ANALYSIS OF WIRELESS DATA NETWORK TRAFFIC

by

Michael Zhonghua Jiang

B. Eng. University of Science and Technology of China, 1995

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

School of Engineering Science

SIMON FRASER UNIVERSITY

April 2000

APPROVAL

Name: Michael Zhonghua Jiang

Degree: Master of Applied Science

Title of thesis: Analysis of Wireless Data Network Traffic

Examining Committee:

Chair: Dr. Mehrdad Saif

_____

Dr. R.H.S. Hardy, Senior Supervisor

_____

Dr. Ljiljana Trajkovic, Senior Supervisor

_____

Dr. Tiko Kameda, SFU Examiner

Date Approved: _____

Abstract

Recent research activities in the area of network traffic have demonstrated that traffic patterns in modern networks are statistically different from patterns generated by traditional traffic models. These new traffic patterns are long-range dependent. Researchers have shown that long-range dependency has considerable impacts on queueing performance, and that it is becoming a predominant factor in solving a number of packet traffic engineering problems.

The main objective of this thesis is to investigate traffic patterns in wireless data networks. By performing measurements, simulations, and analysis, our results demonstrate that wireless data traffic exhibits long-range dependent behavior, and, therefore, is statistically different from patterns generated by traditional traffic models. To evaluate the performance of wireless Cellular Digital Packet Data (CDPD) networks, we use Opnet tool to simulate the Telus (formerly BCTel) CDPD network. In our simulations, we use genuine traffic traces from Telus network. Our simulation results indicate that genuine traffic traces produce longer queues and, thus, require larger buffers in the deployed network's switching elements.

ACKNOWLEDGEMENTS

Nothing is ever done alone. Thus, I would like to thank certain people who helped me along the way with this project. Thanks to my supervisors, R. H. S. Hardy and Ljiljana Trajkovic, who, I feel, were responsible for advancing my knowledge in the area of wireless data network traffic analysis to a new and, hopefully, much higher levels than before. Thanks to Chao Cheng, who gave us great help by maintaining the software used in this project. Deserving of thanks as well is my wife Xin, for her constant support throughout the project, from the beginning to the end.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction

In the past decade, a large number of traffic measurements from deployed packet networks have been collected and analyzed. They include Ethernet local area networks (LAN's) [17], wide area networks (WAN's) [11], common channel signaling networks CCSN/SS7 [5], World Wide Web (WWW) [4], and variable bit rate (VBR) video over asynchronous transfer mode (ATM) [6]. The results were twofold:

- These studies demonstrated that it was possible to clearly distinguish between genuine packet network traffic and traffic generated by widely used mathematical models.
- In contrast to the traditional packet traffic models, aggregate packet streams are statistically self-similar, or fractal in nature. That is, realistic network traffic looks the same when measured over time scales ranging from milliseconds to minutes and hours.

Researchers have shown that long-range dependency has considerable impact on the queueing performance of wired networks [6, 11], and that it is a dominant characteristic for a number of packet traffic engineering problems. In this paper, we demonstrate the impacts of self-similarity on queueing performance in wireless data networks. An equivalent mathematical manifestation of the long-range dependence (LRD) is its property that the underlying traffic process is self-similar. A long-range dependent process is characterized by an autocorrelation function that decays as a power of the lag time, implying that the sum (over all lags) of the autocorrelations diverges [2]. This is why LRD is important for

network traffic analysis: even though the high-lag autocorrelations are individually small and negligible, their cumulative effect is important, and the traffic behavior is drastically different from that of traffic processes considered in past teletraffic research.

Our measurements, simulations, and analysis results demonstrate that wireless data traffic exhibits long-range dependent behavior. Therefore, wireless traffic is statistically different from traffic generated by traditional traffic models. To evaluate the performance of wireless Cellular Digital Packet Data (CDPD) networks, we use the Opnet (Opnet Technologies, formerly Mil. 3) to simulate the Telus (formerly BCTel) CDPD network. In our simulations, we use genuine traffic traces from Telus. Our simulation results indicate that genuine traffic produces longer queues and, thus, requires larger buffers in the network's switching elements than traffics generated by traditional models.

The rest of this thesis is organized as follows. In Section 1, we briefly describe the structure of a Cellular Digital Packet Data Network. Section 2 gives a brief description of long-range dependency. In Section 3, we briefly describe how the genuine wireless data trace was obtained and present the analysis results. Section 4 gives a detailed description of the Simulation Model used for the queueing analysis. This includes the network, node, and process models. In Section 5, we discuss the resulting traffic engineering implications. We conclude with Section 6 and a summary of our work.

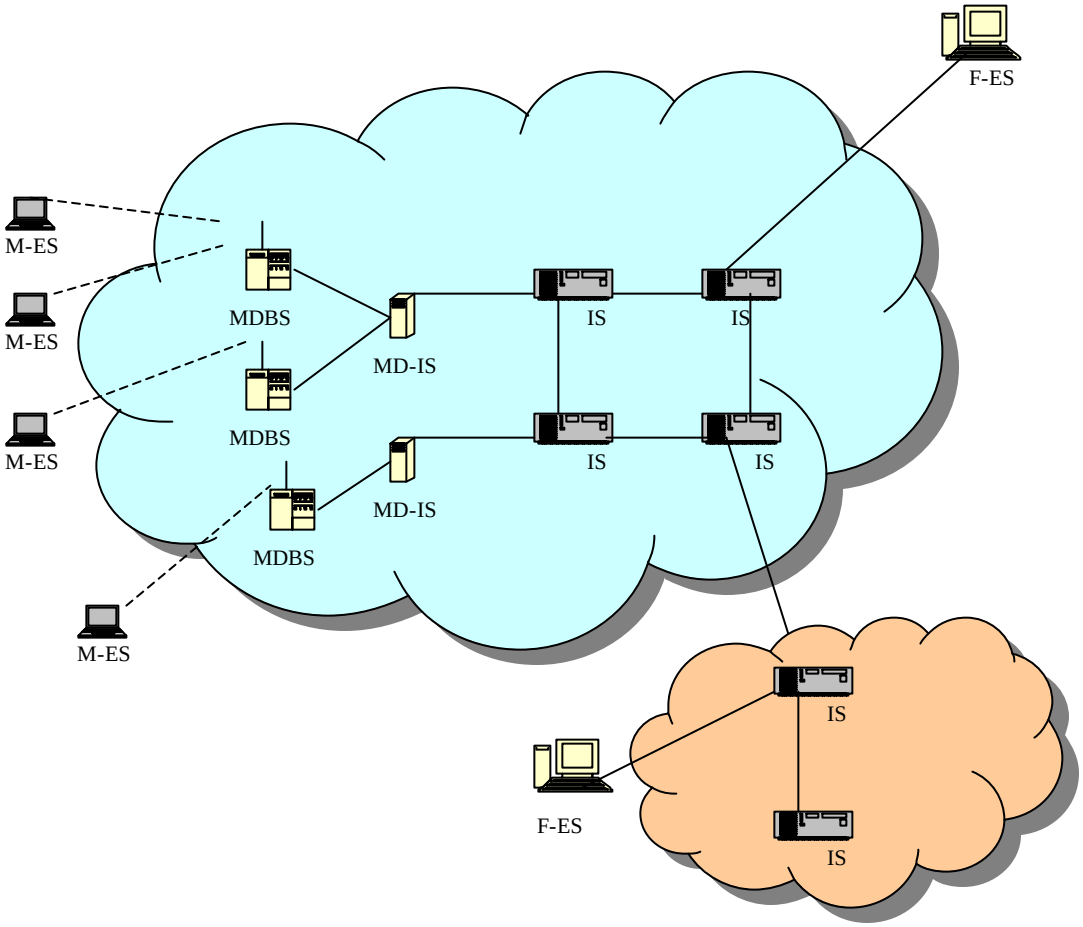
1. Overview of a CDPD network

Cellular Digital Packet Data (CDPD) is a wireless technology that provides packet switched data transfer service using the radio equipment and spectrum available in the existing analog mobile phone system (AMPS) based analog cellular networks. The CDPD System Specification Release 1.0 was originally published in 1993 by several cellular carriers including Ameritech, GTE, and McCaw (later acquired by AT&T). Later, these and other cellular companies jointly formed the CDPD Forum in 1994, and issued Release 1.1 in 1995. The Specification covers the following major areas: 1) CDPD communications architecture; 2) Key interfaces; 3) Protocol stacks; 4) Radio resource control; 5) Mobility management; 6) Accounting management; 7) Support services, and 8) Network management.

The following sub-sections provide a high-level discussion of the CDPD architecture, protocol stacks, and the Airlink Interface. The brief description of the communication architecture serves as background material for CDPD, while the sub-sections on protocol stack and Airlink Interface highlight the protocols and procedures that are the subject of our research.

1.1 CDPD communication architecture

The CDPD communication architecture is based on the OSI Basic Reference Model and it is only concerned with the lower three OSI layers. Fig. 1.1 depicts the network elements as defined in CDPD [1].



M-ES: Mobile End System
MDBS: Mobile Data Base Station (RF to landline relay)
MD-IS: Mobile Data Intermediate System (mobility management)
IS: Intermediate System (routers)
F-ES: Fixed End System

FIG. 1.1. OVERALL CDPD NETWORK ARCHITECTURE.

The purpose of the CDPD network is to allow data transmission between Mobile End Systems (M-ES) and Fixed End Systems (F-ES). As its name implies, an M-ES is a subscriber's equipment that may move between administrative domains, or between network areas. On the other hand, an F-ES is a stationary system that either provides data application services outside the direct control of a CDPD Network Service Provider (NSP), or it provides network support services administered and maintained by a CDPD NSP. A typical application involving an F-ES of the first type (known as an external F-ES) and an M-ES is a terminal inside a police cruiser sending a request to a database system for a driver license validation. Every time a police officer logs on the terminal, a message will be sent to the so-called internal F-ES that provides authentication and authorization services.

The Mobile Data Base Station (MDBS) communicates with a population of M-ES's over the Airlink Interface, providing both data link layer and physical layer functions to a set of paired radio channels within its cell boundary. In CDPD terminology, the channel stream from M-ES to MDBS is called the reverse channel, and the channel stream in the opposite direction is called forward channel. Data link layer functions include radio channel allocation, interoperation with cellular voice channel usage, and radio media access control. At the physical layer, the MDBS transmits and receives data modulated with the Gaussian Filtered Minimum Shift Keying (GMSK) scheme at a bit rate of 19.2 kilobits per second.

The Mobile Data Intermediate System (MD-IS) is a network layer entity that serves a domain area consisting of multiple cells. Each cell is controlled by a MDBS. The MD-IS is the only entity in the CDPD network that keeps track of the location of an M-ES, and it is thus capable of routing data to that M-ES regardless in which cell it is currently located. Within a CDPD network that has more than one MD-IS, such location awareness is made possible by connections between the MD-IS's, therefore allowing routing of data to and from M-ES's that are roaming.

Similar to the MD-IS, the Intermediate System (IS) is also a network layer entity that relays data to another MD-IS or IS. The major difference between the two is that the IS is not aware of the mobility of the M-ES's. IS's are normally off-the-shelf routers that represent the backbone of the CDPD network.

1.2 Protocol stacks and Airlink Interface

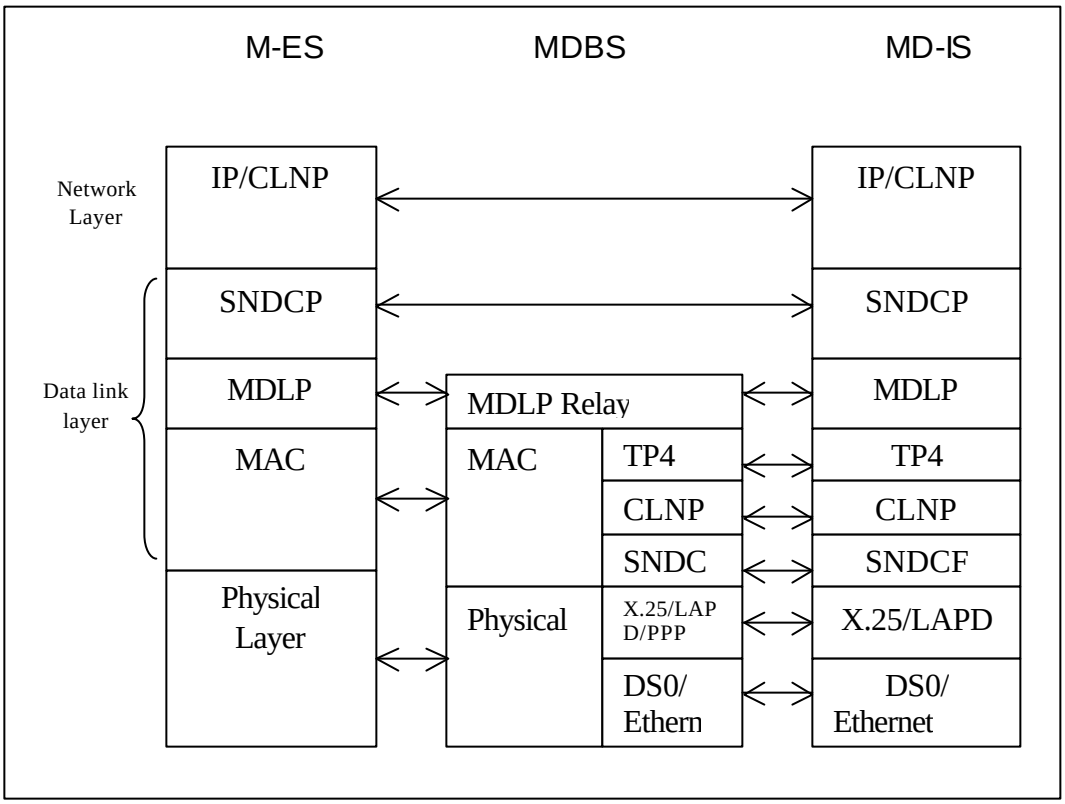


FIG. 1.2. AIRLINK PROTOCOL PROFILE.

