Available online at http://www.csjonline.org/

The Caspian Sea Journal
ISSN: 1578-7899

Volume 10, Issue 1, Supplement 2 (2016) 81-90

Evaluation of the Performance of Congestion Control Methods in Wireless Sensor Networks

Ameneh Gholamhosseinzadeh,
Department of educational Sciences, Amol Ayatollah Amoli, Islamic Azad University, Amol, IRAN
Corresponding Author
Mojtaba Mansoori, *Ph.D.*
Department of educational Sciences, Amol Ayatollah Amoli, Islamic Azad University, Amol, IRAN

ABSTRACT – Wireless sensor networks consist of elements, including sensor, processor and communication elements, which measure the targets in a geographic area. The network can be used at public, commercial, industrial and medical levels. When multiple sensor nodes send packages to a destination node, there will be the possibility of congestion. This leads to the loss of data and excessive consumption of the sensors energy. Therefore, several routing protocols are provided for data transfer and congestion control; each is trying to reduce lost packages and optimal use of energy. In this study, several congestion control methods, including Drop Tail, RED, AVQ were examined and performance of each of the methods was evaluated in the same condition in terms of network topology, number of packages, bandwidth, links, number of buffers and other parameters and compared by calculating the number of packages delivered to the destination node, the number of missing package and so forth. To achieve the above-mentioned goals, NS2 2.35 software simulation environment was used that is an efficient software for wireless sensor networks. Simulated environment was implemented with the same conditions for three congestion control methods of RED, AVQ and Drop Tail. Evaluation and analysis of the results showed that RED algorithm has the best performance and RED Drop Tail has a low performance.
KEY WORDS: wireless sensor network, congestion, congestion control, performance, NS2 simulator

Introduction

Recent developments in microelectronic-mechanical systems (MEMS) and wireless networks enabled the construction of wireless sensor network (WSNs). Wireless sensor network is a network of nodes that receive certain information from the environment in cooperation with each other. This network of many sensor nodes are cheap and low power consumption that are used in military and nonmilitary applications. The applications include monitoring of habitats, pollution detection, water and air quality, smart homes and geographical phenomena [1]. Wireless sensor networks are based on the occurrence of event. The sensors in the network, which are in a distributed range, monitor and measure physical and environmental conditions in collaboration with each other and report this information to the central node. In the case of events in areas controlled by sensors, the sensors are activated and start sending measured data. Simultaneous transmission of data by sensors increases data traffic load, which consequently leads to the phenomenon of congestion on the network. Congestion in the sensor networks results in the loss of sent sensed packages, resuction in the package delivery rate to central node, and finally reduction in the quality of the communication channel and network performance. Congestion in the wireless sensor networks has a negative effect on the network performance, namely it reduces the power and increases the use of energy for the package delivery. This study investigated the detection and congestion control methods proposed in the wireless sensor networks in terms of network performance criteria in NS2 simulation environment. This paper aimed to provide a variety of congestion control methods for wireless sensor network. Since congestion control is one of the factors determining network performance, it is necessary and important to do research on the methods and compare their performance. Several congestion control methods have been provided for different applications. In this paper, the methods are first classified, one sample is selected in each category and assessed in terms of network performance parameters, including package loss rate of a flow and upstream sources service rate of the package production so that the algorithm can be compared in the same conditions and selected the best algorithm. In this study, three algorithms RED, AVQ and Drop are compared.

Methodology

This study used library studies related to the topic of the study, including theses, specialized books, articles and Persian and Latin magazines in the university, National Library, Iran Institute of Science and Information Technology as well as reliable papers in databases available on the internet in order to develop the theoretical framework. Chellaprabha et al. (2012) examined the advanced congestion control in DCCP_TCPLike, in which the efficiency of traffic control algorithm is considered as the best and strongest protocol of the Internet transmission. The researchers measured and analyzed the algorithm performance as the most appropriate method to wireless networks and established the reset controlled (RC) of average network performance of DCCP_TCPLike as the best algorithms in computer networks. The results show that the performance of the model

