

**EEP-703 Network Lab**  
**Assignment 2-Simulating AODV  
and DSDV in NS2:**

**Harshit Kumar Gupta**  
2013EET2369



Computer Technology  
Department Of Electrical Engineering  
IIT DELHI

February 24, 2014

# Contents

<u>PROBLEM STATEMENT</u>	2
<u>ABSTRACT</u>	3
<u>INTRODUCTION</u>	4
<u>SPECIFICATIONS AND ASSUMPTIONS</u>	5
<u>METHODOLOGY</u>	6
<u>Execution Directive</u>	7
<u>Simulation</u>	8
<u>OUTPUT GRAPH</u>	9
<u>RESULTS AND CONCLUSIONS</u>	18

# PROBLEM STATEMENT

Construct wireless networks and conduct simulations in NS-2 for following:

1. Let us consider a 3 node topology (A, B and C), where A is the source and C is the destination.  
The coordinates of A and C are (50, 50) and (400, 50).  
A and C are out of each others range. B is an intermediate node acting as a router.  
Use DSDV and AODV as the underlying protocols. Consider CBR traffic at 200 Kbps. Use the following scenarios in your simulation:
  1. when B is a static router at (200, 50)
  2. when B is a dynamic router moving from (0, 50) to (200, 50) in a straight line at a constant speed
3. Plot throughput at C against time for both scenarios using both AODV and DSDV.
4. Demonstrate hidden node problem and exposed node problem by conducting simulations on suitable topologies.

# ABSTRACT

The entire code has been written as to simulate on a Network Simulator which inputs a 'tcl' file and when compiled with a ns2 Simulator generates a 'trace' and 'nam' file as Output. Furthermore awk tool has been used as to extract the columns of the trace file and to act them as parameters for deciding the effect of bottlenecks in the Network through realising the Packet Size and the inter delay time between the packets. Finally the value of those parameters is plotted using the 'gnuplot' tool so as to get the values through a graph.

# INTRODUCTION

In 1996-97, ns version 2 (ns-2) was initiated based on a refactoring by Steve McCanne. Use of Tcl was replaced by MIT's Object Tcl (OTcl), an object-oriented dialect of Tcl. The core of ns-2 is also written in C++, but the C++ simulation objects are linked to shadow objects in OTcl and variables can be linked between both language realms. Simulation scripts are written in the OTcl language, an extension of the Tcl scripting language. Presently, ns-2 consists of over 300,000 lines of source code, and there is probably a comparable amount of contributed code that is not integrated directly into the main distribution of ns-2 exist, both maintained and unmaintained. It runs on GNU/Linux, FreeBSD, Solaris, and Mac OS X.

AWK is an interpreted programming language designed for text processing and typically used as a data extraction and reporting tool. It is a standard feature of most Unix-like operating systems. AWK was very popular in the late 1970s and 1980s, but from the 1990s has largely been replaced by Perl, on which AWK had a strong influence.

While the gnuplot is a command-line program that can generate two- and three-dimensional plots of functions, data, and data fits. It is frequently used for publication-quality graphics as well as education.gnuplot can produce output directly on screen, or in many formats of graphics files, including Portable Network Graphics (PNG), Encapsulated PostScript (EPS), Scalable Vector Graphics (SVG), JPEG and many others. It is also capable of producing LaTeX code that can be included directly in LaTeX documents, making use of LaTeX's fonts and powerful formula notation abilities. The program can be used both interactively and in batch mode using scripts.

# SPECIFICATIONS AND ASSUMPTIONS

## **Specifications**

1. position of A (0,50)
2. position of B(200,50) in static topology
3. from (0,50) to (200,50) in dynamic topology
4. position of C(400,50)
5. CBR traffic =200Kb

## **Assumptions**

1. ns2 Simulator will be used for compiling the tcl file.
2. Inspite of perl ; 'awk' tool will be used to cut the parameters.
3. 'gnuplot' has been used to plot the values to study bottlenecks.
4. Other throughput degradation factors have been ignored.

# METHODOLOGY

The methodology that is used for developing this project work is defined below:

1. The entire code is written in tcl file format.
2. First of all all the required 3 nodes are created. Out of which 1 is sender and 1 is receiver node.
3. 1 nodes act as Router which transmit packet from sender to Receiver.
4. Links are Wireless and are created as per Specifications above.
5. A CBR traffic source is Setup over the TCP agent.
6. Once the file is compiled it , the Output trace file is generated.
7. nam file is used for simulation of network in given topology.
8. trace file is used for analyis of network

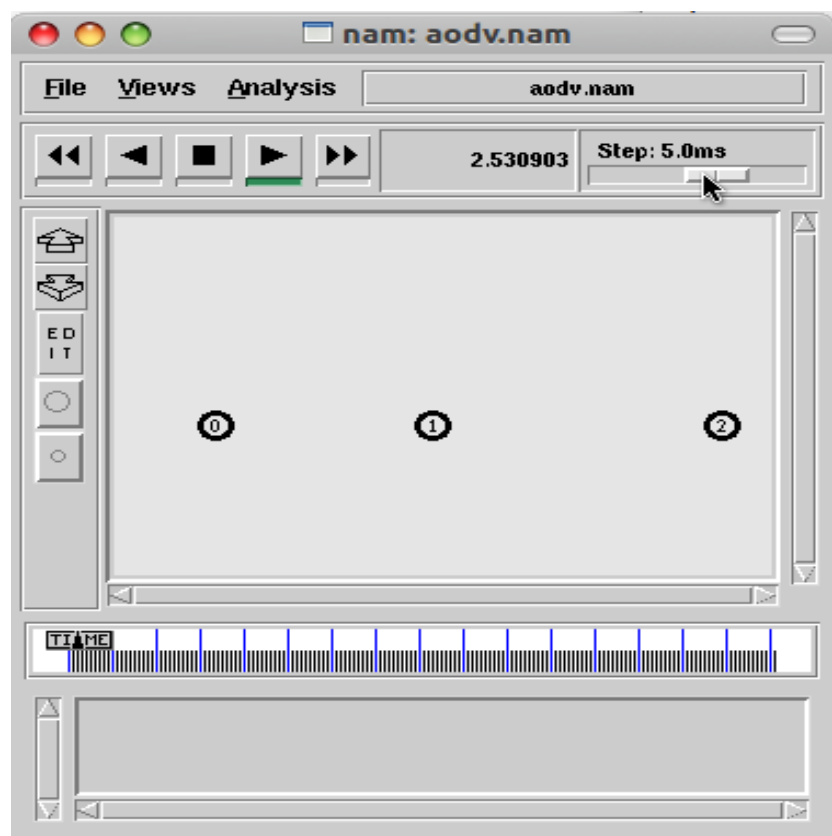
## Execution Directive

The Directions to how do we execute the program as to accomplish the project.

