I J C T A, 9(4), 2016, pp. 1979-1988

Adaptive Backoff Algorithm for Congestion Control in IoT

M. Swarna[1], S. Ravi[2] and M. Anand[3]

ABSTRACT

In this paper, Backoff algorithm for congestion control in IoT is proposed. The simple control mechanism of backoff algorithm makes it more suitable to be implemented in IoT based applications for congestion control. The exponential and Fibonacci backoff algorithm is applied to IoT nodes for congestion control. The IoT nodes change their backoff algorithm according to the data traffic type and load. Nodes transmit data intensively by FIB and convert to BEB as their loads reduce, whereas IoT node which has lower constant traffic load uses BEB backoff. The waiting time and contention window size increment for both Fibonacci and exponential backoff algorithm is analyzed. The performance of both backoff algorithms is compared. The retransmission time should be adaptive to increases the performance.

Keywords: IoT, Backoff Algorithm, Congestion control, Exponential Back off, Fibonacci Back off.

1. INTRODUCTION

Data collected by the Internet of Things (IoT) must be transmitted to servers for processing in order to provide various services. Due to the limited amount of resources in IoT, including network bandwidth, node processing abilities, and server capacities, congestion control in IoT plays is important for meeting service performance requirements. Traditional Internet protocols may not be directly applicable to the constrained environment that IoT devices are expected to deal with, especially given the resource limitations of these devices. To address this, taking into account communication patterns and resource availability, several new protocols have been proposed. The requirements to support IoT communications are substantially different from the design paradigm for current macrocellular networks optimized for human communications. Therefore, the requirements for IoT communications may be best supported by a new architecture and lightweight protocol structure rather than an evolution of the current cellular architecture and protocols. Binary exponential backoff or truncated binary exponential backoff refers to an algorithm used to space out repeated retransmissions of the same block of data, often as part of network congestion avoidance. Examples are the retransmission of frames in carrier sense multiple access with collision avoidance (CSMA/CA) and carrier sense multiple access with collision detection (CSMA/CD) networks, where this algorithm is part of the channel access method used to send data on these networks. In Ethernet networks, the algorithm is commonly used to schedule retransmissions after collisions.

2. RELATED WORKS

The paper describes the design and development of an end-to-end IP based architecture integrating a Constrained Application Protocol (CoAP) over 6LowPAN Contiki based WSN with an HTTP over IP based application. Representational State Transfer (REST) architectures allow IoT and Machine-to-Machine (M2M) applications to be developed on top of web services which can be shared and reused. The paper

[1] Research Scholar, ECE Department, Dr. M.G.R. Educational and Research Institute, Chennai, Email: swarnavinil@gmail.com

[2] Professor & Head, ECE Department, Dr. M.G.R. Educational and Research Institute, Chennai

[3] Professor, ECE Department, Dr. M.G.R.Educational and Research Institute, Chennai

illustrated how the introduction of UDP and the packet overhead compression drastically reduce the mote's power consumption and consequently increase the battery lifetime (Walter Colitti et al., 2011).

In this paper, the key technologies involved in the implementation of Internet of Things and the major application domain where the Internet of Things will play a vital role are described. Radio Frequency Identification, Near field communication, Machine to Machine communication and Vehicle to Vehicle communication technologies that can be used to implement the concept of Internet of Things is discussed in this paper. Addressing and networking issues of IoT is discussed in this paper. In this paper, the technologies used to make Internet of Things a reality is presented (Shashank Agrawal et al., 2013).

This paper proposes a simplified air interface protocol for IoT and a simultaneous access channel for uplink (UL) IoT communication. To support the IoT system concept, it is proposed that a separate lightweight air interface protocol for IoT. Performance results for the proposed simultaneous access channel used for the upper link IoT communication are provided (Chandra S. Bontu et al., 2014)

An IoT messaging protocol that is gaining in popularity and importance is the Constrained Application Protocol (CoAP). In this paper, CoAP was originally specified with a primitive congestion control mechanism, to address its various limitations a new congestion control protocol named CoAP Congestion Control Advanced (CoCoA) is being developed (Rahul Bhalerao et al., 2014).

In this paper, congestion control protocols for wireless sensor networks are analysed in terms of their suitability to detect congestion and notify the concerned nodes so that an appropriate control will be taken. In this paper, depending on the application types, how different mechanisms are used to handle the congestion is discussed (Mohamed Amine Kafi et al., 2014).