RC_DCCP_TCPLike in the network is more efficient. Moreover, the re-control mechanism can be modified and enhanced to improve overall network performance to DCCP_TFRC. Chellaprabha used NS2 simulation [2]. Sergiou et al. (2013) investigated an alternative hierarchical tree route for congestion control in wireless networks. They removed traffic and congestion in wireless sensor networks by two different methods of resources control and traffic control, and built HTAP algorithm with a few simple solutions and lowe calculation operations. Moreover, the algorithm HTAP evaluated different scenarios compared to resources control. The researchers noted that there is no traffic control and traffic bulke on the network. Sergiou et al. also pointed that when a crash occurs in network, HTAP algorithm transmits the packages over alternative networks; that is, if a route is full of traffic, the algorithm automatically replaces another route that is useful for resources control and traffic bulke control. It was found that HTAP algorithm is very simple and efficient and more successful than other algorithms [3]. Sergiou et al. (2014) studied congestion control in wireless sensor networks through dynamic alternative route. They found that a great deal of data creates congestion conditions due to the limitations of resource and wireless network nodes. Therefore, they concluded that the collision algorithms require professional algorithms and therefore investigated the replaced dynamic algorithm in the collision route of their own research. Each wireless sensor network is a combination of several sensor nodes that collects environmental data and send them to the monitor. The network was produced due to the resources, storage resources, processing, energy and communication constraints and limitations. The monitor can be a stationary or mobile node. The mobile node collects, stores, processes and analyzes data moving around the network. When the sensor nodes start sending information to the monitor, it leads to information overload in the monitor where the congestion problem occur. In package switch networks, congestion is the major problem of the network, in which the network capacity is completed in one or more intermediate nodes because of the increased speed of resources. This leads to the loss of the sent package. Therefore, it is necessary to control congestion in order to allocate the resources efficiently. This paper intended to classify and examine congestion management algorithms and protocols for wireless sensor networks and analyze optimal performance variables in these algorithms using NS2 simulator software according to the classification.

Congestion Control Protocols

Recently, various protocols have been provided for congestion control. The protocols are different in congestion recognition, congestion notification and control techniques.

Congestion control protocols are classified into four categories:

- Traffic control: congestion is controlled by reducing the number of packages sent to WSN. Traffic control techniques have two types of additive increase multiplicative decrease (AIMD) or rate-based mechanisms.
- Source control: one of the disadvantages of traffic control mechanism is the reduced data rate. It can be useful to reduce the rate of source data in some applications. For example, in important event-oriented applications, all important information should reach to the well. Source control mechanisms are introduced to fix this flaw that control congestion by increasing the source or using other ways to transfer data to the source. This method works better than traffic control methods. In some networks, the combination of the two methods is used to enjoy the benefits of both.
- Priority congestion control mechanism: some networks use priority MAC techniques for filled nodes in order to achieve access channels. In other words, congestion is controlled by different priorities in different situations in this mechanism.
- Queue assistant mechanism: This mechanism controls the traffic by managing the queue of packages sent.

Table 1 shows some features of congestion control protocols.

Protocol	Publication year	Congestion detection	Congestion notification	Rate setting
[6]RED	2010	Average queue length	Implicit	Rate setting
[4]AVQ	2006	Queue length and application type	-------	Direction switch
[5]DropTail	2001	Queue length and channel position	Explicit	---------

Table 1. Some features of congestion control protocols in the wireless sensor network

Random early detection algoritm (RED)

One of the important mechanisms to provide quality service and congestion control in the networks is the use of active queue management mechanisms and scheduling. With these mechanisms, the traffic is controlled and network performance reduction is avoided. Queue memory management algorithms control the amount of packages in the queue by determining when and which packages should be discarded in the case of congestion. RED is one of the major algorithms of active queue management. The reasons for the popularity of the algorithm is the average delay of low queue and high-performance. However, the performance of RED is heavily dependent on the setting of the control parameters. RED algorithm has two components. The first component estimates the average queue length using exponential weighted average, which can be interpreted as a low pass filter. The second component makes decision on the removal of the package reached. Average weighted queue determines the explosive traffic degree that is allowed to enter the queue. The average queue length is calculated with respect to the idle periods of queue and estimation of the number of m packages that can pass the well during the queue unemployment. After a period of unemployment, the weighted average queue length is calculated. The main feature of RED algorithm is that the average queue length is always kept low, and yet it does not crash with instant explosions and transient congestions.

