Traffic Estimation in mobile ad-hoc networks (MANET)
Syed Mushabbar Sadiq <ics-sym@dsv.su.se>

Abstract
Channel resources are one of the most limiting factors in a radio network. The Medium Access Control (MAC) protocol is responsible for how this radio resource is divided between mobile ad-hoc network nodes. It is important to give the different nodes their fair share. In order to do this, precise knowledge on the traffic situation is needed. If the slots allocation is done with knowledge of the traffic load, more efficient allocation results than just assuming uniform traffic can be achieved. A very important issue is therefore to make estimations on traffic flows that each node handles. In this research paper we look into traffic estimation methods that are applicable in mobile ad-hoc networks. We study existing methods and then present a traffic load estimation method based on an exponential decrease function. Furthermore, we look into two parameters; window size $T_{w}$ and gamma $\gamma$. We analyze and suggest fine tuning of these parameters for estimation of Poisson based traffic load. Finally, performance analysis on the basis of traffic load estimation error in percentage has been presented. This traffic estimation method based on exponential decrease function may also work for other traffic models if fine tuning of the above parameters have been made.

Table of Contents
ABSTRACT ......................................................1

1 INTRODUCTION ........................................1
1.1 RADIO RESOURCE ALLOCATION AND TRAFFIC RATES........................................2
1.2 AIM, MOTIVATION AND OUTLINE OF THE PAPER 3

2 TRAFFIC ESTIMATION METHODS ......3
2.1 ‘TRAFFIC MATRIX OF THE LOAD’ ON EACH LINK BASED ON THE NUMBER OF ROUTING PATHS USING A LINK........................................3
2.2 TRAFFIC ESTIMATION BASED ON NODE DEGREE................................................4
2.3 LIMITATIONS OF EXISTING TRAFFIC LOAD ESTIMATION METHODS .........................4
2.4 A PROPOSED TRAFFIC LOAD ESTIMATION METHOD........................................4

3 IMPLEMENTATION AND EVALUATION ISSUES...............................5
3.1 PERFORMANCE EVALUATION ...................... 5
3.2 SIMULATION ASSUMPTIONS AND CONSIDERATIONS ........................................7

4 SIMULATION RESULTS AND ANALYSIS OF TRAFFIC LOAD ESTIMATION ........................................7
4.1 TRAFFIC LOAD ESTIMATION FOR STATIC AD-HOC NETWORKS................................ 7
4.2 TRAFFIC LOAD ESTIMATION FOR MOBILE AD-HOC NETWORKS................................ 9

5 CONCLUSIONS AND RECOMMENDATION FOR FUTURE WORK ........................................ 13
5.1 RECOMMENDATION FOR FUTURE WORK 14

6 REFERENCES...........................................15

APPENDIX A: MOBILITY MODEL AND MAXIMUM RADIO TRANSMISSION RANGE. ...................... 15
APPENDIX B: UNIFORM POISSON TRAFFIC MODEL...........................................16

1 Introduction
Mobile Ad-hoc Networking (MANET) is one of the most important technologies that have gained interest due to recent advancements in both hardware and software techniques. MANET technology allows a set of mobile users equipped with radio interfaces (mobile nodes) to discover each other and dynamically form a communication network [1]. Other wireless communication systems tend to focus just on the mobility of user devices, not on the network itself. For example mobile phones and W-LAN (Wireless Local Area networks) terminals rely on some kind of fixed and pre-prepared infrastructure, whereas a mobile ad-hoc network is unplanned, self configurable and does not
require any help from fixed-installed communication network infrastructure.

MANET incorporates routing functionality into the mobile nodes so that they become capable of forwarding (relaying) packets on behalf of other nodes and thus effectively become the infrastructure. It can have different kinds of topology. Mobile nodes may be located on ships, trucks, cars, etc, and on very small portable devices. Due to the mobility the topology can change very quickly and unpredictably. Furthermore, new nodes can join and leave the network at any time. The network may get divided into sub-networks. A MANET can be envisioned to have dynamic, rapidly-changing, random, multi-hop topologies which are composed of relatively bandwidth-constrained wireless links.

![Figure 1. Mobile Ad-hoc Network (MANET)](image)

MANETs are of interest because they do not require any prior investment of fixed infrastructure. When a fixed backbone is not available, a readily deployable MANET can be used. In rural areas and third world countries, where basic communication infrastructure is not well established, MANET may be readily deployable at low cost.

For obvious reasons there has also been a great interest from military and defense research organizations in MANET and its applications. This work has been done in co-operation with The Swedish Defense Research Agency (FOI).

### 1.1 Radio resource allocation and traffic rates

In a MANET, topology changes results in changing traffic rates over the links in the network. In a radio network the channel resources are one of the most limiting factors, it is therefore very important that this resource is used as efficiently as possible. The Medium Access Control (MAC) protocol is responsible for how this resource is divided between the nodes (or links).

Two common MAC protocols used in mobile ad-hoc networks are Time Division Multiple Access (TDMA) and Spatial Time Division Multiple Access (S-TDMA). TDMA is a channel access scheme that divides the scarce radio resource into small time durations, known as time slots. These time slots are uniquely allocated to nodes such that only one node can transmit during its allocated time slots. S-TDMA is a collision-free multi-hop channel access protocol. In S-TDMA the same time slot can be allocated to nodes that are sufficiently separated from each other. The sharing of time slots is done for nodes such that no (or small) interference is obtained. This increases the overall capacity of the system, in comparison with classical TDMA, as more than one node can transmit during same time slots. [2]

TDMA is simple and easy to implement, but it is very inefficient from a resource utilization point of view. S-TDMA increases the overall capacity of the system, but it is important to design efficient time slot scheduling algorithms so that different nodes (or links) can get their fair share. In order to do this, precise knowledge on the traffic situation is needed.

