CHAPTER 2

MODELING, SIMULATION AND CROSS-LAYER DESIGN

2.1 NETWORK MODEL

Wireless networks operate continuously in time while modern computers run their programs sequentially and on discrete time steps. Hence, achieving real-time simulation results is all but impossible unless one resorts to sophisticated techniques. However, near real-time behavior can be programmed on computing platforms. The primary objective of this chapter is to develop one such model to achieve a near real-time simulation of a small ad-hoc network. In this chapter, the models that are used for simulations are developed followed by a discussion about cross-layer designs and results comparing two types of scheduling.

The issue focused in this thesis is one of maximizing the throughput of the network under analysis. Unlike some other networks, traffic in the current network is directed towards a gateway node, which in-turn is connected to a wired network. In effect, the problem to be analyzed is the last-mile connectivity problem for cases where the connections to the application area are already established using wired means.

During this optimization exercise, two metrics are considered for throughput maximization. The first metric is the cumulative power expended during transmission by the participating nodes. The second metric is the reduction in the mean data packet delivery time using cross-layer architecture under the constraints of optimized transmission power.
The current problem is explained using the network schematic shown in figure 2.1. A set of ad-hoc nodes are randomly located. A connection to the gateway is possible when the nodes are ready to connect. Each of the nodes communicates with the gateway using wireless means, and for this purpose, both the nodes and the gateway are equipped with wireless cards. Each card is capable of transmitting in any of the 12 channels that are available. The nodes are in control of the transmission power and are capable of varying the power. The gateway is the only infrastructure connected node. Figure 2.1 shows a single gateway for the purposes of convenience. However, several gateways could equally service the ad-hoc network to provide a more efficient but expensive connectivity solution. Each node in the network contends to access the gateway for access to the wired connection. If the network is not time-slotted, the nearest nodes will always be connected but the farthest nodes will starve for connection. One of the aims of network design is to strive for equal allocation of the gateway resources to achieve
fairness for the competing nodes. In addition, the hidden terminal problem in which a terminal transmission collides with an already established connection also remains unsolved.

Hence, it is mandated that the power of the nodes be always optimized such that the nearest nodes transmit at the lowest level and the farthest nodes at maximum power, thereby reducing the aforementioned problems somewhat. The difficulty with this approach is that the node with the maximum power will interfere with the transmission of all other nodes towards the destination. In the case, the farthest node becomes an interferer at the gateway rather than a legitimate user of the gateway. Indeed, the weaker node subdues the stronger node despite the availability of transmission power in the latter.

An unoptimized meshed ad-hoc network with a set of nodes and a single gateway is shown in figure 2.1. The nodes within the maximum power range (marked as $P_{\text{max}}$) are always connected and are called the serviced nodes ($N_s$). Some other nodes (marked $N_b$) in the boundary can potentially be connected later with a time-delay. This delay arises from the time taken to increase the power to $P_{\text{max}}$ by the MAC coordinated function. The remaining nodes (marked $N_u$) are starved and cannot be connected at all at this instant of time.

Therefore, the problem is formulated, wherein the serviced nodes are identified and used as virtual gateways by the other nodes for establishing connectivity with the primary gateway. Fundamentally, this type of a network shares some of the characteristics of a wireless peer-to-peer network.

Now considering figure 2.2 which shows the gateway operating under a reduced transmit power of $P_{\text{opt}} < P_{\text{max}}$. The unconnected node’s transmission power is also reduced such that $P_{\text{min}} << P_{\text{max}}$. 
Intuitively and also based on multiserver queuing theorems, such an arrangement could improve throughput in small networks. It is shown that when an optimum power is used, an increase in throughput and fairness in access are satisfied in this network. The question of determining the optimum power now remains.

The time is slotted according to the scheme discussed in chapter 1, namely into frames where the frames are further subdivided into slots. Each frame is of a temporal duration $t_f$, and each slot extends to a duration of $t_s$. During the frame time $t_f$, the transmit power of the nodes marked $N_f$ (Figure 2.2) is fixed by the algorithm running at the gateway node as $P_{\text{opt}}$. Such a method is the equivalent of a Point Coordination Function (PCF) scheme as discussed earlier.

In effect, the system is time-slotted and a new transmission can commence only at the beginning of a time-slot. A packet is considered to be received without an error if the signal-to-interference plus noise ratio is above
a target signal-to-noise ratio of $\gamma_0$ for the entire duration of the slot in which the packet is transmitted. During the entire operation, each node operates at a transmission power that is less than $P_{\text{max}}$. Under these conditions, now an analytical problem is formulated.

### 2.2 ANALYTICAL PROBLEM FORMULATION

Let $P_i$ denote the transmit power of the $i^{\text{th}}$ node, such that $P_{\text{min}} \leq P_i \leq P_{\text{max}}$ and $\tilde{\delta}_i$ be an indicator variable defined such that

$$
\tilde{\delta}_i = \begin{cases} 
1; & \text{when the } i^{\text{th}} \text{ node is active} \\
0; & \text{when the } i^{\text{th}} \text{ node is non-active}
\end{cases}
\quad (2.1)
$$

Let $G_i$ denote the gain from the $i^{\text{th}}$ node to the gateway. The gain includes any and all of the effects that change the signal strength between the sender and the receiver including path loss, shadowing, multipath fading, antenna gain and coding gain amongst others. Let $\sigma^2$ denote the additive Gaussian white noise power received at the gateway node. Then, if the $i^{\text{th}}$ node is active with $\tilde{\delta}_i = 1$, its signal-to-interference noise ratio (SINR) at the gateway node is given by $\gamma_i$ where

$$
\gamma_i = \frac{P_i G_i}{\sum_{j \neq i} P_j G_j \rho_{ij}^2 \tilde{\delta}_j + \sigma^2} \quad (2.2)
$$

where $\rho_{ij}$ is the cross-correlation between the spreading sequences of the $i^{\text{th}}$ and $j^{\text{th}}$ nodes, both of which are assumed to be active (the correlation vanishes otherwise). In the case of ad-hoc nodes using ISM the band of frequencies, this correlation can be the adjacent channel spread sequence interference.
The powers, denoted as a \((N \times 1)\) vector, \(P = [P_1, \ldots, P_N]^T\), and the indicator vectors, \(\delta = [\delta_1, \ldots, \delta_N]^T\), are defined such that \(P_{\text{min}} \leq P \leq P_{\text{max}}\) and \(\delta \in \{0,1\}\). The joint power control and scheduling problem is to determine, \textit{in each transmission slot}, the feasible set of nodes that can transmit, i.e., the vector \(\delta\), and their corresponding transmit powers, i.e., the vector \(P\). This choice is subject to the additional restriction that the SINR of all of the active nodes should be above the target SINR \(\gamma_0\). This problem is difficult because of the fact that the \(\delta_i\)‘s are binary valued, thereby making the problem a discrete optimization problem. However, the problem becomes more tractable, if the binary valued constraint on the variables \(\delta_i\) is relaxed. In particular, the discrete variable \(\delta_i \in \{0,1\}\) can be replaced by a continuous variable \(\phi_i \in \{0,1\}\) moving the optimization problem out of discrete space into a continuous space.

A physical interpretation for relaxing the variable \(\delta_i\) to be continuous comes about by considering the variable \(\phi_i\) (at a given instant of time) as the probability of transmission by a node. That is, \(\phi_i\) is the probability of the \(i^{th}\) node being active in a transmission slot. Thus, \(\delta_i\) can be treated as a Bernoulli random variable with parameter \(\phi_i\). Thus, the expected SINR for active links at the gateway is the ratio of the received signal power to the expected interference from other active nodes plus noise power. Now, if the \(i^{th}\) node is active, its expected SINR \(\gamma_i\), is given by

\[
\gamma_i = \frac{P_i G_i}{\sum_{j=1}^{N} P_j G_j \phi_j^2 \phi_j + \sigma^2}
\tag{2.3}
\]

Therefore, by relaxing the binary valued variables \(\delta_i\), the problem of joint power control and scheduling is transformed into one of random access with power control (RAPC) to the wireless media, where the objective is to
determine for each node the probability of transmission and the transmit power. In this case, the minimum acceptable SINR criterion for error free packet reception can now be replaced by a minimum expected SINR criterion, \( \gamma_i \) with \( \gamma_i \geq \gamma_0 \).