1. Reach the Path of the Project Folder using `cd assign1`
2. Compile the tcl file by `ns tcl-script.tcl [parameters]`
3. `trace-output.tr` and `nam-output.nam` file is generated.
4. Cut the Parameter columns using `awk` as `awk -f filename.awk filename.tr >parameters.dat`
5. plot the parameters using the `gnuplot` as Write `gnuplot` on Terminal
6. `gnuplot throughput.plot ;gnuplot interarrival.plot`
7. This is how we get the graphical analysis of our network.



## Simulation

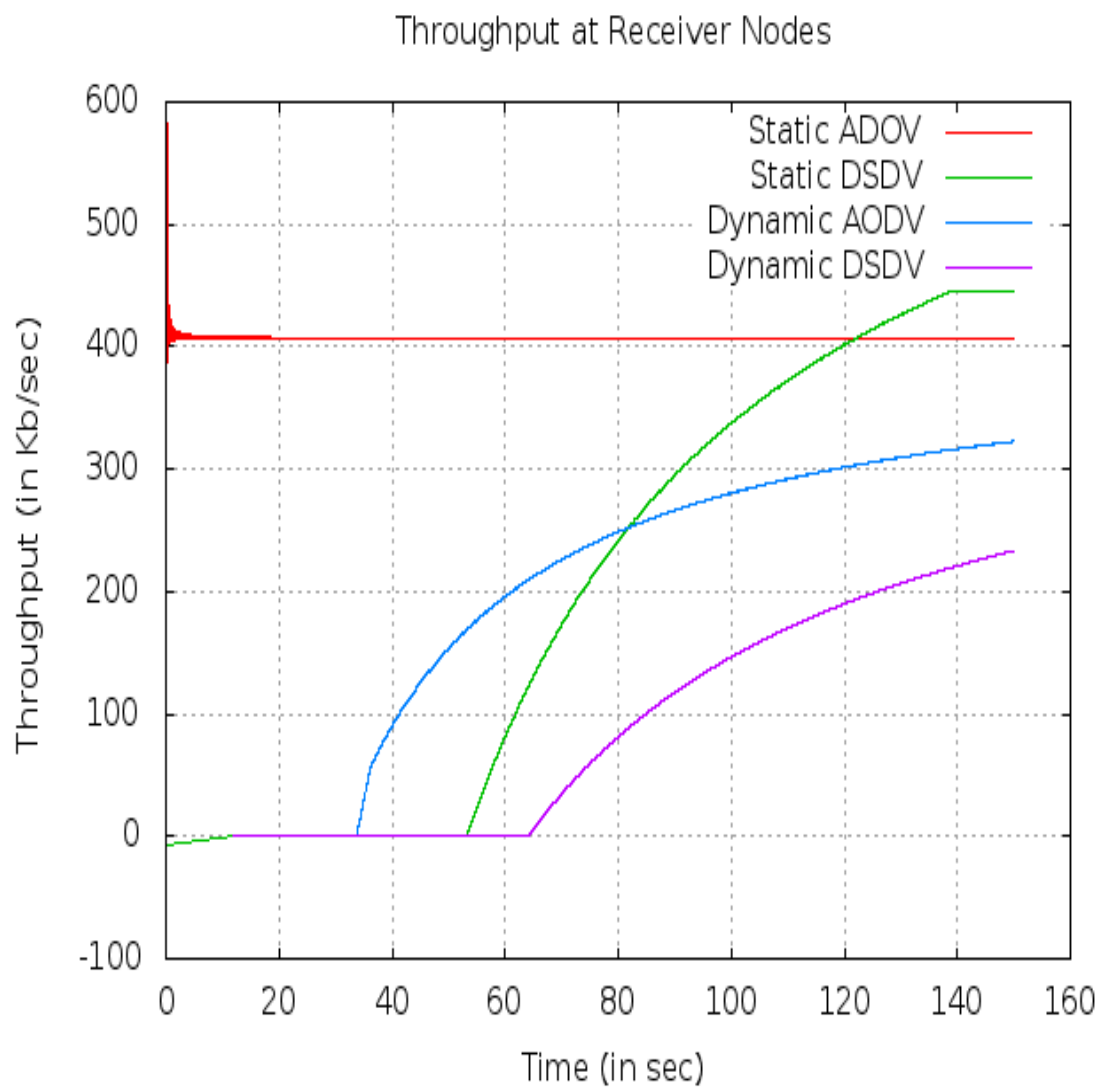


Simulation in NS2

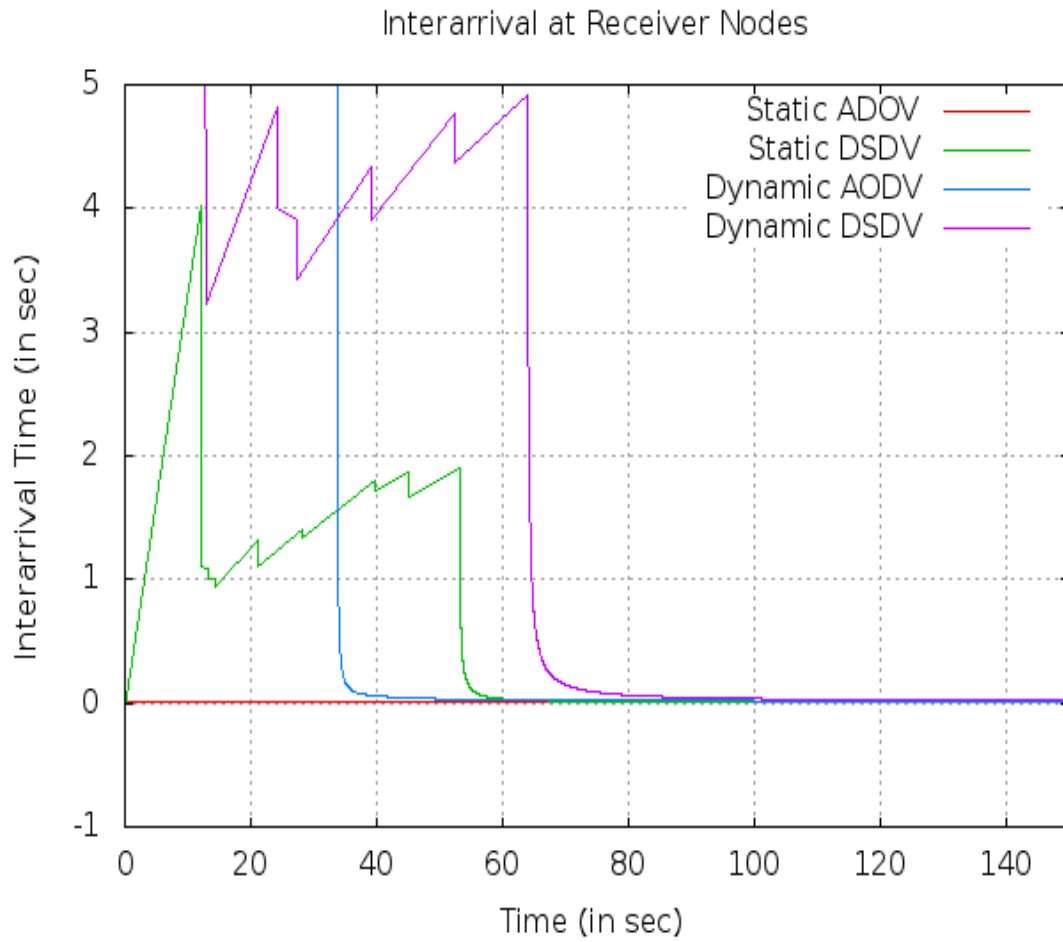