If the time slot allocation is done with precise knowledge of the traffic load, more efficient allocation results than just assuming uniform traffic can be achieved [3][4]. A very important issue is therefore to make accurate estimations of the traffic flows that each node handles.

#### 1.1.1 Traffic sensitive S-TDMA scheduling

There are a number of studies attempting to find good traffic sensitive S-TDMA schedules, based on slots allocation with knowledge of the traffic load. A critical description of some of these studies follows.

In [3], Somarriba has described a traffic sensitive S-TDMA schedule, named Enhanced Schedule (ES). His novel systematic procedure to find transmission schedules in S-TDMA system is based on the traffic load in each link. The approach suggested in his work is a combination of the basic S-TDMA protocol proposed in [2], together with some extra slots allocated to the
heavier loaded links. His work concludes that traffic-sensitive schedules show an improvement of up to 30% in comparison with the Basic Schedule in classical S-TDMA.

Some other attempts have been made by Robertazzi and Shor. [4] They have presented a distributed traffic sensitive algorithm, called ‘The Degree Algorithm’. It uses node degree (the number of active, incident links to a node) as the basis for establishing priority during the process of generating schedules. The basic idea it uses is that the more incident links a node has, the more traffic it will carry and the more slots it should be assigned. It has shown better performance than its counterpart non-traffic sensitive distributed algorithms. [4]

1.2 Aim, motivation and outline of the paper

The aforementioned research work makes it obvious that for good network performance, the nodes (or links) with heavy traffic loads need to access the channel more often than the nodes (or links) with small traffic loads. If time slot allocation is done on the basis of the time-dependent traffic loads on the nodes, then channel/resource utilization could be improved.

How can these time-dependent traffic loads on the nodes or links be estimated? What traffic estimation methods have been used in earlier research work mentioned above? How good traffic estimates they produce? How to evaluate them to know how well they perform? There are many questions that need to be answered. This report will answer the above listed questions. The short description of the forthcoming work, given below, will give a brief idea of how this paper has been organized.

Section 2 is about traffic load estimation methods. We present existing traffic load estimation methods. We then explain their limitations in live networks scenario and propose a traffic load estimation method based on an exponential decrease function.

Section 3 is about Implementation and Evaluation Issues. We present a method that is used in performance analysis of the proposed traffic load estimation method. We also highlight simulation assumptions and considerations.

In Section 4 we present the simulation results and analysis of estimated traffic load in ad-hoc networks. We have analyzed both static and mobile ad-hoc networks. We present fine tuning of the two parameters; Window Size $T_{ws}$ and gamma $\gamma$.

Section 5 is about conclusions and recommendations for further research work.

This paper has two appendices. In the end, references are listed.

2 Traffic Estimation Methods

We will first describe existing traffic load estimation and measurement methods that have been used in mobile ad-hoc networks research, in an attempt to find an optimum slot allocation algorithm. They are explained in sub-sections 2.1 and 2.2.

We will then talk about their limitations that make them inefficient to use in live networks. Finally, in section 2.4, we will propose a new traffic load estimation method.

2.1 ‘Traffic Matrix of the load’ on each link based on the number of routing paths using a link

This method has been used at least in two research studies. [5][6] According to this method the traffic load that can be expected to flow in each link $(i, j)$ is based on the number of routes (or paths) that use the link $(i, j)$ . This method probably works for all forms of routing. However it assumes that the traffic load on each route is equal. So, as long as different routes carry exactly the same traffic, this method gives the correct values of traffic load on the relative scale. Although this might be true for uniform traffic load distribution, it is generally not true.

Let $\lambda_{ij}$ be the traffic load on a link $(i, j)$ . Then

$$\lambda_{ij} = \mu \times \sum \text{paths using the link (i, j)}$$

**equation 2-1**

Where $\mu$ is the traffic load on each route. Since traffic load distribution is uniform, so $\mu$ is given by (for a fully connected network):

$$\mu = \frac{\text{total traffic load}}{\text{number of links}}$$
\[
\mu = \frac{\lambda}{N(N-1)}
\]

equation 2-2

Where \( \lambda \) is the total traffic load over the whole network and \( N \) is the total number of nodes in a mobile ad-hoc network.

Denote the number of paths that use the link \((i,j)\) as \( P_{ij} \). So equation 2-1 implies:

\[
\lambda_{ij} = \mu \times P_{ij} = \frac{\lambda}{N(N-1)} \times P_{ij}
\]

equation 2-3

\( p_{ij} \) depends on the routing protocols used and it is an undirected measure of the traffic load that can be expected to flow through the link \((i,j)\). It is the measure of the traffic load on the relative scale. From the traffic sensitive S-TDMA slot allocation algorithm point of view it does not matter if we measure traffic load as \( \lambda_{ij} \) or as \( p_{ij} \) on the relative scale. It is preferred to use \( p_{ij} \) as an undirected measure of the traffic load on the relative scale as network might not be fully connected and there might be packets loss. Let \( L_{ij} \) be the relative traffic load flowing through the link \((i,j)\). Then:

\[
L_{ij} = P_{ij}
\]

equation 2-4

This \( L_{ij} \) are the correct values of traffic load flowing through the link \((i,j)\) on the relative scale.

The following research studies have been based on this method.

- Traffic Load Matrix based on Minimum Hop Routing (MHA), used by Somarriba. [3][5]
- Traffic Load Matrix based on Most Forward Routing (MFR), used by Robertazzi and Shor. [4][6]

The traffic load matrix gives relative traffic load estimation for each link. However, if we want relative traffic load estimation for each node, then this can be calculated by summing up the load of all the outgoing links of that node.

