SIRE REPORT

YUVRAJ DUTTA
121EC0272

Department of Electronics & Communication Engineering

REVIEW OF OPTIMIZING TRANSMIT INTENSITY IN MULTI CHANNEL CSMA

A Distributed Approximation Framework

Abstract

This review report is based on the above-mentioned paper which generally uses the closed form Bethe approximation framework to find the transmit intensities using the target service rates. An iterative method has been applied to find the empirical service rates which have been used to find the optimal error.

Yuvraj Dutta

121ec0272@nitrkl.ac.in, duttayuv.nitrkl@gmail.com
National Institute of Technology Rourkela
Department of Electronics & Communication Engineering
Final Year
Phone: +917596871995, +8801553844999

Introduction:

Computer Networking refers to interconnected computing devices that can exchange data and share resources with each other. These networked devices use a system of rules, called communications protocols, to transmit information over physical or wireless technologies. Nodes and Links are the basic building blocks of a computer network. Nodes simply refer to data communication equipment like modem, hub, switch or data terminal equipment such as two or more computers or printers. Links may be physical like cables or optical fibers, or wireless networks. In a working computer network, all the nodes must follow a certain protocol so that they can send or receive data through it. There are various kinds of transfer protocols to follow. One of them is CSMA widely known as the Carrier Sense Multiple Access protocol. CSMA is a network protocol for carrier transmission that operates in the Medium Access Control (MAC) layer. It senses or listens whether the shared channel for transmission is busy or not and transmits if the channel is not busy. In general aspect, there is only one channel through which the nodes will try to transmit their data. However, in this assignment CSMA scheme has been dealt with operating on multiple channels. This multi-channel CSMA scheme is quite different from other existing single channel operations. Here we use the Bethe closed form approximation technique to compute our transmit intensity.

Bethe Closed Form Approximation and its significance:

Previously, the main technique that has been used to compute scheduling and other optimal parameters in a CSMA scheme was the MCMC (Markov Chain Monte Carlo) approach. But the main problem is its slow convergence rate due to poor adaptivity to network configuration. Even in a small network it takes a lot of time to find out the behavior of the nodes. In summary, all prior CSMA algorithms suffer slow convergence rate implicitly or explicitly.

In a CSMA network, we can understand the network's behavior using a concept from physics and probability called a Markov random field (MRF), which we refer to as CSMA-MRF. In this model, each link in the network is represented as a vertex, and interfering links are connected by edges. The activity levels of the links correspond to parameters in this MRF model. The service rate of each link is then equivalent to the probability of that link being active in the CSMA-MRF. In the study of MRFs, concepts like the 'Gibbs free energy' and the 'Bethe free energy' are used to compute these probabilities. Research has shown that finding a minimum or zero-gradient point of the Bethe function can give us approximate values for these probabilities. This method has been successful in fields such as computer vision, artificial intelligence, and information theory. The key advantage of this approach is that it provides a way to determine the relationships between the probabilities and the MRF parameters with low computational complexity.

Using the Bethe approximation is easier than other prior algorithms because here we only compute the specific interested cluster of links/individual links of a complex network. We always include the intensity of the other interfering links to find the optimal parameters of our targeted links because the co-relation of the interfering links is vital for the performance of those targeted links. Hence, the Bethe approximation is one kind of iterative method that is like Markov’s iterative method.

Working Methodology:

During CSMA protocol if a node is ready to transmit data to the medium it will at first try to detect the free channel which will be used for transmission. The main reason for this detection is to avoid any kind of possible data collision. Suppose a node tries to send data through a channel and that channel is being used by another node at the same time than this would cause the frames of data to garble resulting in a collision. This makes the protocol error prone. In this report we have reviewed a paper that implements a CSMA scheme that leverages multiple channels. Though, it is not guaranteed that the error will be very less than any other schemes, but it will be less than single channel CSMA scheme. This CSMA scheme leverages the Bethe approximation framework to compute the transmit intensities in multi-channel CSMA systems. Before mentioning the methodology let's get acquainted with some upcoming terminologies in this report.