Fig.1.2. shows that network layer protocol data units (NPDU's) or packets are transmitted across a mobile data link connection between the M-ES and the MD-IS using the Mobile Data Link Protocol (MDLP). The link layer PDU's (called MDLP frames) are sent out as exchanges between the MD-IS and MDBS, and as exchanges between the M-ES and MDBS. The MD-IS to MDBS communication is dictated by the particular sub-network protocol in use. In our simulation, we focus on the communication between the M-ES and MDBS, which use the MAC and physical protocols. The data transmission from M-ES to MDBS are described below:

- *Network Layer (IP/CLNP):* The Sub-Network-Dependent Convergence Protocol (SNDCP) receives the network layer data packet.
- *SNDCP*: With SNDCP, optional header compression is implemented on the header portion of the IP or CLNP packet, and optional V.42 bit data compression is implemented on the data and header portion of the packet. The packet is segmented based on the maximum frame size handled by the link layer entity and an SNDCP header is added to each segment. Encryption is performed on the data portion (not on the SNDCP header) and the segmented packet is forwarded to the MDLP entity.
- *MDLP:* The MDLP layer adds the MDLP header and forwards the frame to the MAC layer entity.
- *MAC*: With MAC, a sequence of frames is converted into a bit stream by inserting at least a single frame flag in between. The bit stream is blocked into consecutive sets of bits and the Reed Solomon encoding is performed. The Reed Solomon code used in CDPD MAC layer is RS(63,47) , with symbol size 6 bits.
- *Forward Channel:* The data and RS code bits (378 bits) are interleaved with an additional 42 bits containing forward synchronization word, decode status, and busy/idle flags to make up a 420-bit block.
- *Reverse Channel*: The data and RS code bits (378 bits) are interleaved with 7 continuity indicator bits to form 385-bit blocks. Multiple blocks (full-duplex M-ES) or a single block (half-duplex M-ES) is prefixed with a

dotting sequence and a reverse synchronization word to form the reverse burst. Each bit is modulated as a symbol of a Gaussian Minimum Shift Keying (GMSK) waveform.

1.3 Medium access control layer

The medium access control (MAC) layer is responsible for the frame transfer between the M-ES and the MDBS. The protocol defined in the MAC layer performs the following three functions: (i) data encapsulation, (ii) medium access management on the reverse channel, and (iii) channel stream timing and synchronization. Tasks associated with data encapsulation include frame boundary delimitation, data transparency, frame synchronization, and error detection and correction (Reed-Solomon encoding). Medium access management on the reverse channel is accomplished by using the slotted non-persistent Digital Sense Multiple Access with Collision Detection (DSMA/CD) scheme. Finally, the last function is provided by transmitting synchronization and timing indicators in the forward channel to allow M-ES's to synchronize to a master microslot clock before starting a transmission.

1.4 Digital sense multiple access with collision detection

In CDPD, the forward channel is contentionless and, therefore, it is always available for the MDBS to broadcast data to a population of M-ES's. In contrast, multiple M-ES's have to compete with each other to access the reverse channel for data transmission to the MDBS. Therefore, the Carrier Sense Multiple Access

with Collision Detection (CSMA/CD) scheme is employed to control the reverse channel access by notifying the M-ES's when the channel is available. Such notification is provided periodically in the forward channel with two types of status flags: Channel Busy/Idle Status and Block Decode Status. The Channel Busy/Idle Status indicates whether the reverse channel is busy or idle. When the channel is busy, an M-ES attempting to transmit has to defer until the channel becomes idle again. This status flag thus provides a mechanism for collision avoidance. The other status flag, the Block Decode Status, indicates if the previous FEC block transmitted by an M-ES was successfully decoded by the MDBS. As a result, this status flag serves to provide collision detection. This scheme is called Digital Sense Multiple Access with Collision Detection (DSMA/CD).

2. Overview of self-similarity

2.1 Introduction to self-similarity

Kolmogorov first introduced self-similar processes in 1941 within a theoretical context. Statisticians do not seem to have been aware of the existence or statistical relevance of such processes, until Mandelbrot and his co-workers introduced them into statistics. The basic idea of self-similarity is much older. Mandelbrot referred to, for example, Leonardo da Vinci's drawings of turbulent flows that exhibit coexistent "eddies" of all sizes and thus possess self-similarity. A geometric shape is called self-similar in a deterministic way, if the same geometric structures are observed, independently of the distance from which one looks at the shape. In the context of stochastic processes, self-similarity is defined in terms of the distribution of the process [2]:

Definition 1 *Let* Y_t *be a stochastic process with continuous time parameter t.* Y_t *is called self-similar with self-similarity parameter H, if for any positive stretching factor c, the rescaled process with time scale ct,* $c^{-H}Y_{ct}$*, is equal in distribution to the original process* Y_t*.*

This means that, for any sequence of time points $t_1, ..., t_k$, and any positive constant c, $c^{-H}(Y_{ct_1}, Y_{ct_2}, ..., Y_{ct_k})$ has the same distribution as $(Y_{t_1}, Y_{t_2}, ..., Y_{t_k})$. Thus, typical sample paths of a self-similar process look qualitatively the same,

irrespective of the distance from which we look at them. In contrast to deterministic self-similarity, it does not mean that the same picture repeats itself exactly as we get closer. It is, rather, the general impression that remains the same.

2.2 Self-similarity in teletraffic engineering

Interest in self-similar arrival processes was stimulated by measurements of Ethernet traffic in Bellcore [8, 9]. The measurements indicate that the Ethernet traffic seems to look the same over the larger time scales (minutes, hours) and over the smaller time scales (seconds, milliseconds); hence the term self-similar traffic. Since then, this feature has been observed in several experiments, for example, in TCP [11], Motion Pictures Experts Group (MPEG) video traffic [6], and WWW traffic [4].

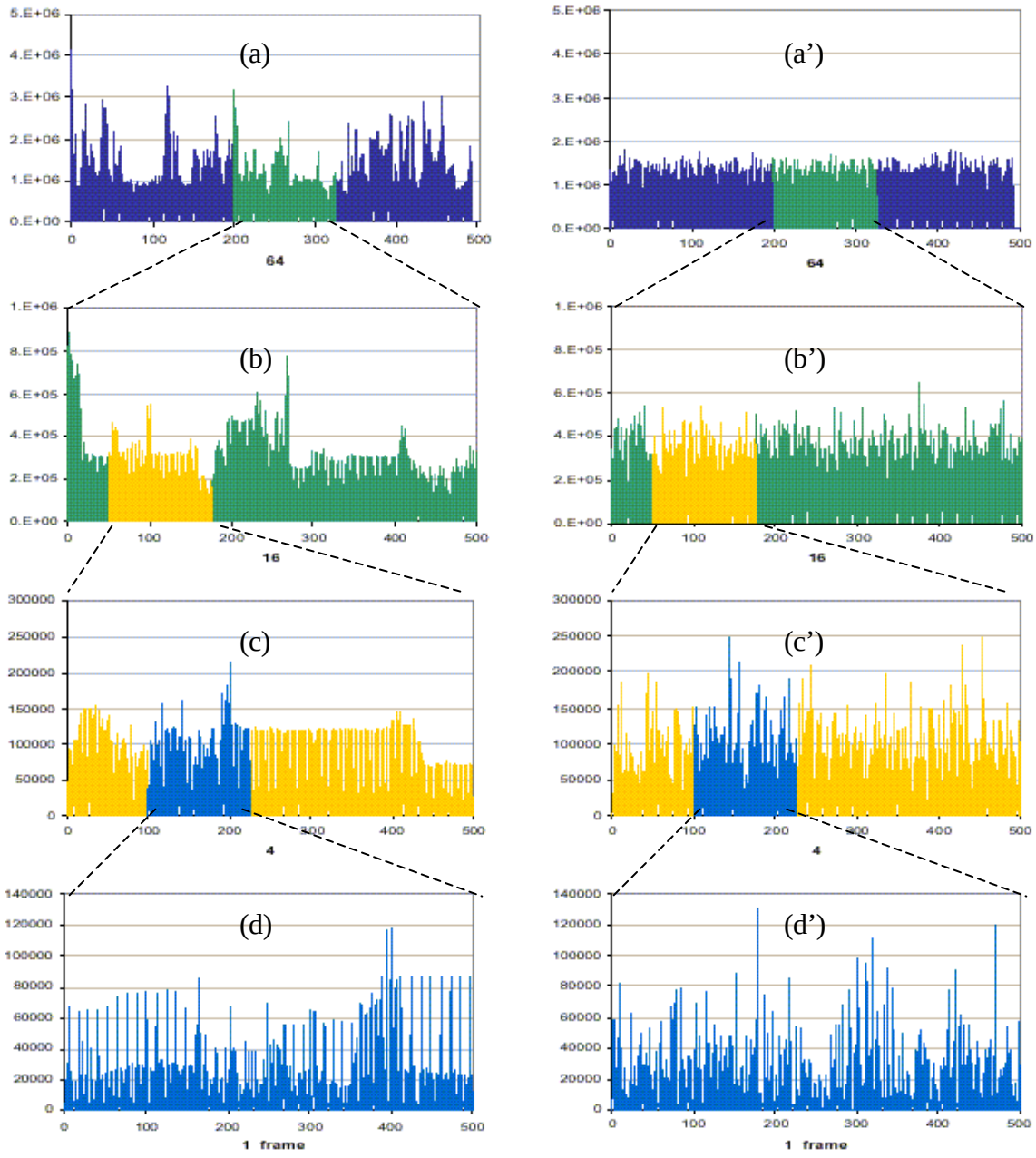


FIG. 2.1. PICTORIAL "PROOF" OF SELF-SIMILARITY.

Fig. 2.1 shows the comparison of Star Wars MPEG traffic and a synthetic traffic from an appropriately chosen Poisson model over four time scales (courtesy of

Velibor Markovski). Scenes (a)-(d) depict a sequence of simple plots of the packet numbers for four different choices of time units for the MPEG traffic trace captured by compressing the Star Wars movie. Starting with a time unit of 64 frames (Fig. 2.1(a)), each subsequent plot is derived from the previous one by increasing the time resolution by a factor of 4 and by concentrating on a randomly chosen subinterval (indicated by different shades of gray). The finest time scale is 1 frame. From Fig. 2.1, we can see that all plots in the left column are "similar", that is, MPEG traffic seems to look the same over the whole spectrum of time scales. More importantly, notice the absence of a fixed length of a "burst". We can observe different burst lengths on different time scales. This time scale-invariant feature of the MPEG video traffic differs from both conventional telephone traffic and from stochastic models for packet traffic. For comparison, the synthetic traffic generated from a comparable (same average packet size and same arrival rate) Poisson process was depicted by the sequence of plots (a')-(d') in Fig. 2.1. This sequence was obtained in the same way as the sequence on Fig. 2.1 (a)-(d). Fig. 2.1 provides a simple method for distinguishing clearly between the measured MPEG traffic data and the traffic generated by the Poisson traffic model. It suggests that more sophisticated traffic models that can capture self-similar characteristic of real network traffic need be used in the prediction of network performance.

The following definition of self-similarity for stochastic processes is widely adopted in teletraffic engineering [11]:

Definition *2. Assume* X_k *to be a wide-sense stationary process with mean* $E(X_k) = \overline{X}$ *and autocorrelation function* $r(i) = C(i) \big/ Var(X_k)$. *Where C(i) is the covariance* of X_k. *Consider next processes* $X_k^{(m)} (m = 1,2,...)$ *that are constructed from* X_k *by averaging over non-overlapping blocks of size* m, *i.e.,* $X_k^{(m)} = \left(\sum_{n=0}^{m-1} X_{km+n} \right) / m$. *The processes* $X_k^{(m)}$ *are also wide-sense stationary, with mean* $\overline{X}$ *and autocorrelation function* $r^{(m)}(i)$. *The process* X_k *is called (asymptotically) second-order self-similar if* $r^{(m)}(i) = r(i)$, *(for* $m, i \to \infty$*).*

Self-similarity manifests itself in a number of equivalent ways:

- Slowly decaying variances: $Var\left\{X_k^{(m)}\right\} \sim m^{-b}$ if $m \to \infty$, where $0 < b < 1$
- A slowly decaying autocorrelation function: $r(i) \sim i^{-b}$ for $i \to \infty$.

Such a process differs from the traffic generated by traditional Poisson models used for modeling voice traffic that has:

- The variance $Var\left\{X_k^{(m)}\right\} \sim m^{-1}$
- $r(i)$ decays exponentially fast, i.e., $r(i) \sim g^{-i}$ for large i and $g < 1$
- The process behaves like (second-order) pure noise for large m, i.e., ($r^{(m)}(i) \to 0$ $(i \neq 0)$ if $m \to \infty$).

An important parameter of self-similarity is the Hurst parameter $H = 1 - b/2$. Self-similarity is implied by $0.5 < H < 1$.

2.3 Tools used to access the self-similar nature of a given trace [15, 17]

- Visual inspection: plots of samples from processes $X_k^{(m)}$ for a wide range of values of m look similar for fractal traffic, while those for short-range-dependent traffic "flatten out" when m gets large;
- The R/S plot or pox plot: plotting $\log[R(n)/S(n)]$ versus $\log(n)$ for various subsets of the available data allows one to determine, by linear regression, the Hurst parameter H. Here *R(n)* is the *adjusted range of X* in the time interval 0 to *n*, *S* is the *sample standard deviation*.
- The variance-time plot: of $\log(\text{var}\{X_k^{(m)}\})$ versus $\log(m)$ allows estimation of $b = 2(1-H)$;
- Whittle's estimator: An estimator based on a maximum likelihood technique due to Whittle, given by Beran [3]. Let $f(l;H)$ denote the power spectrum of X when normalized to have variance 1, and $I(l)$ the periodogram (i.e., power spectrum estimated using a Fourier transform) of x_t. To estimate H, find $\hat{H}$ that minimizes:

$$g(\hat{H}) = \int_{-p}^{p} \frac{I(l)}{f(l;H)} dl$$

If $\{x_t\}$ has length n, then the above integral is readily converted to a discrete summation over the frequencies $l = \frac{2p}{n}, \frac{4p}{n}, ..., 2p$.

- Wavelet analysis (courtesy of Dr. Fred Xue): A property of an LRD process that the spectrum $\Gamma(v)$ takes the following form for a range of frequencies u close to 0:

$$\Gamma(v) \sim c_f |v|^{1-2H}, \; v \to 0,$$

where $c_f > 0$ and $0.5 < H < 1$.

Wavelet analysis is a useful tool for analyzing the scale-dependent properties of data via the coefficients of wavelet decomposition. Since self-similar processes possess similar statistical properties over a large range of scales, the wavelet analysis appears to be a natural analysis tool to test the fractal behavior of a long-range dependent process.