Using RED, missing a single package is sufficient to inform the nodes of the congestion. When a package is lost, the well warns the origin in an uncertain maner that the discarded package faced congestion along the route to destination. In response to this unclear warning, the source reduces its sending rate so that the router's queue buffer is not overflown.
Possible congestion on the network is identified by calculating the weighted average queue length in the algorithm, and in case of detection of congestion in the network, the incoming package is deleted or marked. The average sending time is calculated for each package sent according the following pseudo-code, and necessary warning will be provided over a specific time interval.
For each package arrival
Calculate the averae queu size (avg)
Ih min_{th} <= avg <= max_{th}
Calculate probability p_b
With probability p_b
Mark the arriving package
Else if max_{th} <= avg
Mark the arriving package
Maxth and minth parameters are selected according the desired average queue size.

Drop Tail
One of the simplest mechanisms of queue management is Drop Tail. This method is a complete lack of queue storage management. When a package reaches the end of queue whose sources have been fully used, the package along other all packages are discarded as long as a space is created in the queue.
Benefits of Drop Tail include:
- Implementation of this method for developers and users is simple to understand.
- This method can reduce the number of packages discarded, especially if a large memory size is allocated to the queue. Of course, queues with great length leads to an increased end-to-end delay for all currents that use this queue.
Restrictions of Drop Tail include:
- As long as the queue is not fully completed and the sources are not fully used, Drop tail does not discard the packages. As a result, after the queue completion, until a space is not empty in the queue, it cannot absorb the rest of the traffic. This situation leads to the blockage of queue due to the lack of buffer space to store the incoming packages. Thus, a small number of currents can occupy all buffer capacity and prevent access to the rest of the current to the queue. - Drop Tail causes the lines to remain full for long periods, because the host system, when able to recognize congestion (by discarding packages) be the queue to one hundred percent of its capacity and resources are fully occupied. - Drop tail causes completing the queues for long term intervals because the host systems can detect congestion (by discarding the packages) when the queue reaches its 100% capacity and sources are fully occupied.

AVQ
This method uses a virtual queue whose capacity is less than the actual capacity of the connection. When a package reaches the real queue, virtual queue is also updated to reflect the new package arrival. When the virtual buffer is filled, the packages are discarded/marked in the real queue, then virtual capacity for each connection is changed such that the total inflows to the connection achieves a desired productivity. Suppose that C shows the connection capacity and μ is the optimal productivity of the connection. Then, AVQ method acts in each router as follows:
Router considers a virtual queue whose capacity is $\hat{C}<C$ and the buffer size is the same as the buffer size of the real queue. One virtual queue is inserted in virtual queue for each package arrival (if there is enough space in buffer). If the new package leads to the virtual buffer overflow, the package is discarded from virtual buffer and the real package is also discarded or its' ECN bit is marked. For each package arrival, the virtual queue capacity is updated according to the following relation:
$\hat{C}=\alpha(\mu C-\lambda)$
Where λ is the connection input rate. It should be noted that no real entry and exit of the package is not performed in the virtual queue and only the necessary changes are applied in the virtual queue.
Familiarity with NS2 simulator
NS2 is a discrete event simulator that is used to study the dynamic nature of communication networks. With this simulator, simulation of protocols, system network functions and wireless networks such as TCP and UDP, routing algorithms and so forth are well done.
Assessment conditions
To evaluate and compare the results, it is necessary to implement and provide an environment with the the same conditions. Thus, a network is depicted in NS2 software with the conditions listed in the following table.
The simulated environment was written by programming language Otcl and all functions used were placed in a file to better understand the written plan.

