

AN EFFICIENT MONITORING OF RECOVERING THE PACKETS IN WIRELESS SENSOR NETWORK

John J P. Mamatha C M. Prasanna Lakshmi G S

ABSTRACT:

Wireless Sensor Networks (WSN) are widely used in many application such as environmental monitoring, military , smart home , green computing , etc. WSN consists of numerous tiny nodes powered by very small batteries that are normally non – rechargeable [1].

Recovering of packets in wireless sensor networks have been a challenging issue. Many demonstrations have been implemented to recover the data packet at the receivers end. However due to many number of transmission of packets or retransmissions [2], few packets may be lost completely or partially. And these packets are not only difficult to retrieve but also critical to recover it.

This paper proposes an idea, to implement stop and wait ARQ at the receivers end. The technology of stop and wait ARQ is that, when a packet is sent to the nodes (receiver), the nodes in turn sends an ACK (acknowledgement) to the sender and then the sender on receiving the ACK frame sends another packet. We also deal in this paper to completely recover the data packets which has been lost instead of recovering a part of it.

KEYWORDS: WSN, green computing, nodes, packet, stop and wait ARQ, frame, ACK frame.

INTRODUCTION:

A wireless sensor network consists of many wireless nodes that have the ability to sense the state of other nodes around it [3][4]. In wireless sensor network, packets can be transmitted only if there is cooperation of the nodes to receive and transmit the data packets.

However, many nodes may come and go to and from the node respectively, there are chances of data packets to be lost or failure of nodes. If this occurs

we need to have an algorithm that prevents the loss of packet. It is easier to prevent loss of packet than recovering of data packets once it's lost.

- John J P is currently working as a Asst Professor in Computer Science & Engineering in Nagarjuna College of Engineering and Technology, Bangalore, PH-8970670335. E-mail: johnypaul10@gmail.com
- Mamatha C M is currently pursuing Ph.D program in Computer Science & Engineering in V.T. University, Belgaum, Karnataka PH-9164419899. E-mail: mamathacm@gmail.com.
- Prasanna Lakshmi G S is currently pursuing Ph.D program in Computer Science & Engineering in V.T. University, Belgaum, Karnataka PH-9632832800. E-mail: prasannalakshmi@gmail.com

In addition, we may face a crucial issue. Due to many packets transmitting to nodes[5], the nodes may breakdown partially[6] which may lead to lose of data packet[7] completely or partially[8], hardware lose and attenuation on signal[9].

To avoid the issues stated above, we propose an existing algorithm to be implemented on the receiver's end, to solve almost all the above stated problems.

RELATED WORK:

In wireless sensor networks, the data is transmitted from sensor nodes to base stations. These sensor nodes collect the information and transmit the information to other nodes or base station. In present era, many routing protocols in wireless sensor network have been proposed. But they do not guarantee the complete recovery of packets. This routing protocol can be divided into three types:

1. Data centric
2. Hierarchical based
3. Location based. [10]

Data Centric: It is basically related to software architecture where database plays an important role. Database-Centric or Data Centric combines a general purpose relational database management system

customized in memory or file based data structures and access methods.

Hierarchical based: TCP/IP routing is based on a two level hierarchical routing mechanism in which an IP address is divided into two portions - Network and Host.

Hierarchical based routing is a method of arranging the routers in a hierarchical model. A corporate intranet is a good example of hierarchical based routing.

Location based: These are the services for general class of computer programs used to include suitable controls for location and time data. Location based services has a number of usage in social networking and entertainment services. It is also accessible with mobile devices through the mobile network. LSB is also used to identify a location of a person or object. For example:- Searching a nearest shop, bank etc.

DIRECTED DIFFUSION: [11]

Directed diffusion is based on data centric routing protocol. Base station sends a packet to its neighboring nodes through broadcasting. When any node has information about this packet, the data will be transmitted to the base station.

GEAR (geographic and energy aware routing) [12]

GEAR is based on location – based routing protocol and is often stated as an improved version of directed diffusions.