The aim is to express the problem of random access with power control (RAPC) as a convex optimization problem. Since the primary goal in this thesis is power control, the objective function of the convex optimization problem should involve some metric of the transmit power. The maximum transmit power by a node in a slot represented as \( \| \) \ could be one such metric. However, it is found that optimizing such an objective, subject to SINR constraints, would drive all transmission probabilities to 0, which is not a solution. Hence, the transmission probabilities of a node are to be maximized with \( \min \varphi_i \) or alternatively, one could consider the minimization of \( \max 1/\varphi_i \) (\( \ell^\infty \)-norm). This constraint encourages each node to be active with a high probability; while the power constraint discourages a node from being active when the interference on the channel is high (other nodes are active with high probability). Clearly, the well known power verses delay (or throughput) dilemma for multiple access networks is captured by this problem formulation (Bossert and Weymark 2004).

Given this, the selection of an optimum power that is less than the maximum power is the most practical solution to achieve. This would result in a reduction in the MAC layer collisions. Then, frequency reuse could be employed to increase the throughput of the network by allowing more nodes to be active.

A scalarized version of the multi-criterion optimization problem can be formulated as follows:
Minimize \( \|P\|_{\infty} + \lambda \|1 / \phi_i\|_{\infty} \) \hspace{1cm} (2.5)

Subject to

\[ 0 \leq P \leq P_{\text{max}} \]
\[ 0 \leq \phi_i \leq 1 \]
\[ \gamma_i \geq \gamma_0, \quad \text{and} \quad 1 \leq i \leq N \]

where \( N \) is the number of wireless nodes and \( \lambda \) is a scalarization parameter and the random variable \( \phi_i = [\phi_1 \ldots \phi_N]^T \). Varying \( \lambda \) will generate a family of solutions, which characterize the Pareto-optimal surface for the scalarized problem. Thus, the tradeoff between two “competing” objective functions is quantified. In particular, it captures the power verses delay (or throughput) tradeoff for multiple access networks. The problem in equation (2.5) can be named RAPC-1, as a variation of RAPC.

By introducing dummy variables \( t_P \) and \( t_{\phi_P} \) the problem in (2.5) can be converted to the following form

\[ \text{Minimize} \quad t_P + \lambda \cdot t_{\phi_P}^{-1} \]

subject to

\[ P_i \cdot t_P^{-1} \leq 1 \quad \text{and} \quad 1 \leq i \leq N \]
\[ t_P \cdot P_{\text{max}}^{-1} \leq 1 ; \]
\[ t_{\phi_P} \cdot \phi_i^{-1} \leq 1 \quad \text{and} \quad 1 \leq i \leq N \]
\[ \phi_i \leq 1 \quad \text{and} \quad 1 \leq i \leq N \]
\[ \gamma_0 \cdot \gamma_i^{-1} \quad \text{and} \quad 1 \leq i \leq N \]

where \( \gamma_i^{-1} = P_i^{-2} \sum_{j=1}^{N} P_j G_i^{-2} \rho_{ij}^2 Q_j + \sigma^2 P_i^{-1} G_i^{-1} \) \hspace{1cm} (2.7)
is a polynomial, and the formulation in (2.6) is in the form of a Geometric program (GP) (Cagalj et al. 2004). Positivity constraints are implicit in a GP. Now a \((2N+1) \times 1\) vector is defined as \(X = [x_1, \ldots, x_{2N+2}]^T\), where

\[
x_i = \begin{cases} 
\log(P_i) & 1 \leq i \leq N \\
\log(\varphi_i) & N + 1 \leq i \leq 2N \\
\log(t_p) & i = 2N + 1 \\
\log(t_{\varphi}) & i = 2N + 2 
\end{cases}
\]

(2.8)

where \(\log(.)\) denotes the natural logarithm. Combining (2.8) with (2.6), and taking the logarithm of the objective function and constraints, the following convex optimization problem is obtained:

Minimize \(\{\log(\exp(x_{2N+1}) + \lambda \exp(-x_{2N+2}))\}\)

(2.9)

Subject to

\[x_i - x_{2N+1} \leq 0, \quad 1 \leq i \leq N\]

\[x_{2N+1} - \ln(P_{\text{max}}) \leq 0\]

\[x_{2N+2} - x_i \leq 0, \quad N + 1 \leq i \leq 2N\]

\[x_i \leq 0, \quad N + 1 \leq i \leq 2N\]

where \(\exp(.)\) denotes the exponential function. The convexity of the problem follows from the convexity of the \textit{affine} and \textit{log-sum-exp} functions.

Considering RAPC-1, within the above framework, three more variants of RAPC-1 can be formulated. The motivation behind studying these variants is to investigate the effect of reducing the degrees of freedom on the efficacy of RAPC-1.
2.2.1 RAPC2 (Different power level with a fixed probability of transmission)

In the first variant, RAPC-2, the nodes are allowed to have different transmit power levels, but it is forced to have a common probability of transmission. Thus, RAPC-2 can be formulated as

\[
\text{minimize } \|P\|_\mathcal{X} + \frac{\lambda}{\phi} \\
\text{subject to } \\
0 \leq P \leq P_{\text{max}} \\
0 \leq \phi_i \leq 1 \\
\gamma_i \geq \gamma_0, \quad \text{and} \quad 1 \leq i \leq N
\]

where \( \phi \) is a scalar, and in the definition of \( \gamma_i \) in (2.7) let \( \phi_i = \phi \ \forall \ i. \)

2.2.2 RAPC3 (Common power level with variable probability of transmission)

Another variant of this class of schemes is RAPC-3, where the nodes are allowed to transmit with different probabilities, but are forced to transmit at a common power level. RAPC-3 can be formulated as

\[
\text{minimize } \|P\|_\mathcal{X} + \lambda \left\| \frac{1}{\phi_i} \right\|_\mathcal{X} \\
\text{subject to } \\
0 \leq P \leq P_{\text{max}} \\
0 \leq \phi_i \leq 1 \\
\gamma_i \geq \gamma_0, \quad \text{and} \quad 1 \leq i \leq N
\]
where $P$ is a scalar, and in the definition of $\gamma_i$ in (2.7) let $P_i = P \forall i$.

### 2.2.3 RAPC4 (Common power level with a common probability of transmission)

In yet another variant, RAPC-4, all nodes are assigned a common power level and transmission probability. RAPC-4 can be formulated as

\[
\text{minimize} \quad P + \frac{\lambda}{\varphi}
\]

\[
\text{subject to}
\]

\[
0 \leq P \leq P_{\text{max}}
\]

\[
0 \leq \varphi \leq 1
\]

\[
\varphi_i \geq \varphi_0, \quad \text{and} \quad 1 \leq i \leq N
\]

where $P$ and $\varphi$ are scalars, and in the definition of $\gamma_i$ in (2.7) let $P_i = P$, $\varphi_i = \varphi \forall i$. RAPC-2,3 and 4 can be easily cast into geometric programs, and hence into convex optimization problems by following a procedure similar to that for RAPC-1. Therein lies an important benefit of these kinds of schemes.

### 2.2.4 Geometric Programming Applied to RAPC

The primal GP is:

\[
\text{(GP)} \quad V_{GP} := \text{minimize} \quad g_0(t)
\]

\[
\text{subject to} \quad g_k(t) \leq 1, \quad k = 1, 2, \ldots, p
\]

\[
t_i > 0, \quad i = 1, 2, \ldots, m
\]

\[
(2.13)
\]

where $g_0(t) = \sum_{j=1}^{n_0} c_j t_1^{a_{l_j}} \ldots t_m^{a_{m_j}}$
\[ g_k(t) = \sum_{j=1}^{n_k} c_j t_1^{a_{j1}} \cdots t_m^{a_{mj}}, \quad k = 1,2,\ldots,p \] (2.15)

Given exponents \( a_{ij} \) for the \( i \)th variable in the \( j \)th product term, \( i=1,\ldots,m \) and \( j=1,\ldots,n_p \), are arbitrary real constants and term coefficients \( c_j \) are positive. Here, \( g_0(t) \) is called the objective function and \( g_k(t) \) the \( k \)th constraint function, where \( n_0 \) is the number of product terms in the objective function and \( (n_k-n_{k-1}) \) is the number of product terms in the \( k \) constraint function. Therefore, the number of the total product terms is \( n_p \), where \( p \) is the number of constraint functions. The vector \( t \) contains \( m \) variables, denoted by \( t_1,\ldots,t_m \).

