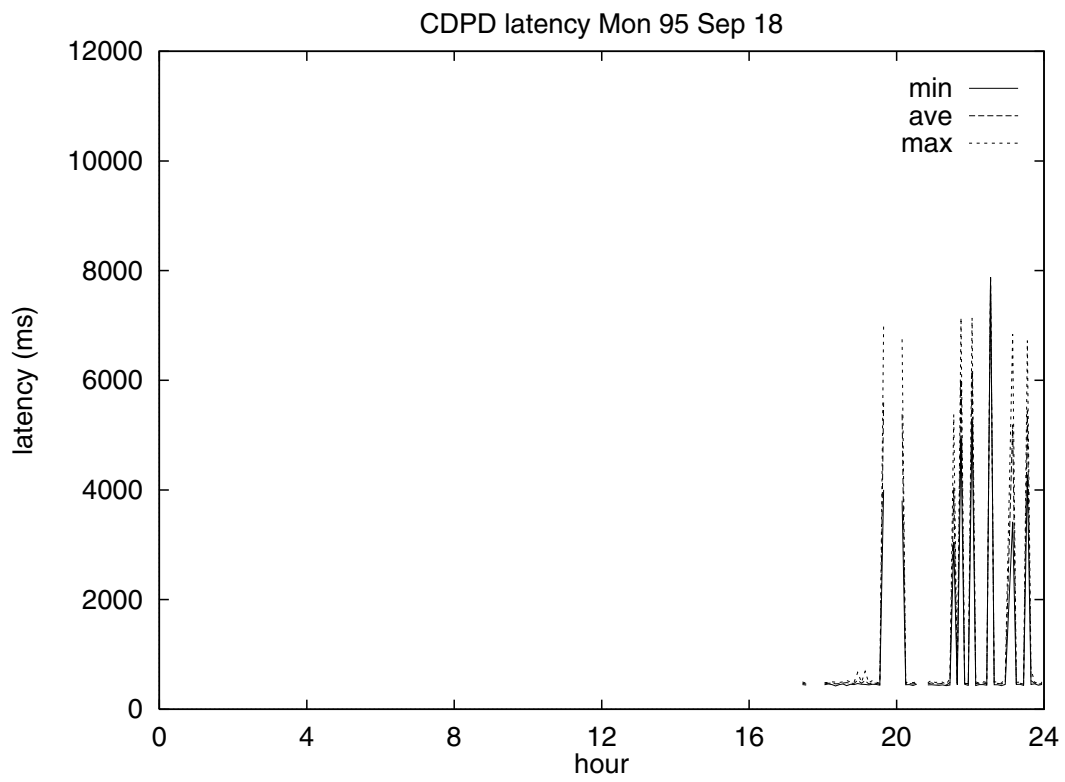
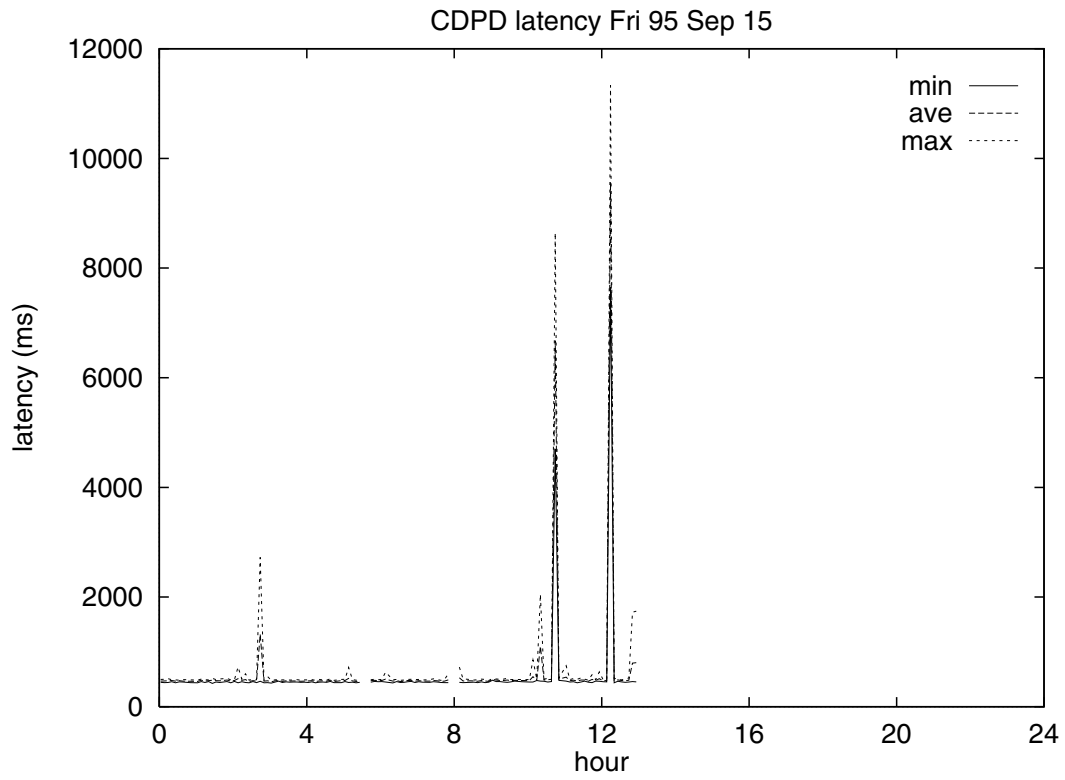