A wavelet-based estimator suggested in [14] is based on a spectral estimator designed by performing a time average of the $|d_x(j,k)|^2$ at a given scale:

$$\hat{\Gamma}_x(2^{-j}v_0) = \frac{1}{n_j}\sum_k |d_x(j,k)|^2,$$

where n_j is the number of wavelet coefficients at octave *j*. The coefficient $|d_x(j,k)|^2$ measures the amount of energy in the analyzed signal about the time instant $2^j k$ and frequency $2^{-j}n_0$. An estimator for the Hurst parameter *H* can be found from a simple linear regression of $\log_2(\hat{\Gamma}(2^{-j}v_0))$ on the scale level *j*:

$$\log_2(\hat{\Gamma}_x(2^{-j}v_0)) = (2\hat{H} - 1)j + \hat{c}.$$

In this project, we will use the R/S and variance-time plots to analyze the wireless data network traffic trace. These two methods will be discussed in detail in Section 3.3.

2.4 Impacts of self-similarity on teletraffic engineering

Experimental evidence for self-similarity in various types of data network traffic is already overwhelming and continues to grow. Clearly, the phenomenon can no longer be ignored in teletraffic engineering. So far, simulations and analytical studies have shown that it may have a considerable impact on network performance that could not be predicted by the traditional short-range-dependent models. However, its implications on, for example, connection admission policies or network dimensioning are still not well understood. In particular, a very practical question is how queueing and traffic shaping (such as leaky bucket used for policing in an ATM node) affect self-similarity [9, 11, 12, 13, 17, and 18].

3. Traffic measurements

3.1 The traffic monitor

The monitoring system used to collect data for the present study was built by AirLink Communications, Inc. The AirLink CDPD Protocol Analyzer is an effective tool used to isolate problems and to measure the performance of the Airlink Interface. It monitors the CDPD traffic between a base station and mobile end systems, allowing users to quickly diagnose problems. For each packet seen on the RF channel, the Analyzer records a timestamp (accurate to within 10 ms), the packet length, the status of the Airlink Interface, and the encrypted data in each packet. The accuracy of our timestamps is lower than the time stamp achieved in measuring the Ethernet traffic trace from Bellcore (100 μs). This is due to the limitations of the CDPD Protocol Analyzer. Nevertheless, as we will show in Section 3.3, the impact of the lower-accuracy on the statistic analysis of the data is tolerable, even though taking the measurements over larger time-scales (10 ms) has imposed some smoothing.

3.2 The network environment of BCTel's CDPD network

BCTel's mobile data network is a typical commercial CDPD network, where base stations are located on top of high-rise buildings, and the users are mostly police and fire departments. The traffic used for our analysis was collected on the reverse channel in a CDPD cell.

A snapshot of the network configuration at the time the data set was collected is given in Fig. 3.1. During our measurement, there were about 10 mobile end

systems registered in that cell. This network cell consisted of one MDBS connected to the CDPD backbone network, implying that almost all the traffic on the Airlink in this cell was visible from our monitoring point, except the traffic generated by the hidden terminals.

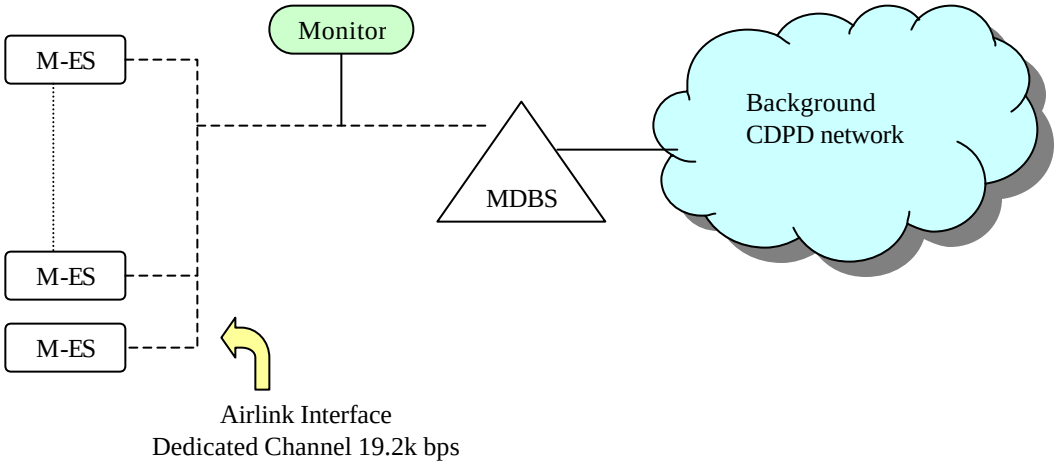


FIG. 3.1. CDPD NETWORK ENVIRONMENT FOR OUR MEASUREMENT.

We obtained three sets of data from BCTel. Due to the limitation of the resources, only one set of traffic data was long enough for statistical analysis. Table 3.1 gives a summary description of the traffic data analyzed in Section 3.3.

Table 1. BCTel's CDPD network traffic Trace.

BCTel's CDPD network traffic Trace				
Measurement period	Total number of bytes	Total number of MAC packets	Network utilization	Traffic load
Start of Trace: 14:56:37.56, 06-12-98 **Start of Trace: 15:24:46.88, 06-12-98**	152439	1281	5.29%	1016bps

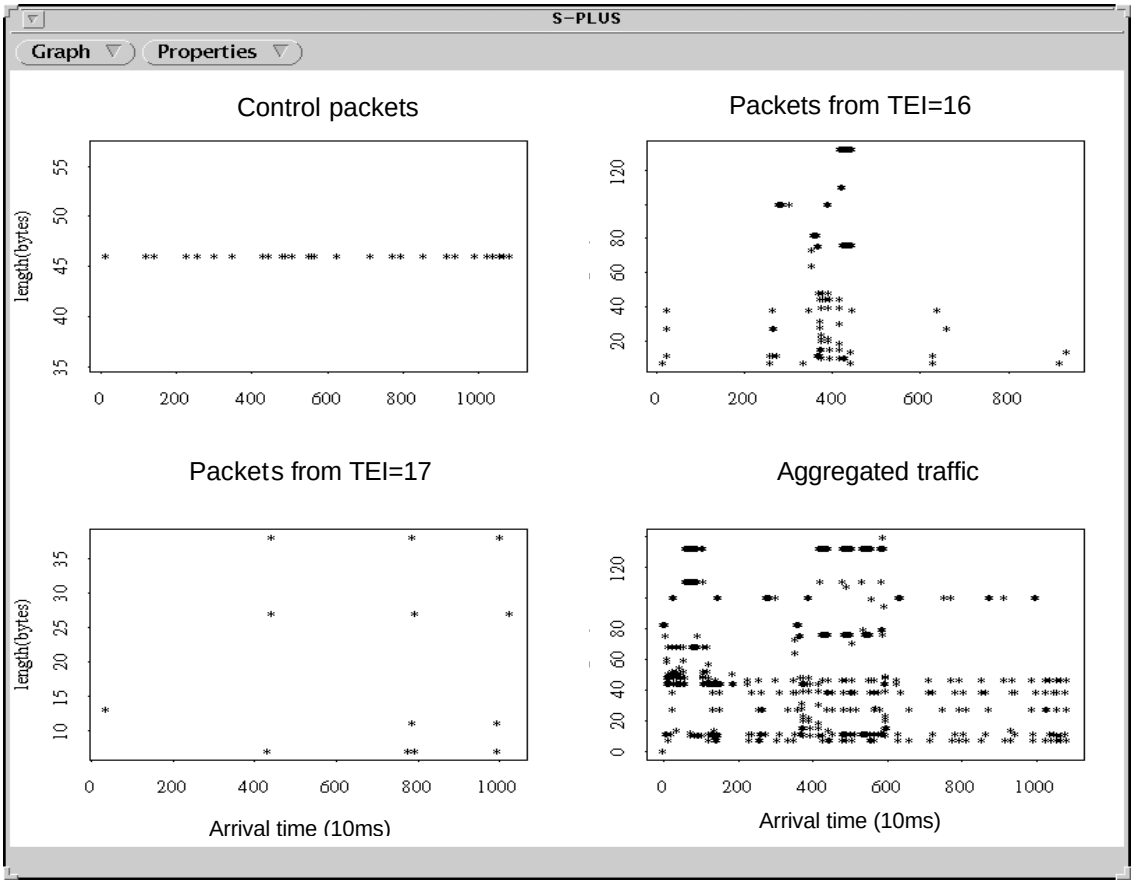


FIG. 3.2. MEASURED CDPD NETWORK TRAFFIC TRACES.

Traffic data were collected on the BCTel's CDPD network in downtown Vancouver area from 14:56:37.66 to 15:24:46.88 on June 12, 1998. Fig. 3.2 shows the packet arrival process. During the period this data was collected on 10 M-ES's appeared in this cell, some M-ES's were more and some were less active. The M-ES with Terminal Equipment Identification (TEI) number 16 is an example of an active system. The M-ES with TEI = 17 is less active. In Section 3.3, we will use the aggregated traffic for the statistical analysis. From the viewpoint of the MDBS, the aggregated traffic is the total input traffic.

3.3. Statistical analysis of the measured data

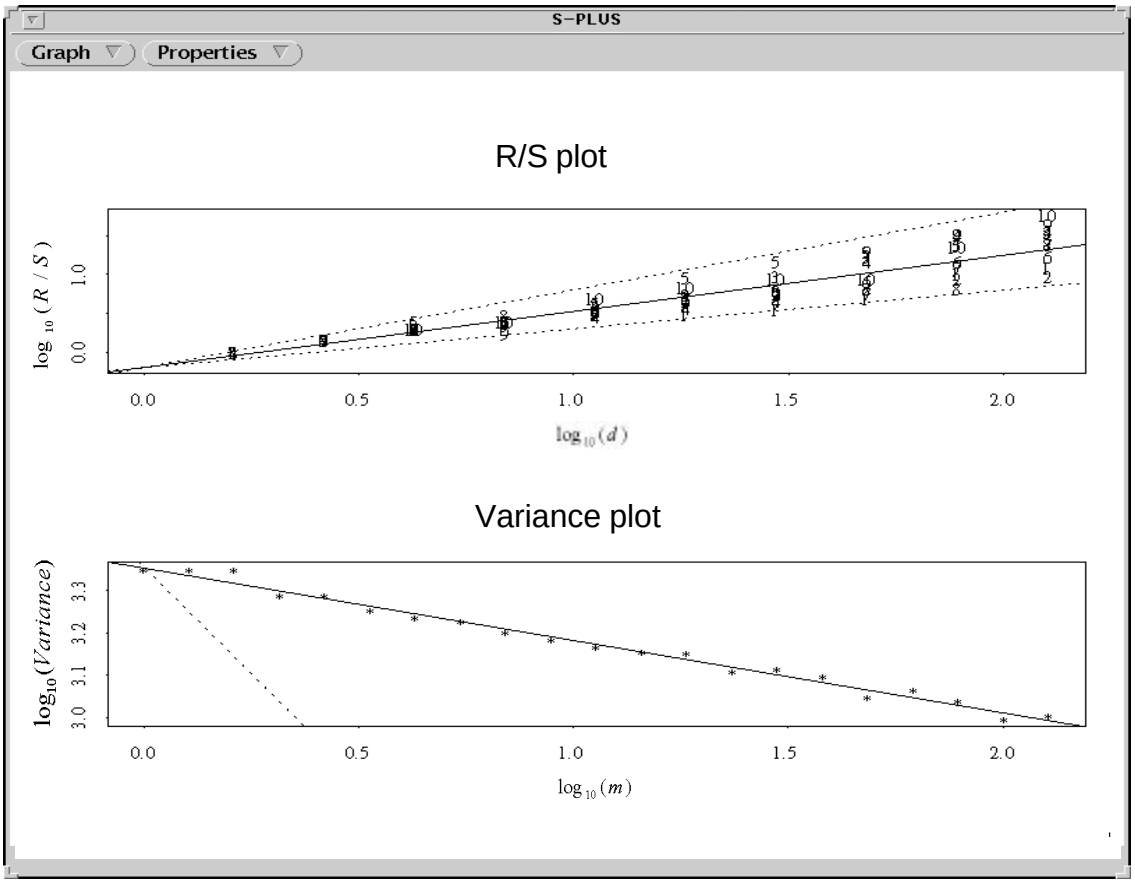


FIG. 3.3. GRAPHICAL METHODS FOR CHECKING THE SELF-SIMILARITY.

In this project, we used two graphical methods to test the self-similarity of the wireless data traffic trace: R/S and variance-time plots. We now give an overview of these two methods [2, 9].

Graphical R/S analysis consists of taking logarithmically spaced values of n (starting with $n \approx 10$), and plotting $\log(R(n)/S(n))$ versus $\log(n)$ results in the

rescaled adjusted range plot (also called the pox diagram of R/S [8]). When the Hurst parameter H is well defined, a typical rescaled adjusted range plot starts with a transient zone representing the nature of short-range dependence in the sample, but eventually settles down and fluctuates in a straight zone of a certain slope. Graphical R/S analysis is used to determine whether such asymptotic behavior appears in the data. To affirm self-similarity, the zone should be located between two lines with slopes ½ and 1, respectively. For practical purpose, the most useful feature of the R/S analysis is its relative robustness against changes of the marginal distribution.

As an illustration of the usefulness of the graphical tools for detecting self-similarity in our measured trace, Fig. 3.3 depicts the plot of R/S and the variance-time curve. The Hurst parameter can be estimated directly from the corresponding R/S plot. One can see that the value of the asymptotic slope of the R/S plot is clearly between 0.5 and 1 (lower and upper dotted lines, respectively), with a simple least-squares fit giving $\hat{H} \approx 0.80$.

The so-called variance-time plots are obtained by plotting $\log(\mathrm{var}(X^m))$ vs. $\log(m)$ ("time") and by fitting a simple least squares line through the points in the plane, while ignoring the small values for m. If the estimate $\hat{b}$ of the asymptotic slope is between –1 and 0, the process is self-similar. The estimate for the degree of self-similarity is given by $\hat{H} = 1 - \hat{b}/2$.

In Fig. 3.3, the variance-time curve, which has been normalized by the corresponding sample variance, shows an asymptotic slope that is distinctly different from –1 (dotted line) and is estimated to be about – 0.2, resulting in an estimate $\hat{H}$ of the Hurst parameter H of $\hat{H} \approx 0.90$.