The program solves polynomial GP, and also removes the polynomial with this understanding. The dual to GP (2.15) is

\[
\text{(GD)V}_{GD}: \quad \text{maximize} \quad \prod_{j=1}^{n_p} \left( \frac{c_j}{X_j} \right)^{\lambda_j,}\prod_{k=1}^{p} \lambda_j \lambda_k
\]

Subject to \( \sum_{j=1}^{n_0} x_j = 1 \)

\( \sum_{j=1}^{n_p} x_j a_{ij} = 0, \quad i = 1,2,\ldots,m \)

\( x_j \geq 0, \quad j=1,2,\ldots,n_p \) (2.16)

Where \( \lambda_k = \sum_{j=1}^{n_k} x_j, k = 1,2,\ldots,p \) (2.17)

For a (primal) GP having \( m \) variables \( (t_i) \), \( p \) constraints and \( n_p \) (polynomial) product terms. It is seen that (dual) GD has \( n_p \) non-negative variables \( (x_j) \) in \( m+1 \) linear equations. In the literature the degree of difficulty of a GP is defined by:
degree of difficulty = \( n_p - m - 1 \).

Let \( F(x) \) denote the negative of the logarithm of the objective function of \((GD)\), i.e.,

\[
F(x) = \sum_{j=1}^{n_0} x_j \ln \left( \sum_{k=1}^{c_{ij}} p + 1 x_j \ln \sum_{j=n_{k-1}}^{n_k} c_{j,k} \right)
\]  

(2.19)

It is seen that \((GD)\) is a linearly constrained convex programming problem,

\[
\text{minimize} \quad F(x) \\
\text{subject to} \quad A x = b, \quad x \geq 0,
\]

where the coefficient matrix is given by

\[
A = \begin{bmatrix}
1 & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 \\
& a_{1,1} & \cdots & a_{1,n_0} & a_{1,n_0-1} & \cdots & a_{1,n_1} & \cdots & a_{1,n_{p-1}} & \cdots & a_{1,n_p} \\
& \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
& a_{m,1} & \cdots & a_{m,n_0} & a_{m,n_0-1} & \cdots & a_{m,n_1} & \cdots & a_{m,n_{p-1}} & \cdots & a_{m,n_p}
\end{bmatrix}
\]  

(2.20)

and right side hand

\[
b^T = (1, 0, ..., 0) \in \mathbb{R}^{m+1}
\]  

(2.21)

Note that the first \( n_0 \) entries of the first row of \( A \) are ones, and data \( a_{ij} \) are stored from row 2 to row \( m+1 \) in \( A \).

### 2.3 MMANET SIMULATION MODEL DEVELOPMENT

Based on the dual requirements of power control and traffic carrying capacity, two types of models are developed in this work.
The first model is called the Power Control simulation Model (PCSM) and is developed using MATLAB.

The second model is called the Transmission Simulation Model (TSM) and is developed using NS2.

Details of these models and associated models used in this work are presented in the following pages.

2.3.1 Graph Theory

Graph theory is one of the most powerful tools in wire line network research. In this theory, the network is mapped into a graph, with users corresponding to nodes and wired connections corresponding to edges (or arcs). The edges can be chosen to be unidirectional if the transmission is simplex, thereby creating a directed graph, or bi-directional when the transmission is duplex, creating an undirected graph. In general, all the links are associated with equal weights but this is not a necessary criterion to use this technique in network analysis. When interested in the per-link communication rates, then the links can be weighted (Diestel 2000). While these observations apply for wired networks, wireless networks pose additional problems. The links in wireless networks are not fixed but are interference limited, and the interference is, in turn, dependent on the transmission power of the nodes in the vicinity of the receiver/transmitter pair. To account for these features, a network model called the Physical Model (Gupta and Kumar 2000) is selected for the analysis presented here. Components in this model include the channel gain matrix that describes the propagation of signal rate from node to node. The channel gain matrix also takes into account the receiver function $f_R(.)$ that defines the capabilities of the receiver and acts as a “catch-all” for several of the receiver properties.
2.3.2 Simulation Model Components

The physical model represents an ad-hoc network of n nodes as a collection \(X_1, X_2, \ldots, X_n\). Each node is equipped with a transmitter, a receiver and an infinite buffer, capable of communicating with some or all of the other nodes, using single-step or multi-hop routing methods. Nodes are assumed to be simplex transmission modes meaning nodes cannot transmit and receive simultaneously. Each transmission occupies the entire allocated bandwidth \(W\) of the channel. It is assumed that the source node \(X_i\) is capable of transmitting with any power \(P_i \leq P_i^{\text{max}}\), where \(P_i^{\text{max}}\) is the maximum power of the node. Let the destination node, \(X_j\), be equipped with a receiver that is characterized by the receiver function \(f_R(\cdot)\). The channel gain between the nodes \(X_i\) and \(X_j\) is represented as \(G_{ij}\). Therefore, when \(X_i\) transmits with power \(P_i\), the node will receive the signal with a power \(G_{ij}P_i\).

In the case where there are \(n\) nodes and \(m\) gateways, the channel gain matrix is defined as a two-dimensional \(n \times m\) matrix \(G = [G_{ij}]\). The elements along the diagonal are unimportant since a simplex mode of transmission is assumed and they are always set to zero that is, \(G_{ii} = 0\). Also, another important assumption is made that the elements of the gain matrix are frequency independent, but allow the gain to be time varying.

The receiver of each node is designed to be subjected to thermal noise, background interference from various standard noise sources such as other networks, and interference from other users. It is to be noted that the interference caused by \(X_i\) to \(X_j\) is also determined by the link gain \(G_{ij}\). Thermal noise and background interference are jointly represented by the power spectral density \(\eta_i\) for node \(X_i\). Thus, the noise vector is defined as \(H = [\eta_1 \eta_2 \ldots \eta_n]^T\).
SINR is the ratio of the resultant signal received at the receiver node to the noise interference. Let \( \{X_t : t \in \mathbb{T}\} \) be the set of transmitting nodes at a given point of time. Each node \( X_t \) transmits with a power of \( P_t \). Let \( X_j \) be a node, where \( j \not\in \mathbb{T} \), attempting to receive the information from node \( X_i \), \( i \in \mathbb{T} \). Then the SINR at node \( X_j \) will be given by

\[
\gamma_{ij} \triangleq \frac{P G_i}{\sum_{j \in \mathbb{T}} P_j G_{ji} + W \eta_i}
\]  

(2.22)

where \( \eta_i \triangleq \sigma^2 \) and \( W \) is the bandwidth that was not considered in equation (2.4). The rate of transmission from node \( X_i \) to node \( X_j \) under the SINR \( \gamma_{ij} \) is given by

\[
R_{ij} = f_R(\gamma_{ij})
\]  

(2.23)

where the receiver function \( f_R(.) \) reflects the capabilities of the receiver and the total performance metric. These channel models reflect the parameters that can be fixed in the ensuing protocol simulations.

2.3.3 Rate Adaptation Model

As mentioned earlier, the receiver function can be set to account for the various operation characteristics of the receiver. For example, when the transmitter continuously adapts to the receiver SINR, then the receiver function becomes

\[
f_R = W \log_2 \left( 1 + \frac{1}{\tau} \gamma_{ij} \right)
\]  

(2.24)
In this equation, with \( \tau = 1 \) the receiver achieves Shannon’s capacity. With \( \tau > 1 \), the equation approximates the maximum data rate that meets a given BER requirement under a specific modulation and coding scheme. Until now, it has been assumed that the receivers used are capable of variable-rate adaptation, as defined by equation (2.24) with \( \tau = 1 \), meaning that transmitters automatically adjust the transmission rate to match the SINR at the receiver and achieve the Shannon bound. Such an assumption implies that the available rates are not restricted to a finite set of values. In this analysis, the effects of discrete-transmission rates are alone analyzed.

2.3.4 Common Transmission Rate Model

If all the nodes use a common rate, the receiving node \( X_j \) will be able to decode the signal from \( X_i \) with a negligible error provided the SINR \( \gamma_{ij} \) is constantly above a given threshold \( \gamma_T \), i.e., \( \gamma_{ij} \geq \gamma_T \). Otherwise, the signal is lost. However, this model is not taken up for analysis in this thesis.