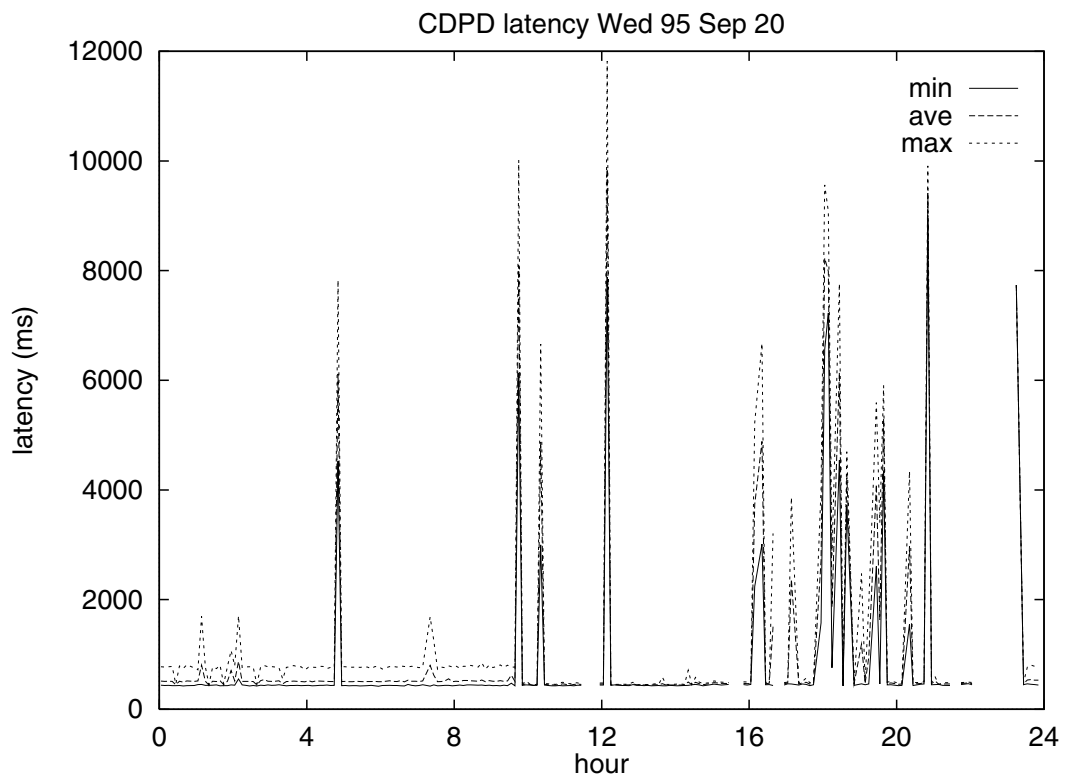
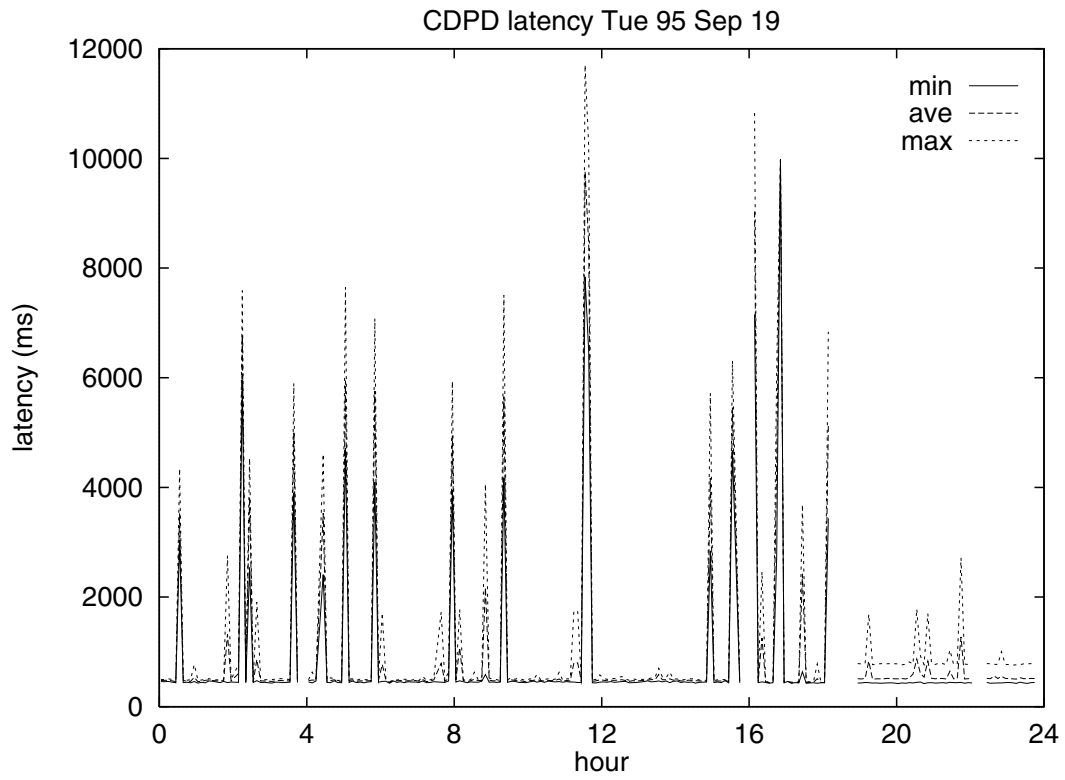