These two graphical methods suggest that the traffic sequence is self-similar with self-similarity parameter $\hat{H} \geq 0.80$. According to M. Roughan and D. Veitch [15], if $H \geq 0.6$, the data trace can be affirmed as self-similar even if the trace has some degree of non-stationarity. Thus, our estimate of H demonstrates that wireless data traffic tends to have self-similar behavior, and, therefore, is statistically different from traffic generated by traditional traffic models.

4. Simulation models

While approaches described in Section 3 convincingly establish the presence of LRD in the genuine CDPD network traffic trace (see Fig. 3.3), its significance to queueing performance and traffic engineering for the CDPD network may not be obvious. For example, it has been argued by some that LRD has no practical impact and need not be incorporated into performance models.

In this section, we give a detailed description of the network model we used to investigate the impact of LRD on CDPD network performance. At first, we introduce simulation tool used for simulations. In this project, we used OPNET Modeler for modeling and simulation of CDPD networks. Its object-oriented modeling approach and graphical editors mirror the structure of actual networks and network components, and, hence, our model intuitively maps to BCTel Mobility's system.

To provide useful data, network models must combine accurate descriptions of topology, data flow, and control flow. OPNET utilizes separate model formats for each of these three model types.

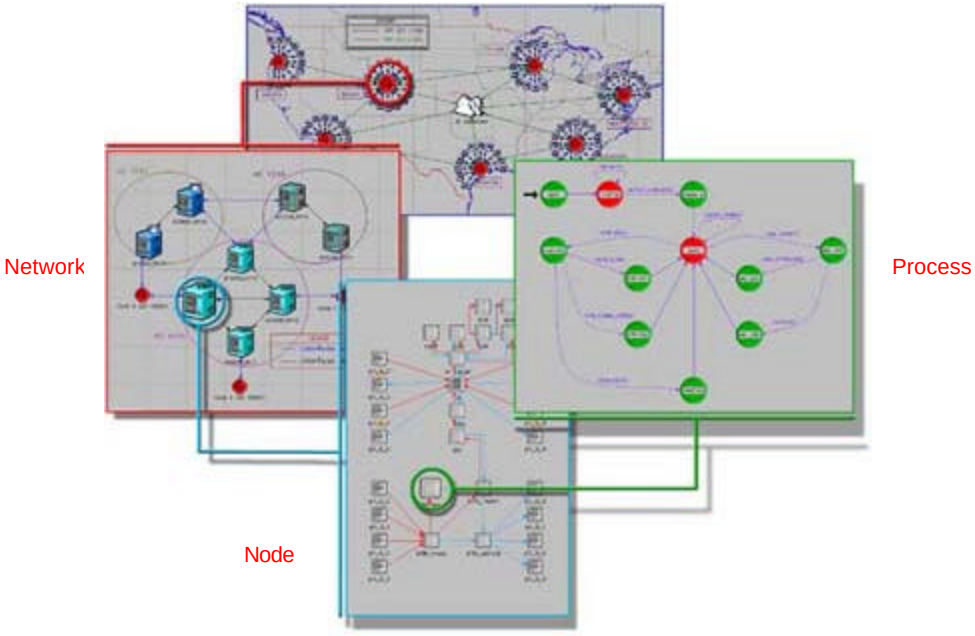


FIG. 4.1. HIERARCHICAL STRUCTURE OF AN OPNET MODEL.

Three layers form the hierarchical structure of an OPNET model: OPNET Network Layer Models define the position and interconnection of communicating entities, or nodes. Each node is described by OPNET Node Model, which typically depicts the interrelation of processes, protocols, and subsystems. The functionality of each Node is defined by an OPNET Process Model. The following are the phases of the modeling and simulation cycle of OPNET:

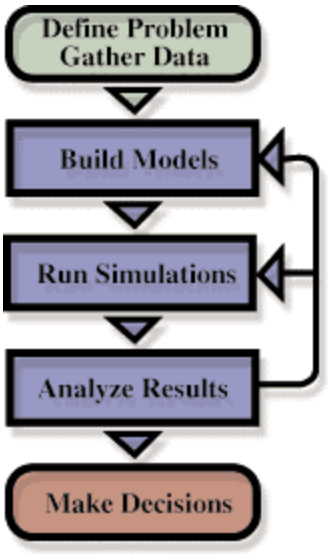


FIG. 4.2. OPNET MODELING AND SIMULATION CYCLE.

- **Model Building and Configuration**
  - Network Editor: defines or changes network topology models
  - Node Editor: defines or changes data flow (systems architecture) models
  - Process Editor: defines or changes control flow (behavioral logic) models.
- **Running Simulations**
  - Simulation Tool: defines and runs simulation studies using models constructed with OPNET Editors
  - Interactive Debugging Tool: interacts with running simulations.
- **Analyzing Results**
  - Analysis Tool: displays and compares statistical results.

The topology of our CDPD network is shown in Fig. 4.3. Only the Media Access Control (MAC) layer of the CDPD protocol is simulated.

4.1 CDPD network model

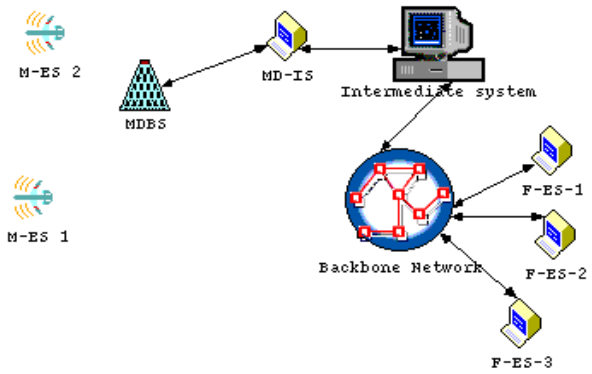


FIG. 4.3. CDPD NETWORK MODEL.

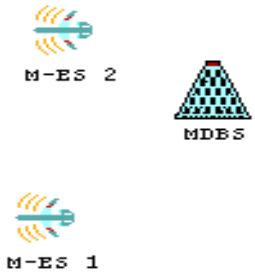


FIG. 4.4. NETWORK MODEL FOR THE AIRLINK INTERFACE.

The two M-ES's send packets to the MDBS via the reverse channel at a center frequency of 825 MHz. They receive packets from the MDBS via the forward channel at a center frequency of 870 MHz.

In the network model of the Airlink Interface shown in Fig. 4.4, the two M-ES compete for the bandwidth (825 MHz ~ 825 MHz + 30 KHz). Each M-ES generates data packets according to the genuine trace collected from the

operational CDPD network. Other traditional traffic models are also available from the OPNET libraries, such as Poisson and Gaussian models. The MDBS arbitrates activities on the channels it hosts at the MAC sub-layer, much like an Ethernet hub.

4.2 Node models

The M-ES node model shown in Fig. 4.5 is composed of a radio receiver and a radio transmitter, both working at 19.2 Kbps. The center frequency of the receiver (forward channel) is 870 MHz, with a bandwidth of 30 KHz. The center frequency of the transmitter (reverse channel) is 825 MHz, with a bandwidth of 30 KHz.

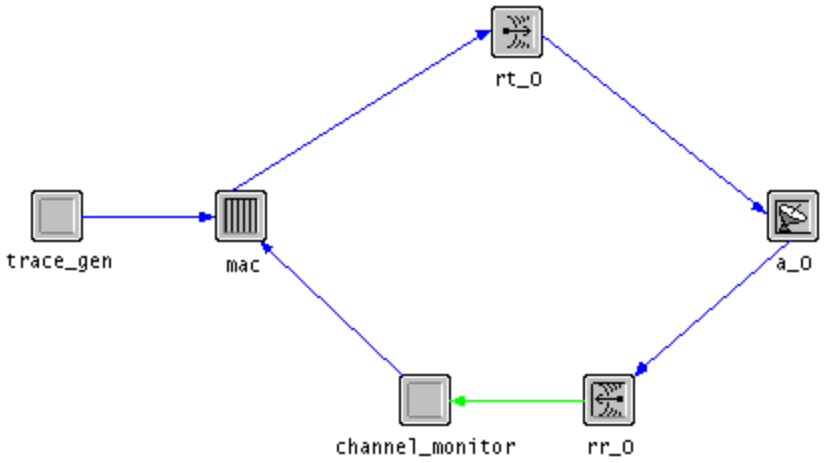


FIG. 4.5. NODE MODEL FOR M-ES.

The radio transmitter is operating at a center frequency of 825 MHz. The radio receiver is operating at a center frequency of 870 MHz. Both have bandwidth of 30 KHz.

The processor (mac) includes the digital sense multiple access (DSMA) logic. It retrieves the information about the reverse channel from the forward channel (*busy/idle* and *decode status* flags). If the reverse channel is busy or a collision occurs, the M-ES will back off for a random time period and will try to retransmit again. The generator generates packets according to our genuine traffic trace. It can also use the traditional traffic models supported by OPNET libraries. The M-ES model also destroys the packets it has received.

The MDBS model shown in Fig. 4.6 is more complex. It is connected to the external network (the wired part of a CDPD network) with a pair of transmitter and receiver. It receives packets from the M-ES via a radio receiver operating at a center frequency of 825 MHz (reverse channel), and transmits packets via a radio transmitters at a center frequency of 870 MHz (forward channel).

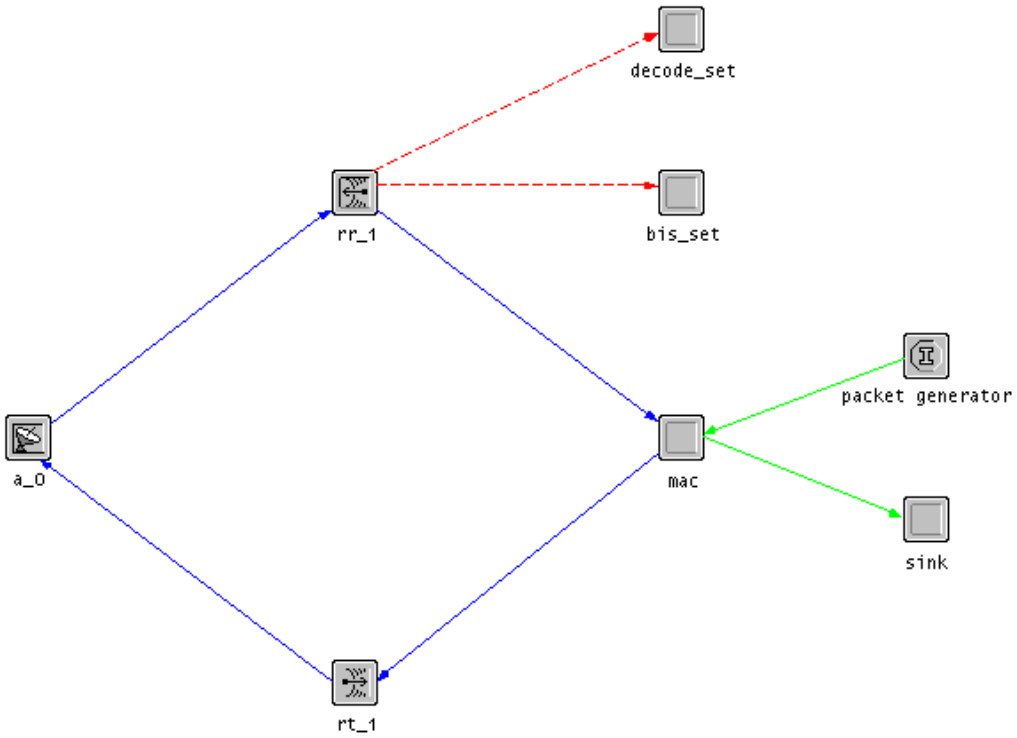


FIG. 4.6. MOBILE DATA BASE STATION (MDBS) NODE MODEL.

This is the interface between the wireless part and the wired part of a CDPD network. The MDBS model also includes a processor (mdbs_mac) that implements the DSMA logic for the MDBS. It sets the *busy/idle* and *decodes status* flags on the forward channel according to the status of the reverse channel. The MDBS also receives the packets and measures certain statistics such as BER, the RF power level, the signal-to-noise ratio, and the end-to-end delay.

The model is highly parameterized. In each M-ES, users may modify parameters such as transmission rates, packet length (which can be fixed, read in from a

trace, or described stochastically), and the radio transmission modulation method. Where applicable, parameters are set to the default values specified by the CDPD standard.

We improved the initial model by incorporating a Rayleigh fading model into the radio transceiver pipeline to induce bursts of bit errors in the transmitted packets. This type of bit error model allows a direct specification of the BER and the burstiness of the errors in the channel [7].

4.3 Process models

4.3.1 Cdpd_mes_mac Process Model

The cdpd_mes_mac process model performs media access control for the CDPD MAC interface in an M-ES. The role of cdpd_mes_mac is to accept data packets from higher layer protocols (MDLP), to encapsulate this data into MAC frames, and to transmitted these frames through the reverse channel to the MDBS using the first-in-first-out scheduling scheme.

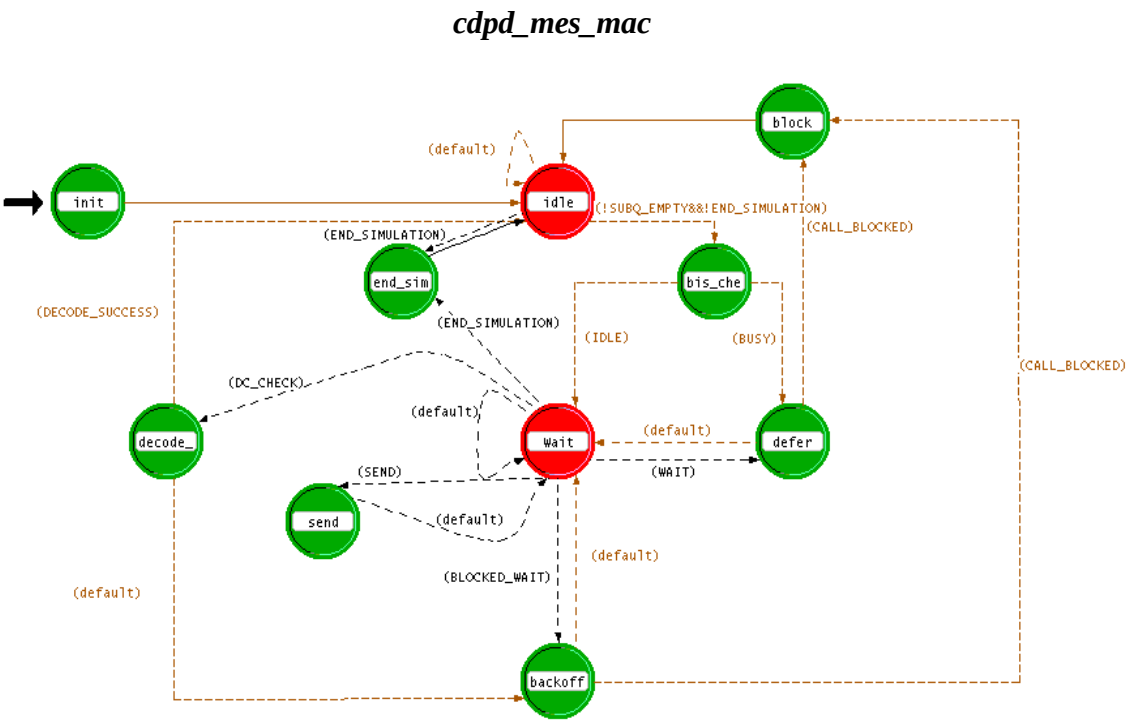


FIG. 4.7. FSM OF CDPD_MES_MAC.