This paper presents first evaluation results for a mechanism that improves the communication between cloud services and resource-constrained IoT devices. If the use of advanced congestion control mechanisms improves the quality of service in terms of higher throughput and faster processing of requests by the network is evaluated. In this paper the amount of requests that can be processed in parallel increases and the time it takes for clients to complete their tasks decreases is achieved. The advanced congestion control mechanisms achieve this by calculating optimized retransmission time out timers and adjusting the backoff behavior dynamically (August Betzler et al., 2014). In this paper, a node and path traffic prediction model to predict and minimize the congestion is proposed.

In this paper prediction of packet generation due to network congestion from both periodic and event data generation is discussed. Simulation using NS-2 and Matlab is used to demonstrate the effectiveness of the proposed algorithm. In this paper, Congestion zones should be determined according to the analyzed effectiveness. In this paper, the change in network utility analyzed depending on the design of the congestion zone (Ga-Won Lee et al., 2014).

In this paper, a flow control optimization problem for wireless sensor networks with lifetime constraint and link interference in an asynchronous setting is proposed. This paper makes use of the interference set to model the spatial contention between links and formulate the problem as a nonlinear constrained optimization problem. The proposed algorithm can achieve the maximum utility. Extensive simulations are conducted to demonstrate the efficiency of proposed algorithm and validate the analytical results (Jiming Chen et al., 2010)

This paper proposes a cryptographic solution against insider threats through a distributed capability-based access control in IoT. The capability-based approach offers benefits in terms of distributed management, support for delegation, traceability of the access, authentication chains to extend scalability and support of standard certificates based on Elliptic Curve Cryptography (ECC). This has offered the possibility to exploit the IoT potential in terms of end-to-end connectivity based on IPv6, access to the resources through Constrained Application Protocol methods and finally to provide an access control solution on top for CoAP Resources (Jośe L. Herńandez-Ramos et al., 2013).

This paper presents the Identity Authentication and Capability based Access Control (IACAC) model with protocol evaluation and performance analysis. To protect IoT from man-in-the-middle and denial of service (Dos) attacks, the concept of capability for access control is introduced. Finally, the proposed protocol is evaluated by using security protocol verification tool and verification results shows that IACAC is secure against aforementioned attacks. This paper addresses challenges in IoT and security attacks to give an actual view of IoT networks (Parikshit N. Mahalle et al., 2013).

2.1. Overview of IoT protocols

An IoT device may consist of several interfaces for connections to other devices, both wired and wireless. These include (i) I/O interfaces for sensors (ii) interfaces for internet connectivity (iii) memory and storage interfaces and (iv) audio/video interfaces etc.

- IoT protocols
- Link layer

The data is physically sent over the network's physical layer or medium is determined by link layer protocols. Nodes on the same link exchange data packets over the link layer using link layer protocols. Link layer determines in what way the packets are coded and signaled by the hardware device over the medium which the host is attached. Some link layer protocols which are relevant in the context of IoT are 802.3 Ethernet, 802.11 WiFi, 802.16 WiMax, 802.15.4 LR-WPAN and 2G/3G/4G mobile communications.

2.2. Network/internet layer

IP datagrams are sends from the source network to the destination network by network layers. This layer performs the host addressing and packet routing. Host identification is done using hierarchical IP addressing schemes such as IPv4 or IPv6.

2.3. Transport layer

The transport layer protocols provide end-to-end message transfer capability independent of the underlying network. The transport layer responsible functions such as error control, segmentation, flow control and congestion control. The congestion control capability of TCP helps in avoiding network congestion and congestion collapse which lead to degradation of network performance.

3. EXPONENTIAL BACK-OFF ALGORITHM

If collision occurs between two nodes, nodes wait for some time to retransmit. After nth collision, random value (*k*) taken from {0,……2*n*-1}, node waiting time calculated by k*Tslot times for retransmission. In exponential back-off algorithm, minimum packet size is considered as slot. Consider two IoT nodes have number of packets to transmit. In slot1, both IoT nodes try to transmit data simultaneously collision will occur. In slot2, both IoT nodes retransmit data with probability of ½ (node1 transmit data and node2 does not retransmit). In slot3, both IoT nodes try to transmit data again collision will occur. In slot4, node1 transmit data with probability of ½ and node2 transmit data with probability of ¼ (node1 transmit data). In slot5, node1 definitely transmit data, node2 transmit data with probability of 1/4. If collision occurs, in slot 6, node1 transmit data with probability of ½ and node2 transmit data with probability of 1/8. If node1 transmit data successfully, the probability of node2 sending data successfully half with each collision. In exponential back algorithm, node2 might not transmit data until node1 get idle and waiting time for node2 to transmit data is high. As the number of retransmission attempts increases, the delay increases exponentially.