### 2.2 Traffic Estimation based on Node Degree

Another traffic load estimation method that can be used in mobile ad-hoc networks is based on node degree. Node degree is the number of active, incident links to a node. In [4][6] Robertazzi and Shor describe how their Degree algorithm uses node degree to allocate time slots based on the relative traffic estimation of a node. Their degree algorithm uses node degree as the basis for establishing priority. It assumes that the more incident links a node has, the more traffic it may carry and the more time slots it must be assigned.

### 2.3 Limitations of existing traffic load estimation methods

The above described methods, have limitations that make them inefficient in a real live scenario. They assume equal (uniform) traffic load distribution on each path. If the non-uniform traffic load distribution is unknown, then they do not work at all. In practical live networks we can not assume uniform traffic load distribution, so we need better methods for traffic estimation that should be independent of the type of traffic load distribution.

In the next section 2.4, we will suggest a new traffic load estimation method that is based on simple mathematical calculations, that works for both uniform and non-uniform traffic load distribution. In fact, it does not care about the type of traffic load.

However, one important thing to notice is that the method explained in section 2.1, based on traffic matrix of the load, can be useful as a reference method as it gives correct values of traffic loads on the relative scale when uniform traffic load distribution is used.

### 2.4 A proposed Traffic Load Estimation Method

If we measure the inter-arrival time between two packets, say \( \Delta t \) seconds, and size of packet, say \( \Delta L \) bits.

Then the traffic load through the node, say \( L_{est} \) be give by:
Another way is to take time average over time, say $T_{ws}$. This implies:

$$L_{est} = \left( \frac{\Delta L}{\Delta t} \right) \times \frac{\text{bits}}{\text{seconds}}$$

**Inter-arrival time between two packets**

![Diagram](image)

Where $L_{est}$ is the estimated traffic load through the node, $n$ is the number of packets that have arrived in the time $T_{ws}$.

A design parameter here is the time $T_{ws}$. We call it the window size. Simulation studies will help us to choose appropriate value of $T_{ws}$.

### 2.4.1 Exponential decrease traffic load estimation function

The problem with time average is that it gives equal importance to all packets that have been received in window size $T_{ws}$. But the importance of traffic load measured at a link, decreases with time. For instance the packet that has arrived most recent, say at time $t_n$, should be given more weight compared with packets that have been arrived earlier in time. This importance of packets can be controlled and evaluated via exponential decrease estimation function:

$$L_{est} = c \sum_{t_i = t_n}^{t_1} e^{(\gamma t_i)} \quad t_i \in T_{ws}$$

**equation 2-6**

Where $L_{est}$ is the estimated traffic load and $t_i$ is the time since arrival of the packet. $T_{ws}$ and $\gamma$ are the design parameters that will be decided during simulation studies. $c$ is a constant and controls relative load factor. $c$ is equal to relative load factor times $\gamma$. Relative load factor is a constant value for the whole network during the window size $T_{ws}$. It is used to normalize estimated traffic load values so that reasonable comparison can be made.

It is $\gamma$ that controls how much more importance should be given to recent packet arrivals compared with packets that have been arrived earlier in time. Larger value of $\gamma$ means to give more importance to recent packet arrivals. However, small $\gamma$ means to give all packets almost same preference. We need to choose an appropriate value of $\gamma$ during simulation studies so that we can have best estimation of traffic load $L_{est}$.

### 3 Implementation and Evaluation Issues

This section will look into implementation and performance evaluation issues. We will present how performance analysis of the proposed traffic load estimation method has been done.

#### 3.1 Performance Evaluation

The relative traffic load estimation method described in section 2.1 can give correct values of traffic load on the relative scale if uniform traffic is generated through the simulated mobile ad-hoc network. So, for the evaluation purposes, we can populate the simulated network with a uniform Poisson based traffic load distribution such that the method described in section 2.1 gives correct values of the traffic load on the relative scale. These correct values can then be compared with results obtained from our traffic estimation modules (implemented via the...
exponential decrease traffic load estimation function).

### 3.1.1 Correct Values (True Values) of traffic load on the relative scale

As said, ‘Correct Values’ of traffic load are based on routing information as well as the type of input traffic. In this case it is uniformly distributed Poisson data, which means the traffic load on each route (or path) is equal.

From this point on we call ‘Correct Values’ of traffic load as the ‘True values’ of traffic load.

Say, are the true values of relative traffic load on each node $i$, which are calculated as the sum of relative traffic load on all the outgoing links $(i,j)$ of the node $i$. $L_{ij}$ is the relative traffic load on the outgoing link $(i,j)$ of the node $i$. (See equation 2-4), then, true values of relative traffic load, for node $i$ are give by:

$$L_{True}(i) = \sum_{j=1}^{n} L_{ij}$$

**equation 3-1**

where $n$ is the total number of outgoing links from node $i$.

### 3.1.2 Estimated Values of traffic load

‘Estimated Values’ of traffic load, say $L_{est}(i)$, in packets per second are estimated as described in sub-section 2.4 for each node $i$. We have used both methods as they are inter-linked with each other. Time Average is a special case of Exponential Decrease Load Estimation Function when same importance has been given to all packets, i.e. when $\gamma = 0$ and $c = (1/T_{ws})$, i.e.:

$$L_{est} = \frac{1}{T_{ws}} \left[ e^{-0(0)} + e^{-0(0)} + \ldots + e^{-0(0)} + e^{-0(0)} \right]$$

**equation 3-2**

which is the time average i.e. the number of packets that have arrived in the time $T_{ws}$.