Transmit Intensity: Refers to the rate at which devices attempt to transmit data on the network.

Service Rate: The service rate of a link $\mathbf{i}$ is the long-term fraction of the time it is active.

Load: It refers to the normalized service rate with respect to maximum service rate in a network.

For our proposed CSMA scheme we are considering a wireless network comprising $\mathbf{N}$ links operating across $\mathbf{M}$ channels. Each link is constrained to transmit on a single channel at any given time. To model the interference between links, we employ the well-established conflict graph model.

In this model, we represent the network as a graph $\mathbf{G(V, E)}$, where:

- $\mathbf{V}$ is the set of vertices, each corresponding to a wireless link in the network.
- $\mathbf{E}$ is the set of edges, representing potential interference between links.

Two links $\mathbf{i}$ and $\mathbf{j}$ are considered to interfere if they cannot transmit simultaneously on the same channel without causing signal degradation. In the conflict graph, this interference relationship is represented by an edge connecting the vertices corresponding to links $\mathbf{i}$ and $\mathbf{j}$.

This graph-based representation provides a clear and efficient way to capture the complex interference relationships in the network. It forms the foundation for analyzing and optimizing the network's performance, particularly in terms of channel allocation and transmission scheduling.

$$\boldsymbol{Ni} \triangleq \{\boldsymbol{j} : (\boldsymbol{i}, \boldsymbol{j}) \in \boldsymbol{E}\}$$

represents the set of links that interfere with the $\mathbf{i}^{\text{th}}$ link.

Further, $\boldsymbol{di} \triangleq |\boldsymbol{Ni}|$**,** represents the number of links that interfere with $\mathbf{i}^{\text{th}}$ link.

In a Carrier Sense Multiple Access (CSMA) scheme with slotted time, we represent the transmission status of a link $\mathbf{i}$ at any time t using $\boldsymbol{\beta i(t)} \in [\boldsymbol{M}] \triangleq \{\mathbf{0}, \mathbf{1}, \ldots, \boldsymbol{M}\}$ Specifically:

- $\boldsymbol{\beta i} = \mathbf{0}$ if link i is not transmitting.
- $\boldsymbol{\beta i} = \boldsymbol{m}$ if link i is transmitting (i.e., active) on channel $\boldsymbol{m} \in [\boldsymbol{M}]$**.**

We denote the schedule at time t by $\boldsymbol{\beta(t)} \triangleq [\boldsymbol{\beta i(t)} \in [\boldsymbol{M}] : \boldsymbol{i} \in \boldsymbol{V}]$**.** A Schedule is only valid if and only if two interfering links don't pass data on the same channel at a same time i.e.

$$\boldsymbol{\beta i(t)} \neq \boldsymbol{\beta j(t)},$$

The set of all feasible schedules denoted by $\boldsymbol{F}$ is given by

$$\{\boldsymbol{\beta o} \in [\boldsymbol{M}]\boldsymbol{N} : \forall\, (\boldsymbol{i}, \boldsymbol{j}) \in \boldsymbol{E}, \boldsymbol{\beta i} \neq \boldsymbol{\beta j}\ \boldsymbol{whenever}\ \boldsymbol{\beta i}\, \boldsymbol{\beta j} > \mathbf{0}\}$$

During the transfer of data, a link can only pass one data packet per time slot.

The adaptive CSMA protocol operates on a time-slotted basis and employs a distributed mechanism for channel access. Each link **i** is associated with a real-valued parameter $\mathbf{r_i}$, termed the transmission intensity, which quantifies the link's aggressiveness in capturing available channels.