Table 2. General characteristics of the network designed for evaluation

Row	Feature	Value
1	Simulation environment size	500*500
2	Number of sensors	50
3	Number of wells	1
4	The protocol used	UDP
5	Simulation time	2"
6	The stored amounts	Trace: MAC, Route, Agent
7	Radio standard	802.11
8	Location of sensors energy supply	Embedded battery
9	Physical layer model	914 MHz
10	Routing protocol	AODV
11	Initial energy of the well	100
12	Operating range	100m
13	Initial energy of the sensor	10
14	Operating range	50m
15	Maximum buffer length	30
16	Average buffer capacity	25
17	Bandwidth	
18	Use of energy in sending the package	0.024

As shown in Table 2, the simulated environment has 50 nodes and a well where all nodes will send their information packages to the above-mentioned node.

Figure 1 shows a graphical view of a simulated program that is stored in a file with the extension .nam. Figure 1 presents a view of the output file of the transactions, including a file with the extension .tr. The file on will be the basis of all evaluations.

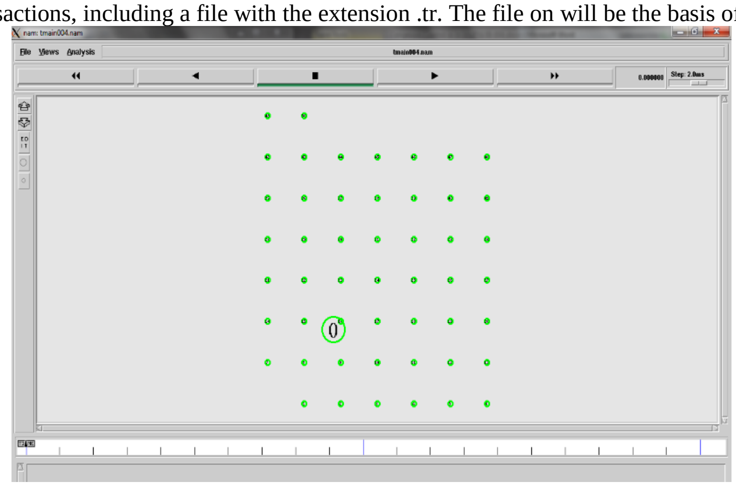
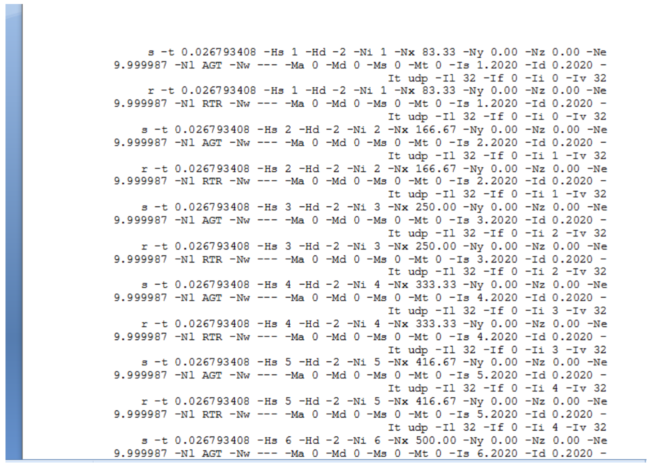


Figure 1. A graphical view of a simulated program

In Figure 2, Node 0 indicates the well that is greater than the other nodes and other nodes are randomly scattered around the wells. All transactions and positions, including successful and unsuccessful storage are presented in the figure. Figure 3 is a view of the program where the black circles indicate the operating range of the sensor, the black dotted lines indicate the sending of the package and the black squares show the occurrence of the congestion and packages overflow.



```
     s -t 0.026793408 -Hs 1 -Hd -2 -Ni 1 -Nx 83.33 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 0 -Iv 32
     r -t 0.026793408 -Hs 1 -Hd -2 -Ni 1 -Nx 83.33 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 1.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 0 -Iv 32
    s -t 0.026793408 -Hs 2 -Hd -2 -Ni 2 -Nx 166.67 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 2.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 1 -Iv 32
    r -t 0.026793408 -Hs 2 -Hd -2 -Ni 2 -Nx 166.67 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 2.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 1 -Iv 32
    s -t 0.026793408 -Hs 3 -Hd -2 -Ni 3 -Nx 250.00 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 2 -Iv 32
    r -t 0.026793408 -Hs 3 -Hd -2 -Ni 3 -Nx 250.00 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 3.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 2 -Iv 32
    s -t 0.026793408 -Hs 4 -Hd -2 -Ni 4 -Nx 333.33 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 4.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 3 -Iv 32
    r -t 0.026793408 -Hs 4 -Hd -2 -Ni 4 -Nx 333.33 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 4.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 3 -Iv 32
    s -t 0.026793408 -Hs 5 -Hd -2 -Ni 5 -Nx 416.67 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 5.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 4 -Iv 32
    r -t 0.026793408 -Hs 5 -Hd -2 -Ni 5 -Nx 416.67 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 5.2020 -Id 0.2020 -
                                        It udp -Il 32 -If 0 -Ii 4 -Iv 32
    s -t 0.026793408 -Hs 6 -Hd -2 -Ni 6 -Nx 500.00 -Ny 0.00 -Nz 0.00 -Ne
9.999987 -Nl AGT -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 6.2020 -Id 0.2020 -
```