### 3.1.3 Relation between Method 1: ‘Time Average’ and Method 2: ‘Exponential Decrease Load Estimation Function’

**Equation 2-6** implies:

$$L_{est} = c \left[ e^{-\gamma_1} + e^{-\gamma_2} + \ldots + e^{-\gamma_{n-1}} + e^{-\gamma_n} \right]$$

Time average is a special case of exponential decrease load estimation function when same importance has been given to all packets, i.e. when $\gamma = 0$ and $c = (1/T_{ws})$, i.e.:

$$L_{est} = \frac{1}{T_{ws}} \left[ e^{-0(0)} + e^{-0(0)} + \ldots + e^{-0(0)} + e^{-0(0)} \right]$$

**equation 3-2**

### 3.1.4 Performance Evaluation Criteria

Let $L_{est}(i)$ be the relative estimated values of the traffic load and $L_{true}(i)$ be the relative true values of the traffic load for node $i$. Then, the absolute error difference between the true and estimated values of traffic load for node $i$ is given by:

$$\epsilon(i) = \left| L_{true}(i) - L_{est}(i) \right|$$

**equation 3-3**

Relative Traffic Load Estimation Error (in Percentage) has been used as the basis of most of the decisions to choose design parameters. Relative traffic load estimation error, $\epsilon_{rel}(i)$ for node $i$ is given by:

$$\epsilon_{rel}(i) = \left| \frac{L_{true}(i) - L_{est}(i)}{L_{true}(i)} \right|$$
Most often percentage error is used and relative traffic load percentage error, $\varepsilon_{\text{per-rel}}(i)$ is given by:

$$
\varepsilon_{\text{per-rel}}(i) = \left(\frac{L_{\text{true}}(i) - L_{\text{est}}(i)}{L_{\text{true}}(i)}\right) \times 100
$$

for node $i$.

**equation 3-4**

In this case we need to choose design parameters (Window Size $T_{\text{ws}}$ and $\gamma$) so that $\varepsilon_{\text{per-rel}}(i)$ gets minimum for all nodes i.e. $\sum \varepsilon_{\text{per-rel}}(i)$ should be minimum.

### 3.2 Simulation assumptions and considerations

This research work has been done in co-operation with FOI [7]. The simulation software aquarius has been used to simulate networks and generate Poisson distributed uniform traffic load. Each simulation run consists of 12 hours, i.e. $12 \times 60 \times 60 = 43200$ seconds. Three instances of traffic load have been generated for all analyzed networks, with increasing values of overall network load ($\lambda = 20, 40, 60$ packets/seconds). Traffic estimation modules have been written in MATLAB.

**Routing:** Minimum Hop Routing has been used. In Minimum Hop Routing, the route with minimum number of hops has been used when a packet has to be transferred from source to destination.

**MAC Protocol:** Only plain TDMA has been used as the MAC protocol. The reasons for using TDMA are that it is simple and easy to implement. Moreover from a traffic estimation point of view it does not matter if we use TDMA or S-TDMA. Any MAC protocol can take advantage of traffic load estimation if feedback between MAC and traffic load estimation modules has been designed and used.

**Mobility:** We have also considered node mobility as the node which once may carry heavy traffic load, when it changes its position, it may then carry low traffic load (or nothing) and vice versa. It is therefore important that the number of time slots allocated to a node (or link) must be updated regularly. In order to do so we must know the time-dependent traffic loads of the nodes.

As a first step, static ad-hoc networks have been analyzed with no mobility. In static networks, the topology does not change during the simulation period as nodes do not move. Analysis of static ad-hoc networks will help us in making general conclusions. For example, how long time it takes to get appropriate values of estimated load. They may tell us if our implementation of traffic load estimation modules is correct.

In step two, mobile ad-hoc networks have been analyzed. In order to generate movement of mobile ad-hoc network nodes, a simple 2-dimensional random walk mobility model has been used. Please look in Appendix A: about how this mobility model has been implemented.

**Geographical Area:** Network nodes movement is bounded in a square and the length of the square is 1000 meter for all analyzed networks. This is supposed to give reasonable scenario which is close to real military case where 10 to 20 mobile nodes are scattered within 1000 meter square boundary.

**Average maximum radio connectivity distance:** The radio connectivity between mobile nodes is chosen such that at least 95% of all node pairs can communicate by single- or multi-hop connections. The average maximum transmission range has been estimated at FOI (We refer to appendix A.2). It is approximately same for the different network scenarios when they are bounded within a 1000 meter square boundary if the simulation time is large.

### 4 Simulation results and analysis of traffic load estimation

#### 4.1 Traffic Load Estimation for Static Ad-hoc Networks

Following design parameters will be optimized to get the estimated traffic load values for static ad-hoc networks so that estimation error gets minimum: Gamma $\gamma$ that controls the importance
given to most recent packets arrival and Window Size $T_{\text{ws}}$, that answers for how long time we need to count packets arrival at a particular node.

Our simulation results indicate that estimation error gets minimum when $\gamma$ is zero and $T_{\text{ws}}$ is maximum. This is what should be expected for Poisson distributed traffic load. When the window size becomes large, the number of packets that have been received during this time becomes large. If $n$ is the number of packets that have been received during window size $T_{\text{ws}}$ and $\lambda$ is equal to mean arrival rate then:

$$\lambda = E(\text{Poisson load}) = \left(\frac{n}{T_{\text{ws}}}\right) \text{ packets per seconds}$$

**equation 4-1**

### 4.1.1 Optimum values of Gamma $\gamma$

We refer to exponential decrease traffic load estimation function [equation 2-6] as described in section 2.4.1. During simulation studies, we choose $\gamma$ so that estimation error gets minimum.

In Figure 3, we plot traffic load estimation error in percentage versus $\lambda$.