After initializing the state variables, the process waits in the state “idle” for a packet arriving from the upper layer protocol. Once a packet arrives, the packet will be encapsulated to a sequence of blocks (with fixed length 378 bits) and the FSM transitions to the state “bis_check”. In this state, the FSM will check the ‘bis_flag’ on the forward channel. Flag = 0 indicates that the reverse channel is idle. Data transmission can begin by a transition to the state “wait, and by waiting a MIN_WAIT_TIME before sending out the block. MIN_WAIT_TIME is a configurable system parameter with a default value of 0. If the ‘bis_flag’ indicates that the reverse channel is busy, the FSM will transition to the “defer” state. In the “defer” state, the M-ES performs a random backoff and tests the ‘bis_flag’ again. The M-ES FSM will be in the “backoff” state when attempting to retransmit one or more blocks that have been marked as a decode failure by the MDBS. When the M-ES enters the “send” state, it transmits a block through the reverse channel. “Decode_wait” state is a temporary state in which the M-ES waits a brief duration of time to determine whether the last block of its last transmission was successfully received by the MDBS or not. Time allocated for ”decode_check” is 7 microslots (21.875 ms).

Detailed description of the transition conditions, state variables, and functions of each state, are given in TABLES 2 - 4.

TABLE 2. cdpd_mes_mac Transition Condition Descriptions.

cdpd_mes_mac Transitions	
Condition	Details
!SUBQ_EMPTY&&!END_SIMULATION	A packet waits to be transmitted at the MAC module
END_SIMULATION	The simulation has completed.
CALL_BLOCKED	A packet failed to be transmitted.
DECODE_SUCCESS	A packet was received successfully.
IDLE	No packet is being transmitted over the reverse channel.
BUSY	A packet is being transmitted over the reverse channel.
DC_CHECK	A packet was sent out. It is time to check the decode status.
WAIT	A packet is being transmitted over the reverse channel.
SEND	No packet is being transmitted over the reverse channel. It's time to transmit a new packet.
BLOCKED_WAIT	The last packet transmitted was blocked.

TABLE 3. cdpd_mes_mac State Variables.

cdpd_mes_mac State Variables	
Bis	Flag used to indicate the busy/idle status of the reverse channel.
access_attempts	Number of transmission attempts for the current frame being transmitted.
delay_dist_ptr	A pointer to the distribution function which controls the random back off time.
Monitor_id	The ID of the monitor object in this node.
node_id	The object ID of the parent node model.
blk_counter	A counter for how many times the current frame was blocked from transmission.
Pkptr	Pointer to the packet currently being transmitted.

TABLE 4. cdpd_mes_mac Process State Definitions.

cdpd_mes_mac Process State Definitions	
Init	This is the first state entered by the process model and the initial interrupt should be the **begin simulation interrupt**. The activities performed in this state include: 1. Initializing the **bis** state variable. 2. Determine the **object ID** of the MAC's node module, as well as that of the attached **channel_monitor**. 3. Load the appropriate **PDF**'s for the random back off time.
Idle	This state serves as a waiting point for new transmission requests to arrive from the higher layer. It is entered after initialization or upon completion of processing for a transmission request.
end_sim	The simulation is completed. Release the resources.
bis_check	When the process model enters this state, processing of a new transmission request is about to begin. The higher layer data is already encapsulated into a CDPD MAC layer frame.
Defer	Computes the random number of slot times to defer according to the truncated binary exponential defer procedure. A timer in the form of a self interrupt is set to mark the end of the back off period. After the defer period elapses, the process model can move to the Wait state or to the Block state depending on the values of the blk_counter variable.
Wait	This state is essentially empty and serves as a waiting point for new transmission requests to arrive from the higher layer
Send	When the process model enters this state, processing of a new transmission request is about to begin. The higher layer data is already encapsulated into a CDPD MAC packet, and this packet is taken from the head of the request queue and sent out through the output line. After completing these actions, the process model will go back to the Wait state and wait for the response of the MDBS to check if the packet was received successfully. When DC_CHECK is high, the process model will enter decode_check state and will check the status of the decode flag on the forward channel.
Backoff	Computes the random number of time slots to backoff according to the truncated binary exponential backoff procedure. A timer in the form of a self interrupt is set to mark the end of the back off period. After the backoff period elapses, the process model can move to the Wait state or to the Block state depending on the values of the blk_counter variable.
decode_check	In this state, the process model will retrieve the decode flag from the forward channel. If decode flag is 1, the last packet

	transmitted was received successfully and the process model will enter Idle state again and wait for the request for the next transmission. If the decode flag is 0, the process model will enter the backoff state, and blk_counter will increase by 1.
Block	A collision has occurred. Transmission is continued for the jam time, after which the transmission will be aborted. Any physical layer events are ignored during the jam period and all the parameters will be reset to the default values.

4.3.2 Cdpd_trc_gen model

cdpd_trc_gen

FIG. 4.8. FSM OF CDPD_TRC_GEN.

To run a “trace driven” simulation, we need to input measured CDPD network traffic trace into the network model. This is done by the cdpd_trc_gen process model. The state “Init” initializes the variables and loads the genuine traffic data into a structure called ‘list’. It will examine the first line of the list to obtain the arrival time and packet size for the first frame. It will create a frame with the packet size and set up a self-interrupt at the arrival time. Then, the FSM will transition to the state “Idle”. In the “Idle” state, M-ES first transmits the first frame

generated by the “Init” state to the cdpd_mes_mac when a self-interrupt happens. Then, the packet data length and the arrival time are generated according to the values on the next line of the list. The generated packet is then forwarded to cdpd_mes_mac, and “Idle” schedules another packet arrival (i.e., self interrupt) before going back to sleep again.

Detailed description of the transition conditions, state variables, and functions of each state, are given in TABLE 5 and TABLE 6.

TABLE 5. cdpd_trc_gen State Variables.

cdpd_trc_gen State Variables	
Name	Description
line_index	Indicates which line has been read from the script file.
line_list_ptr	A pointer to the list read in from the script file.
list_size	Number of lines in the file.
field_list_ptr	Pointer to the fields obtained by decomposing the list.

TABLE 6. cdpd_trc_gen Process State Definitions.

cdpd_trc_gen Process State Definitions	
Name	Description
Init	This is the first state entered by the process model and the initial interrupt should be a **begin simulation interrupt**. The activities performed in this state include: 1. Initializing the **line_index** state variable. 2. Loading the script file into a list structure. Initialize the pointer to this list. 3. Decompose the first line of the list into a field structure. Also initialize the field pointer.
Idle	In this state, when a self interrupt happens, the process model will generate a packet and schedule a self interrupt at the generation time of the next packet.

4.3.3 M*dbs_mac* process model

Mdbs_mac is the peer part of cdpd_mes_mac inside the MDBS. The MDBS has continuous access to the reverse channel stream. The MDBS notifies the listening M-ES's of the presence of data on the reverse channel and of the decode status of the incoming data blocks from the reverse channel, by using the busy/idle flag and the decode status flag in the forward channel stream.

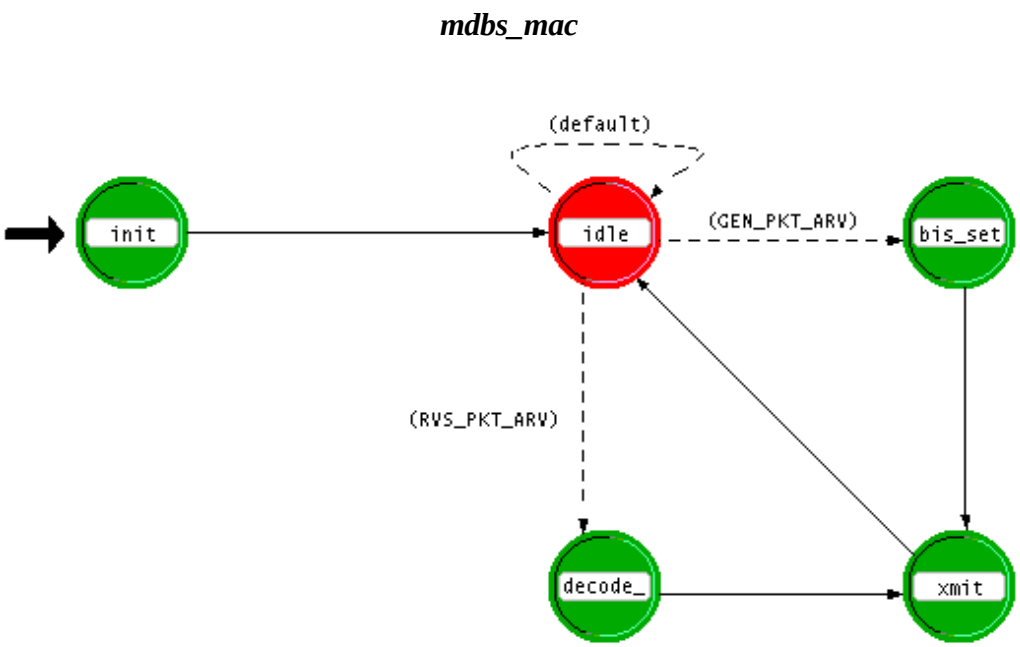


FIG. 4.9. FSM OF MDBS_MAC.

The implementation of the detection of data on the reverse channel must be such that M-ES transmissions within 8 bit times (0 in our model) of the last bit of an idle flag should result in setting the next busy/idle flag. MDBS, after receiving the block from the reverse channel, decodes the data block within two microslots and

sets the next decode status flag with the result. Once the flag is set to indicate decode success, it remains in that state until a decode failure occurs.

In the FSM of mdbs_mac process model, “init” state initializes the variables and then transitions to the “idle” state. In the “idle” state, the MDBS monitors the status of the reverse channel and the decode status of the incoming data blocks from the reverse channel. According to the status, the MDBS sets the busy/idle flag and decodes status flag on the forward channel in the ”bis_set” and “decode_set” state, respectively. In the “Xmit” state, the MDBS sends the packet on the forward channel and discards the packet received from the reverse channel. For the detailed description of the transition conditions, state variables, and functions of each state, please refer to TABLE 7 and TABLE 8.

TABLE 7. mdbs_mac Transitions.

mdbs_mac Transitions	
Condition	Details
GEN_PKT_ARV	A packet arrived from the idle generator.
RVS_PKT_ARV	A packet arrived from the reverse channel.

TABLE 8. mdbs_mac Process State Definitions.

mdbs_mac Process State Definitions	
Name	Description
Init	Performs all necessary initializations.
Idle	Used to wait for incoming packets.
bis_set	Set the busy_idle flag according to the status of the reverse channel.
decode_set	No collision has occurred, and the packet can be sent to its destination. (In this simulation model, the incoming packet is simply destroyed.)

TABLE 9. Configurable M-ES MAC Parameters.

Parameter	Default Value	Description
Max_TX_Attempts	13	The number of times an M-ES will observe Busy/Idle flags in order to gain access to a CDPD channel before declaring the channel congested.
Min_Wait_Time	0	The minimum amount of "microslots" an M-ES must remain in an Idle state before transmitting blocks. A microslot is the time between Busy/Idle flags, or 60 bits.
Max_Blocks	64	The maximum amount of blocks in one continuous transmission burst.
Max_Entrance_Delay	35	The maximum amount of microslots (60 bit times = 3.125 milliseconds) that the M-ES will wait when attempting to re-access a channel for an initial burst.
Min_Count	4	Due to decode failures, an M-ES will attempt to retransmit in no less than [2 (to the 4^{th} power)]-1 microslots time intervals.
Max_Count	8	Due to decode failures, an M-ES will attempt to retransmit in no more than [2 (to the 8^{th} power)]-1 microslots time intervals.

In TABLE 9, the default values of the configurable parameters we used in our simulation are listed. The values were suggested by the CDPD Forum and can be found in [1].

4.4 Cellular model custom pipeline stages

For the radio links in OPNET, an open architecture is provided to allow developers to specify customized behavior for each individual link and on a per-transmission basis. This architecture is referred to as the transceiver pipeline because it provides a conduit connecting a transmitter to one or more receivers.

The simulation kernel manages the transfer of packets by implementing a series of computations that each models particular aspects of link behavior. The sequence of the computations and their interface are standardized for each type of a link. However, each computation, referred to as a pipeline stage, is performed outside the simulation kernel by a user-supplied procedure, called a pipeline procedure. In this manner, OPNET provides an open and modular architecture for implementing link behavior.

A link's underlying implementation can be thought of as a sequentially executed set of pipeline stages. The pipeline stage sequence of a link is executed once for each packet transmission that is submitted at the source of the link. In other words, when a packet is sent to a transmitter, the simulation kernel proceeds to call appropriate pipeline stages to process the packet. Certain pipeline stages are executed when the packet is transmitted, and others are executed later due

to the delay associated with the traversal of the link and transmission of the packet.