# OUTPUT GRAPH

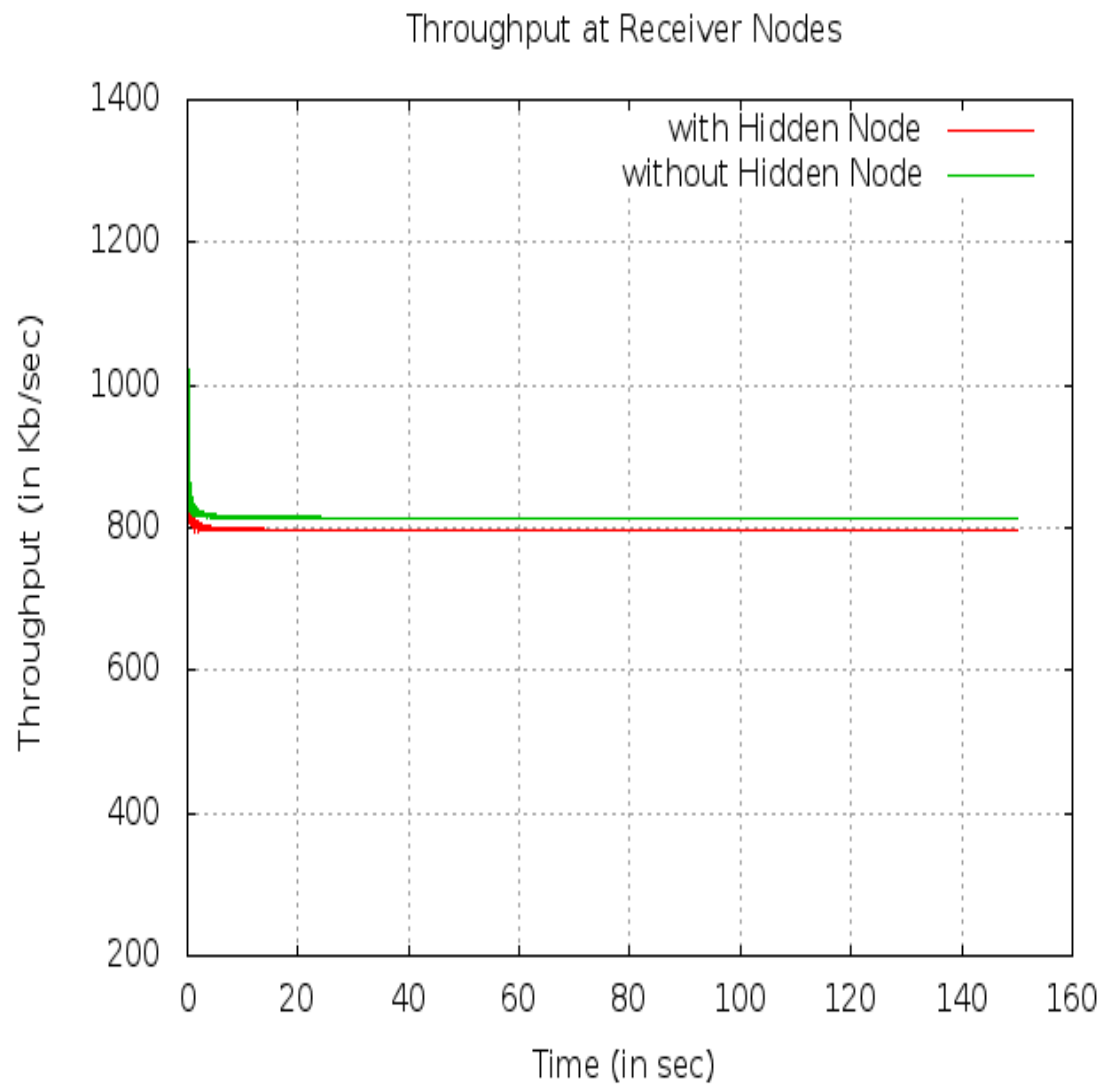
Throughput Analysis in AODV and DSDV



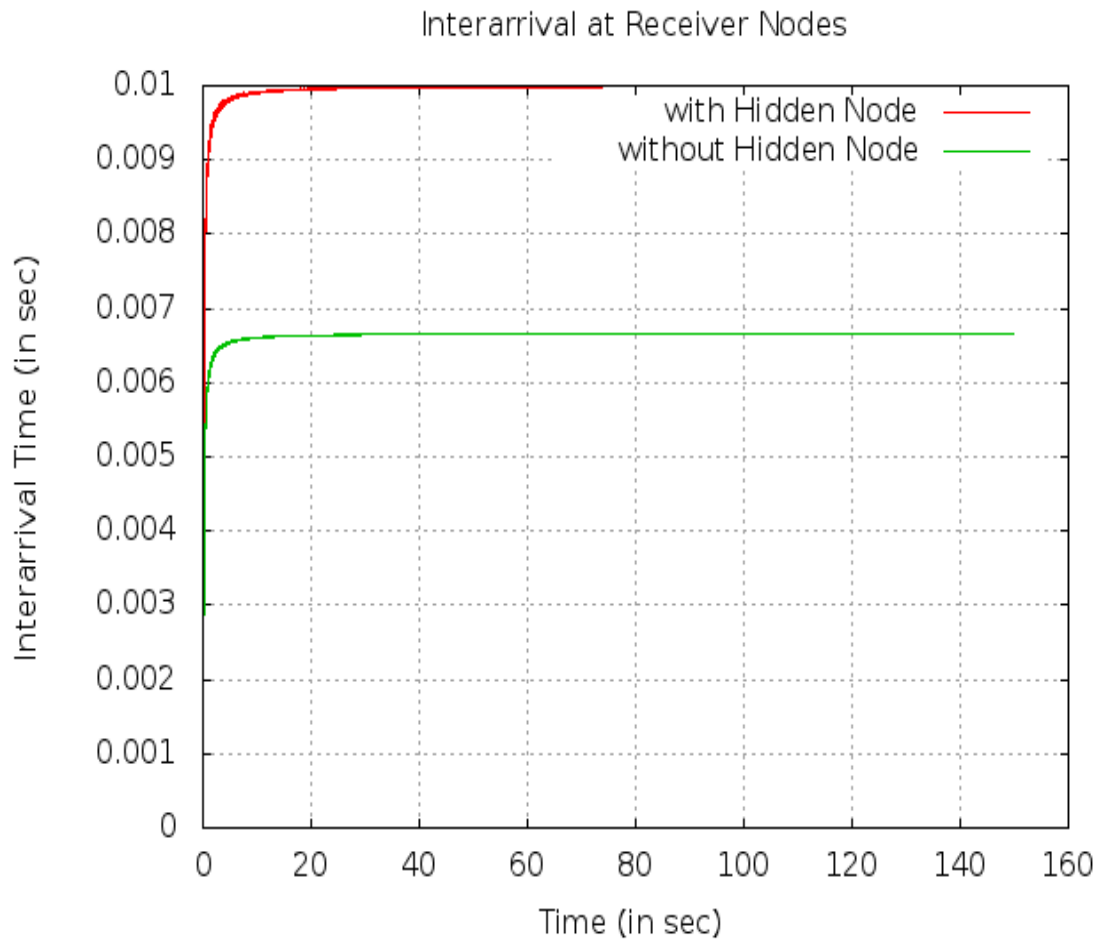
## InterArrival Time Analysis in AODV and DSDV



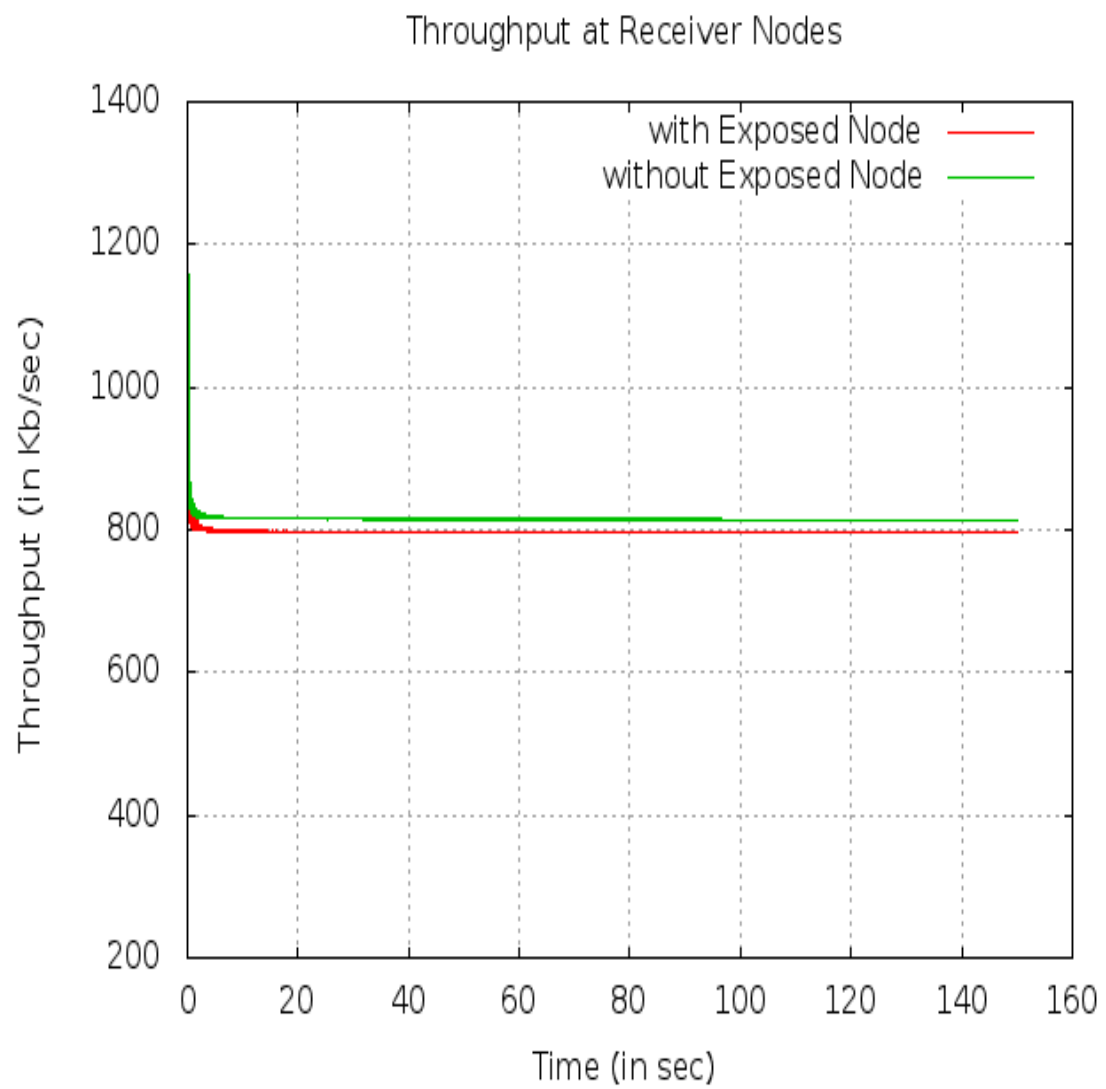
## Throughput Analysis in Hidden Node Problem