Traffic load estimation error, say $\varepsilon_{\text{per-rel}}$, has been calculated as:

$$\varepsilon_{\text{per-rel}} = \left(\frac{1}{n} \sum_{i=1}^{n_{\text{node}}} \left(\frac{L_{\text{true}}(i) - L_{\text{est}}(i)}{L_{\text{true}}(i)}\right) \times 100\right)$$

**equation 4-2**

Where $s$ means total number of samples and $n_{\text{node}}$ means total number of nodes. This plot shows that estimation error increases as $\gamma$ increases and vice versa. Best estimation can be made when $\gamma$ is as small as possible (i.e. when $\gamma$ is close to zero). The plot also indicates that estimation error decreases as we have more packets to analyze.

This plot is for three instances of twenty nodes static ad-hoc network with overall network traffic load $\lambda = 60$ packets/sec, 40 packets/sec, and 20 packets/sec. The ideal plot of a single node has also been shown. In ideal case there is no packet loss and we know exact value of traffic load $\lambda$, which is 2 packets/sec. This gives an idea of how the curve should look like for a single node in ideal case. However, the analyzed networks are far from ideal situation. First, there is packet loss of about 5%, so we do not know precise value of overall network traffic load $\lambda$. Second, different nodes of the static ad-hoc networks have different traffic load and there is a wide difference on the relative scale. Third, in our analyzed networks relayed traffic does not follow Poisson characteristics. Still we can see that the traffic load estimation error in analyzed static ad-hoc networks follows the ideal curve and slope of $y = \left(\frac{1}{2} \lambda\right)$ on the log scale. It means that $\varepsilon_{\text{per-rel}} \propto \frac{1}{\sqrt{\text{packets}}}$

![Figure 3. Analysis of Gamma $\gamma$ values when Window Size is constant – plot of Traffic Load Estimation Error versus $\lambda$ / $T_{\text{ws}}$](image)

### 4.1.2 Optimum values of Window Size $T_{\text{ws}}$

We need to find the optimum value of $T_{\text{ws}}$, so that we can get minimum estimation error. We will analyze packets that have arrived during $T_{\text{ws}}$. We will use $\gamma$ equal to zero for best estimation. See equation 3-2, exponential decrease traffic load estimation function will become the time average for window size $T_{\text{ws}}$. 

![Figure 3. Analysis of Gamma $\gamma$ values when Window Size is constant – plot of Traffic Load Estimation Error versus $\lambda$ / $T_{\text{ws}}$](image)
In Figure 4, we plot traffic load estimation error $T_{ws} \left( \frac{\text{seconds}}{} \right) \times \lambda \left( \frac{\text{packets}}{\text{seconds}} \right)$ versus $T_{ws} \times \lambda \left( \text{packets} \right)$. 

For each $T_{ws}$, traffic load estimation error $e_{per\_rel}$, has been calculated according to equation 4-2. The plot shows that estimation error decreases as $T_{ws}$ increases and vice versa. Best estimation is made when $T_{ws}$ becomes large as expected for Poisson distributed traffic load. The plot also indicates that estimation error decreases as we have more packets to analyze.

The ideal plot of a single node has also been shown. As said earlier, in ideal case there is no packet loss and we know exact value of traffic load $\lambda$. But the analyzed networks are far from ideal situation for reasons stated above. However, traffic load estimation error follows the slope of ideal curve i.e. $y = \left( -\frac{1}{2} x \right)$ on the log scale.

![Log Plot - 20 Nodes Static Ad-hoc Networks: Analysis of "Window Size"](image)

**Figure 4.** Analysis of Window Size when Gamma $\gamma$ is zero – plot of Traffic Load Estimation Error versus Window Size times Lambda

This plot also indicates that we do not have to take very long time averages for acceptable estimation error. It depends on what is the acceptable estimation error, but in general time averages with 1000 packets (or more) achieve estimation error below 5%.

Figure 4 also indicates that traffic load estimations get better for high traffic loads when $T_{ws}$ is constant. For high traffic loads, $\lambda$ gets higher and this gives more packets to analyze within a window size. As a result estimation gets better.

In general, traffic load estimation error decreases as packets to be analyzed increases. Packets to be analyzed will increase whenever $T_{ws}$ or $\lambda$ will increase (or both).

### 4.2 Traffic Load Estimation for Mobile Ad-hoc Networks

In this section, for mobile ad-hoc networks, we will also attempt to find the optimum values of $T_{ws}$ to count number of packets that have arrived during this time and optimum values of $\gamma$ that controls the importance given to packets so that best estimation can be made.

Simulation results indicate that (for exponential decrease function) the optimum value of $\gamma$ increases as network nodes mobility increases and hence decreases the effective window sizes $T_{ws}$.

Furthermore, in contrast with static ad-hoc networks, we cannot use maximum window sizes $T_{ws}$ to get best results by taking time averages. In fact, the optimum value of $T_{ws}$ depends upon the speed of the network nodes.

#### 4.2.1 Optimum values of Window Size $T_{ws}$

We need to find the optimum value of $T_{ws}$, to count for packets that have arrived in that time duration so that we can get minimum relative load estimation error.