2.3.5 Traffic Model of MMANET

The fundamental assumption that is made is that all the packets that are created in the simulation have a single destination – the primary gateway. Therefore, in this work, the cases of multicasting or broadcasting are not considered and are candidates for future work. Also, no assumptions are made on the traffic needs of the network.

The various channel models used in the simulation determine the positions of the nodes. The channel model also determines the propagation properties of the signal. They ultimately specify the gain matrix. Note however that the quality of the communication does not depend on the location of the nodes but on the power of the received signal, which is one of the most important characteristics of the model with one infrastructure
gateway and multiple routing gateways, with multiple nodes of traffic generators.

The routing task is performed in the model using either a proactive or a reactive scheme, or a combination of both. Examples of proactive protocols that have been suggested for wireless networks are OLSR, DSDV, and WRP. Examples of reactive protocols include DSR, LAR, RDMAR, AODV, and TORA. Hybrids incorporating a combination of both these protocols can be found in ZRP and LANMAR (Bambos and Kandukuri 2002).

From the discussion above, the current analysis is required to be carried out in two phases.

- In the first phase, addresses issues relating to handling transmitter based power control, rate of transmission and receiver based bit error rate (BER) dependence on SINR. The media or the channel will be simulated with standard packages available. This phase simulated with MATLAB, and is called the power control simulation model (PCSM).

- In the second phase, the questions regarding handling traffic needs with capabilities of the nodes and a modified MAC protocol implementations. The second phases is simulated with network simulator (NS2) version 2.29 models and are called transmission simulation models (TSM).

Each of these models is explained in detail in the following sections.
2.4 POWER CONTROL SIMULATION MODELS

As mentioned earlier, the power control algorithm is one of the most crucial aspects of this work. The simulation of power control is performed using a toolkit developed in MATLAB / SIMULINK. The MATLAB toolkit consists of four main modules:

- The transmitter module
- The receiver module
- The channel module
- The measuring devices module

The parameters used in the PCSM block of the simulations are listed in table 2.1. This block is used to change dynamically the parameters of PCSM at run time.

Table 2.1 Power Control Simulation Model (PCSM) Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Data 1</th>
<th>Data 2</th>
<th>Data 3</th>
<th>Data 4</th>
<th>Data 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode/Data Rate</td>
<td>1 Mbps</td>
<td>2 Mbps</td>
<td>5.5 Mbps</td>
<td>11 Mbps</td>
<td>11 Mbps</td>
</tr>
<tr>
<td>Packet Size</td>
<td>256 bits</td>
<td>512 bits</td>
<td>1024 bits</td>
<td>2048 bits</td>
<td>4095 bits</td>
</tr>
<tr>
<td>Channel Numbers</td>
<td>1 to 11</td>
<td>1 to 11</td>
<td>1 to 11</td>
<td>1 to 11</td>
<td>1 to 11</td>
</tr>
<tr>
<td>Channel type</td>
<td>AWGN/ Raleigh</td>
<td>AWGN/ Raleigh</td>
<td>AWGN/ Raleigh</td>
<td>AWGN/ Raleigh</td>
<td>AWGN/ Raleigh</td>
</tr>
<tr>
<td>Channel Sig/noise ratio (Es/N)</td>
<td>-30 to 100 db</td>
<td>-30 to 100 db</td>
<td>-30 to 100 db</td>
<td>-30 to 100 db</td>
<td>-30 to 100 db</td>
</tr>
</tbody>
</table>
A detailed schematic of the simulation model developed for PCSM is shown in Figure 2.3. The parameters selected are displayed below the system parameter box. Here, six adjustable parameters are displayed, namely, the

- Rate of transmission
- Packet size
- Short preamble
- Channel number
- Channel type
- Signal to Noise power (Es/No)

![System Parameters Diagram](image)

**System Parameters**
- Rate = 11Mbps
- Packet Size = 1024 Bytes
- Short Preamble is off
- Channel Number = 11
- Channel Type = AWGN
- Channel Noise (Es/No) = 10 dB

**Wireless model**
**Data Rates of**
1Mbps/2Mbps/5.5Mbps/11Mbps

![Network Diagram](image)

**Figure 2.3 A Wireless Power Control Simulation Model**
2.4.1 PCSM Model Terminologies

Before looking into some the details of the simulations, the terminology and their definitions used in the module are provided here

- **Network number**: A unique identifier for the wireless network block
- **Number of nodes**: Number of nodes connected to the network, affects the size of send, receive and schedule inputs and outputs of the wireless block
- **Data rate**: Transmission speed of the network.
- **Minimum frame size**: A message shorter than this number will be padded
- **Transmit power**: According to the standard, the transmission power is limited to a maximum of 1000 mW in USA, 100 mW in Europe and 10 mW in Japan
- **Receiver signal threshold**: If the received energy is above this value, then the medium is regarded as busy
- **Path-loss exponent**: The number ‘\(\lambda\)’ can lie between 2 and 4
- **ACK timeout**: The time a sending node will wait for an ACK message before retransmitting it
- **Retry limit**: The maximum number of times a node will try to retransmit a message before giving up
- **Error coding threshold**: A number in the interval [0, 1] which defines the percentage of block errors in a message that the coding can handle. For example, certain coding schemes can fully reconstruct a message if it has less than 10% block errors.
2.5 **PCSM TRANSMITTER MODULE**

The Sub-modules of PCSM transmitter module are schematized in figure 2.4. This module consists of 5 Sub-modules, namely,

- PCSM source data generator (random integer generator)
- Framing and preamble circuit, and pulse shaping circuit
- VAR block and power measurement
- modulate and spread
- Pulse shaping circuit

![Figure 2.4 Transmitter Module of PCSM](image)

**2.5.1 PCSM Source Data Generator Sub-module**

The source data is generated using a random data signal generator and is constructed such that the output bit pattern is random. This sub-module generates random uniformly distributed M-ary integers in the range [0, M-1]. The output of this block is a set of random bytes. The parameters used in this sub-module are given below.
Table 2.2  Source Data Generator Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Starting of Range</th>
<th>increment</th>
<th>Ending of Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-ary number</td>
<td>001</td>
<td>+1</td>
<td>255</td>
</tr>
<tr>
<td>Initial seed</td>
<td>12345</td>
<td>+2</td>
<td>99999</td>
</tr>
<tr>
<td>Sample time</td>
<td>PPDU_frame period</td>
<td>Packet size</td>
<td>max</td>
</tr>
<tr>
<td>Samples per frame</td>
<td>1024</td>
<td>128</td>
<td>4095</td>
</tr>
</tbody>
</table>

*Preamble Protocol Data Unit [PPDU]*

2.5.2  Converter Sub-Module

This Sub-module converts bytes to bit blocks and maps a vector of integers to a vector of bits. The first bit of the output vector is the most significant bit (MSB). The number of bits per integer value defines how many bits are mapped from each integer. The input can be either a scalar or a frame based column vector. The parameter that can be passed is the number of bits per integer. The default value is 8 bits.

2.5.3  Framing and Preamble Circuit Sub-module

This block is built with 6 DSP constant routines, each of which has variable parameter consisting of constant value sample modes that are frame-based with a frame-period. This parameter gives the simulator the freedom to select the overhead bits on the simulation. Also, each slot timing can be varied. A set of concatenation matrix blocks are used to merge data (refer to Figure 2.5 and Table 2.3).
2.5.4 PCSM Modulate and Spread Sub-module

Before the signal reaches this module, the transmit bits are to be tapped. The next block is modulated and spread, which then selects the data rate from the system block and modulates the data from the source. It is possible to implement different channel data rates in this module.
2.5.5 PCSM Pulse Shaping Circuit Sub-module

As shown in Figure 2.26, the pulse shaping filter consists of a Root Raised Co sinus (RRC) with a roll-off factor of 0.22. The length of the FIR filter is determined by 96 coefficients. This module contains the following parameters: number of filter taps, RRC roll-off factor and over-sampling factor that defines the number of samples per chip. Table 2.4 shows the parameter settings that were used in these simulations.