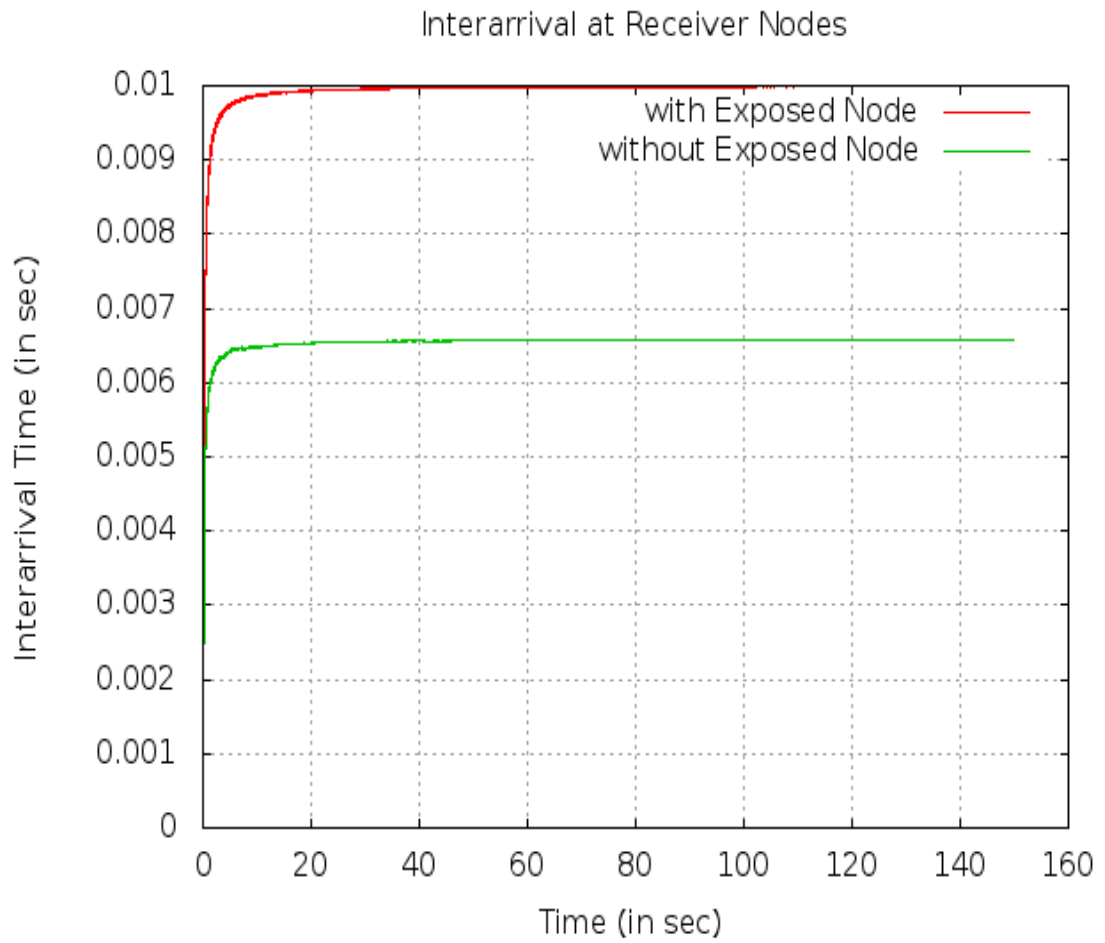
InterArrival Time Analysis in Hidden Node Problem