In case of static networks, where topology of networks does not change as nodes do not move, there is static non-changing routing matrix. (Refer to section 4.1.2) In those cases the longer the $T_{ws}$, the better the results are. However, in case of mobile ad-hoc networks where nodes are moving and topology of network depends on the movement of nodes, we often have changes in routing matrices that depends on the speed of the nodes. In these scenarios we cannot achieve better results by simply counting number of packets for longer $T_{ws}$.
Figure 5 presents analysis of optimum values of $T_w$ when $\gamma$ is equal to zero. Optimum window size $T_{opt}$ has been calculated so that estimation error gets minimum for whole network. First for each window size $T_w$, estimation error $\varepsilon_w$ has been calculated by:

$$\varepsilon_w = \frac{1}{n} \sum_{i=1}^{n_{max}} \left( \frac{L_{True}(i) - L_{est}(i)}{L_{True}(i)} \right) \times 100$$

for each window size $T_w$

**equation 4-3**

Then, a sample of optimum window size $T_{opt}$ has been chosen for which $\varepsilon_w$ is minimum i.e.:

$$T_{opt} \in \min_{T_w} \left( \frac{1}{n} \sum_{i=1}^{n_{max}} \left( \frac{L_{True}(i) - L_{est}(i)}{L_{True}(i)} \right) \times 100 \right)$$

Finally, the optimum window size $T_{opt}$ that is plotted along the y-axis in Figure 5 has been calculated as the sample mean, i.e.

$$T_{opt} = \frac{1}{S_{samples}} \sum_{s=1}^{S} \left( T_{opt,s} \right)$$

**equation 4-4**

We can see that optimum values of window size $T_{opt}$ depend on network nodes mobility. For less mobile network, rather longer window size produces better results, where as a rather small window size produces better results for highly mobile networks.

This behavior was expected as changes in network topology and hence routing tables cause nodes, which once carry heavy traffic loads, to carry low traffic loads or no traffic at all and vice versa. There is high probability for fast moving network nodes to have many such changes compared with slow moving network nodes. Many such changes cause to use small window size to count for number of packets when estimating traffic load for best possible results in terms of minimum percentage error of traffic load estimation. In general, window size decreases as network nodes mobility increases.

Figure 5 also indicates that $T_{opt}$ depends on traffic load. Figure 6 is an attempt to present analysis of $T_{opt}$ independent of traffic load and network nodes speed. On the y-axis it plots

$$\frac{1}{T_{opt}} \left( \frac{\text{seconds}}{} \right) \times \lambda \left( \frac{\text{packets}}{} \right) - \frac{1}{\text{packets}}$$

and on the x-axis it plots

$$\frac{\text{speed} \left( \frac{\text{meters}}{} \right)}{\text{seconds}} \times \frac{\text{RadioRange} \left( \frac{\text{meters}}{} \right)}{\lambda \left( \frac{\text{packets}}{} \right)} - \frac{1}{\text{packets}}$$

*RadioRange* is the average maximum radio transmission range.

**Figure 5.** Optimum Window Size versus Nodes Mobility for 10 nodes mobile ad-hoc networks

**Figure 6.** Log plot: Analysis of Optimum Window Size
4.2.2 Analysis of Estimation Error for optimum window sizes

In Figure 7, we plot traffic load estimation error for optimum window sizes versus nodes mobility. Estimation error has been calculated for optimum window sizes $T_{opt}$ using equation 4-2. Optimum window sizes, $T_{opt}$ has been calculated using equation 4-4.

![Figure 7. Estimation Error for Optimum Window Size versus nodes mobility.](image)

We can see that estimation error increases as nodes mobility increases. This is because $T_{opt}$ decreases as network nodes mobility increases. When we have small $T_{opt}$, it means that we have less packets to analyze.

![Figure 8. Log Plot - Analysis of Estimation Error for Optimum Window Sizes](image)

In order to get independence from network nodes mobility, we plot estimation error versus (1/packets) in Figure 8. We can see that estimation error decreases as we have more packets to analyze or when (1/packets) decreases. Estimation error approximately follows the slope of $y = (1/3)x$ on the log scale. That means estimation error, say $\varepsilon_{per\_rel}$ approximately satisfies:

$$\varepsilon_{per\_rel} \propto \frac{1}{(\text{packets})^{1/3}}$$

4.2.3 Optimum values of Gamma $\gamma$

We refer again to exponential decrease traffic load estimation function as described in section 2.4.1.

During simulation studies, we need to choose $\gamma$, that controls how much more importance should be given to recent packet arrivals, so that our estimated traffic load matches true values or provides us the best possible traffic load estimation.

In Figure 9, we plot optimum values of $\gamma$ versus nodes mobility. Optimum values of $\gamma$, say $\gamma_{opt}$, have been calculated for large window size $T_{ws}$. $\gamma_{opt}$ has been chosen so that estimation error gets minimum for whole network. For each gamma value, estimation error $\varepsilon_{\gamma}$ has been calculated by:

$$\varepsilon_{\gamma} = \frac{1}{n} \sum_{i=1}^{n} \frac{L_{\text{True}}(i) - L_{\text{est}}(i)}{L_{\text{True}}(i)} \times 100 \quad \text{for each } \gamma.$$  \hspace{1cm} \text{equation 4-5}

Then, a sample of optimum gamma, $\gamma_{opt}$, has been chosen for which $\varepsilon_{\gamma}$ is minimum i.e.:

$$\gamma_{opt} \in \min_{\forall \gamma} \left( \frac{1}{n} \sum_{i=1}^{n} \frac{L_{\text{True}}(i) - L_{\text{est}}(i)}{L_{\text{True}}(i)} \times 100 \right)$$

Finally, the optimum gamma $\gamma_{opt}$ that is plotted along the y-axis in Figure 9 has been calculated as the sample mean, i.e.
\[ \gamma_{opt} = \frac{1}{S_{\text{samples}}} \sum_{s=1}^{S} (\gamma_{opt_s}) \]

\text{equation 4-6}

Figure 9 shows that optimum value of \( \gamma \) in mobile ad-hoc networks increases as nodes mobility increases. When nodes mobility increases, we expect more changes in network topology and hence more changes in routing tables. Many such changes cause to use high value of \( \gamma \) that gives more important to the most recent packet arrivals and hence effective window sizes decreases.