The protocol functions as follows:

1. In each time slot **t**, a single link **i** is randomly selected for potential state update.
2. The selected link **i** senses all channels to monitor the transmission schedule $\boldsymbol{\beta(t-1)}$ from the previous time slot.
3. Based on this information, link **i** updates its transmission status $\boldsymbol{\beta i(t)}$ for the current time slot.
4. Concurrently, all other links **j ≠ i** maintain their transmission status from the previous time slot, such that $\boldsymbol{\beta j(t)} = \boldsymbol{\beta j(t-1)}$.

The update rules for the selected $\mathbf{i^{th}}$ link are following:

- Link **i** will identify the available channels from the (t-1) time instance. (i.e. the channels free from its interfering links)
- If no channel is idle in the previous (t-1) time slot then link **i** will not transmit. $\boldsymbol{\beta i = 0}$
- Else if K channels are idle

- With probability $\frac{1}{1+K exp(\text{ri})}$, $\beta i = 0$ link **i** doesn't transmit
- With probability $1 - \frac{1}{1+K exp(\text{ri})}$, link **i** transmits on one of the $\boldsymbol{K}$ channels (randomly chosen)

Remarks: This mentioned adaptive CSMA is very much straight forward cause here we are only considering one link that can update its status, but we can also consider multiple links status update. Moreover, we can also work on with continuous time slot instead of slotted one.

Algorithm :

This research addresses the challenge of optimizing transmit intensities $\{\mathbf{r}\}\mathbf{i} \in \boldsymbol{V}$ in CSMA networks to achieve target service rates $\{\mathbf{s}\}\mathbf{i} \in \boldsymbol{V}$ for all links, thereby enhancing overall network performance and resource utilization.

Considering the Bethe approximation framework the algorithm for computing transmit intensity at link i follows:

1. Consider the desired service rate $\{\boldsymbol{sj}\}\boldsymbol{j} \in \boldsymbol{Ni}$ from the interfering links
2. Compute the transmit intensity $\boldsymbol{exp(ri)}$ given by

$$\left(\frac{\mathbf{1} - \boldsymbol{M u_i}}{\boldsymbol{u_i}}\right)^{\boldsymbol{d_i - 1}} \prod_{j \in N_i} \frac{\boldsymbol{u_i - (M-1) u_{ij}}}{\mathbf{1} - \boldsymbol{M(u_i + u_j - (M-1)u_{ij})}}$$

Where $u_i \triangleq \frac{u_i}{M}$ is the service rate normalized with the number of channels and u_{ij} is the corelated normalized transmit intensity which is given by

$$\frac{\boldsymbol{u_i + u_j - 1 + \sqrt{4(M-1)u_i u_j + (u_i + u_j - 1)^2}}}{\boldsymbol{2(M-1)}}$$

Algorithm for Network Topology:

1) Take N=100 nodes
2) Select the 2D square size inside which nodes will be randomly distributed
3) Select the maximum distance between any two nodes
4) Select the maximum clique size (Using the **ELsclique** Function to find the maximum clique size)
5) Use the undirected graph to draw the topology.