Throughput Analysis in Exposed Node Problem

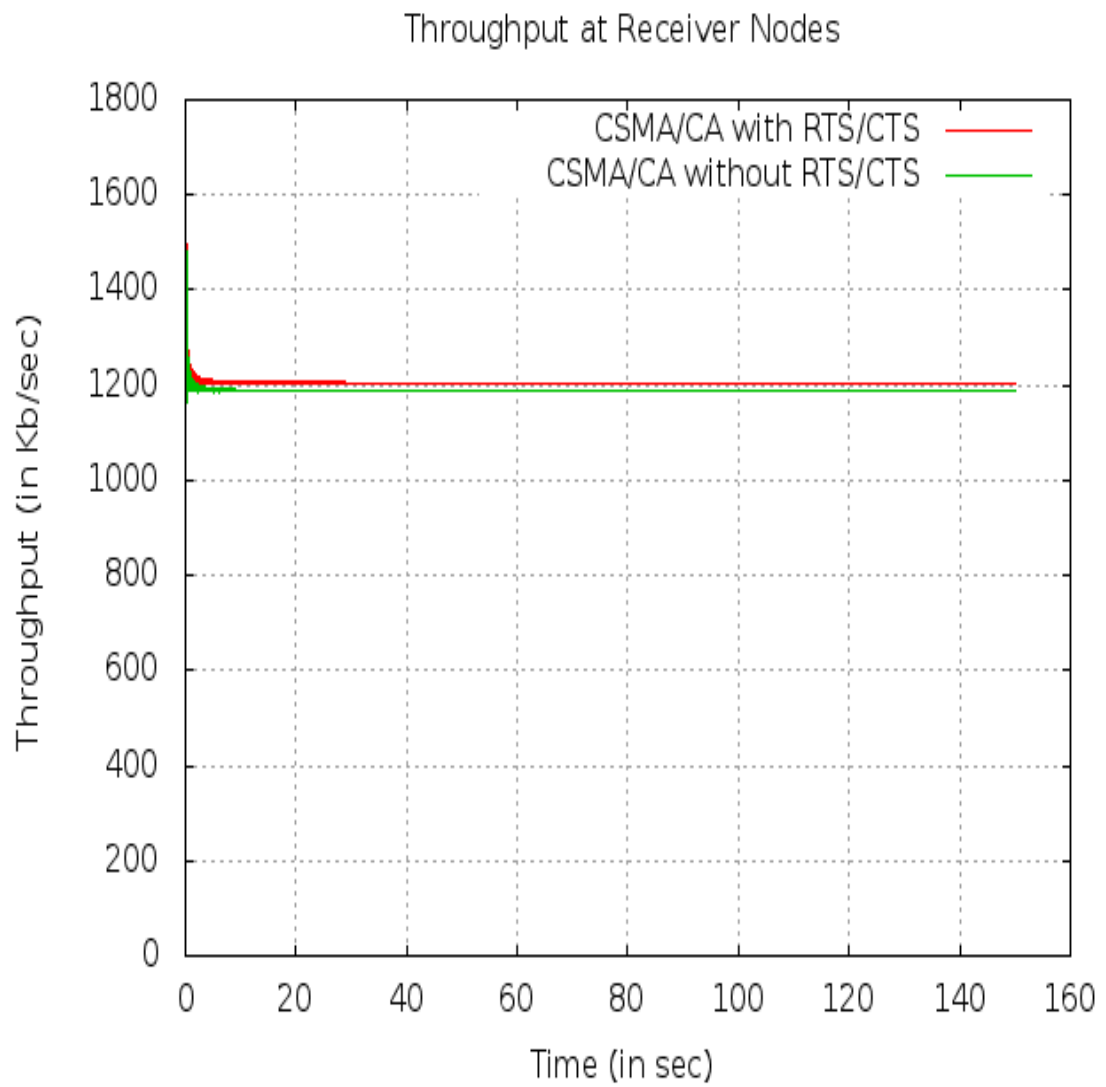


InterArrival Time Analysis in Exposed Node Problem

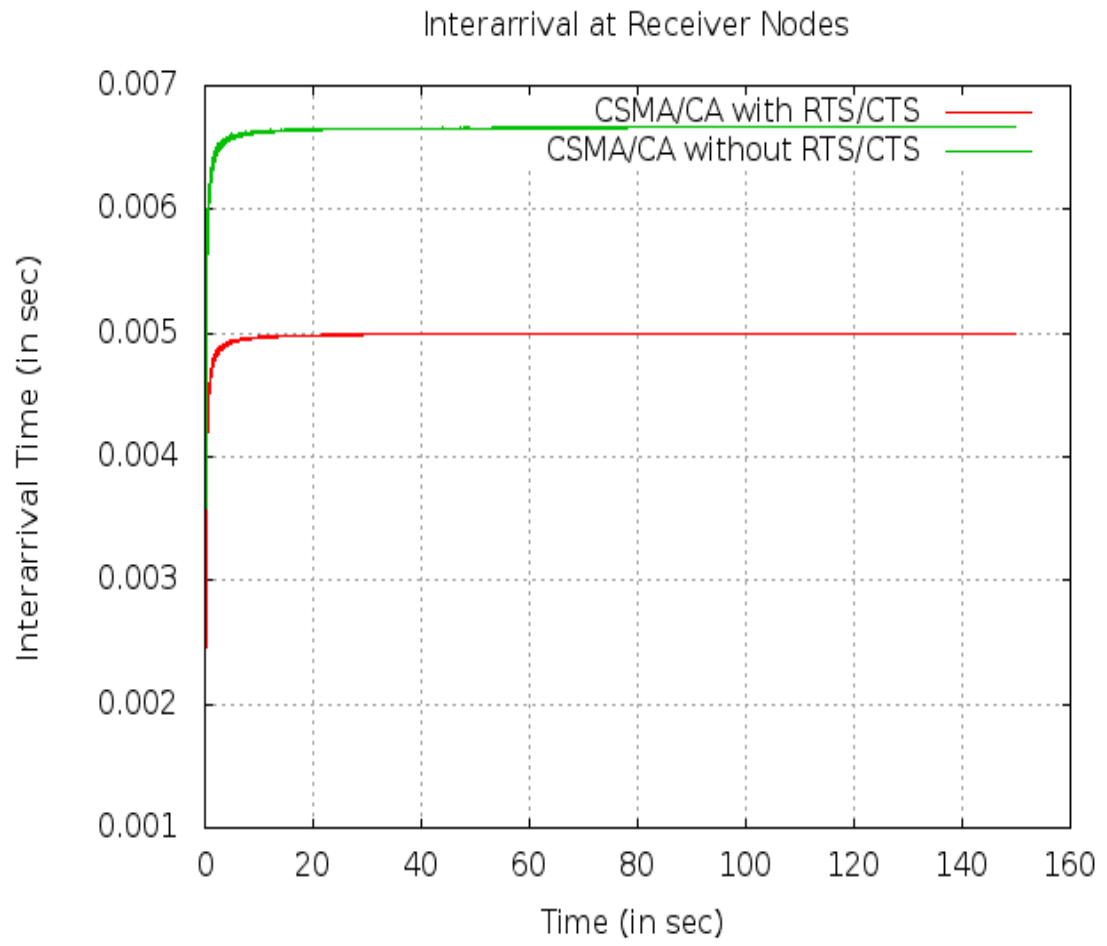


Throughput Analysis in CSMA/CA





InterArrival Time Analysis in CSMA/CA



## RESULTS AND CONCLUSIONS

1. For CBR traffic analysis over TCP and UDP client .tr(trace) file is used.
2. We used 'awk' utility to know performance parameters in network by extracting data from trace file.
3. For simulation of network .nam(network animator file)is used which gives proper visualization of transfer of packets and loss of packets in network topology.
4. Finally we use gnuplot to show the Graph of throughput and inter arrival time.
5. Plotted graphs show differnt behaviour of Network under differnt Protocol.