Figure 9. \textit{Optimum values of Gamma} \( \gamma \) versus Nodes Mobility \textit{[Large Window Size has been used]}

Figure 9 also indicates that \( \gamma_{opt} \) depends on traffic load. Figure 10 presents an attempt for analysis of \( \gamma_{opt} \) independent of traffic load and network nodes speed.

On the y-axis it plots
\[ \gamma_{opt} \left( \frac{1 \text{ packet}}{\text{seconds}} \right) \]
and on the x-axis it plots
\[ \frac{\text{speed}}{\text{seconds}} \]
\[ \text{RadioRange} \left( \frac{\text{packets}}{\text{seconds}} \right) \sim \left( \frac{1 \text{ packet}}{\text{seconds}} \right) \]

4.2.4 Analysis of Estimation Error for optimum gamma

In Figure 11, we plot traffic load estimation error for optimum gamma verses mobile ad-hoc network nodes mobility. Estimation error has been calculated for optimum gamma \( \gamma_{opt} \) using equation 4-2. Optimum gamma \( \gamma_{opt} \) has been calculated using equation 4-6.

We can see that estimation error for optimum gamma \( \gamma_{opt} \) increases as nodes mobility increases. This is because \( \gamma_{opt} \) increases as network nodes mobility increases. Higher value of \( \gamma_{opt} \) means we are giving more weight to more recent packet arrivals and effective window size becomes small.

In order to get independence from network nodes mobility, we plot estimation error verses \((1/\text{packets})\) in Figure 12. Similar to section 4.2.2, we can see that estimation error is
proportional to number of packets. Estimation error decreases as we have more packets to analyze or when \((1/\text{packets})\) decreases. Figure 12 also shows that estimation error approximately follows the slope of \(y = (1/3)x\) on the log scale. So, just like earlier results presented in section 4.2.2 of ‘estimation error for optimum window sizes’, estimation error for optimum gamma, say \(\varepsilon_{\text{per rel}}\) approximately satisfies:

\[
\varepsilon_{\text{per rel}} \propto \frac{1}{(\text{packets})^{\frac{1}{3}}}
\]

**Figure 12. Log Plot - Analysis of Estimation Error for Optimum Gamma**

Comparing Figure 8 and Figure 12, we can also conclude that both traffic load estimation methods (see section 3.1.2) i.e. ‘simple time average with optimum window sizes’ and ‘exponential decrease estimation function with optimum gamma’ give almost same estimation error.

### 4.2.5 Comparison of Estimated and True traffic load

In case of mobile ad-hoc networks, topology of network changes and hence loads at nodes change too. So we need to have periodic estimation of traffic load. The ideal thing is; of course, to do load estimation as often as possible.

In this section a plot has been presented in Figure 13. It presents Estimated and True values of traffic load verses time for 10 nodes mobile ad-hoc network; nodes mobility is 10 meters/second, Poisson parameter lambda \(\lambda\) is equal to 20 packets per second and load update time is 5 seconds. This plot is only for two nodes; each sub-plot on left column is for node 0 and each sub-plot on right column is for node 1. It demonstrates the effect of window size \(T_{ws}\) as it changes. The optimum value of window size \(T_{opt}\) is 9.415 seconds. The plot shows that how estimated traffic load differs from true values of traffic load below and above \(T_{opt}\).

**Figure 13. Effect of window size \(T_{ws}\) on Estimated Traffic Load, (Gamma \(\gamma = 0\))**

### 5 Conclusions and Recommendation for Future Work

The proposed exponential decrease traffic load estimation method is a general traffic estimation method. It is intended for any kind of traffic load distribution. It depends on selection and optimization of following parameters: Gamma \(\gamma\) and Window Size \(T_{ws}\).

In this research work only Poisson based, uniformly distributed traffic load, has been estimated and analyzed. It is important to realize
that true values of traffic load in our simulation only changes when routing changes due to changes in network topology as a result of mobility. Above parameters have been analyzed only for Poisson based traffic load and our results and conclusions are valid only for uniformly distributed Poisson based traffic load. However, the proposed traffic load estimation method based on exponential decrease function may also work fine for other traffic models if fine tuning of the above parameters have been made.

The following conclusions can be drawn:

1) Estimation of traffic load in static ad-hoc networks is much better compared with estimation of traffic load in mobile ad-hoc networks.

   When large window size (that gives more than 1000 packets to analyze) is used estimation error gets below 5% in static ad-hoc networks (See Figure 3 and Figure 4). On the other hand in case of mobile ad-hoc networks the best result gives estimation error about 10% for low mobile network (See Figure 7 and Figure 8). Estimated load percentage error rate in mobile ad-hoc networks increases with increase in mobility.

2) Traffic load estimation error in both static and mobile ad-hoc networks depends on traffic load. For high traffic load we have got better load estimation than low traffic load. This is because we have more packets per time unit available for traffic analysis in case of high traffic load (For static ad-hoc networks, see Figure 3 and Figure 4. For mobile ad-hoc networks, see Figure 8).

3) For static ad-hoc networks, the larger the window size $T_{ws}$, the better the estimation (See Figure 4).

   For mobile ad-hoc networks, the optimum value of window size $T_{ws}$ depends on nodes mobility as well as traffic load. It decreases as nodes mobility increases. It also decreases as traffic load increases (See Figure 5).

4) For static ad-hoc networks, the best value for $\gamma$ is 0 (See Figure 3).

   For mobile ad-hoc networks, the optimum value of $\gamma$ depends on nodes mobility as well as traffic load. It increases as nodes mobility increases. It also increases as traffic load increases (See Figure 9).

   When nodes mobility increases, changes in network topology increases and we face more changes in routing tables. Many such changes cause to use high value of $\gamma$ so that it gives more important to the most recent packet arrivals and hence effective window sizes decreases.