PROPOSED ALGORITHM:

Our proposed algorithm is based on stop and waits ARQ implementing at the receivers end. The packets may be lost due to broken nodes in its transmission path. Based on our survey we consider three models gossiping [13], DSLPC (Data Salvation Low Power Consumption) [10] and Leo [9].Gossiping is a model that selects transmission path on the other hand Leo is a model that defines the transmission priority based on the packet types.

The packet with highest priority chooses the shortest path and the other packet chooses the path with higher energy. The figure 1 below determines the rate of packet loss due to node damage [10].

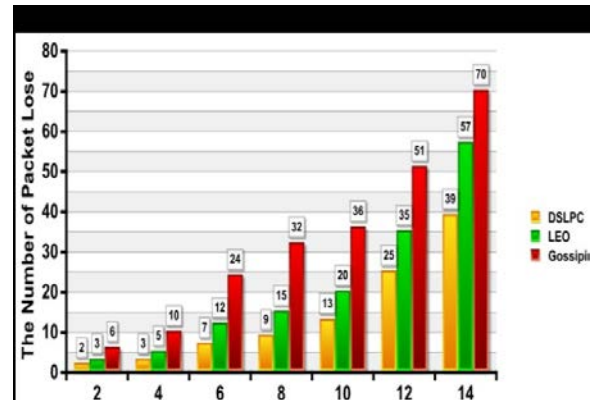


Figure 1: packet loss due to node damage

In our algorithm, we are concerned that packets are not lost. Even if it is lost, a duplicate of the lost packet is sent by the sender. The sender transmits a frame and then waits for a reply from the receiver end before sending another frame. The receiving end replies with an acknowledgement (ACK) if the frame is error free otherwise it may do two of the following things.

1. Many reply with a negative acknowledgement (NACK)
2. May wait for the sender to time out.

The figure 2 below illustrates a simple timing chart of a stop n wait ARQ flow control[14]

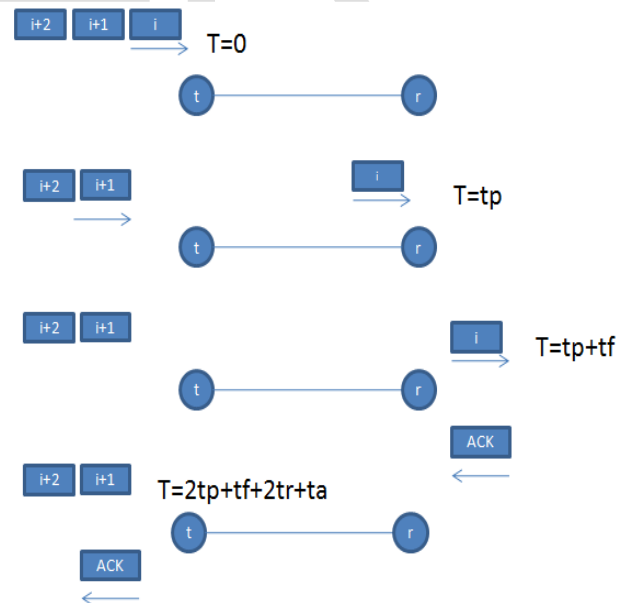


Figure 2: a simple timing chart of a stop n wait ARQ flow control

The stop and wait protocol is very simple and less expensive. In figure 2, we assume two frames I and i+1. Frame I is transmitted at T=0.

Assume t_f and t_p be the time required to enter all bits of the packet and the propagation time of the packet between the transmitter (t) and the receiver (r). It takes as long as $T=t_f+t_p$ to transmit a frame. When frame i arrives, the receiver processes the frame as long as t_r and generate an acknowledgement. The acknowledgement packet takes $T=t_a+t_p$ to be received by the transmitter where t_a =time required to enter all bits of acknowledgement packet, and it takes t_r for processing it at the receiver. Therefore total time for transmitting a frame completely is

$$T=t_f+2t_p+2t_r+t_a.$$

In the below figure 3, illustrates a schematic representation of stop and wait ARQ.