## Performance Analysis of All Protocols

### Simulation result of STATIC AODV

Throughput : 407.102 bps  
Interarrival Time : 0.0133232 sec

### Simulation result of DYNAMIC AODV

Throughput : 322.707 bps  
Interarrival Time : 0.0168049 sec

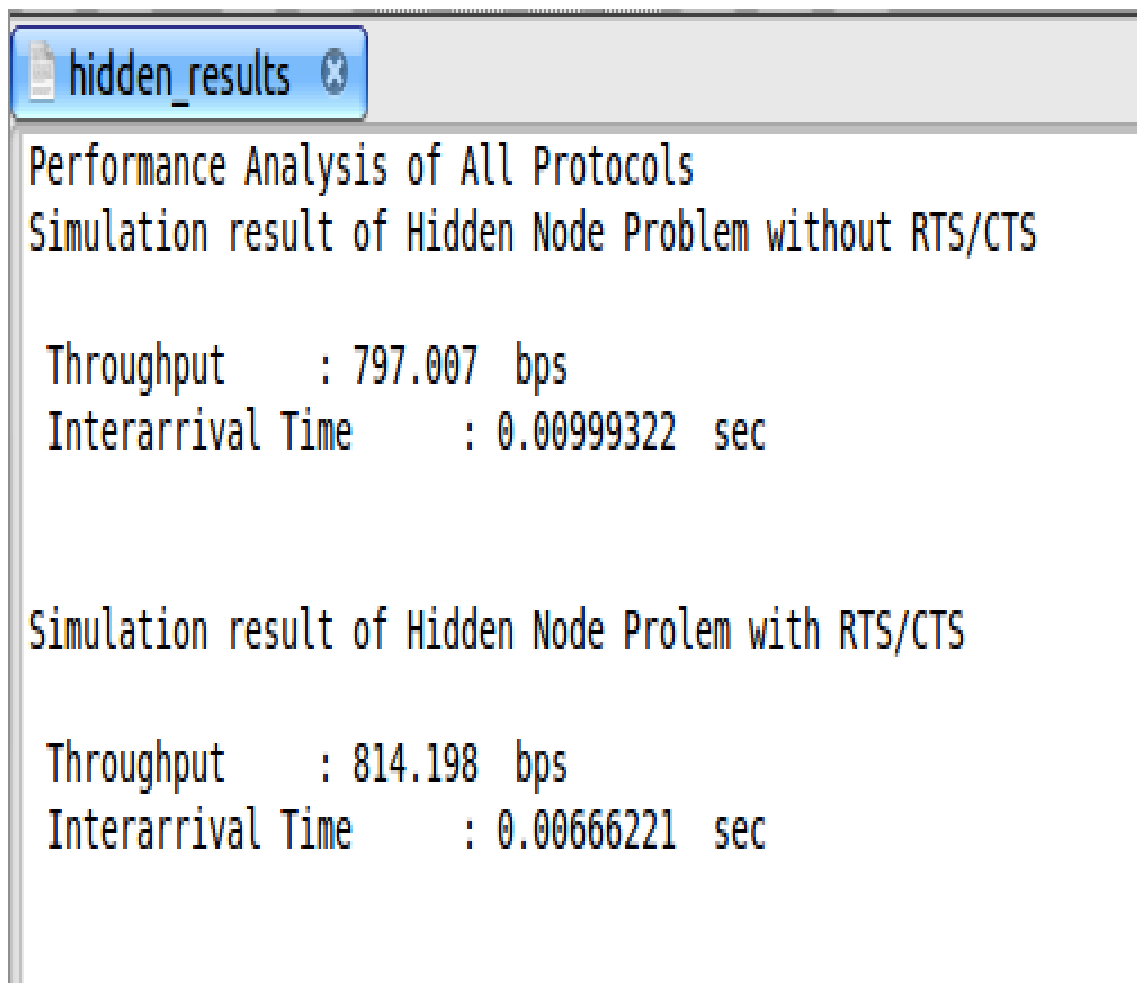
### Simulation result of STATIC DSDV

Throughput : 445.95 bps  
Interarrival Time : 0.00664431 sec