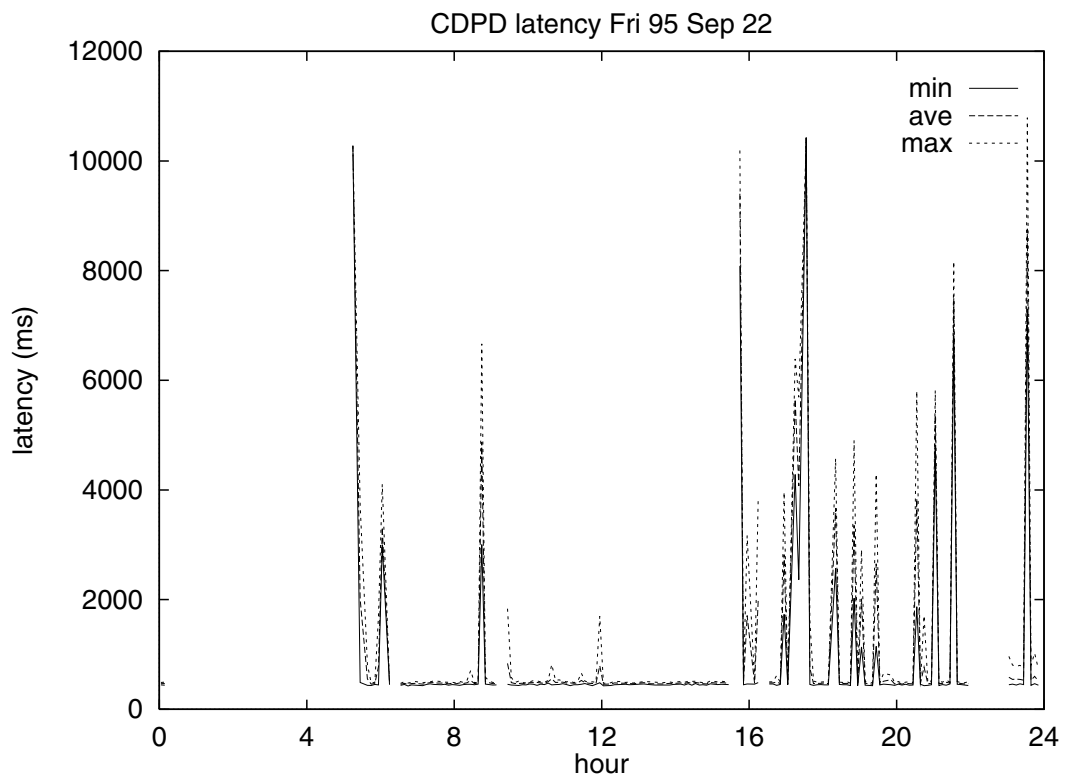