3.1. Drawbacks of Exponential back-off Algorithm

In exponential backoff algorithm, after each collision, contention window sizes increases quickly. If Contention widow sizes increases, waiting time of nodes to access channel also increases. Because of high widow sizes, channel bandwidth also lost. Because of large timeout value, long wating time before retransmission issued to Nodes.

3.2. Fibonacci Back-off Algorithm

Fibonacci series defined as $fib\ n = fib\ n - 1 + fib\ n - 2$, $fib\ 0 = 0$, fib 1 = 1, $n \geq 0$. In Fibonacci back-off algorithm, waiting time increases slowly. In Fibonacci backoff algorithm, the increment factor is decreased if more collision occurs between IoT nodes. Table 1 show, after certain time ratio of successive Fibonacci terms limited into a certain range. The FIB range is desired by the network according to the traffic load. The waiting time for IOT node is in FIB range. If traffic is high (low) in network, the low (high) FIB range is selected. Consider four different IoT nodes have packets for transmission and FIB range is 5. In slot1 four nodes transmit data simultaneously. Collision will occur, then waiting time for nodes (1 to 4) assigned respectively as 1, 2, 3 and 5. In slot2, node1 transmit data successfully, other nodes (2 to 4) waiting time respectively as 1, 2 and 4. In slot3, node 2 transmits data successfully. After that node 1 and node 2 waiting time is reset by network as 1 and 2. In slot4, node1 and node3 transmit data simultaneously collision will occur. Node1 and node3 waiting time reset as 3 and 4. Now, waiting time for nodes (1 to 4) reset respectively as 3, 1, 4 and 2. If collision occurs in the network, the FIB value for both nodes is reset by the network.

3.3. Drawbacks of Fibonacci back-off Algorithm

Because of small timeout value, the unnecessary retransmission will occur in nodes.

3.3.1. *Congestion in IoT*

In internet, the Transmission Control protocol is used for reliable communications at transport layer. For short communication of IoT, TCP is not enough. Connection setup for TCP required more time. Another drawback in TCP is congestion control, TCP is responsible for end-to-end congestion control, but the amount of data transfer is very small in IoT, so TCP congestion control is impractical. In IoT, Congestion is occurred due to simultaneous messages from several nodes. Congestion is mainly seen in machine to machine and vehicle to vehicle communication.

Table 1

Iteration	*Ratio of Successive Fibonacci terms*
1	1
2	1
3	2
4	1.50
5	1.75
6	1.65
7	1.65
8	1.65
9	1.65
10	1.65
11	1.65
12	1.65
13	1.65
14	1.65
15	1.65

Congestion occurs in IoT in two principle situations:

- Large number of IoT Devices transmit data at the same time
- IoT nodes roaming onto a other network, then network node is increased.

3.3.2. *Congestion control in IoT*

In network, if mobility management procedures or session management procedures occurs, then network will reject the IoT Device's request with a back-off timer to the device, so that the IoT Device does not re-attempt the request for the specific period of time indicated in the backoff timer.

Different types of control for the existing back-off timer

1. APN based congestion control: The network discards the Session Management requests from devices to a certain APN to control the amount of traffic.
2. Mobility management congestion control: The network rejects Mobility Management requests from IoT Devices.
3. Network set Low Access Priority Indicator (LAPI) in "low priority" IoT Devices, where the application(s) can stand longer access delays. The LAPI used by the network to reject such an IoT Device from access, and assign a back-off timer preventing the device from immediately repeating the access attempt. LAPI provides an Extended Wait Timer which provides the ability for the mobile network to reject a request with a longer back-off timer

The IoT Device support both APN based congestion control and mobility management congestion control and LAPI.

Proposed system hardware Setup

1) Connect IoT Node - 1 Tx from Rs232 to Rx1 of Arduino & Rx from Rs232 to Tx1 of Arduino
2) Connect IoT Node - 2 Tx from Rs232 to Rx2 of Arduino & Rx from Rs232 to Tx2 of Arduino
3) Provide Power suppy (5V) and Gnd to the MAX232 boards from Arduino board.