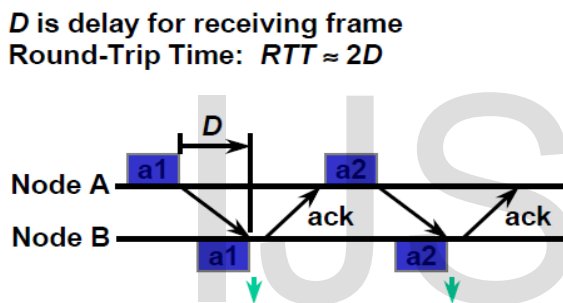


Figure 3: Schematic representation of stop and wait ARQ.

Since the receiver sends ACK and NACK, both the frames are also at risk. These transmitted frames may be lost or damaged. If it is lost, the sender site times out and retransmit the frame again.

If it is damaged, the sender site would be confused with the damaged acknowledgement frame that to which actual frame it belongs. this problem can be solved by using 1 bit sequence number .the sequence number can take on the values 0 and 1 and the sequence numbers are used alternatively to each frame [15] as shown below in figure 4.

Thus , when the sender retransmit frame 0, the receiver can determine that it is seeing a second copy of frame 0 rather than the first copy of frame 1 and therefore can ignore it .

The sender places a sequence number (SN) in the frame header. The receiver acknowledges with the next frame expected (NFE) value. The figure below illustrates it.

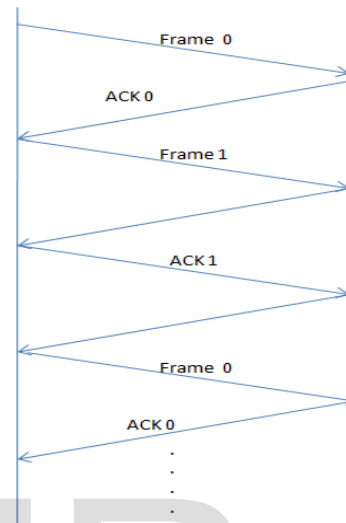


Figure 4: Transmission of frames using sequence numbers.

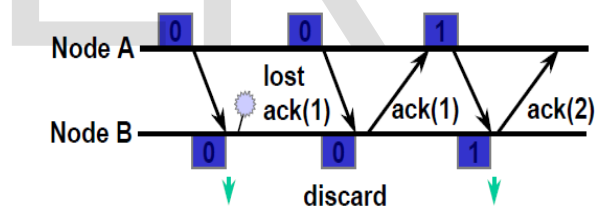


Figure 5: illustration of acknowledgement frame being lost.

In the above figure 5 we demonstrate how the acknowledge which have been lost when it is transmitted from node B to node A, due to time out again the node A will send request to node B and waits for its acknowledgement.

To retransmit the frame from node A to node B we need extra field for the SN and NFE. That SN and NFE require extra bits in the packet as shown below in figure 6.



Figure 6: Structure of a packet.

STOP AND WAIT POSSIBLE SCENARIOS:

In the figure 7 illustrates 4 possible cases in stop and wait ARQ. The figure depicts the protocol behavior. The sending side is the left line and the receiving side is the right line. Time flows from top to bottom.

Figure 7 (a) shows the situation in which the ACK is received before time expires, (b) and (C) show the situation in which the original frame and the ACK ,respectively are lost and (d) shows the situation timeout fires too soon. Lost frame is the frame that was corrupted while transmitting, that this corruption was detected by an error code as mentioned before (1 bit sequence number) on the receiver, and that frame was discarded [15].