![Figure 2.6 PCSM Pulse Shaping Sub-module](image)

**Table 2.4 PCSM Pulse Shaping Sub-module Parameter table**

<table>
<thead>
<tr>
<th>Number of filter Taps</th>
<th>Filter order</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRC Roll-off Factor</td>
<td>.3 to .9</td>
</tr>
<tr>
<td>Over sampling factor</td>
<td>sample_per_chip</td>
</tr>
<tr>
<td>Sample weight</td>
<td>1 to 9</td>
</tr>
</tbody>
</table>

2.5.6 PCSM Variance Block and Power Measurement Sub-module

This is the variance block that is used to calculate the variance of the vector elements. If the switch, “running variance” is selected, the block returns the variance of the input elements over a period of time. Settings on the “fixed-point” can only applied when the block inputs are fixed point
signals. Transmitter power measurement is done using a numeric display of input values. Formats of short, short_e, hex etc. can be used. Similarly, sampling time can also be set in this modules.

2.6 PCSM RECEIVER MODULE

The receiver modules are drawn in Figure 2.7. The received signal power level is measured with the numeric meter. The center frequency mixer is used to condition the signal for the filter function. Receiver pulse shaping filter carries some important design parameters and is set with care.

Figure 2.7 PCSM Receiver Modules

The receiver power measurement block measures the signal power received at the transmitter after wave propagation through the channel module. This module consists of 6 sub-modules, namely,

- Rx front sub-modules
- Sync to chip sub-modules
- Symbol boundary/frame boundary alignment sub-modules
• Demodulate and despread sub-modules
• Deframing sub-modules

2.6.1 PCSM Receiver Front

Figure 2.8 shows the receiver front. This block consists of 3 modules:

• Mixer module
• Pulse shaping filter
• Gain amplifier

Mixer module is used when an IF is required to be generated. The pulse shaping filter is similar to the filter at the transmitter end.

![Figure 2.8 PCSM Receiver Front Sub-module](image)

<table>
<thead>
<tr>
<th>Table 2.5 PCSM Receiver Filter Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of filter Taps</td>
</tr>
<tr>
<td>RRC Roll-off Factor</td>
</tr>
<tr>
<td>Over sampling factor</td>
</tr>
<tr>
<td>Sample weight</td>
</tr>
</tbody>
</table>
Table 2.6 PCSM Receiver Gain Amplifier Parameters

<table>
<thead>
<tr>
<th>Gain</th>
<th>1/Samples_per_chip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplication</td>
<td>Element-wise[K*u] or Matrix</td>
</tr>
<tr>
<td>sample time</td>
<td>Inherited/External</td>
</tr>
<tr>
<td>Signal Data Type</td>
<td>via dialogue/back propagation</td>
</tr>
<tr>
<td>Calculation</td>
<td>Floor/ceiling</td>
</tr>
</tbody>
</table>

2.6.2 PCSM Sync to Chip Sub-module

Figure 2.8 shows the PCSM sync-to-chip sub-module. This module consists of delays for discrete-time input that can be varied by fixed integer number of sample periods. The module is down-sample block that uses sampling techniques with a sample factor of k and sample offset of k-1. The module operates on a sample based mode.

![Figure 2.8 PCSM Sync to Chip Sub-Module](image)

Figure 2.9 PCSM Receiver Sync to Chip Sub-Module

2.6.3 PCSM Symbol Boundary/Frame Boundary Alignment Sub-module

Figure 2.9 shows the receiver sync to chip sub-module. An integer delay module is used to insert discrete time input delays by a fixed integer
number of sample periods. The adjustable parameters are the delay [filter_delay_chips_correction] with initial conditions and reset port raising/falling edge or none.

### 2.6.4 PCSM Demodulate and Despread Sub-module

Figure 2.10 shows the signal concatenation techniques. This module demodulates with input as the chips and PPDU as the output. The output is tapped to provide input to the measurement block. A concatenation matrix is used for inter-connection of transmit modules.

![Figure 2.10 PCSM Signal Concatenation Block](image)

**Figure 2.10 PCSM Signal Concatenation Block**

### 2.6.5 PCSM Measuring Module

Figure 2.11 shows the measuring devices module. This module consists of several instrumentation modules.
Figure 2.11 PCSM Instrumentation Block showing Different Equipments

Figure 2.12 PCSM Measuring Devices

Figure 2.12 shows the different measuring devices used in the simulation model.

Figure 2.13 shows the signals at the transmit modules for channel numbers 1 and 6.
Figure 2.13  PCSM Channel Signal at the output of the Transmitter
2.7 PCSM MODEL NUMERICAL PARAMETERS

The list of numerical parameters that can be used with the PCSM model during the execution of the simulations are as follows. These parameters are tunable depending upon the type of network that is being designed:

a) The rate of transmission was varied between 1 Mbps, 2 Mbps, 5.5 Mbps and 11 Mbps. The rate of adaptation (R_b) is based on receiver feedback implemented in the software.

b) Packet sizes were selected from 512, 1024, 2048 and 4095 bits. It was found that larger packet sizes improved the throughput of the network if the frame rate was less than 0.08. Larger packet sizes were also more useful in networks with plenty of overhead traffic due to routing requirements. However, when the system that is very dynamic might be better off using smaller packets due to increased packet loss in a dynamic environment.

c) Signal power levels were varied between 1.0 mW, 2.0 mW, 3.45 mW, 4.8 mW, 7.25 mW, 10.6 mW, 15.0 mW, 36.6 mW, 75.8 mW and 281.8 mW. The corresponding ranges for each of these power levels (P_i) (obtained using two-ray model simulations) were 4 m, 6 m, 8 m, 9 m, 10 m, 11 m, 12 m, 15 m, 18 m and 25 m respectively. The task of the power control algorithm is to choose one of these power levels to create an optimal connection.

d) The Additive White Gaussian Noise (AWGN) channel model was used uniformly throughout the simulations.

e) Receivers were designed with differing signal to noise interference ratio (SINR) of -92 dBm, -90 dBm, -85 dBm, and
-80 dBm for each of the aforementioned data rates respectively ($R_b$). The signal to noise ratio was varied between -10 to 18 dB ($E_b/N_0$).

f) MAC module of the link layer was modified in the design of new optimizing techniques in power control 4 levels ($P_i$) resulting in a MANETMAC protocol.

g) The MANETMAC was operated as a reactive protocol. Past studies have shown that reactive protocols improve capacity in small networks.

2.8 TRANSMISSION SIMULATION MODEL

The simulation in the earlier section is designed to calculate the transmit power for a set of given channel conditions, and BER at the receiver. The Transmission Simulation Model (TSM) model is next used to simulate the node mobility, rate of connection and scheduling. In the design of the model topology, mobility parameters and queue size at the nodes are assigned as scene parameters.

The TSM simulations from NS2 are obtained in the form of a trace file. The results in the trace file are analyzed using PERL, a report generator language designed to extract string and patterns from files and xgraph for screen display.

2.8.1 TSM Simulator

The Network Simulator (NS2) is the simulation tool used for the simulations of mobile ad-hoc nodes in this work. It is an object-oriented simulator, written in C++, with an OTcl interpreter as a front-end and is free to use. The operating environment required is UNIX like. The simulations
here were run on a windows machine with a cygwin UNIX emulator. The
simulator defines a class hierarchy and also compiled hierarchy in C++. The
two OTcl and C++ hierarchies are closely related to each other.

2.8.2 TSM Link Layer

The link layer object is responsible for simulating the data link
protocols. Many protocols can be implemented within this layer including
packet fragmentation and reassembly, and the reliable link protocol. Another
important function of the link layer is to set the MAC destination address in
the MAC header of the packet. In the current implementation, this task
involves two separate issues: finding the next–hop–node’s IP address
(routing) and resolving this IP address into the correct MAC address (ARP).
Normally for all outgoing (into the channel) packets, the packets are handed
down to the LL by the routing agent. The LL hands down packets to the
interface queue. For all incoming packets (out of the channel), the MAC layer
hands up packets to the LL, which is then handed off at the node entry point.
2.8.3 TSM Routing Layer

The Address Resolution Protocol (ARP) module receives queries from the link layer. If ARP has the hardware address for the destination, it writes this address into the MAC header of the packet. Otherwise, it broadcasts an ARP query, and caches the packet temporarily. For each unknown destination hardware address, there is a buffer for a single packet. In the event that additional packets to the same destination are sent to ARP, the earlier buffered packet is dropped. Once the hardware address of a packet’s next hop is known, the packet is inserted into the interface queue.
2.8.4 TSM Queue at the Gateways

The *Drop Tail Queue* class is implemented as the interface queue that maintains the order in which the packets arrive. Whenever the queue gets filled, the newly arriving packets get lost.