System connection Diagram

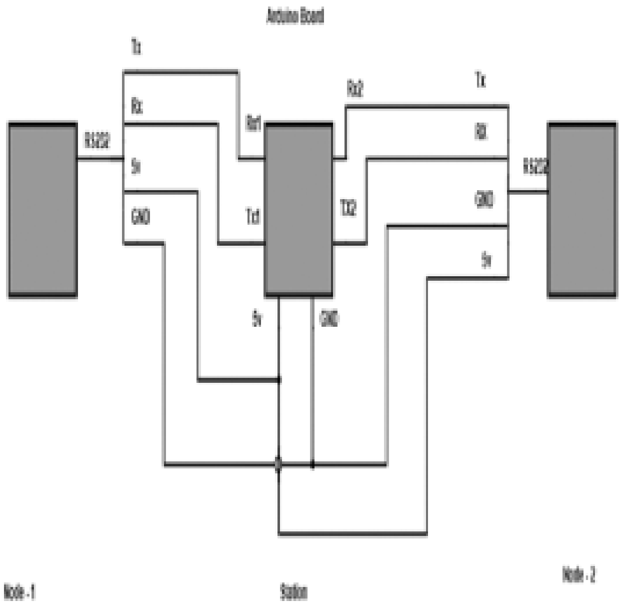


Figure 1:

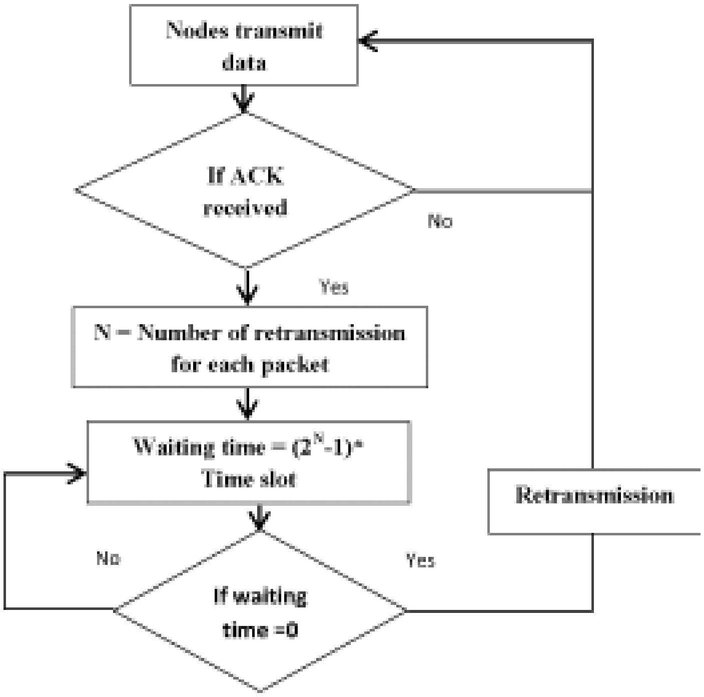

Nodes transmit data
If ACK received
No
Yes
N = Number of retransmission for each packet
Waiting time = (2^N-1)* Time slot
Retransmission
No
Yes
If waiting time =0

Figure 2:

The code flow Send Data by Exponential backoff algorithm is shown in figure 2. Time slot is dependent upon the network traffic and size.

Proposed system station flow

Serial ports are initialized for communication of IoT node1 and IoT node 2. Slot signal send to the IoT node1 and IoT node2. Slot time is wait time for data from IoT node1 and IoT node2. If both IoT nodes send data during this time, send nack to both the Nodes to signal collision and discard the data.If only one of the nodes send data during the Slot Time, then send ack to that Node and display the received data.

The code flow SendData by Fibonacci backoff algorithm is shown in figure 3.

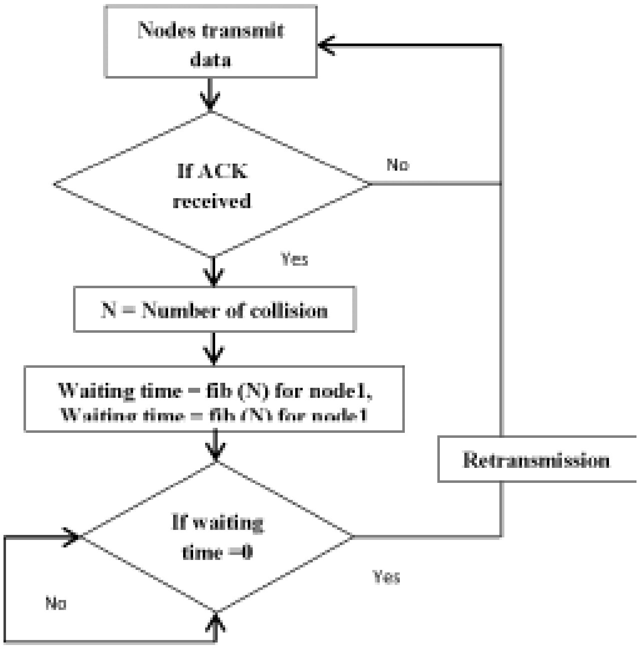


Figure 3:

Implemented hardware

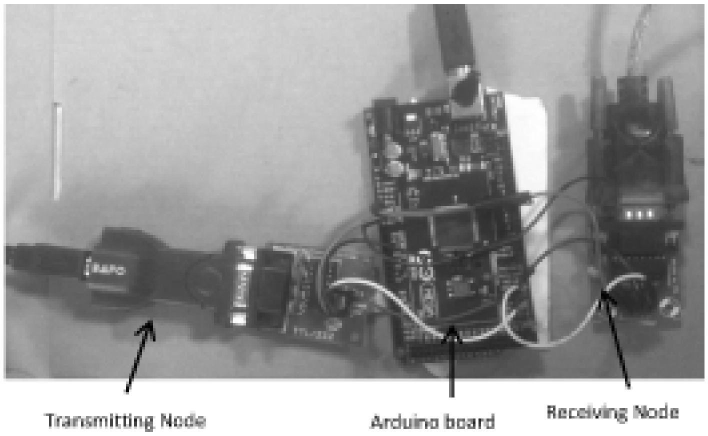


Figure 4:

4. SIMULATION SET UP AND RESULTS

The simulation part has three main menus such as open the serial port, Select Exponential or Fibonacci backoff algorithm, and SendData. The first option is used to open the serial port. The second option is used to select the Exponential (OR) Fibonacci algorithm. The Third option is used to start transmission to the station. In simulation the USB address (for both IoT nodes) using dmesg in the terminal window is obtained. The obtained USB address updated in Node.py program for both IoT nodes. The connection made as per the circuit diagram of proposed system. In simulation firstly node2.py program executed on receiver IoT node. Secondly node1.py program executed on transmitter IoT node. The corresponding algorithm (Exponential (or) Fibonacci) is selected in both IoT nodes. The transmission is started on Hyperterminal.

Throughput Vs mobility speed with traffic rate 10 packets/s

Table 2 show throughput of a network. Binary Exponential Backoff denoted as (BEB) and Fibonacci Backoff denoted as (FIB). Throughput of network improved in FIB compared with BEB, even number of nodes also increased.

Table 2

Speed (m/s)	Network with 20 nodes		Network with 30 nodes	
	Throughput (kBps) of BEB in percentage	Throughput (kBps) of FIB in percentage	Throughput (kBps) of BEB in percentage	Throughput (kBps) of BEB in percentage
2	62	76.1	57.14	71.4
4	64.5	78.5	57.14	71.4
6	66.6	80.95	57.14	71.4
8	69	83.3	59.5	73.8
10	69	85.7	59.5	71.4
12	71.4	90.4	59.5	71.4
14	71.4	95.2	59.5	71.4
16	71.4	95.2	59.5	76.1
18	73.8	95.2	59.5	76.1
20	78.5	100	59.5	76.1