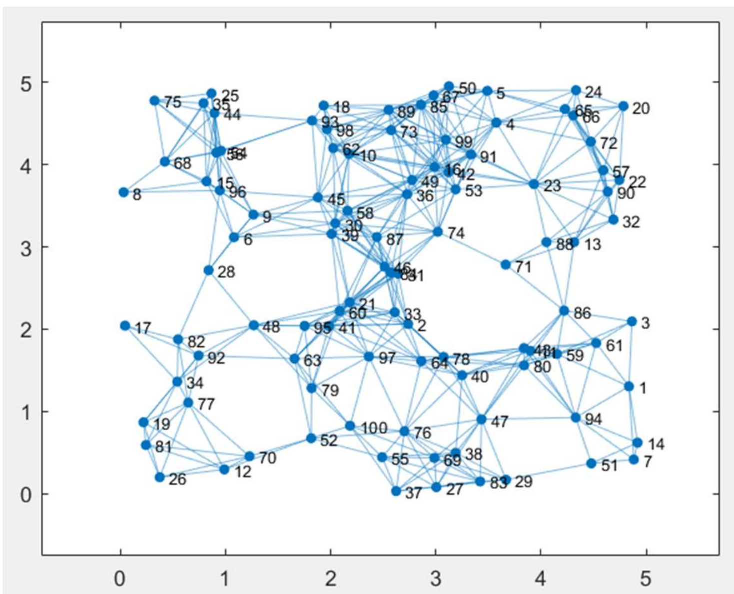

Fig 1: Random Graph Topology for N=100 links, maximum clique size=9, maximum distance=1, Box size =5x5

Algorithm for Error vs Different Load:

1) Code for Network Topology
- Remember to use the same network topology for different available channel

2) Let take channel options M= [2,4,6,8,10]
- For a specific channel calculate the maximum possible service rate by the mentioned formula
- Take some load values load= [0.1:0.1:0.9]
- From those load values calculate the target service rate using Target_service_rate=load*maximum_service_rate
- Using those specific target service rates calculate transmission intensities

3) The target transmission intensities will now be used to empirically calculate service rates

4) After that error will be calculated for all the loads and graph of error vs load will be plotted

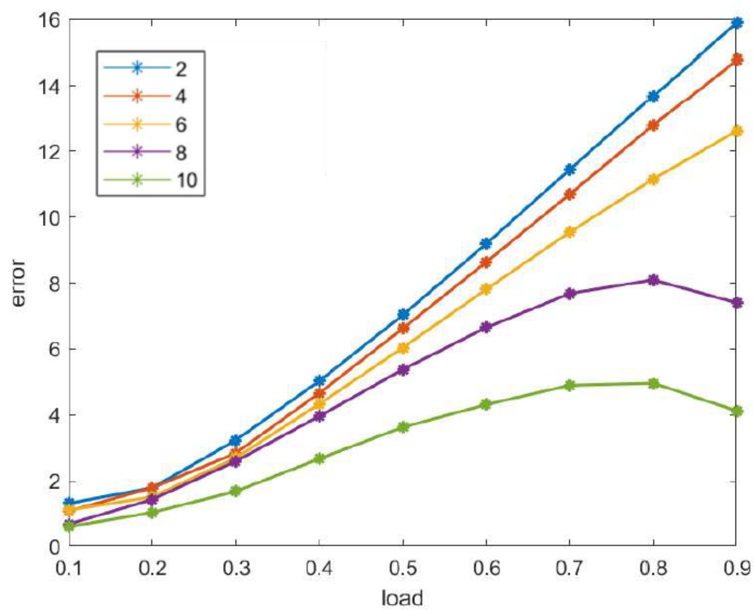

Fig 2: Error vs Load graph for different channels

Algorithm for Error vs Different network size (Constant Density):

For constant density we will increase the box size in proportion to the number of links.

1) Must consider different network size e.g. N= [40 60 80 100 120] having channel M=6.
2) For different loads we will consider the different network topology using different box size e.g. [5.1,6.123,7.17,7.98,8.66]
- Calculate the empirical service rates from the target service rates