Figure 2. A view of the output of transactions

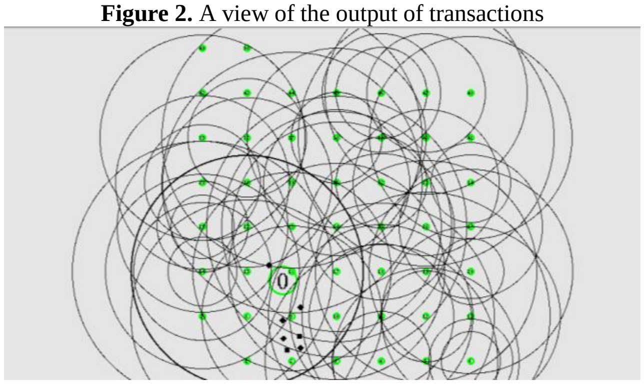


Figure 3. A graphical view of the range of transactions and congestion occurence

Description of the transaction output file

As shown in Figure 3, the output of transactions carried out shows the lines of origin and destination and package position. All future evaluations will be based on this file. In the simulated environment, the packages are stored in the queue by sending each package to the well, and will not be entered the queue in the case of overflow. Figure 4 shows the queuing of the network.

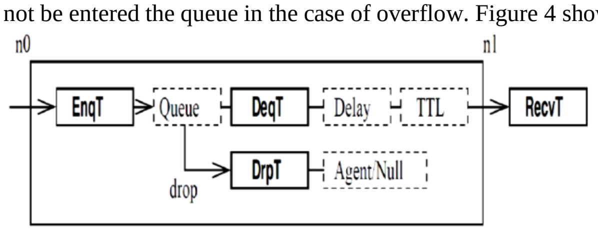


Figure 4. Queue formation in the sensor and well communications

As shown in Figure 4, Enqt records the data of packages that enters the link and is linked to the input queue. If the packages overflow, the information in the packages (which has been lost) are controlled by Drpt. Drpt contains packages information that are removed from the queue and finally Recvt gives us information about the packages reached to the destination.

In NS simulation software, transactions are stored on a ascii file meaning that events taken place during the process of simulation along with the time of occurrence are recorded in the file. This file contains 12 fields as Figure 5.

Event	time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**

Figure 5. Fields stored in the transaction file

Description of the fields are as follows.
1-The first field shows the closed position that is equivalent to four possible symbols +, -, r or d.
r: (Recive) shows that the successful reaching of the package.
+: Entering the packages into the queue (enqueued)
-: Exiting the package from the queue (dequeued)
d: Lossing- dropping from the queue (dropped)
2. The second field is the time of occurrence of each transaction.
3. The third field is the input node of the link where the transaction takes place.
4. The fourth field is the output node of the link where the event occurs.
5- The type of package shows (TCP or CBR, etc.).
6. Package size
7. Flags values
8- Current identifier. An identifier is allocated to each current flow; for example, we have set 1 for TCP one and 2 for CBR.
9. The source address that is displayed as node.port.
10. The destination address that is displayed as node.port.
11. The package sequence number in the network layer protocol.
12-ID is unique to each package.