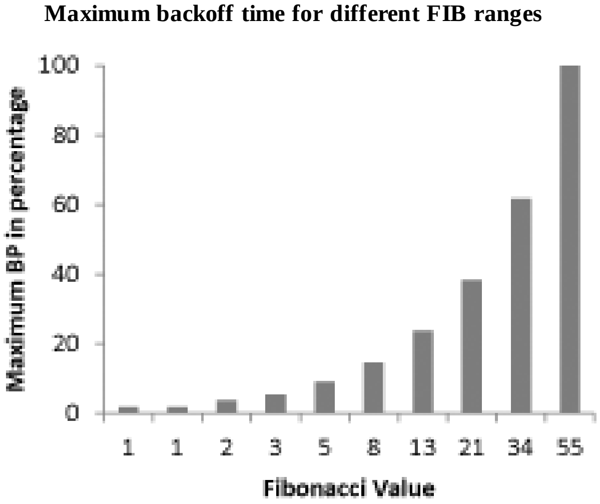


Figure 5

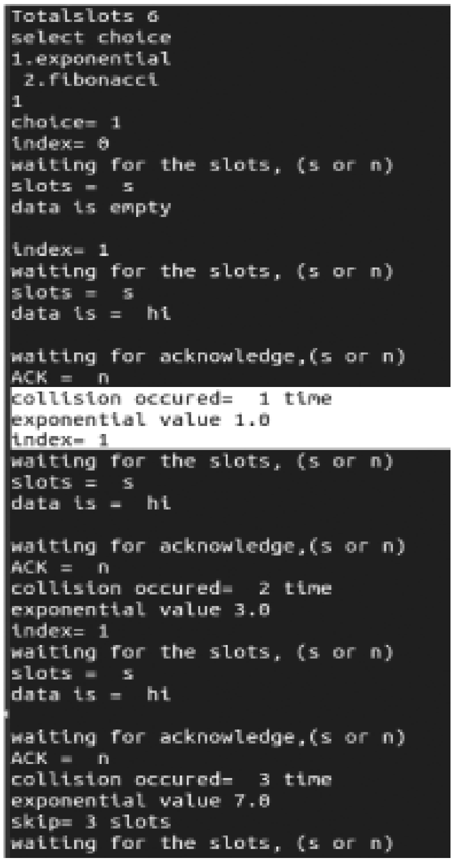
5. OUTPUT SCREEN SHOTS

Figure 6(a) shows the exponential backoff algorithm. Figure 6(b) shows the Fibonacci backoff algorithm. In figure 6(a) after the 3rd collision waiting time increased as 7. In figure 6(b) after the 3rd collision waiting time increased as 2.

Figure 7 shows how the backoff time varied with number of collision increases.

Figure 8 shows the packet delay for various loads. The traffic load is varied with increasing time period. At *t* = 0, traffic load value is 0.51 and packet delay for all algorithm is same (20). After some time traffic

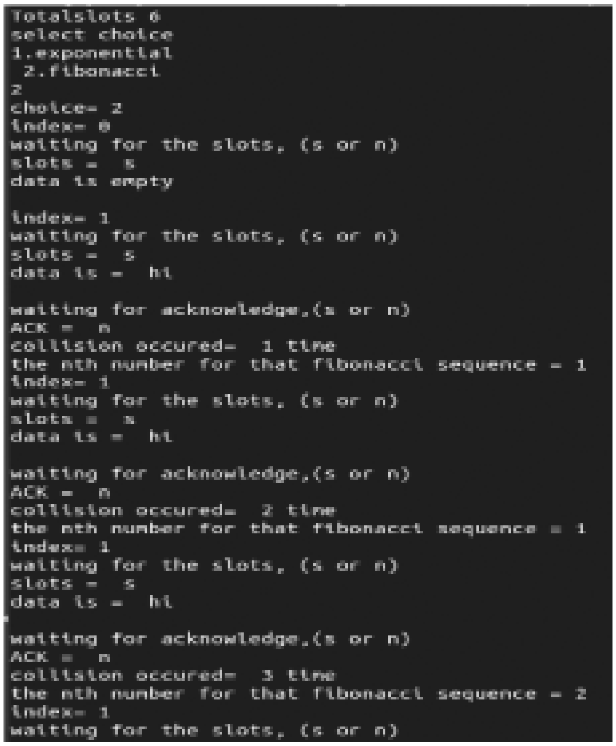
Node - 1 output:



```
Totalslots 6
select choice
1.exponential
 2.fibonacci
1
choice= 1
index= 0
waiting for the slots, (s or n)
slots =  s
data is empty

index= 1
waiting for the slots, (s or n)
slots =  s
data is =  hi

waiting for acknowledge,(s or n)
ACK =  n
collision occured=  1 time
exponential value 1.0
index= 1
waiting for the slots, (s or n)
slots =  s
data is =  hi

waiting for acknowledge,(s or n)
ACK =  n
collision occured=  2 time
exponential value 3.0
index= 1
waiting for the slots, (s or n)
slots =  s
data is =  hi

waiting for acknowledge,(s or n)
ACK =  n
collision occured=  3 time
exponential value 7.0
skip= 3 slots
waiting for the slots, (s or n)
```

Figure 6(a):