3) Plot the error vs different network size graph

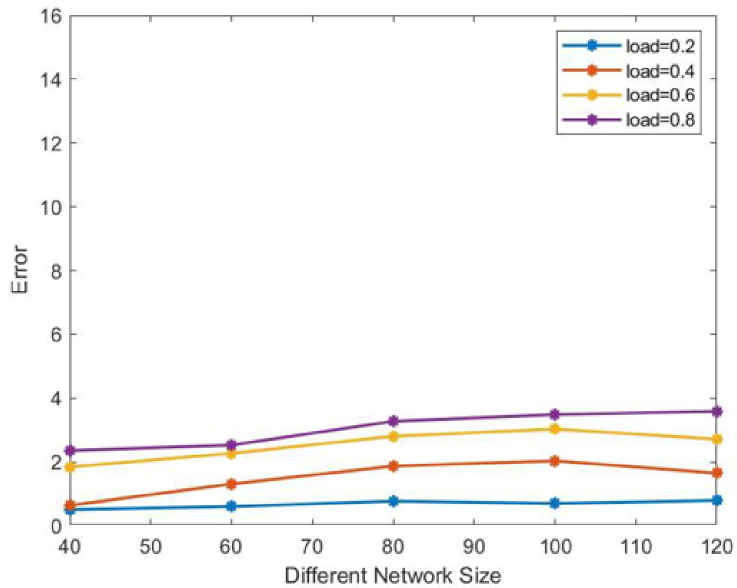

Fig 3: Error vs Different Network for different loads (Constant Density)

Algorithm for Error vs Different network size (Different Density)

For different density we will have to increase the no of links in a constant box size.

1) Must consider different network size e.g. N=[40 60 80 100 120] having channel M=6.
2) For different loads we will consider the different network topology using same box size e.g. 5x5

- Calculate the empirical service rates from the target service rates

3) Plot the error vs different network size graph

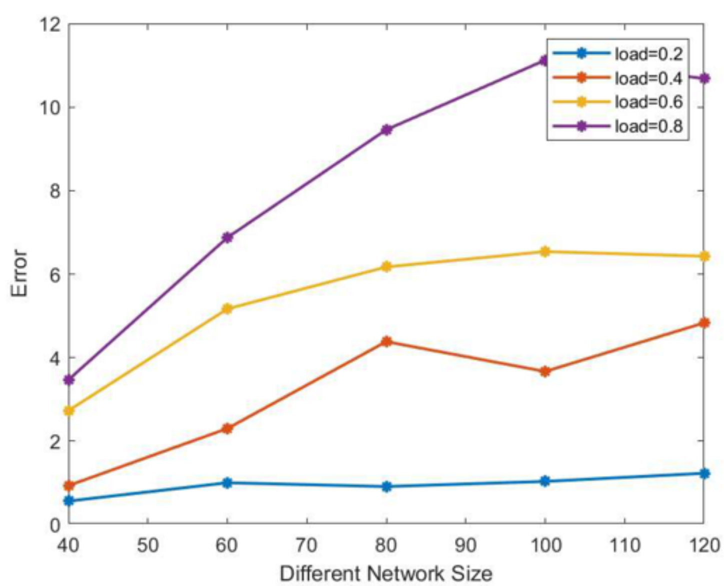

Fig 4: Error vs Different Network for different loads (Different Density)

Observations:

- For error vs different loads graph, we can see that the error significantly increases for increasing load. Moreover, for the increasing channel size our error also increases substantially. The total error will remain in between (1-16) %. This will happen because the more load amount the more links will remain active for a specific channel size and hence error increases rapidly. Besides, less amount of channel size also increases the error due to lack of choice for the links during Markov's iterative method. This also implies that error is a composite function of channel size and load of the links.

- For error vs different network size (constant density) we can get somewhat constant error for different topologies. As the number of links increases, the box size also increases. This helps to keep a constant density in the topology as a result during the iterative method we can get an approximately linear graph. Moreover, the error increases for increasing load because in that case the links will remain more active eventually leading to a larger error than the less load values. Error will remain constant in between (0-4) % at most.

- For error vs different network size (different density) we get rapidly increasing error for increasing network size. As the number of links increases, the box size remains constant. As a result, the density will eventually increase referring to rapid increase in error. Here also we can observe the graphs for higher load will have higher error sequence than lower load due to the link activation time. The error range will reside in between (1-12) % accordingly.

References:

1) **CSMA Using the Bethe Approximation: Scheduling and Utility Maximization**
   Se-Young Yun, Jinwoo Shin, and Yung Yi, Member, IEEE

2) **Free Energy Approximations for CSMA Networks**
   B. Van Houdt

3) **Wikipedia**

4) **ELSclique Function**
   YANHAO WEI