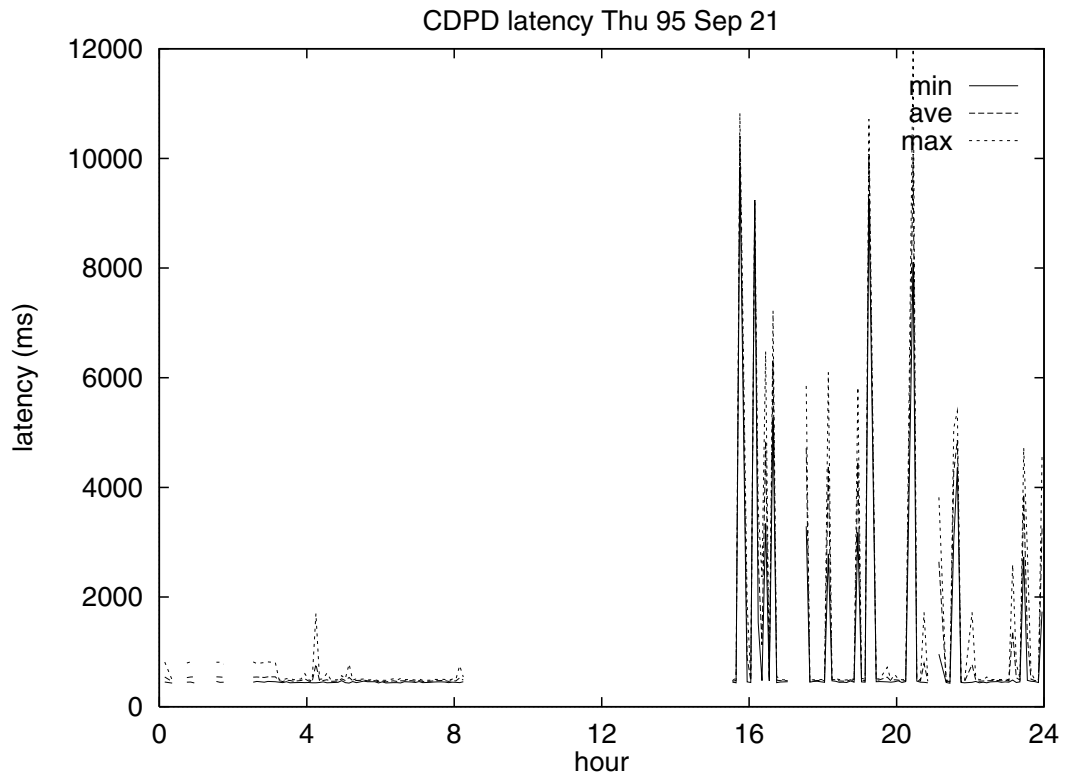


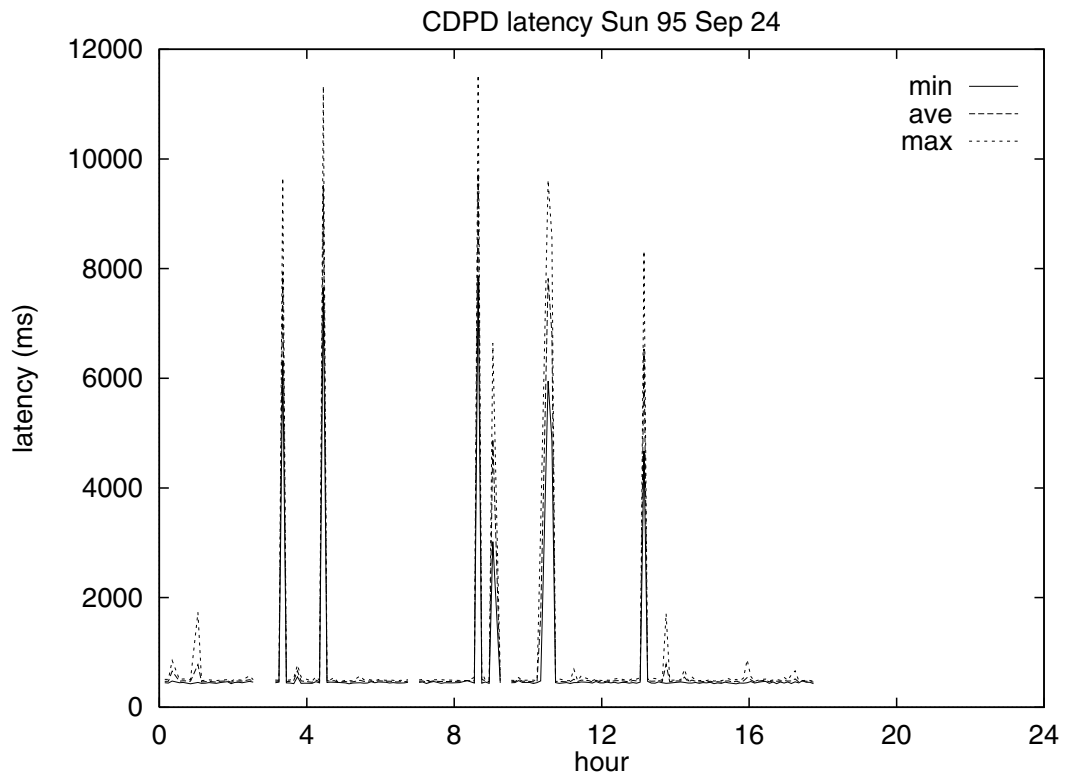