2.8.5 TSM Mac Layer

The distributed coordination function (DCF) MAC protocol has been implemented at the mobile nodes. It uses a RTS/CTS/DATA/ACK pattern for all unicast packets and simply sends out DATA for all broadcast packets. The implementation uses both physical and virtual carrier sense. The PCF has been implemented at the gateway nodes.

2.8.6 TSM Network Interfaces

The network interface serves as a hardware interface which is used by the mobile node to access the channel. The wireless shared media interface is implemented as the class *Phy/WirelessPhy*. This interface, subject to collisions and the radio propagation model, receives packets transmitted by other node interfaces to the channel. The interface stamps each transmitted packet with the meta-data related to the transmitting interface such as the transmission power, wavelength etc. This meta-data in packet header is used by the propagation model in the receiving network interface to determine if the packet has minimum power to be received and/or captured and/or detected (carrier sense) by the receiving node. The model approximates the Direct Sequence Spread Spectrum (DSSS) modulation interface.

In the development of the TSM model using network simulator-2, the following assumptions are made:
- A mobile node definition that determines the spatial location of each node as a function of time. Different types of mobility are tried: linear, random and periodic patterns ($N_i$).

- Each source is designated as a constant bit rate source (CBR) ($R_b$).

- The topology definitions are then input as random, chain, star (Hub and spokes)

- Standard routing algorithms like the Destination Sequenced Distance Vector (DSDV), Dynamic Source Routing (DSR), Temporarily Ordered Routing Algorithm (TORA) and Ad-hoc On demand Distance Vector (AODV) algorithms are tested at the gateway nodes to see the effect of each on the throughput of the network.

Figure 2.15 shows a general schematic of ad-hoc nodes. The simulations were run on small networks with less than 32 nodes and with two nodes as the infrastructure gateways. The nodes were placed in an area of 400 m by 500 m.

![Figure 2.15 Schematic of the TSM Development Model](image-url)
In Figure 2.15 the node marked 2 is the gateway node and node marked 3 is the sink node equivalent to the internet sink. So that the traffic carried by the link 2-3 measures the throughput of the network.

2.9 TSM TRAFFIC DESIGN NUMERICAL PARAMETERS:

The number of nodes in the network was chosen from the set of \[8, 12, 16, 24, 32\] for different simulation runs. The nodes in the ad-hoc network are confined to a 400 unit \( \times \) 400 unit square region. Initial locations (x and y coordinates, z coordinate = zero) of the nodes are obtained using a uniform distribution. Each mobile is assumed to move continuously, without pausing at any location. Each node moves with an average speed of \( v \). The actual speed is uniformly distributed in the range \( v - \alpha \) and \( v + \alpha \) units/second, where, \( \alpha = 1.5 \), when \( v < 15 \) and \( \alpha = 2.5 \), when \( v > 15 \) average speeds of \( v \) in the range 1.5 to 32.5 units/sec. Each node makes several “moves” during the simulation. A node does not pause between moves. During a given move, a node travels a distance \( d \), where \( d \) is exponentially distributed with mean 20 units. The direction of movement for a given move is chosen randomly. For each such move, for a given average speed \( v \), the actual speed of movement is chosen uniformly distributed between \( |v - \alpha|, v + \alpha \). If during a move (over chosen distance \( d \)), a node “hits” a wall of the 400×400 region, the node bounces and continues to move after reflection, for the remaining portion of distance \( d \). Two mobile hosts are considered disconnected if they are outside each other’s transmission range. The transmission range is decided based on the simulation results of section 2.6. All nodes have the same transmission range. For the simulations, transmission range values of 40, 60, 80, and 90 units were used. The wireless links between mobile nodes have 1,2 Mbps. In this simulation, simulation time is inversely proportional to the average speed. For instance, simulations for average speed 1.5 unit/sec run 5000 seconds of execution, whereas about 2000 seconds for average speed 4.5 units/sec. As the average speed is increased, for a given simulation time,
the number of moves simulated increases. Thus, although the simulations at different speeds are for the same mobility model, as speed is increased, a particular configuration (for instance, partition) that may not have occurred at a lower speed can occur at the higher speed. On the other hand, a configuration that did occur at a lower speed lasts a shorter time when the speed is higher.

For the simulation, a sender and a destination gateway are chosen randomly. Any data packets that cannot be delivered to the destination due to a broken route are simply dropped and are treated as lost. Under the circumstances a retry limit is applied. The source generates 10 data packets per second (on average), with the time between two packets being exponentially distributed. The data rate was chosen low to speed up the simulation. However, this has the impact of sending small number of packets between two route discoveries (as compared to when the source continuously sends packets). This, in turn, results in higher number of routing packets per data packet.

To overcome the partial visibility problem that Mobile Ad-Hoc Networks (MANET) exhibit, data link-level (level 2 OSI) routing protocols are considered. Examples of such protocols in popular use are the Ad-hoc On-Demand Distance Vector (AODV) routing protocol reactive and the Destination-Sequenced Distance Vector (DSDV) routing protocol proactive. A common feature these protocols share is its complexity. Another consideration is that LAN are massively deployed and its protocols and applications are robust and well known. An Ethernet LAN characteristic is the total visibility issue, that is, a broadcast frame reaches all members of the LAN, however, IEEE 802.11x MANETs do not have it. The proposed MMAC protocol is a link-level layer protocol that transforms for the higher layers a 802.11x MANET into an Ethernet LAN, maintaining compatibility with other protocols.
The DSR routing agent is in the sender node’s entry point, forcing all packets received by the node to be handed down to the routing agent. This model is required for implementation of piggy-backed routing information on data packets which otherwise would not flow through the routing agent. The DSR agent checks every data packet for source-route information. It forwards the packet as per the routing information. All nodes that participate in forwarding this reply to the source node create a forward route to destination. This state created from each node from the source to destination is a hop-by-hop state and not the entire route as is done in source routing.

The data values needed for this analysis like the packet delivery ratio and the end-to-end delay are evaluated from this trace file. In order to evaluate end-to-end delay, the packets that are originating from various sources with their identification are recorded with their time stamp. Whenever the same identification with a receive operation is found in the trace file, the time stamp for the event is obtained and the time difference is evaluated. This information is collected from all packets that have reached the destination and then the mean of the time is evaluated as the average end-to-end delay. To evaluate the packet delivery ratio, the number of packets originating from the various sources and the number of packets that have successfully reached the destination are counted. These two values are used for getting the packet delivery ratio. Figure 2.16 and Figure 2.17 show the model in operation.
Figure 2.16 TSM Packet Dropping in the Queue of Node 2.

Figure 2.17 TSM Gateway and Mobile MMANET
2.10 CROSS LAYER DESIGN

While the conclusions in this thesis are substantiated with numerical simulations, some of the aspects of cross-layer design (CLD) are exemplified using a formal theoretical basis in the following pages.

A wireless ad-hoc network consisting of $N$ nodes, and a set $\{L\}$ of logical links with a finite capacity of $C_i$ bits per second is considered. The set $\{L\}$ of links connecting the nodes of the network are assumed to be directed. In addition, the connectivity is assumed to be symmetric implying that

$$l_{(i,j)}\in\{L\} \quad \text{if and only if} \quad l_{(j,i)}\in\{L\} \quad (2.25)$$

Further, a static topology is assumed, where the nodes are considered immobile for a fixed period of time. Finite capacity links imply an implicit mechanism for restricting the transmission rate through each channel. Otherwise, problems such as collisions would not occur. To simplify the mathematical formulation, the small variations in the link capacity depending on external variables (like the weather) is not accounted. This assumption is entirely justifiable since the network parameters (and performance) change on a time-scale that is far shorter than the rate at which capacities change.

Clearly, wireless channels are shared media and are interference-limited, where links contend with one another for exclusive access to the channel. The collisions that arise out of this competition are handled using a conflict graph in the present model. It turns out that the maximum feasible rate region at the link layer is the convex hull of the corresponding rate vectors of independent sets within this conflict graph. In order to include the effect of rate constraints at the network layer, a multi-commodity flow variable that corresponds to the link capacities allocated to the flows towards different destinations is considered. Under these assumptions and
simplifications, the resource allocation problem reduces to a utility maximization problem with schedulability and rate constraints.