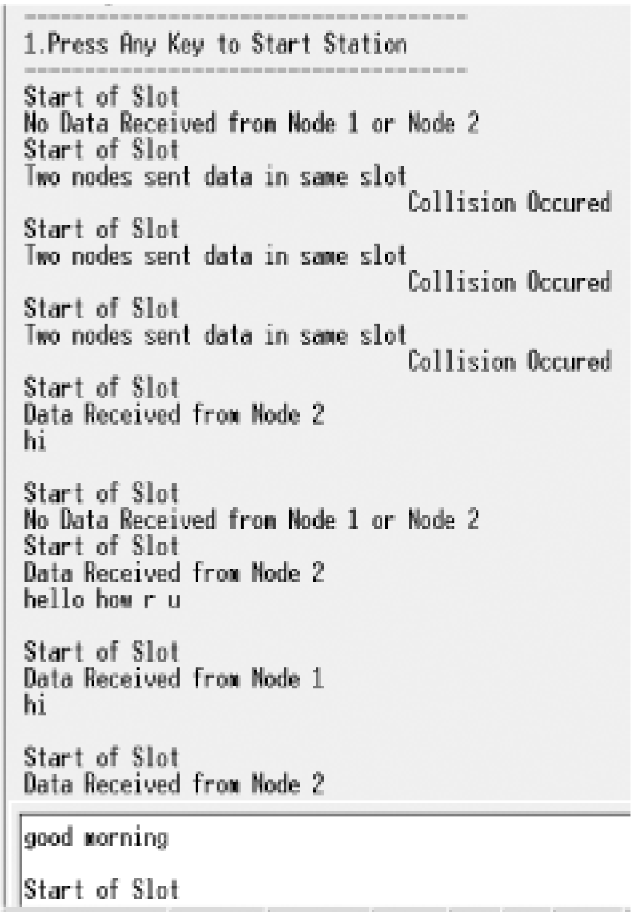
Node – 2 output:



```
Totalslots 6
select choice
1.exponential
 2.fibonacci
2
choice= 2
index= 0
waiting for the slots, (s or n)
slots =  s
data is empty

index= 1
waiting for the slots, (s or n)
slots =  s
data is =  hi

waiting for acknowledge,(s or n)
ACK =  n
collision occured=  1 time
the nth number for that fibonacci sequence = 1
index= 1
waiting for the slots, (s or n)
slots =  s
data is =  hi

waiting for acknowledge,(s or n)
ACK =  n
collision occured=  2 time
the nth number for that fibonacci sequence = 1
index= 1
waiting for the slots, (s or n)
slots =  s
data is =  hi

waiting for acknowledge,(s or n)
ACK =  n
collision occured=  3 time
the nth number for that fibonacci sequence = 2
index= 1
waiting for the slots, (s or n)
```

Figure 6(b):

Station output:



```
-----------------------------------------
1.Press Any Key to Start Station
-----------------------------------------
Start of Slot
No Data Received from Node 1 or Node 2
Start of Slot
Two nodes sent data in same slot
                                   Collision Occured
Start of Slot
Two nodes sent data in same slot
                                   Collision Occured
Start of Slot
Two nodes sent data in same slot
                                   Collision Occured
Start of Slot
Data Received from Node 2
hi

Start of Slot
No Data Received from Node 1 or Node 2
Start of Slot
Data Received from Node 2
hello how r u

Start of Slot
Data Received from Node 1
hi

Start of Slot
Data Received from Node 2

good morning

Start of Slot
```

Figure 6 (c):

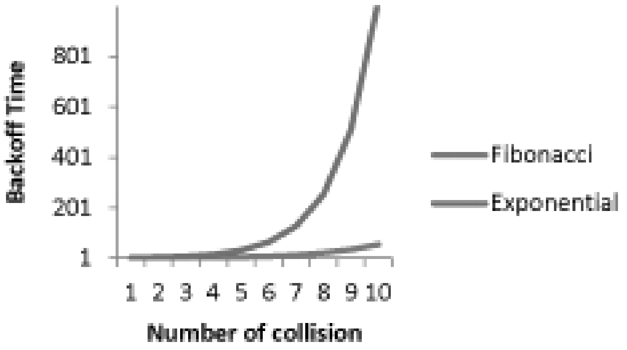


Figure 7:

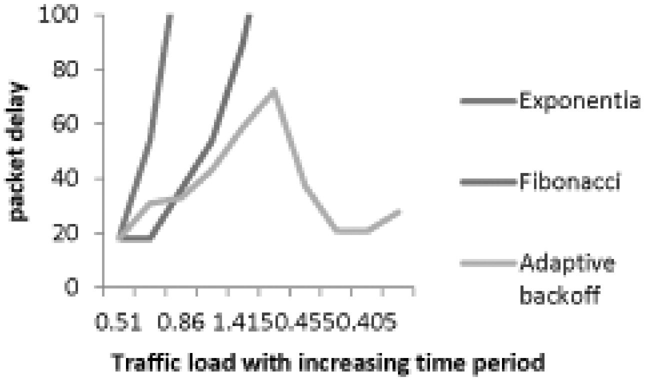


Figure 8:

load increased and decreased. Exponential backoff algorithm rapidly increases the packet delay even for low traffic load. Fibonacci backoff algorithm slowly increases the packet delay even for low traffic load. Adaptive backoff algorithm changed the packet delay according to the traffic load.