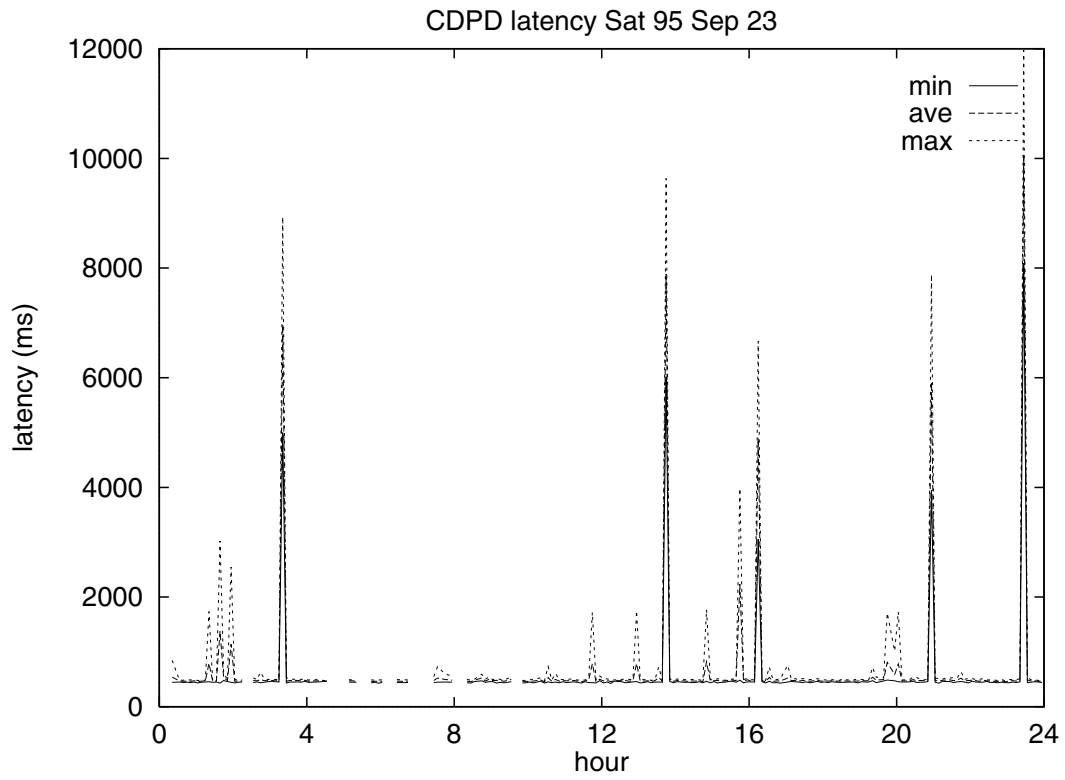












## **CDPD Throughput Measurements**