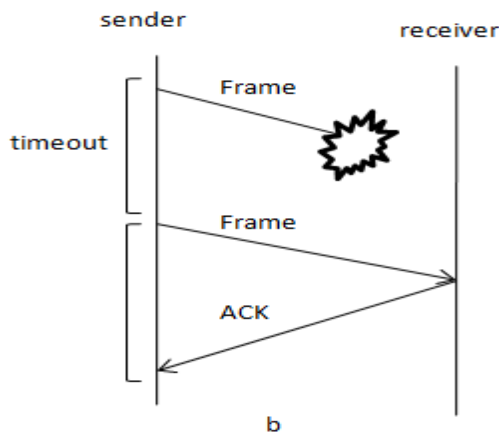
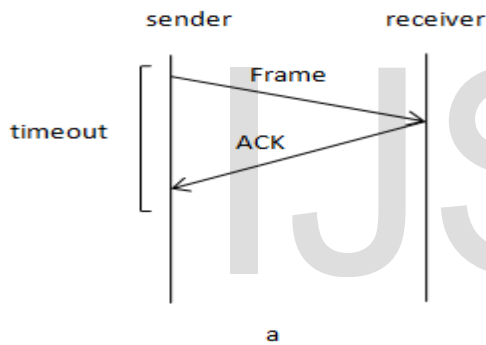
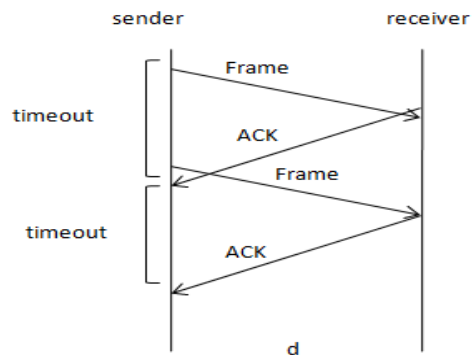
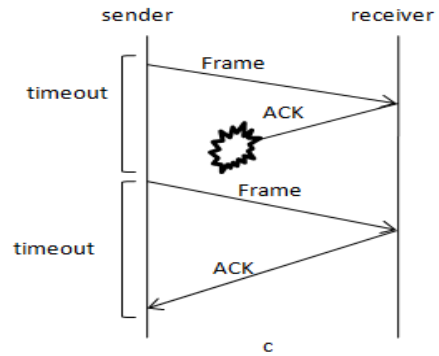


Figure 7: Different scenarios for the stop and wait ARQ.

DESIGN: STOP AND WAIT ARQ [16]:

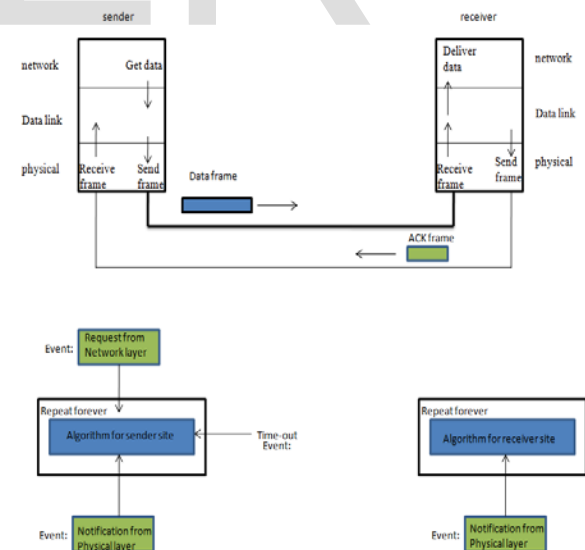


Figure 8: Stop and wait ARQ design.

Figure 8 shows the design of the Stop and Wait ARQ protocol. The sending device keeps a copy of the last frame transmitted until it receives an

acknowledgement for that frame. A data frame uses a seqNo (sequence number); an ACK frame uses an ackNo (acknowledgement number). The sender has control variable which we call Sn (sender, next frame to send), that holds the sequence number for the next frame to be sent (0 or 1).

The receiver has a control variable which we call Rn (receiver, next frame expected), that holds the number of next frame expected. When a frame is sent, the value of Sn is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. When a frame is received, the value of Rn is incremented (modulo-2), which means if it is 0, it becomes 1 and vice versa. Three events can happen at the sender site; one event can happen at the receiver site. Variable Sn points to the slot that matches the sequence number of the frame that has been sent, but not acknowledged; Rn points to the slot that matches the sequence number of the expected frame.