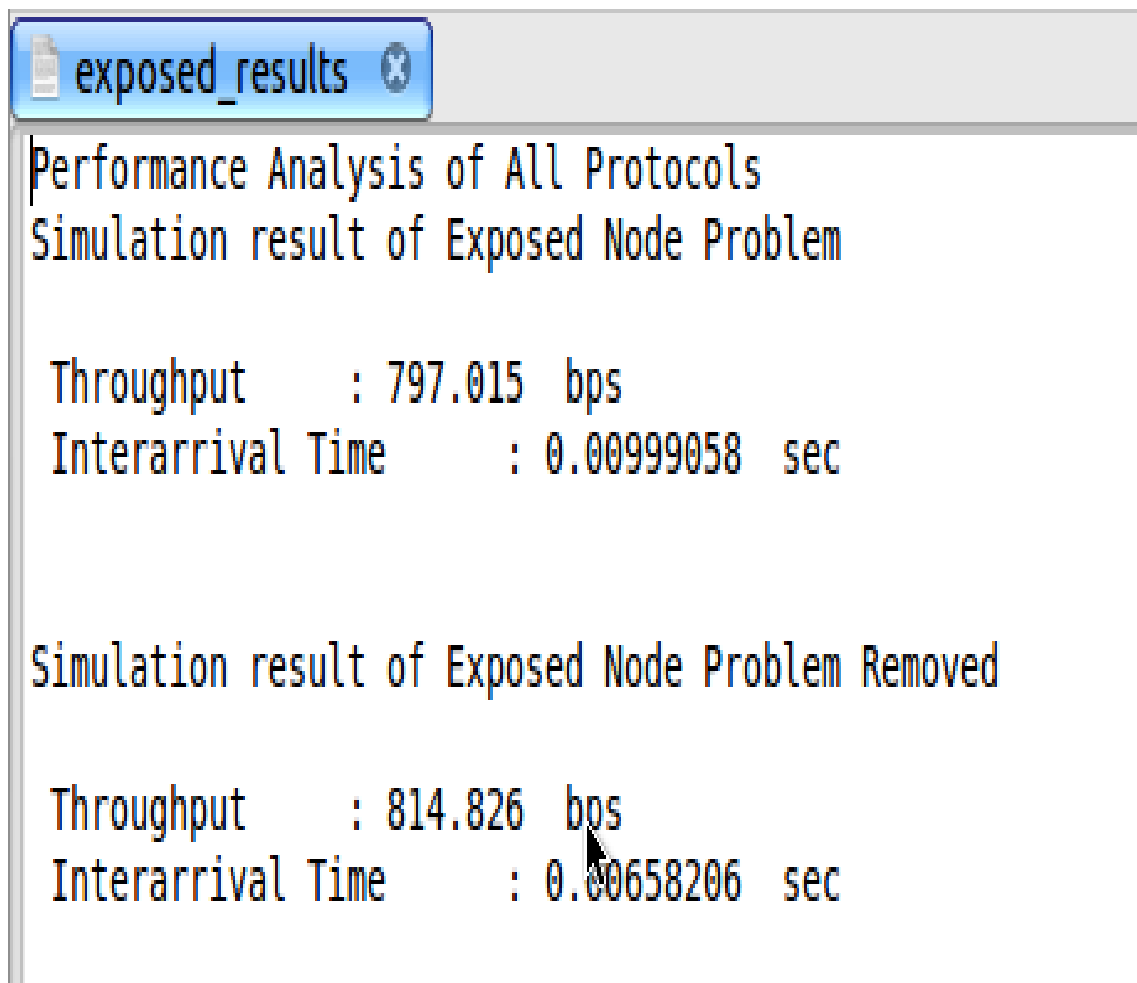
### Simulation result of DYNAMIC DSDV

Throughput : 233.47 bps  
Interarrival Time : 0.0231469 sec

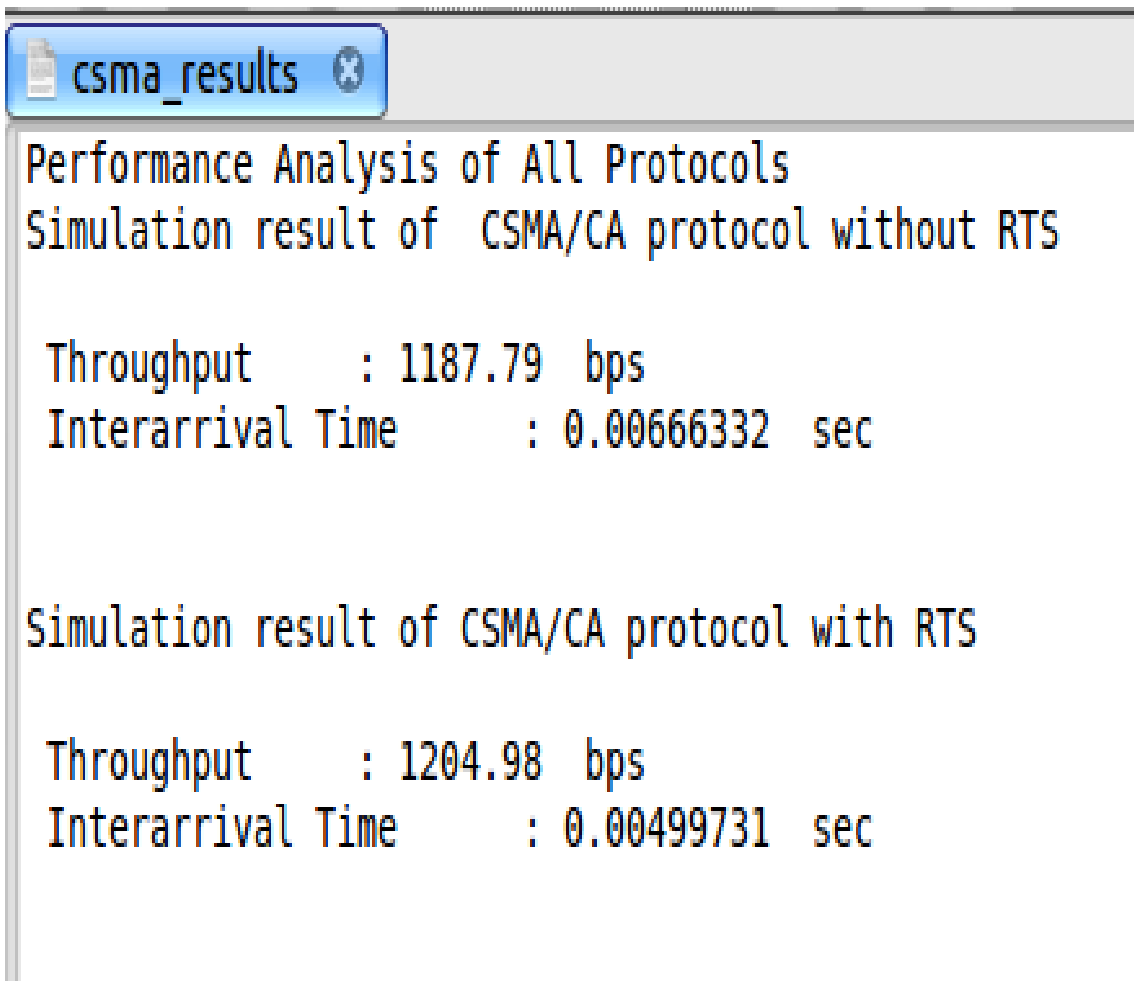
Simulation Result Analysis in AODV and DSDV



Simulation Result Analysis in Hidden Node Problem



Simulation Result Analysis in Exposed Node Problem



Simulation Result Analysis in CSMA and CA Problem