4.4.1 Transceiver pipeline stages in OPNET

The radio transceiver pipeline consists of fourteen stages:

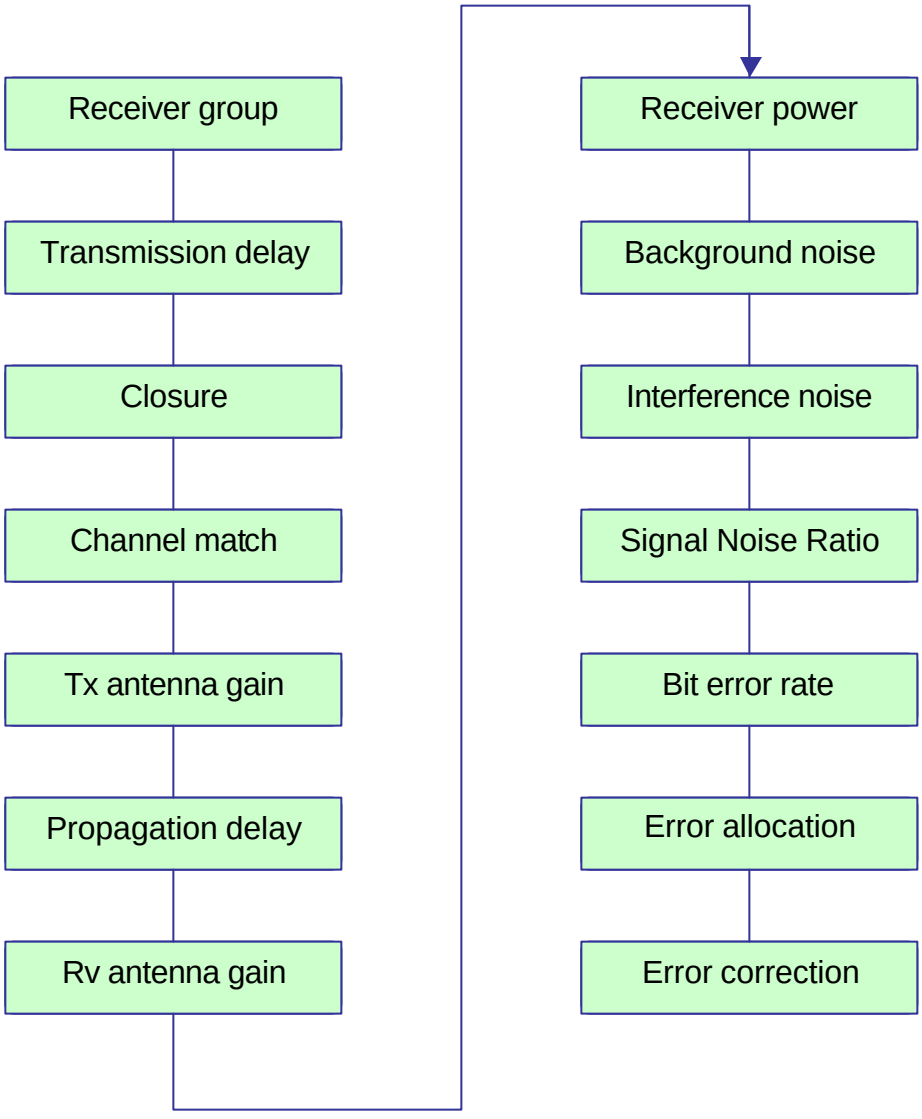


FIG. 4.10. RADIO TRANSCEIVER PIPELINE STAGES.

In our work, we implemented a fading channel into our network model to simulate a bursty wireless channel. The following is a brief introduction of the pipeline stages relative to our work:

Stage 4: Transmitter antenna gain

The purpose of this stage is to compute the gain provided by the transmitter's associated antenna, based on the direction of the vector leading from the transmitter to the receiver. This result is typically used in Stage 7 for receiver power computation.

Stage 6: Receiver antenna gain

The purpose of the receiver antenna gain stage is to compute the gain provided by the receiver's associated antenna.

Stage 7: Receiver power

The receiver power stage is specified by the "power model" attribute of the radio receiver. It is invoked immediately after the return of the receiver antenna gain model, with no simulation time elapsing in between. The purpose of this stage is to compute the received power of the arriving packet's signal (in Watts).

Stage 8: Background noise

By using this stage, OPNET is able to represent the effect of all noise sources, except for other concurrently arriving transmissions, since these are accounted for by the interference noise stage (Stage 9).

Stage 10: Signal-to-noise ratio

It is specified by the “snr model” attribute of the radio receiver. The purpose of SNR stage is to compute the current average power SNR statistics for the arriving packets. This calculation is usually based on values obtained during earlier stages, including received power, background noise, and interference noise. The SNR of the packet is an important performance measure that supports determination of the receiver’s ability to correctly receive the packet’s content. The result computed by this stage is used by the kernel to update standard output statistics of receiver channels and also by later stages of the pipeline.

Stage 11: Bit error rate (BER)

The BER stage is used to derive the probability of bit errors during the past interval of constant SNR. This is not the empirical rate of bit errors, but the expected rate usually based on the SNR. In general, the bit error rate provided by this stage is also a function of the type of modulation that is used for the transmitted signal.

Stage 13: Error correction

This is the final stage of the OPNET radio transceiver pipeline. Based on the results of former stages, now is the time to determine whether or not the arriving packet can be accepted and forwarded via the channel's corresponding output stream to one of the receiver's neighboring modules in the destination node. This usually depends on the result computed in the error correction stage, and on the ability of the receiver to correct the errors affecting the packet. Based on the determination of this stage, the kernel will either drop the packet or allow it to proceed into the destination node.

4.4.2. Uniqueness of mobile radio environment in a cellular system

In most networks the data is transmitted on wire or via a fiber-optics cable. In mobile data networks, radio transmits data over the air. This so-called air-interface, in contrast to wire or fiber optics, is prone to bit errors introduced into the transmitted data. For a radio link, the power loss of the transmitted signal is due to three factors: propagation attenuation, severe fading, and background noise. All three have severe impact on the signal-noise ratio (SNR) of the received frame arrived at the base station. For example, in a mobile radio environment, the bit error rate (BER) can be dynamic and can be as high as 10^{-2}, or even higher compared to the 10^{-10} observed in fiber optical networks [7]. Also, a fading signal will induce bursts of bit errors in the transmitted packets. This type of bit error model will allow a direct specification of the BER and the burstiness of the errors in the channel. The highly dynamic environment presents a unique challenge to the implementation of CDPD networks [1, 16]. An accurate simulation model for a wireless network should be able to reflect the propagation

attenuation, fading, and background noise of the wireless channel. In this section, we give a brief introduction to each.

The propagation attenuation

In general, the propagation path loss increases not only with frequency but also with distance. In free space, the propagation path loss would be 20 dB/decade [7]. Therefore,

$$C \propto R^{-2} = aR^{-2} \tag{4.4.2-1}$$

where C = received carrier power
R = distance measured from the transmitter to the receiver
a = constant .

The decibel expression of the difference in power reception at two different distances R_1 and R_2 is:

$$\Delta C(dB) = C_2 - C_1(dB) = 10\log\frac{C_2}{C_1} = 20\log\frac{R_1}{R_2} \ . \tag{4.4.2-2}$$

In the mobile radio environment, the propagation path loss is 40dB/dec. This means that a 40 dB loss at a signal receiver will be observed by the mobile unit as it moves form 1 to 10 km. Therefore,

$$C \propto R^{-4} = aR^{-4} \tag{4.4.2-3}$$

$$\Delta C(dB) = C_2 - C_1(dB) = 10\log\frac{C_2}{C_1} = 40\log\frac{R_1}{R_2} \ . \tag{4.4.2-4}$$

In a real mobile radio environment, the propagation path-loss slope varies as

$$C \propto R^{-g} = aR^{-g} \ . \tag{4.4.2-5}$$

Usually, g lies in the interval (2, 5] depending on the actual conditions, and it cannot be lower than 2 (corresponding to the free-space condition) [7].

Fading

Since the antenna height of the mobile unit is lower than its typical surroundings, and the carrier frequency wavelength is much less than the sizes of the surrounding structures, multi-path waves are generated. At the mobile unit, the sum of the multi-path waves causes a signal-fading phenomenon. The signal fluctuates in a range of about 40 dB (10 dB above and 30 dB below the average signal) [1, 7]. The fading in a mobile radio environment consist of two parts: $m(t)$ is called *local mean, long-term fading, or log-normal fading* and its variation is due to the terrain contour between the base station and the mobile unit. Another factor $r_0(t)$ is called *multi-path fading, short-term fading,* or *Rayleigh* fading and its variation is due to the waves reflected from the surrounding buildings and other structures. A fading signal $r(t)$ can be characterized by $m(t)$ and $r_0(t)$ in this way

$$r(t) = m(t)r_0(t) \tag{4.4.2-6}$$

where $m(t)$ is the contour of the fading signal and $r_0(t)$ represents the detail of the signal.

Noise level

One type of noise that cannot be avoided is the thermal noise kTB at a temperature T. At 290 K (17^0C) and a bandwidth B = 30 kHz, it is equal to –129 dB. (k is Boltzmann's constant, and $kT = -174 dBm/Hz$ at $T = 290K$). If we assume that the received front-end noise is 9 dB, then the noise level is –120 dB. This is the lower bound on the noise level at the receiver end. Two other types of

man-made noise should also be considered: the ignition noise generated by the vehicles, and the noise generated by 800 MHz emissions. An amplifier will amplify a mobile radio signal received by a receiving antenna. The received signal is also affected by the amplifier noise.

We incorporated all of the above mentioned factors into our wireless channel model. Our work was based on the analysis by William C.Y. Lee, "Mobile Cellular Telecommunications Systems" [7]. The mobile radio propagation is:

$$P_r = P_t - 156 - 40\log r_1 + 20\log h_1 + 10\log h_2 + G_t + G_m \qquad (4.4.2\text{-}7)$$

where
$P_r =$ received power
$P_t =$ transmitted power
$r_1 =$ the distance from the mobile unit to base station
$h_1 =$ height of the base station antenna
$h_2 =$ height of the mobile unit antenna
$G_t =$ base station antenna gain
$G_m =$ mobile unit antenna gain.

According to this general formula, we incorporated the propagation delay, fading, and background noise models into three OPNET pipeline stages. They are stage 7 (receiver power), stage 8 (background noise) and stage 10 (snr stage). The code is listed in the Appendix, in files "cdpd_power.ps.c", "cdpd_bkgnoise.ps.c", and "cdpd_snr.ps.c". The purpose of these custom pipeline stages is to model the physical environment in which a cellular system operates. The end result is to provide SNR calculations that take into account path loss in cellular environments and average fading effects.

4.5 Simulation results

In this section, we will describe a series of simulation experiments that demonstrate the practical significance of LRD in the queueing performance of a CDPD network. We will also show that the use of standard queueing approximations to set CDPD operating parameters may lead to gross performance problems.

In each M-ES node model in our simulation, we implemented a queueing system with the following characteristics: first in first out, infinite waiting room, and arrivals that can be taken either from an ideal packet generator built in OPNET or from an actual CDPD network traffic traces (trace driven simulation). The input traces consist of the measured inter-arrival times and packet sizes. The results shown below are obtained with a single 18-min long CDPD network traffic trace. Why an 18-min trace? The reason is that 18-min is the longest active time in our measured traces. The underlying assumption in engineering practice used in our research work is that the traffic environment is stationary over such time scales. While this assumption is not always satisfied in practice, it does appear to be a reasonable hypothesis for the traces we have measured. In our experiments, we also ran simulations using input traces with exponentially distributed inter-arrival time (Poisson arrival process) and traces from bursty ON/OFF traffic source models. Thus, we will be able to compare the queueing performance of the CDPD network with different input traffic: actual trace, Poisson arrival traffic, and a trace generated by ON/OFF models.

4.5.1. Queueing delay analysis

We begin our investigation with one single simulation and observe the queueing performance of the network. The network scenario is similar to the genuine BCTel's CDPD network environment, with the channel bandwidth of 19.2 kbps.

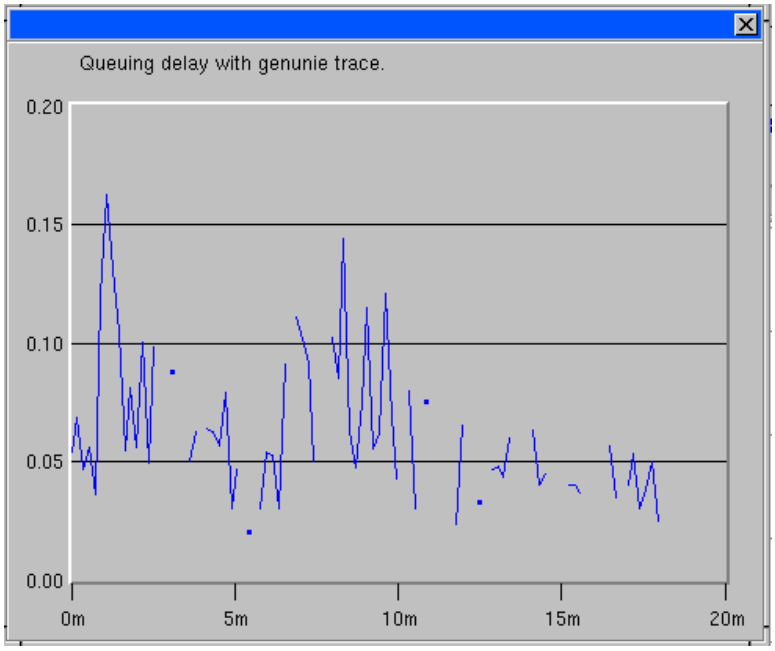


FIG. 4.11. QUEUEING DELAY WITH ACTUAL TRACE.

The result of a single simulation with genuine trace is shown in Fig. 4.11. It shows how many seconds a packet has to wait in the queue before it can be served. From the presented graph, we can conclude that the queueing delay is fluctuating between 0.02 and 0.16 seconds. Notice that the network utilization is about only 5.5%.

For the sake of comparison, we reran the simulation with traffic trace whose inter-arrival times are exponentially distributed. The inter-arrival time has the same mean as the actual trace used in the first simulation: 0.94 sec.