STOP AND WAIT PERFORMANCE ANALYSIS:

$$\text{Efficiency} = \frac{\text{TRANS}}{\text{TRANS} + \text{ACK} + 3\text{PROP}}$$

Putting in numbers for 10 mbps Ethernet

1. Packet size: 1518 bytes
2. ACK size: 64 bytes
3. Packet tx time : 1.2144ms
4. ACK tx time: 51.3us

$$\text{Efficiency} = 95.95\%$$

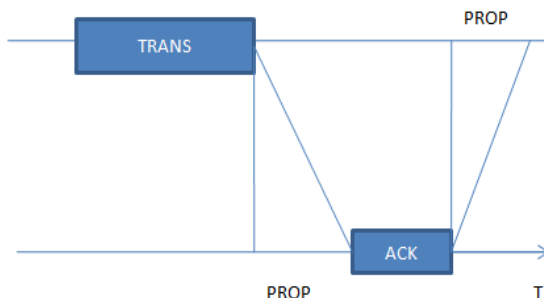


Figure 9: Performance analysis of stop and wait ARQ.

STOP AND WAIT ARQ ALGORITHM AT SENDER SITE:

Algorithm at node A (sender) to send to node B (receiver):

1. Sn=0; // frame 0 should be sent first
2. Cansend=true; //allow first request to go
3. While(true) //repeat forever
4. {
5. Waitforevent(); //sleep until an event occurs
6. If(event(request to send) AND cansend)
7. {
8. Getdata();
9. Makeframe(Sn); //the seqNo is Sn
10. Storeframe(Sn); //keep copy
11. Sendframe(Sn);
12. Starttimer();
13. Sn=Sn+1;
14. Cansend=false;
15. }
16. Waitforevent(); //sleep
17. If(event(arrivalnotification)) //ACK has received
18. {
19. Receiveframe(ackNo); //receive the ACK frame
20. If(not corrupted AND ackNo == Sn)
21. {
22. Stoptimer();
23. Purgeframe(Sn-1); //copy is not needed
24. Cansend=true;
25. }
26. }
27. If(event(timeout)) //timer expired
28. {
29. Starttimer();
30. Resendframe(Sn-1); //resend a copy check
31. }
32. }

ANALYSIS:

We first notice the presence of Sn, the sequence number of the next frame to be sent. This variable is initialized once (line 1), but it is incremented every time a frame is sent (line 13) in preparation for the next frame. However, since this is modulo-2 arithmetic, the sequence numbers are 0,1,0,1 and so

on. Note that the processes in the first event (sendframe, storeframe, and purgeframe) use a Sn defining the frame sent out. We need at least one buffer to hold this frame until we are sure that it is received safe and sound. Line 10 shows that before the frame is sent, it is stored. The copy is used to resending a corrupt or lost frame. We are still using the cansend variable to prevent the network layer from making a request before the previous frame is received safe and sound. If the frame is not corrupted and the ackNo of the ACK frame matches the sequence number of the next frame to send, we stop the timer and purge the copy of the data frame we saved. Otherwise, we just ignore this event and wait for the next frame to happen. After each frame is sent, a timer is started. When the timer expires (line 27), the frame is resent and the timer is restarted.

STOP AND WAIT ARQ ALGORITHM AT RECEIVER SITE:

Algorithm at node B (receiver) to receive from node A (sender):

```
1. Rn=0;           //frame 0 expected to arrive first
2. While(true)
3. {
4.   Waitforevent(); //sleep until an even occurs
5.   If(event(arrivalnotification)) //data frame arrives
6.   {
7.     Receiveframe();
8.     If(corrupted(frame));
9.     Sleep();
10.    If(seqNo == Rn) //valid data frame
11.    {
12.      Extractdata();
13.      Deliverdata();
14.      Rn=Rn+1;
15.    }
16.    Sendframe(Rn);
17.  }
18. }
```

ANALYSIS:

First all arrived data frames that are corrupted are ignored. If the seqNo of the frame is the one that is expected (Rn), the frame is accepted, the data are delivered to the network layer, and the value of Rn is incremented. However there is one subtle point here. Even if the sequence number of the data frame does not match the next frame expected, an ACK is sent to

the sender. This ACK, however, just reconfirms the previous ACK instead of confirming the frame received. This is done because the receiver assumes that the previous ACK might have been lost; the receiver is sending a duplicate frame. The resent ACK may solve the problem before the time out does it.