Simulation of the for congestion control methods

Above-mentioned methods were selected for evaluation. For each method, thse separate programs were written with the same conditions listed in Table 4-4. In programming, the intended method algorithm was implemented resulting in two output files for each of the algorithms. The two output files show the conditions and the number of packages sent, packages reached the destination, the missing packages and time and bulk of packages.

Management of the output files of algorithms

Output files of algorithms have high bulk and management and extraction of mentioned fields are very time-consuming. For example, the output file of the transactions related to DropTail algorithm is 27MB. Thus, it was necessary to resolve the problem using Linux utilities. In this study, the utility AWK is used.

The runtime of the program is 200 units that need to be controlled in every 50 units in addition to the beginning and end of the algorithm so that the algorithm applications can be determined during runtime. For this purpose, four separate programs are used to control the output files.

Figure 6 commands used to control the output file is provided at the time of 0 to 50.

Figure 6 shows the commands used to control the output file of the algorithm at the time of 0-50 units.

```
BEGIN{
FS=" "}
{nl++}

{if($3<=0.50)
{
 if($1=="d"){l=l+1}

 if($1=="r"){r=r+1}
}
}
END{print "Number Of Recived Messages="r,   "Number Of Lost Messages="l}
```

Figure 6. The extraction program of packages from 0-50

Figure 7 shows the commands used to control the output file of the algorithm at the time of 50-100 units.

```
BEGIN{FS=" "}
{nl++}
{if($3>=0.50)
{
if($3<=1.00)
{
 if($1=="d"){l=l+1}

 if($1=="r"){r=r+1}
}}}
END{print "Number Of Recived Messages="r,   "Number Of Lost Messages="l}
```

Figure 7. The extraction program of packages from 50-100

Moreover, a separate program is written for 100-150 and 150-200 times.

Drop Tail algorithm

As shown earlier, DropTail algorithm is a very simple method to control congestion. In the algorithm, when a package reaches the bottom queue whose sources have been fully used, the package along with all other packages are discarded until a space is provided in the queue. Implementation of this algorithm was performed in TMain004.tcl file. The two output files of TMain004.tr and TMain004.nam are created by running the above-mentioned file.

Table 3. presents the results of the algorithm implementation during according to the above-mentioned conditions.

Title	Number
Number of total sent packages	49681
Number of total received packages in destination	23770
Number of total missing packages	25911
Number of packages in 0-50	7945
Number of missing packages in 0-50	6376
Number of packages in 50-100	7369
Number of missing packages in 50-100	10106
Number of packages in 100-150	5425
Number of missing packages in 100-150	5195
Number of packages in 150-200	3262
Number of missing packages in 150-200	3860