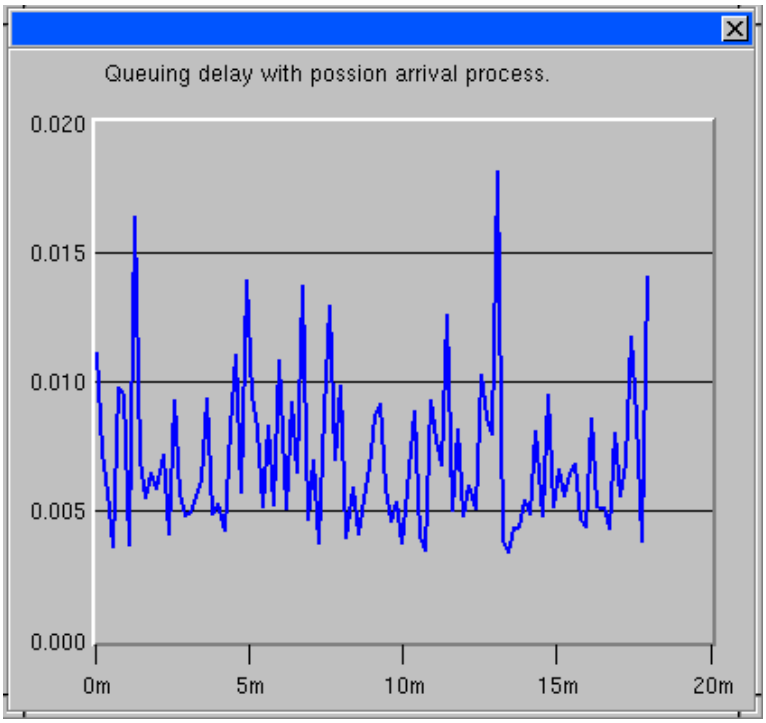


FIG. 4.12. QUEUEING DELAY WITH POISSON ARRIVAL TRACE.

Fig. 4.12 shows the result of a single simulation. In contrast to the delay curve obtained with the actual trace, Fig. 4.12 predicts the queueing delay to be in the interval: 0.005 ~ 0.015 seconds, which is much smaller than the 0.02 ~ 0.15 second interval observed in Fig. 4.11. By keeping the network scenarios in the simulations identical, we are able to isolate the effects of the inter-arrival times on queueing and separate them from effects due to the changing environment of the underlying network.

Next, we investigated the effect of the network utilization on the queue delay performance. Graph (A) in Fig. 4.13 is the average delay versus utilization plot obtained with the original trace. From the viewpoint of teletraffic engineering, the "knee of the curve" is of particular interest. As shown in Fig. 4.13, there is a sharp rise in the average delay around 30 ~ 50% utilization. In contrast to the delay graph obtained with the actual trace, graph (B) predicts the sharp rise when utilization exceeds 80%.

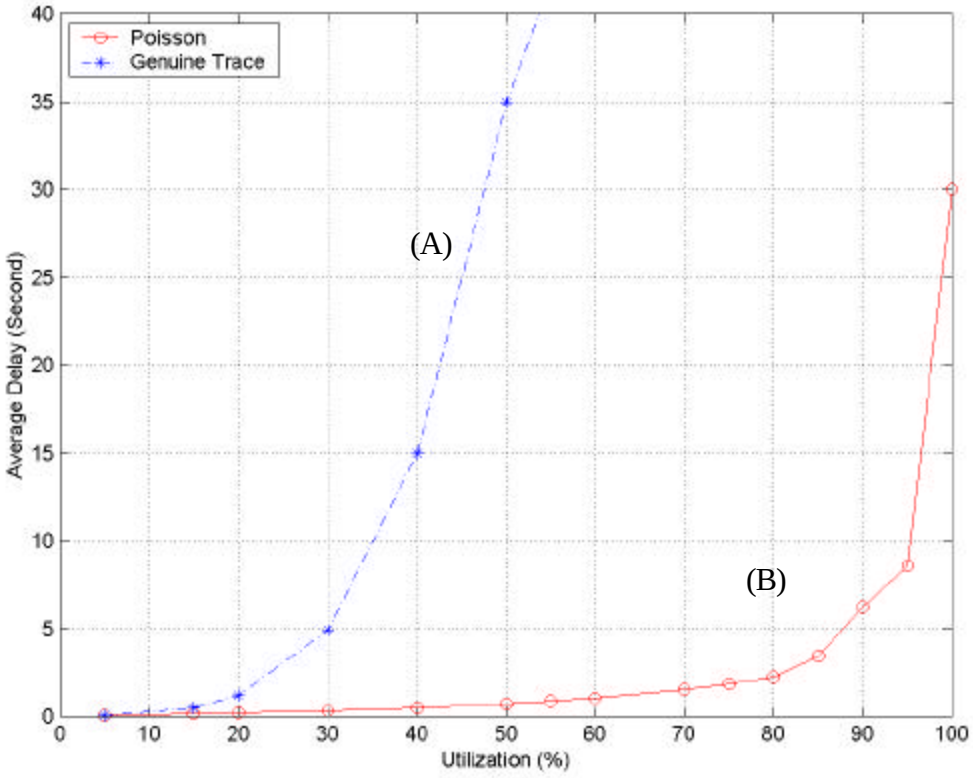


FIG. 4.13. AVERAGE DELAY VS UTILIZATION I

In order to identify the features of the input trace that contribute to the sharp rise in delays at relatively low utilizations, we repeated the simulation experiments with the input trace generated by an ON/OFF traffic model [14]. The model has

the same traffic load to the CDPD network as the actual trace and the same exponential inter-arrival process as used in the last experiment.

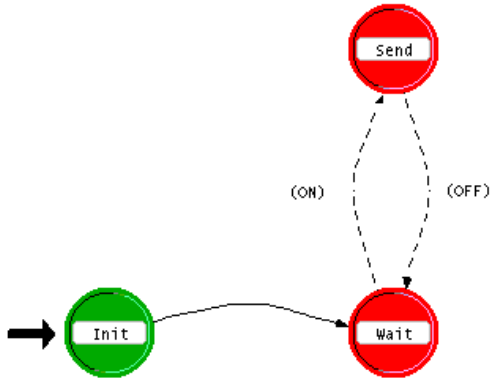


FIG. 4.14. ON/OFF OPNET TRAFFIC SOURCE MODEL.

The structure of the ON/OFF traffic source model we used in our simulation is shown in Fig. 4.14. In this model, three parameters are essential for determining the level of the burstiness of the generated traffic. They are mean duration of the 'ON' state, mean duration of the 'OFF' state, and the number of frames sent out during one 'ON' period. In our simulation, durations of the ON-period and the OFF-period are exponential with mean *b* and *a,* respectively. In this ON/OFF traffic model, the packet size is fixed to 127 bytes based on the measurements.

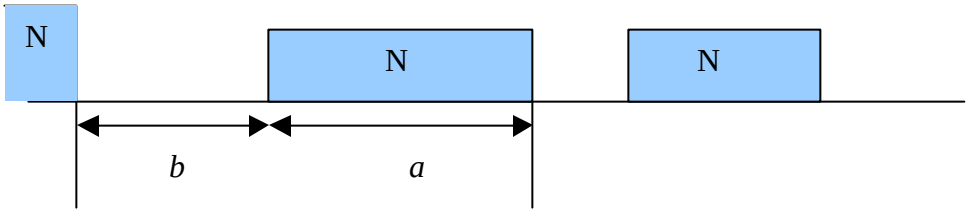


FIG. 4.15. TRACE GENERATED BY ON/OFF TRAFFIC SOURCE MODEL.

The relationship of the three parameters is shown in Fig. 4.15. Here, *a* is the average duration of the ON - period, *b* is the average duration of the OFF - period, and *N* is the number of frames sent during the ON - period. In our first simulation run, we chose *a* = 0.01 and *b* = 9.99. This implies that ON - period << OFF - period. Thus, the generated traffic can be thought as very bursty [14]. On the other hand, to keep the same traffic load level to the network, *N* is set to be 11.

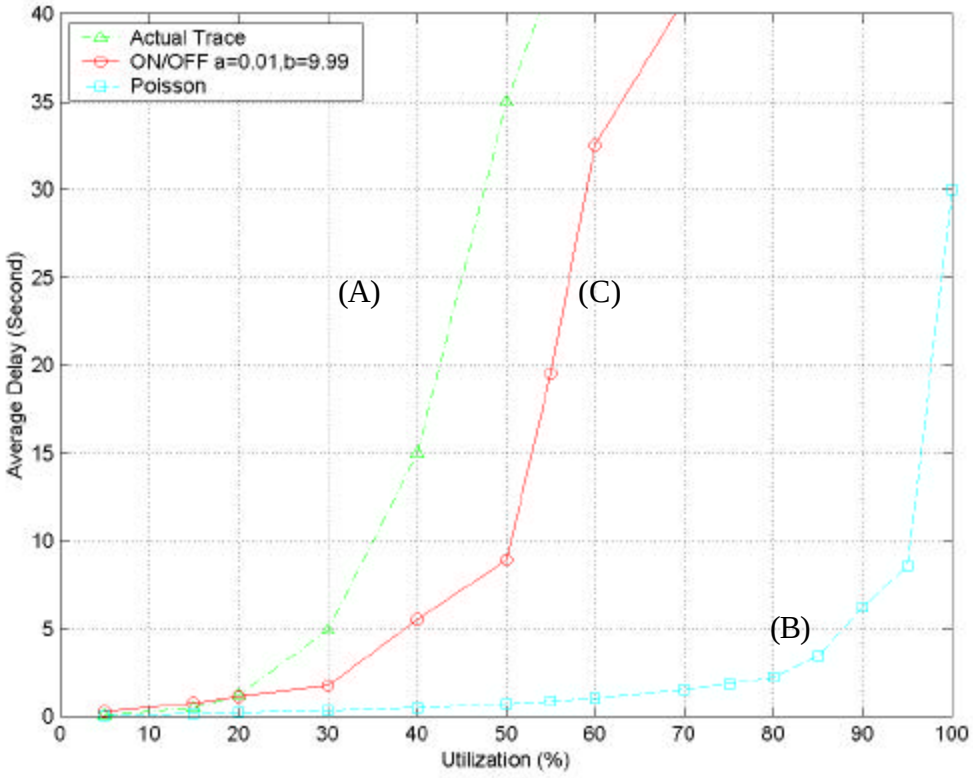


FIG. 4.16. AVERAGE DELAY VERSUS UTILIZATION II

Fig. 4.16 shows the queueing performance of the simulated network with different input traffic types. As a function of the network bandwidth utilization, we plot the average delays for the actual traffic data (graph (A)), for the traffic trace generated by our ON/OFF source (graph (C)), and for a Poisson source (graph (B)). Graphs (A) and (B) are as shown in Fig. 4.13, and the difference between them is obvious. Comparing graphs (B) and (C), we conclude that the queueing performance of the traffic generated by an ON/OFF source is closer to the performance of our actual network traffic data. This is to be expected, since ON/OFF source reflects the bursty characteristic of the genuine network better than a Poisson source. Nevertheless, we still see that there is a discernible

difference between the two cases. Such discrepancies (in this case around 10% on the utilization) may not be acceptable in practical traffic engineering. Thus, additional safety margins have to be incorporated into capacity estimates.

In the next experiment, we repeated two simulation runs with an input trace generated by the ON/OFF source with a = 5, b = 5 and a = 8, b = 2, respectively. We keep the unit time (a + b = 10 sec) unchanged from the ON/OFF source we used in the last simulation. By changing the ratio of a to b, we can modify the burstiness of the generated traffic. Intuitively, we can say that the traffic generated by ON/OFF source with a/b = 1 is less bursty than traffic generated by ON/OFF source with a/b = 0.5.

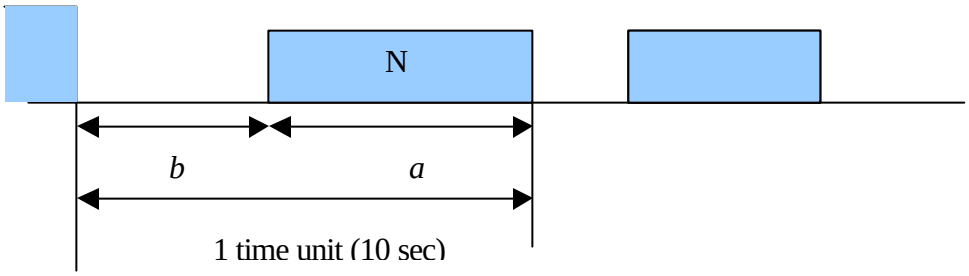


FIG. 4.17. ADJUST BURSTY LEVEL IN AN ON/OFF MODEL.

Fig. 4.17 shows the relationship between a, b, and the time unit. One can choose *b* and *a* independently.

The delay curves obtained with different input traffic traces are shown in Fig. 4.18. The delay performance graphs corresponding to higher a and b ratio (graphs (D) and (E)) are somewhat different from the result of the ON/OFF

source with lower a and b ratio (graph (C)). They depart even further from the delay performance of the actual network traffic trace (graph (A)).

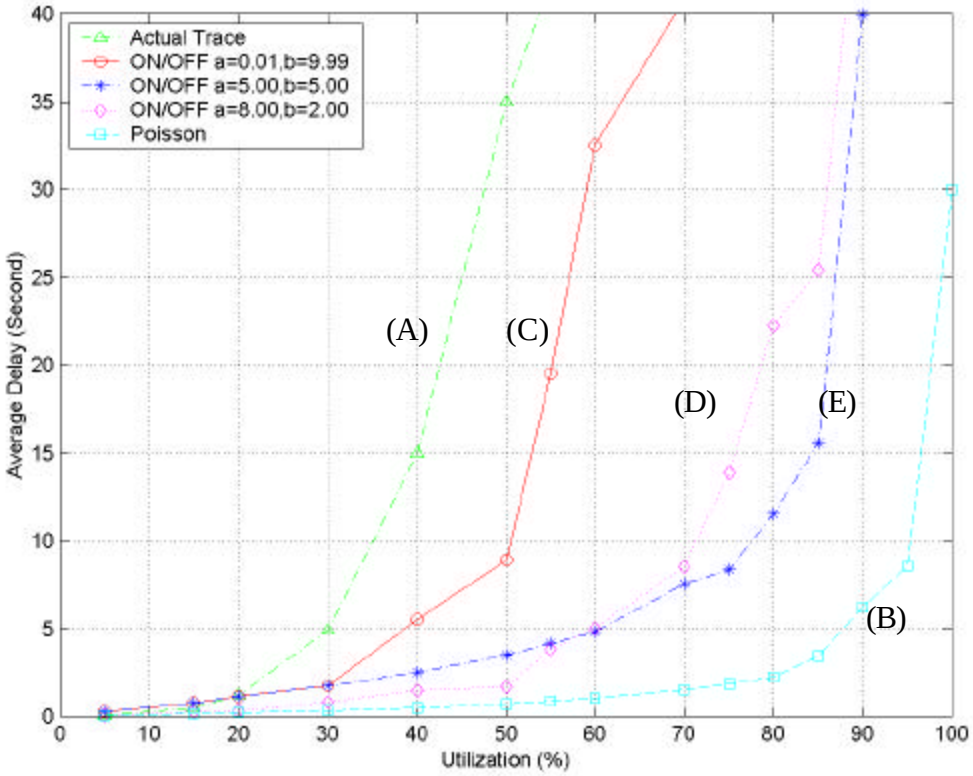


FIG. 4.18. AVERAGE DELAY VERSUS UTILIZATION III.