2.10.1 CLD Schedulable Constraints and Rate Constraints

As discussed earlier, the network is constantly striving to resolve the interference in its links. The following evident constraints dictate the operation of the network:

- A node that shares a common link cannot transmit or receive simultaneously
- Links that do not share nodes can transmit/receive. Such nodes are called independent nodes.

An interference model is used to simulate the competition for the transmission medium in a wireless network where multiple channels are available for transmission (Kodialam and Nandagopal 2003). For example, simultaneous communications in a 802.11 compatible Network Interface Cards (NIC) are allowed over 12 channels.

Under the interference model defined above, a conflict graph (Jain et al 2003) is built that captures the instantaneous contention relationship amongst the links. In the conflict graph, each vertex represents a link, and an edge between two vertices denotes the contention between the two corresponding links. To avoid errors, these links cannot transmit at the same time. For example, figure 2.18 shows an example of a wireless ad-hoc network and its conflict graph.
2.10.2 Network Model

An illustration of the nodal for a network consisting of four nodes is provided here. A chain topology is employed with the links numbered from 1 to 6. Each of the links is symmetric and the links are shown in figure 2.19.

From this diagram, one could create a new set of nodes consisting of independent sets of vertices. The elements of this set do not contain any common edges. For example, the set of nodes \{1,5\} are independent. In
general, it is assumed that the set of independent nodes, as the set \{e\}. An independent set \{e\} is an \(|L|\)-dimensional rate vector \(r^e\), where the \(i^{th}\) entry is

\[
\begin{align*}
    r_i^e &:= C_i & i \in \{e\} \\
    r_i^e &:= 0 & i \notin \{e\}
\end{align*}
\]

and where \(C_i\) is the capacity of the \(i^{th}\) link.

The feasible region in the link layer referred to earlier, is defined as the convex hull of these rate vectors \(r^e\) and is represented as \(\hat{R}\).

\[
\hat{R} := \left\{ r : r = \sum S_e r^e \text{ where} \begin{cases} S_e > 0 \text{ and} \sum S_e = 1 \end{cases} \right\} 
\]  \hspace{1cm} (2.27)

If \(y\) is the link flow vector, then according to schedulability constraint, \(y\) should belong to the region \(\hat{R}\).

\(S_e\) denotes the transmission schedule of the network at a given instant of time. Then, the transmission schedule of each source node during a period of time is fixed by these constraints. Now consider:

- \(D\) denotes the set of destination nodes of network layer flows
- \(f_{i,j}^k\) be the allocated capacity flow to destination \(k\)
- \(x_{i,j}^k\) be the flow generated from \(I\) to \(k\) through \(j\).

and

\[
x_{i,j}^k \geq 0
\]

Since the flow should not exceed the summation of the capacities of incoming and outgoing flows, the constraint
\[ x^k_{ij} \leq f^k_{ij} - \sum_{j \neq k} f^k_{jl} \]  \hspace{1cm} (2.28) \\

where \( i, \in \mathbb{N} \) and \( k, \in \mathbb{D} \) \( j \neq k \).

Thus, equation (2.28) is the rate constraint for resource allocation. It can be interpreted using the multi-commodity flow variable as it flows to different destinations across the network. Under this interpretation, the amount of link capacities allocated to the various destinations in the network during the schedule in question becomes the multi-commodity flow variable.

2.10.3 CLD Problem Formulations

A link is denoted by \( l \) and \( 1 \in L \). It is assumed there is at most one flow between any node and a destination pair \([i,k]\) (This restriction can be relaxed by induction in the general case). Let \( S_{i,k} \) denote a network layer flow between the source destination pairs \([I,k]\).

A utility function \( U \) is defined with the following properties

- continuously differentiable
- monotonically increasing
- strictly concave.

Let \( U_s(X_s) \) denote the utility attained by a source \( s \) at a source rate of \( x_s \). Thus the problem reduces to the selection of source rates \( X_s \) with allocated capacities \( f^k_{ij} \), such that they solve the global problem of rate maximization:

\[
\max_{x_s \geq 0, f_{ij} \geq 0} \sum_s U_s(x_s) \hspace{1cm} (2.29)
\]
This optimization problem is subject to

\[ x_{i,j}^k \leq \sum_{i,j,0} f_{i,j} - \sum_{i,j} f_{j,i} \]  \hspace{1cm} (2.30)

where the flow \( f \in \mathbb{R} \) \hspace{1cm} (2.31)

Solving the system problem (2.29)-(2.30) directly,

- Equation (2.29) states the net utility of the flow \( x \) from \( I \) to \( k \) via \( j \) is the variable to be maximized.
- Equation (2.30) states that the flow \( x \) should be less than or equal to the net channel capacities allocated.
- Equation (2.31) states that the flow \( x \) should belong to the region of flow.

To solve this system problems directly, the coordination among possibly all sources and links in the network is needed. Since (2.27) is a convex optimization problem with strong duality, distributed algorithms (Kunniyur and Srikant 1996, for example) can be used. Hence, the Lagrange dual problem is formed and also interpret the resulting algorithm in the context of joint design of congestion control, routing and scheduling. Typically, an algorithm of optimal flow control (Low and Lapsley 1999) is taken for analysis.

### 2.10.4 Formulation of Dual Algorithm for CLD

Let \( D(p) \) be the dual to the problem of equation (2.27) with the constraints of (2.28) and (2.29), then
Min $D(p) = \max \sum U_s(x_s) - \sum_{(k,i,j):L} p_i^k(x_i^k) - p_i^k \left( \sum_{(k,i,j):L} f_{i,j}^k - \sum f_{ji}^k \right)$

subject to $f \in \tilde{R}$ \hfill (2.32)

By introducing the Lagrange multiplier $p_i^k$ for the source $i$ and destination $k$, equation (2.32) can be decomposed into two problems

\begin{align*}
D_1(p) &= \max \sum_{x_s \neq 0} U_{s(xs)} - \sum P_{s(xs)} \hfill (2.33) \\
D_2(p) &= \max p_i^k \sum f_{i,j}^k - \sum f_{ji}^k \\
\text{subject to } f \in \tilde{R} \hfill (2.34)
\end{align*}

Here, $p_i^k$ is the congestion price for the first sub-problem in congestion control (Low and Lapsley 1999), and the second sub-problem is the routing and scheduling. Thus, the flow global optimization problem decomposes into separate “local” optimization problem of transport and network/link layer with congestion price as the interaction parameters. As is well known, local optimization problems are far more tractable than global optimization problems. Indeed, the difficulty of solving such problems in global maxima can be understood from the knowledge that the identification of global minima for a two dimensional function has not been solved thus far.

To address the congestion control, let us consider the equation (2.34) once again:

\begin{align*}
D_1(p) &= \max \sum_{x_s \neq 0} U_{s(xs)} - \sum P_{s(xs)} \hfill (2.35)
\end{align*}
A simple solution exists for \( x_s \) when LHS \( D_1(p) \) tends to zero (min)

\[
x_{s(p)} = u_s^{t-1}(p_s)
\]  

(2.36)

Equation (2.35) adjusts the source rate according to the congestion price of the source node. It should be noted that this is different from TCP that uses the aggregate price along its path.

Equation (2.34) can also be rewritten as

\[
p_i^k \left( \sum f_{i,j}^k - \sum f_{j,i}^k \right) = \sum f_{i,j}^k (p_i^k - p_j^k)
\]  

(2.37)

Then, the price adjustment for a pair of source destination pair \((i, k)\)
is given by

\[
p_i^k (t + 1) \left[ p_i^k (t) + \gamma_i (x_i^k p(t)) - \sum f_{i,j}^k (p(t)) - \sum f_{j,i}^k (p(t)) \right]
\]

where \( \gamma_i \) is a positive scalar step size. This step size is the most important parameter for optimization.

2.10.5 Operation of the CLD Algorithm

If the demand \( x_i^k (p(t)) \) for bandwidth at source node \( i \) for a flow to destination exceeds the effective capacity

\[
\sum f_{i,j}^k - \sum f_{j,i}^k , \text{ then the price } p_{ki} \text{ will raise, which in turn will reduce the demand and increases the effective capacity.}
\]

Thus, the dual algorithm motivates a joint congestion control, routing and scheduling design, wherein, at the transport layer sources ‘s’
individually adjust their rates according to the local congestion price and simultaneously, nodes $i$ individually update their prices according to equation (2.38). At the network/link layer nodes, the scheduling and route work/link layer data flows are then solved accordingly.