CONCLUSION:

In this paper, we propose stop and wait ARQ algorithm to be implemented in receiving nodes so as to ignore the packets that has been damaged or lost. And wait for the sender to send the packet. Subsequently, we show how the packets have been lost. We also show the 4 possible cases that arise in stop and wait ARQ. The efficiency of this algorithm was found to be 95.57%. With the help of possible algorithms mentioned above, stop and wait ARQ can be implemented at the receivers end in order to minimize packet loss to the best of our model extent. Thus we conclude that stop and wait ARQ algorithm is the simplest and effective protocol.

References:

- [1]. Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "A survey on wireless sensor networks", IEEE Communication Magazim (2007) 108-114.
- [2]. MAC Protocols for Cooperative Diversity in Wireless LANs and Wireless Sensor Networks. IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 16, NO. 1, FIRST QUARTER 2014.
- [3]. S. Misra, V. Tiwari, M. Obaidat, "LACAS: learning automata-based congestion avoidance scheme for healthcare wireless sensor networks," IEEE Jnl. Selected Areas in Comm, Vol. 27, pp. 466-479 (2009).
- [4]. O. Abraham, M. Fapojuwo, D. Siva, "A hybrid routing protocol for wireless sensor networks based on a two-level clustering hierarchy with enhanced energy efficiency," Proceedings of WCNC, pp. 2051-2056 (2008).
- [5] X. Feng, "Qos challenges and opportunities in wireless sensor/actuator networks," Sensors 2008, pp.1099-1110 (2008).
- [6] A.O. Fapojuwo, C.T.Alejandra, "Energy consumption and message delay analysis of QoS

enhanced base station controlled dynamic clustering protocol for wireless sensor networks,” IEEE Transactions

On Wireless Communications, vol.8, no.10, pp. 5366-5374 (2009).

[7] X. Huang, H. Zhai, Y. Fang, “Robust cooperative routing protocol in mobile wireless sensor networks,” IEEE Transactions on Wireless Communications, vol.7, no.12, pp. 5278-5285 (2008).

[8] D. Kim, J. Cho, “Active Caching: A transmission method to guarantee desired communication reliability in wireless sensor networks,” IEEE Communication Letters, vol.13, no.6, pp.378-380 (2009).

[9] S. Misra, P.D. Thomsinious, “A simple, least-time, and energy efficient routing protocol with one-level data aggregation for wireless sensor networks,” The Journal of Systems and Software 83, pp. 852-860 (2010).

[10]. Chu-Hsing Lin, Hung-Yan Lin, Wei-Bin Lee - Routing Protocols with Power Saving and Data Salvation for Wireless Sensor Networks, 2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications

[11]. C. Intanagonwiwat, D. Estrin, F. Silva, J. Heidemann, and R. Govindan, “Directed Diffusion for Wireless Sensor Networking,” IEEE/ACM Transactions on Networking, vol.11, no.1, pp. 2-16 (2003)

[12] P. Dias Thomsinious, Sudip Misra, “A simple, least-time, and energy-efficient routing protocol with one-level data aggregation for wireless sensor networks,” The Journal of Systems and Software 83, pp. 852–860 (2010).

[13]. L.J. Garcia Villalba, A.L. Sandoval Orozco, A. Trivino Cabrera, C.J. Barenco Abbas, “Routing Protocols in Wireless Sensor Networks,” Sensors, vol.9, Issue 11, pp.8399-8421 (2009)

[14]. Nadir F. Mir, Computer and communication networks – Pearson.

[15]. Computer Networks – A system approach edition 3 – Larry L. Peterson and Bruce S. Davie.

[16]. Data communication and Networking edition 4. Behrouz A Forouzan.