6. CONCLUSION

The design and development of IoT nodes with congestion control mechanism is described in this paper. Analysis and simulation results show that the proposed control achieves comparable delay performance and better throughput performance. Adaptive backoff mechanism is implemented in between IoT nodes. Simulated results show how backoff algorithm adapts with various loads. The hardware implementation of adaptive backoff algorithm is achieved. Various Backoff timers for IoT congestion control are reviewed. In proposed method, packet delay is varied according to the load of IoT nodes. The throughput also increased with proposed algorithm.

REFERENCES

[1] Walter Colitti, Kris Steenhaut and Niccolò De Caro "Integrating Wireless Sensor Networks with the Web", http:// couchdb.apache.org/. 2011, hinrg.cs.jhu.edu

[2] Shashank Agrawal and Dario Vieira "A survey on Internet of Things", Abakós, Belo Horizonte, vol. 1, no 2, pp. 78–95, maio 2013–ISSN:2316–9451

[3] Chandra S. Bontu, Shalini Periyalwar, and Mark Pecen "Wireless Wide-Area Networks for Internet of Things", IEEEvehicular technology magazine | MARCH 2014.

[4] Rahul Bhalerao, Sridhar Srinivasa Subramanian and Joseph Pasquale "An Analysis and Improvement of Congestion Control in the CoAP Internet-of-Things Protocol",2014, hinrg.cs.jhu.edu.

[5] Mohamed Amine Kafi, Djamel Djenouri, Jalel Ben-Othman, and Nadjib Badache "Congestion Control Protocols in Wireless Sensor Networks: A Survey", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 16, NO. 3, THIRD QUARTER 2014.

[6] August Betzler, Carles Gomez, Ilker Demirkol and Matthias Kovatsch "Congestion Control for CoAP Cloud Services", 2014 - ieeexplore.ieee.org.

[7] Ga-Won Lee, Sung-Young Lee and Eui-Nam Huh "Congestion Prediction Modeling for Quality of Service Improvement in Wireless Sensor Networks", Sensors 2014, 14, 7857-7880; doi:10.3390/s140507857.

[8] Jiming Chen, Weiqiang Xu, Shibo He, Youxian Sun, Preetha Thulasiraman and Xuemin (Sherman) Shen "Utility-Based Asynchronous Flow Control Algorithm for Wireless Sensor Networks", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 28, NO. 7, SEPTEMBER 2010.

[9] Jośe L. Herńandez-Ramos , Antonio J. Jara, Leandro Mar´ýn, and Antonio F. Skarmeta "Distributed Capabilitybased Access Control for the Internet of Things", Journal of Internet Services and Information Security (JISIS), volume: 3, number: ¾, 2013 - isyou.info.

[10] Parikshit N. Mahalle, Bayu Anggorojati, Neeli R. Prasad and Ramjee Prasad "Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things", Journal of Cyber Security and Mobility, Vol. 1, 309–348, 2013 - vbn.aau.dk

[11] Scalable Network Technologies, Inc., QualNet 4.0 product tour, 2006.

[12] Atheros Communications, AR5002 product bulletin, 2007.

[13] H. Zhai and Y. Fang, "Performance of Wireless LANs Based on IEEE 802.11 protocols." 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communication Proceedings, pp 2586-2590, 2003.

[14] Finch, S. R. "The Golden Mean." Mathematical Constants. Cambridge University Press, pp. 5-12, 2003.

[15] L. Bononi, et al., "A differentiated distributed coordination function MAC protocol for cluster-based wireless ad hoc networks", Proceedings of the 1st ACM international workshop on performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, pp. 77-86, 2004.

[16] F. Cali', et al.., "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement", Proc. INFOCOM'98, San Francisco, CA, March 29 - April 2, 1998, pp. 142-149.

[17] K.W. Chin and D. Lowe, "Simulation study of the IEEE 802.15.3 MAC," in Proceedings of the Australian Telecommunications and Network Applications Conference (ATNAC '04), Sydney, Australia, 2004.

[18] H. Chen, Z. Guo, R. Yao, and Y. Li, "Improved performance with adaptive Dly-ACK for IEEE 802.15.3 WPAN over UWB PHY," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E88-A, no. 9, pp. 2364–2372, 2005.

[19] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control, April 1999, RFC 2581.

[20] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Patridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. WrocLawski, and Lixia Zhang. Recommendations on Queue Management and Congestion Avoidance in the Internet, April 1998, RFC 2309.