### 2.10.6 Numerical Model of CLD

Figure 2.20 shows a network of 6 nodes named A, B, C, D, E, F. A and B form a member of source set and E is the destination.

Two network layer flows are considered
- A to F
- B to E

with the utility function $U_s(x_s) = \log(x_s)$

All the channels are considered to be of a fixed capacity. This can be relaxed as an elastic channel. Therefore, the devices at the nodes become fixed rate devices. In turn, the links considered are of fixed capacity. The symmetric links $\{(C, E),(E,C)\}$ and $\{(B,F),(F,B)\}$ are of capacity 1 unit and all others are
considered to have 2 units. Note that although the topology is symmetric, two cases are considered that demonstrate different properties of the network depending on the routing and intermediate node pathways present.

Using this network, two algorithms are simulated, namely,

- Perfect scheduling
- Distributed scheduling

### 2.10.7 Perfect Scheduling of CLD

To evaluate the algorithm source rate and congestion price of each flow in perfect scheduling, the step size is fixed as $\gamma = 0.1$. Results from simulations that employ perfect scheduling are presented in chapter 3 in section 3.

### 2.10.8 Distributed Scheduling of CLD

When a distributed scheduling is employed, the evolution of source rates and congestion prices are similar to those of perfect scheduling. Similar to the previous case, the values converge quickly to a neighborhood of stable values. However, some prominent differences persist. These discussions are provided in chapter 3 and the differences are shown in chapter 3 (section 3 Figure 3.26).

In this sub-chapter, a model that can be used for the joint design of congestion control routing and scheduling for wireless ad-hoc networks is demonstrated. The results are derived under the framework of network utility maximization along with the application of dual-based decompositions. The formulation of two algorithms for fixed wireless channels (single rate wireless devices) as a utility maximization problem clearly shows the overall system degradation due to suboptimal design in one layer. This model can thus be
used to troubleshoot design issues in practical implementations. The rate optimization model can also be extended to elastic models and applied to the link layer.

2.10.9 Architectural Design of CLD-interface for Implementation

There are two types of interaction models that can be considered for CLD: (a) *synchronous models* and (b) *asynchronous models*. Another method of classification is using the private and public types of data in the protocols. Private data are shared synchronously and are collected internally during normal operations. The demand for private data occurs by issuing a query for data to other layers. Upon issuance of the query, the protocols wait for the result. On the other hand, asynchronous sharing is characterized by the occurrence of specified conditions to which protocols are willing to react. These conditions are occasional and random. In particular, despite the fact that the events are not deliberate, the protocols must expect their occurrence and return to continue their inherent functions. The CLD-interface is in turn responsible for delivering eventual random occurrences to the right layers. Thus, two types of events can occur, namely, *internal* and *external events*. The internal events are generated directly inside the protocols. For example, the routing protocol can identify a broken route and notifies the rest of the stack about this “broken route” event whenever it discovers the failure of a preexisting route. On the other hand, an external event is discovered inside the CLD-interface on the basis of the occurrence of an event provided by the subscriber protocols. An example of an external event is a condition on a ad-hoc node energy level. The external resident protocol will subscribe for a “battery-low” event specifying an energy threshold to the CLD-interface. In return, the interface will notify the physical layer protocol when the battery power falls below a given value and also prepares the transport layer routing protocol for this event. The ad-hoc controller simply provides the current battery level value, but it is not in charge of checking the threshold and for the
notification of related events. This is done by the CLD efficiently. Another example is the topology information collected by a routing protocol. This is more important for the link layer as in the case of MANETMAC, where the decision to fix gateways for a specific time frame is important. Thus, the CLD-interface becomes the provider of shared data that appears independent of its origin to other layer protocols, thereby, making it usable by each protocol.

As the CLD-interface represents a level of unidirectional treatment of cross-layer interactions, an agreement for a common representation of data and events inside the vertical component is a fundamental requirement to guarantee these loosely-coupled interactions. Thus, the CLD-interface works with abstractions of data and events, intended as a set of data structures that comprehensively reflect the relevant portions from cross-layering information interaction and special conditions used throughout the specific stack.

The event of exporting internal data into CLD-interface abstractions is accomplished by using call-back functions, which are well defined and installed by the protocols inherently. A call-back function is a procedure that is registered to a library at one point in time. The call-back function is invoked by the CLD-interface when required. Each call-back consists of instructions to encode private data into an associated CLD-interface abstraction. This is the most crucial opportunity extended by the protocol designers for transparently accessing protocol internal data.

2.10.10 Cross-layer Interface Design

The CLD interface is developed in object oriented ‘C++’ Language. This consists of data structures and functions as defined in any object oriented languages. For example, the following notation is typically adopted:
\textit{CLD\_object.method : \textit{(input)} → \textit{(output)}}

The CLD interface does not generate shared data but acts as an intermediary interface. Protocols synchronize on an abstract representation of internal data (namely CLD data) where one \textit{producer proprietary} protocol specifies a call-back function to export its private data to the abstract representation

\textit{CLD\_data.seize : \textit{(callback())} → ()}

But the \textit{consumer} protocols access the shared data with read only permissions using

\textit{CLD\_data.access : ()} → \textit{(abstract Data)}

In reference to the earlier example, routing agents play the role of the producer protocol, since they export routing tables data contents into an abstract graph representation. Thus, other consumer protocols resident in other layers could gather network topology information calling the method, (described above) which in turn can invoke the call-back function registered by the routing agent. This makes the interaction between producer and consumer protocols loosely coupled, avoiding direct protocol dependencies.

Another functionality of the CLD-interface is to cope with asynchronous interactions. In the case of internal events, the role of the CLD-interface is to collect interactions from external events, based on notifications, and on occurrence, vertically dispatch event occurrences to the specific subscribers. A protocol \textit{subscribes} for a cross-layer event (namely CLD event) by initiating the function

\textit{CLD\_event.subscribe : \textit{(handler())} → ()}

The subscriber protocol will specify a handler function, like in the case of a battery low event, to the CLD event handler that will then used by the CLD-interface to notify occurrences and to trigger event handling.
In short, subscriber protocols play again the consumer role, while producer protocol notify event occurrences by calling

\[ \text{CLD\_event.notify} : () \rightarrow () \]

The CLD-interface is in charge of maintaining a subscription list, like low receiver level threshold, high power interference etc., for each kind of cross-layer event, dispatching occurrences to the correct subscribers to take appropriate actions.

In the case of external events, like low battery, shadow region etc, the CLD-interface acts additionally as an event notifier also. The idea is that some protocols might be interested in conditions that are not directly verified by other protocols. To this end, subscriber protocols instruct the CLD-interface on procedures to detect the event. The detection rules are embedded in a monitor function, which periodically checks the status of the cross-layer abstractions under inquiry. When the monitor detects the specified condition, the CLD-interface dispatches the information to the subscriber protocol. A protocol initiates the monitoring of an external event by passing a monitor and a handler function to the XL-interface through the following method of the target data abstraction

\[ \text{CLD\_data.setMonitor} : (\text{monitor}.\text{handler}) \rightarrow () \]

The CLD-interface serves this call by spawning a persistent computation.

In order to implement CLD-interface, a simulation framework is essential and the best simulator available is the Network Simulator NS2 (v. 2.29), and a library of objects and abstractions, called ‘ns_lib_rout’. Together, they provide a set of simple, cross-platform C++ classes along with header files (.h) that allow the development of network protocols and applications. Such developments are relatively easy and have been proven several times in the past. Currently, the libraries support real platforms,
Linux flavors (Redhat, Suse, Mandrake etc), as well as in the NS2 simulation environment. The link state routing protocols are also available.

Figure 2.21 Cross Layer Data

Figure 2.22 Cross Layer Events

Figure 2.21 shows Cross Layer Data and Figure 2.22 shows events. Data and events make the model distinguish between known to unknown incidents. TCL hooks are used to achieve objective in simulation models.

Thus the objective of this chapter to develop a model to achieve a near real-time simulation of a small ad-hoc network is achieved. Also, in this chapter, the models used for simulations have been shown to be capable of being implemented in real world situations. The cross-layer designs and
results are also of significant importance based on the concepts of scheduling and allocation of resources for higher rate of transmission. With the framework constructed and implemented, the issue of the maximizing the network throughput is tackled next and the results are presented in chapter 3.