In Fig. 4.18, graph (A) is the result from Genuine traffic, (B) is from Poisson arrival traffic source, (C) is from ON/OFF source with a = 0.01, b = 9.99, (D) is from ON/OFF source with a = 8.00, b = 2.00, and (E) is from ON/OFF source with a = 5.00, b = 5.00.

4.5.2. Queueing delay discussion

As can be seen, the delay performance obtained with the actual trace is different from the delay predicted by Poisson arrival processes. More significantly, the disparity between graphs (A) and (B) indicates that the short-range dependent traffic source model will grossly underestimate queueing delays at moderate and high utilizations. On the other hand, this experiment suggests that although bursty ON/OFF model can emulate the genuine network traffic trace better, a renewal arrival process with a single distribution cannot simply explain the burstiness of traffic. For example, the queueing performance graphs obtained by our 'best model' (graphs (A) and (C)) differ significantly. Such discrepancies (around 10% on the utilization axis) may not be acceptable in practical traffic engineering. Thus additional safety margins have to be incorporated into capacity estimates, given that this error estimate may be optimistic. These discrepancies arise because our ON/OFF source model assumes exponential distributions for its ON and OFF periods (or more generally, finite variance distributions). In recent years, it has been recognized that multiplexing a large number of these sources results in aggregate traffic that is inconsistent with traffic measurements from working networks. Thus, continuing this work and implementing more accurate traffic source models would be fruitful.

4.5.3. Buffer overflow probability

In the previous experiments, the buffer we chose was infinite. In this section, we investigate the relationship between buffer overflow probability, the size of the buffer, and the utilization of the network. In our first experiment, the buffer size was fixed to 5 packets. We first investigate the relationship between buffer overflow probability and network utilization. Fig. 4.19 shows a plot of $\log P(q > x)$, where q is the length of queue.

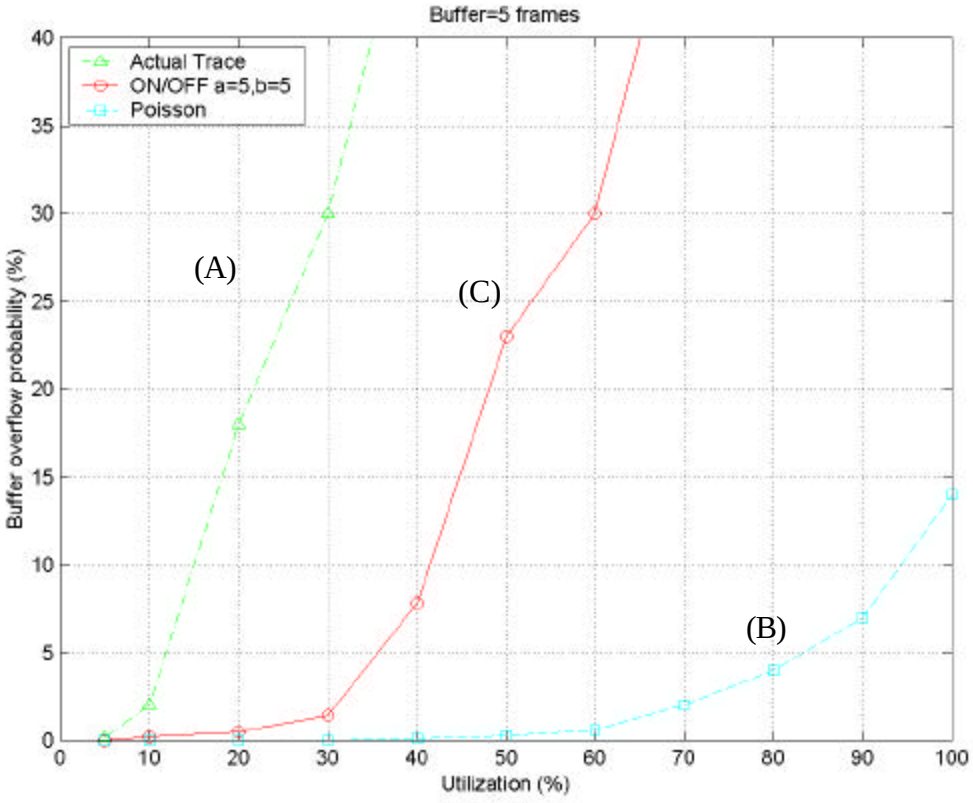


FIG. 4.19. BUFFER OVERFLOW PROBABILITY VS. UTILIZATION.

Here, graph (A) is the buffer overflow probability for the actual trace, and graph (B) is for Poisson arrival source, and (graph C) is for the ON/OFF traffic source model with a = 5, b = 5. Even at relatively low network utilization (20 %), the buffer overflow probability obtained by a genuine traffic source is much larger than from a Poisson traffic source model. Furthermore, an ON/OFF model again predicts queueing performance more accurately than a Poisson traffic model. In our second experiment, the network utilization was kept fixed and we investigated the queue length distribution vs. buffer size. We have considered a utilization of 0.4, corresponding to the knee of graph (A) in Fig. 4.18.

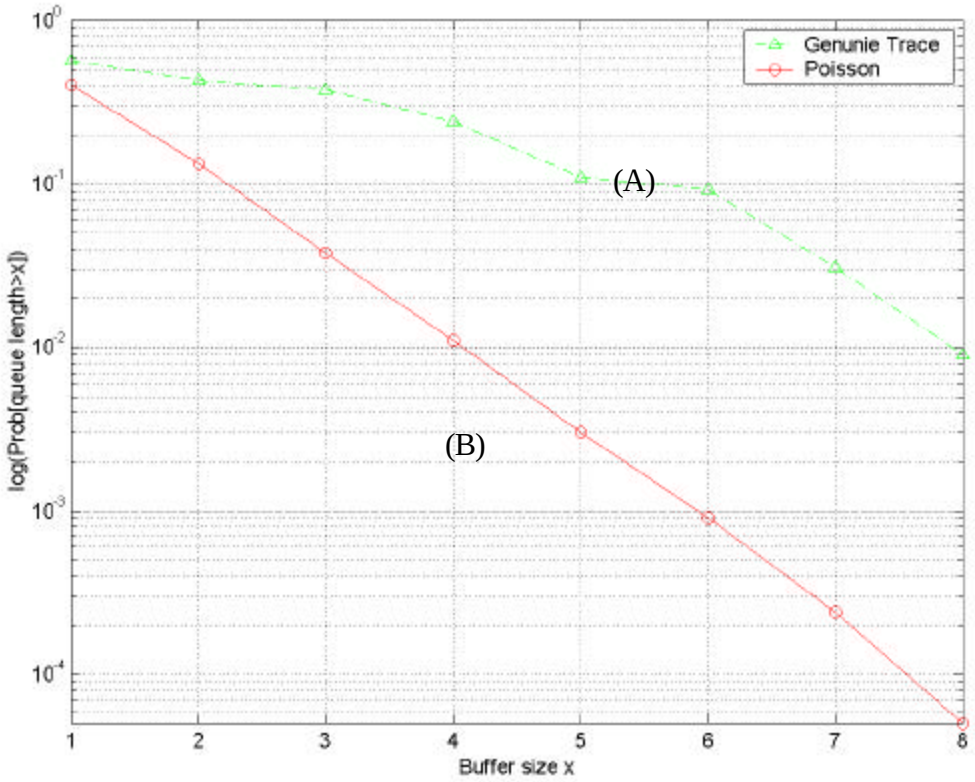


FIG. 4.20. QUEUE LENGTH DISTRIBUTION VS. BUFFER SIZE.

Graph Fig. 4.20 (A) shows the queue length distribution with genuine traffic. Graph (B) is the result when using a Poisson arrival traffic source. It is easy to observe a range of queue lengths of an exponential type (i.e., linear region on the semi-log plot), with constant decay, which falls much faster than the result obtained by using the genuine traffic source. Thus, the simulation results indicate that the tails of the queue length distributions decay more slowly than the exponential rate predicted by a Poisson model. This property can be expected to have considerable impact in engineering. For example, buffers sized on the basis of conventional model may result in under-provisioning. A combination of increased buffers and reduced utilizations may be necessary to achieve acceptable loss rates. The under-provision may introduce additional errors in admission strategies that are based on equivalent bandwidth schemes. Once again, the slow decaying behavior is not existent in short-range dependent models with constant or exponential service times, and shows that the use of standard queueing approximations to set CDPD operating parameters may lead to gross performance problems.

4.5.4. Self-Similar traffic modeling

Collectively, the results obtained from our simulation on how self-similarity impacts performance of wireless data networks leads to the next question: how to actually model network traffic accurately and realistically? In other words, what are traffic models that capture the characteristics observed in measured network traffic? Does this mean “the end of simple traffic models”?

To answer these questions, recall that by *a traffic characteristic*, we mean a property that has been uncovered by a rigorous statistical analysis of measured traffic data and

(i) Can be estimated,

(ii) Can be given a meaningful physical interpretation in the context of the network where the data was collected,

(iii) Has been demonstrated to have a dominant impact on problems related to network engineering.

With 10 or more parameters, traditional traffic modeling yields little insight into the true nature of the traffic, stands no chance of satisfying the three requirements mentioned above, and is of limited practical use (with the possible exception for obtaining numerical results). It turns out hat there exist long-rang dependent processes where on parameter (the Hurst parameter) suffices to capture the self-similarity observed in measured traffic data. These processes are known as Fractional Gaussian Noise (FGN) and Fractional ARIMA Models. For example, LAN traffic can be successfully modeled using a FGN process with three parameters: mean, variance, and Hurst parameter [17]; fractional ARIMA processes with 4 or 5 parameters seems to describe VBR video traffic reasonably accurately [6].

5. Conclusions

In this thesis, by measurements, simulations, and analysis, we demonstrated that CDPD network traffic exhibits long-range dependent (LRD) behavior. By simulating the wireless data network using actual traffic traces, we concluded that genuine traffic produces longer queues and thus requires larger buffers in the network's switching elements.

We used two graphical methods for testing for the long-range dependency of the wireless data traffic trace: R/S plot and variance-time plot. R/S plot shows that $\hat{H} \approx 0.80$. Variance-time plot indicates that $\hat{H} \approx 0.90$. Thus, our results suggest that the wireless data traffic tends to have self-similar behavior, and is statistically different from traffic generated by traditional traffic models.

In this thesis, we used trace-driven simulation experiments to demonstrate that long-range dependence is an important traffic characteristic, and, if ignored, typically results in overly optimistic performance predictions and inadequate network resource allocations. As can be seen from our simulation results, the delay performance obtained with the genuine trace is different from that predicted by Poisson arrival processes. In the circumstance of moderate and high network utilizations, short-range dependent traffic sources (Poisson sources) model underestimated queueing delays grossly.

Another outcome of this work is that, even though ON/OFF model was proposed to be more accurate for capturing the burstiness of network traffics, the difference of network performance between the simulation results we obtained with traffic generated by ON/OFF models and genuine traffic are still obvious. Thus, implementing more accurate traffic source models would be very fruitful in further research. The new traffic models should be able to capture the long-range dependence characteristic of measured network traffic.

References

1. J. Agosta and T. Russell, *CDPD: Cellular Digital Packet Data Standards and Technology*. Reading, MA: McGraw-Hill, 1996.

2. J. Beran, *Statistics for Long-Memory Processes (Monographs on Statistics and Applied Probability).* London: Chapman and Hall, 1994.

3. J. Beran, "Statistical methods for data with long-range dependence," *Statistical Science*, vol. 7, no. 4, pp. 404 - 427, 1992.

4. M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes," *IEEE/ACM Trans. Networking,* vol. 5, no. 6, pp. 446 - 456, Dec. 1997.

5. D. E. Duffy, A. A. McIntosh, M. Rosenstein and W. Willinger, "Statistical analysis of CCSN/SS7 traffic data from working CCS subnetworks," *IEEE J. Select. Areas Commun.*, vol. 12, no. 3, pp. 544 - 551, April 1994.

6. M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. ACM SIGCOMM'94*, London, U.K., Aug. 1994, pp. 14 - 26.

7. W. C.Y. Lee, *Mobile Cellular Telecommunications Systems.* New York, NY: McGraw-Hill, 1989.

8. W. Leland and D. V. Wilson, "High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection," in *Proc. IEEE INFOCOM'91*, Bal Harbour, FL, 1991, pp. 1360 - 1366.

9. W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, no. 1, pp. 1 - 15, Feb. 1994.

10. K. Park, G. T. Kim, and M. E. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," in *Proc. 4th Int. Conf. Network Protocols (ICNP'96)*, Oct. 1996, pp. 171 - 180.

11. V. Paxson, "Empirically-derived analytic models of wide-area TCP connections," *IEEE/ACM Trans. Networking*, vol. 2, pp. 316 - 336, Aug. 1994.

12. V. Paxson, “Growth trends in wide-area TCP connections,” *IEEE Network*, vol. 8, no. 1, pp. 8 - 17, 1994.

13. V. Paxson and S. Floyd, “Wide-area traffic: the failure of Poisson modeling,” in *Proc. ACM SIGCOMM ’94*, London, U.K., Sep. 1994, pp 45 - 56.

14. K. Sohraby, “On the theory of general ON/OFF sources with applications in high speed networks,” in *Proc. IEEE INFOCOM’93*, San Francisco, CA, Apr. 1993, pp. 401 - 410.

15. M. Roughan and D. Veitch, “Measuring long-range dependence under changing traffic conditions,” *in Proc. IEEE INFOCOM’99*, New York, NY, Mar. 1999, pp. 338 - 341.

16. M. Sreetharan and R. Kumar, *Cellular Digital Packet Data.* Norwood, MA: Artech House, 1996.

17. W. Willinger, M. S. Taqqu, W. E. Leland, and D. V. Wilson, “Self-similarity in high-speed packet traffic: analysis and modeling of Ethernet traffic measurements,” *Statist. Sci*., vol. 10, no. 1, pp. 67 - 85. 1995.

18. W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. “Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level,” *IEEE/ACM Trans. Networking*, vol. 5, pp. 71 - 86, Feb. 1997.

19. C. Partridge, “The end of simple traffic models,” *IEEE Network*, pp. 3 – 24, Sep. 1993.