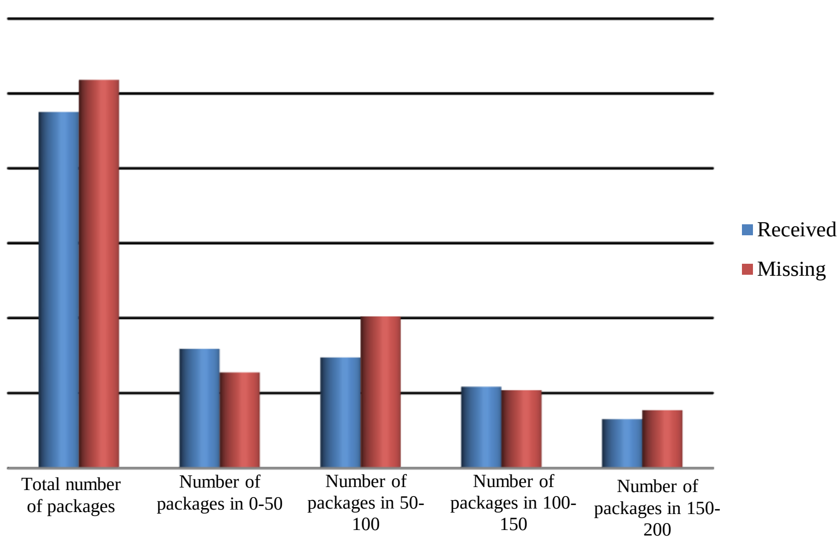


Figure 6. Comparison of the packages via Drop Tail algorithm

RED algorithm

RED is one of the major algorithms of active queue management. The reasons for the popularity of the algorithm is the average delay of low queue and high-performance. However, the performance of RED is heavily dependent on the setting of the control parameters. RED algorithm has two components. The first component estimates the average queue length using exponential weighted average, which can be interpreted as low pass filter. The second component makes decision on the removal of the package reached. Average weighted queue determines the explosive traffic degree that is allowed to enter the queue. Implementation of this algorithm is stored in TMainRed.tcl file where the output files of 18 MB TMainRed.nam and TMainRED.tr are located.

Table 4. shows the results of RED algorithm.

Title	Number
Number of total sent packages	45411
Number of total received packages in destination	31348
Number of total missing packages	14063
Number of packages in 0-50	10913
Number of missing packages in 0-50	3090
Number of packages in 50-100	11236
Number of missing packages in 50-100	4116
Number of packages in 100-150	6550
Number of missing packages in 100-150	2397
Number of packages in 150-200	4626
Number of missing packages in 150-200	1460

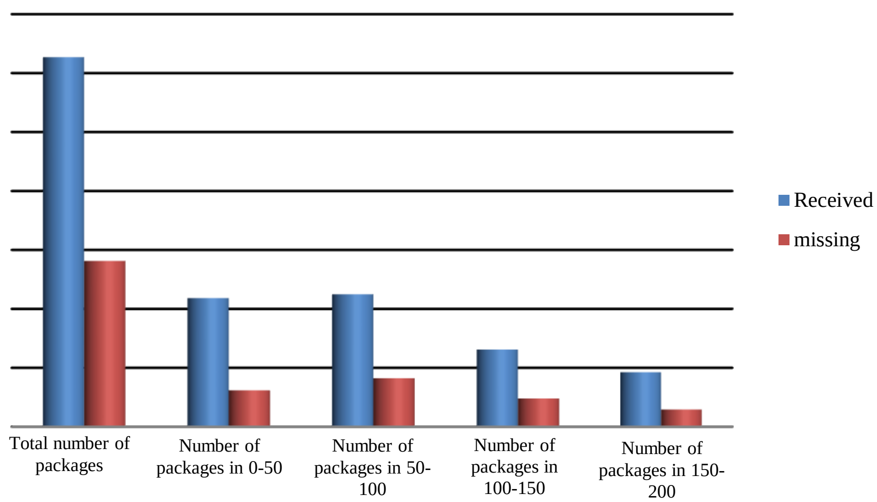


Figure 7. Comparison of the packages with RED algorithm

AVQ algorithm

This method first uses a virtual queue whose capacity is less than the actual capacity of the connection. When a package reaches the real queue, virtual queue is also updated to reflect the new package arrival. When the virtual buffer is filled, packages on the actual queue are discarded/removed, then virtual capacity for each connection is changed such that the total inflows to the connection reaches a desired productivity. Implementation of this algorithm is stored in the TMainAVQ.tcl file where the output files of 18 MB TMainAVQ.tr and TMainAVQ.nam are located. Table 5 shows the results of the AVQ algorithm.

Table 5. Results of AVQ algorithm

Title	Number
Number of total sent packages	50603
Number of total received packages in destination	33894
Number of total missing packages	16709
Number of packages in 0-50	10311
Number of missing packages in 0-50	4140
Number of packages in 50-100	12457
Number of missing packages in 50-100	4723
Number of packages in 100-150	10406
Number of missing packages in 100-150	2513
Number of packages in 150-200	4722
Number of missing packages in 150-200	1303