5) Traffic load estimation error in mobile ad-hoc networks also depends on nodes mobility (See Figure 7 and Figure 11). When tried to get independence from network nodes mobility and traffic load, we found that estimation error is proportional to $\frac{1}{(\text{packets})^{\frac{1}{3}}}$ (See Figure 8 and Figure 12).

6) In case of mobile ad-hoc networks, both traffic load estimation methods (see section 3.1.2) i.e. ‘simple time average with optimum window sizes’ and ‘exponential decrease estimation function with optimum gamma’ give almost same estimation error (See Figure 8 and Figure 12).

The above conclusions suggest that traffic load estimation is possible in ad-hoc networks, but, fine tuning of above described parameters is needed for optimum performance.

5.1 Recommendation for Future Work

5.1.1 Analysis of more realistic packet data traffic models

A number of traffic studies have shown that Poisson process is not very well suited for packet data traffic models as packet inter-arrivals are not exponentially distributed. Also packet data traffic is bursty, statistically self-similar and
fractal in nature. So other data traffic models, such as E-Mail Traffic Arrival Model and/or WWW Traffic Arrival Model should be analyzed and parameters for proposed exponential decrease traffic load estimation method should be analyzed in this scenario.

5.1.2 Average Network Throughput and Average Network Delay

To further analyze proposed exponential decrease traffic load estimation method, feedback should be designed and implemented between traffic sensitive MAC algorithm (S-TDMA) and exponential decrease load estimation function.

Traffic sensitive MAC algorithm can then get estimated load from feedback function and can assign slots based on estimated load. Then we may be able to analyze how much over all performance of the network has been improved. The performance measures will then be Average Throughput and Average Network Delay.

5.1.3 Take account of Queue Lengths

Queue lengths of the active as well as neighboring nodes may also be considered in future research work. When MAC algorithm allocates time slots, more weight or high priority may be given to longer queue for better results. Neighboring Queues can also be considered when prioritizing or allocation time slots.

6 References


Appendix A: MOBILITY MODEL AND MAXIMUM RADIO TRANSMISSION RANGE.

A.1 MOBILITY MODEL USED

To generate mobile ad-hoc network nodes movement, a simple 2-dimensional random walk mobility model has been used. It has been assumed that all nodes move independently according to the same mobility model and use the same values for the constants. To avoid that the nodes diffuse away from each other, we limit the nodes coordinates to be in a square:

\[
(x(t), y(y)) \in \Omega = \{[0,l] \times [0,l]\}
\]

where \(l\) is the side of the square. If a node hits the boundary of square, it will reflect back like a ball. We assume that the reflection is in-elastic. The speed \(v\) of a node is assumed to be constant, while the direction of a node’s movement \(\theta(t)\) is assumed to be given by

\[
\dot{\theta}(t) = -\alpha \omega(t) + \beta \mu(t) \quad \omega(0) = 0
\]
\[ \dot{\theta}(t) = \omega(t) \quad \theta(0) = U(0, 2\pi) \]

where \( \alpha \) and \( \beta \) are constants, \( \mu_\theta(t) \) is a white Gaussian process with mean \( 0 \) and variance \( 1 \), and \( U \) is a uniform distribution. The coordinates of a node are then given by

\[
\begin{align*}
\dot{x}(t) &= \nu \cdot \cos(\theta(t)) \quad x(0) \in U(0, l) \\
\dot{y}(t) &= \nu \cdot \sin(\theta(t)) \quad y(0) \in U(0, l)
\end{align*}
\]

Figure 14, below, presents an example of paths generated by this mobility model.

\[ \text{Figure 14. Path of three nodes generated by the random walk mobility model with } \alpha = 1 \text{ [1/s], } \beta = \frac{\pi}{30} \text{ [rad/s}], \text{ and } \nu = 20 \text{ [m/s].} \]

**A.2 Maximum Radio Transmission Range**

During simulation at FOI, the transmitter power is chosen to get the radio connectivity between mobile nodes so that in each network scenario, at least 95% of all node pairs can communicate by single- or multi-hop connections. The average maximum transmission range (in meters) for networks with 2 to 24 mobile nodes has been estimated. The nodes were moving around for 12 hours in a 1000 square meter area. Individual node speed was 20 meter/seconds.


The average maximum transmission range is approximately same for the different network scenarios if the simulation time is large.

**Appendix B: Uniform Poisson Traffic Model**

Most of the present mobile ad-hoc network research uses Poisson distribution based traffic models. This model has been used in aquarius, for traffic generation purposes. In case of Poisson(\( \lambda t \)) distribution, the number of arrivals \( N(t) \) in a finite time \( t \) is given by:

\[
P\{N(t) = n\} = (\lambda t)^n e^{-\lambda t} \frac{n!}{n!}
\]

Where \( \lambda = \) mean arrival rate

\( N(t) = \) number of arrivals in interval \((0, t)\)

**Figure 15. Shape and probabilities of the Poisson distribution with \( \lambda = 10 \)**

The inter-arrival times are independent and obey the \( \text{Exp}(-\lambda) \) distribution:

\[
P\{\text{interarrival time} > t\} = e^{-\lambda t}
\]

\( \lambda = \) mean arrival rate

\[ \lambda = E[X] \]

Expected Value of \( X = E[X] \)

\[ \lambda = \text{Weighted Average Value.} \]

Variance of \( X = Var[X] \)

\[ \lambda = \text{Standard Deviation of } X = \text{std}[X] = \text{sqrt}(Var[X]) = \text{sqrt}(\lambda). \]

If \( X \) is a Poisson random variable and \( \lambda \) is it’s parameter, then:

[Image of Figure 15]