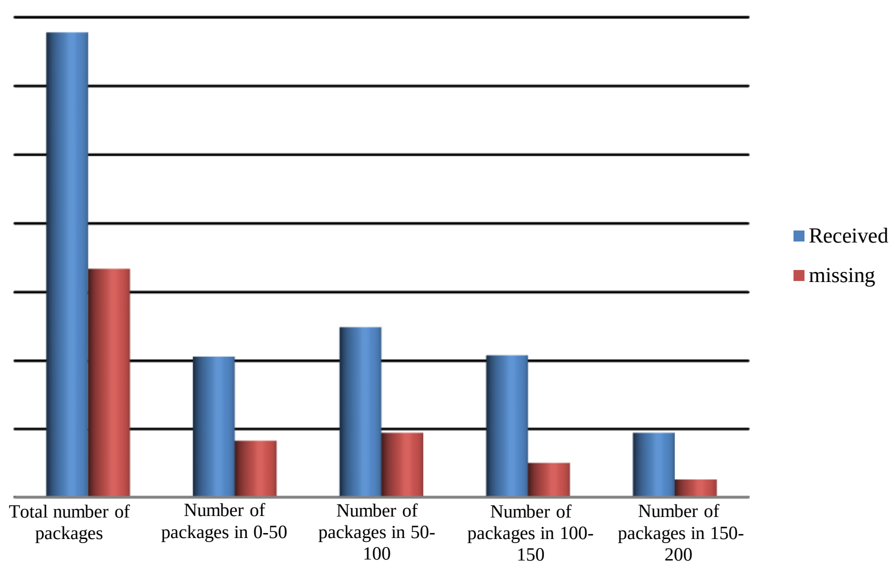


Figure 8. Comparison of the packages with AVQ algorithm

Conclusion

Recent developments in wireless sensor network technology led to its application in different areas. Unlike wired networks, congestion is greater in the wireless networks. Hence, one of the key challenges in wireless sensor networks is the congestion in the networks that results in a waste of energy in a sensed node with limited battery and the increased the package loss rate. Thus, the quality of the service is affected and network performance decreases. It is highly sensitive in sensitive applications such as medical care and thus the occurrence of congestion should be avoided as much as possible and controlled in the case of increased traffic and congestion. This study analyzed and implemented the congestion control protocols in wireless sensor networks. The main question of the study is the congestion control protocol and selection of the optimal protocol in a simulated environment. The main focus of the research is on the assumption that when multiple sensors simultaneously start sending packages to the well, then it will be a very high probability of congestion. For this purpose, several methods have been proposed for congestion control in this study, most of the methods were studied in detail. Here, some of the methods were analyzed in a simulation environment. NS2 simulation software was used, in which all programs related to algorithms were written in TCL language and were separately analyzed in a similar condition. 50 sensors with the operation range of 50 meters and 100 buffer energy and a well with operating range of 100 meters and 100 buffer energy were used for simulations environment. These were assumed to be constant for all congestion control protocols. The three criteria of the number of packages sent per unit of time, the number of incoming packages to the destination and the number of the lost packages were calculated for each of the protocols. The written output files were used to calculate the numbers and due to the high bulk of the output files, the commands for utility AWK were used. For this purpose, the numbers of above items were calculated according to the four sets of written code. Comparing above numbers, these numbers were compared in different conditions in order to select the most efficient protocol. Among the protocols RED, AVQ and Drop Tail, RED protocol has the best performance and Drop Tail protocol has the lowest performance.

References

1. Ms.D.F.Jenolin Flora, Dr.V.Kavitha, and Mrs.M.Muthuselvi, “A Survey on Congestion Control Techniques in Wireless Sensor Networks” IEEE PROCEEDINGS OF ICETECT 2011.

2. B.Chellaprabha, Dr.S.ChenthurPandian, and Dr. C. Vivekanandan, “ Enhanced Congestion Control in DCCP_TCP like” IEEE-20180 26-28 July 2012, Coimbatore, India.

3. C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," presented at the Proceedings of the 2nd international conference on Embedded networked sensor systems, Baltimore, MD, USA, 2004. 145

C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: overload traffic management using multi-radio virtual sinks in sensor networks," presented at the Proceedings of the 3rd international conference on Embedded networked sensor systems, San Diego, California, USA, 2005.

4. Shijun Zhao, Panpan Wang, Simulation Analysis of Congestion Control in WSN Based on AQM, International Conference on Mechatronic Science, Electric Engineering and Computer,2011.