A new approach for measuring latency and bandwidth in wireless Ad hoc network

[1]*Thottadi Hemendra (M.Tech CSE)* Email: hemagiri.thottadi@gmail.com
Maharaja Vijayaram GajapathiRaj College

I. Abstract:

Normally latency in the previous works on only on mesh topology, but on wireless sensor networks but always experimentally calculation and rectification of latency in the WSN (wireless sensor networks) is a big challenge. Once the WSN generated statically or dynamically the router or central servers have to face big difficult to maintain the peers information. Once the path(s) established for transmission the source and destinations will be under one session for logging (further verification). The moment the data transmission is initiated our approach will check the session for any existing session which is matching with the current transmission .If it matches and the communication will not duplicated and ignored and also log will not be updated. If any session is not matching with current transmission session then this session finds the path for transmission and also calculates the average latency for that session's all communications.

The ip's(addresses or node recognition) will always be fluctuated and we have a challenge in maintaining the sessions, so peers will be identified by unique generated id and will be tracked in log.

(index terms: Latency, WSN , transmission ,session)

II. Introduction:

We choose wireless Ad hoc network for the average latency calculations for session's broad casting. ***Session***, normally whenever the transmission path(s) established for communications one session is established and the whole information will be saved in log for avoiding duplicate was not there in the previous work.

In this paper we had a new approach called **session builder**. This approach is handy for WSN infrastructure for finding out the sessions that removes the need of trace backing (one parameter in this case is avoiding retransmission). This in turn helps us by providing central monitoring repository. So that each time we go for a new selection node the necessity of finding out for available paths will be reduced. This algorithm is also useful for determining whether a particular node is a router based on the session and log. So that the data monitoring is assured by not transferring the packets of data to that particular node if it is a router.

In this work we also had another approach for router monitoring. The main aspect behind this algorithm is that it checks for the copies of sessions. If same data is being transmitted it nullifies the path and uses the session that has been formerly build. This approach is known as ***cloning*** or casting. If the data is not similar to the previous one then it will be automatically updated in the log table.

Practical example: let us take to set of aggregated flows. Source as p1 and destinations are {p2, p6}, source as p2 and destinations {p3, p4} and Central monitoring system will start session for first aggregated flow. Once the session is established **SESSION BUILDER** will start checking in the past log for any available session. And if that session is matching then **SESSION BUILDER** will reuse that session till aggregated flow and it will calculate the average latency and update in the log.

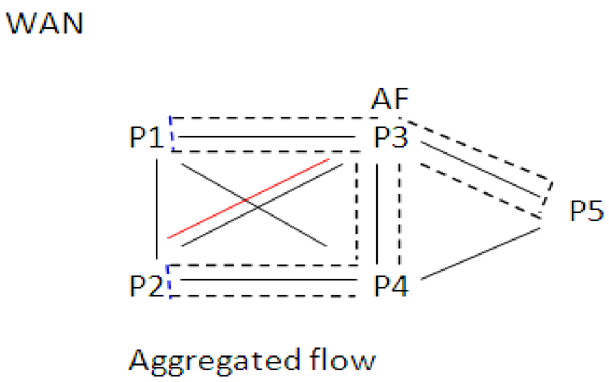
SESSION BUILDER will create the sessions/ selected transmission with central repository control and the sessions will be updated in session vector. All the sessions are unique generations from range 1 to 5!.

For our first set of communication the source as P1 and destinations as P2 and P5, now **SESSION BUILDER** will establish the paths. In this case the available paths for transmission are p1-p3 – p6, p2-p3, and p2-p4-p3. **SESSION BUILDER** establishes a session for communication. And checks for the log if any matching session for the above paths. If system finds any existing or matched log the session will be reused for synchronized broadcasting and delivers the packets from p1 to p6 by finding the path and update the time of communication, average latency for all the session's path. Finally calculate the average latency and marks for the threshold value.

If the session is identified with the following duplicated attributes:

- Source
- Destination
- Same packets
- But not with bandwidth (in this case bandwidth is simulated for the network)

The session will be identified as closable for ignoring the duplicated transmission



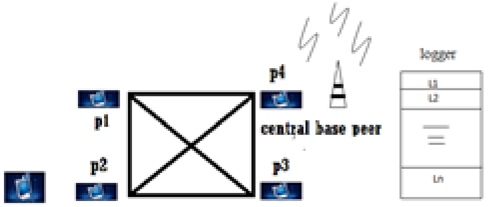
III. Literature survey:

LATENCY:

Latency is a measure of time delay (***which is in synchronization with bandwidth and packet size***) undergone by a system which is the exact definition of latency which relies on the system and the time being measured. In Communication, the lower bound of latency is decided by the medial being used for communication. In a dependable two-way communication, latency bounds the maximum rate that information can be transmitted, as there is a border on the amount of information that is "in-flight" at any one moment.

Different latencies available include:

i. Communication Latency
ii. Audio Latency
iii. Operational latency
iv. Mechanical Latency
v. Computer hardware and operating system latency



Algorithm (NACO):

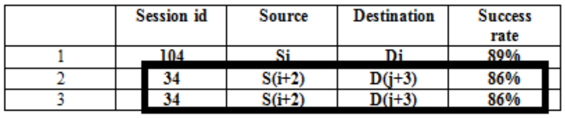
This network analization on controlling and optimization (NACO) is the technique to calculate aggregate latency based on growing sessions. Once the network is deployed for first time there will not be any log initially. The sessions will be logged and updated in the log table with the following sequence.

Router node	Source node	Destination node	Session id	Latency	File size	Bandwidth
Node 1	N5	N6	104	1406	376	0.0341394
Node 0	N2	N4	13	1594	262	0.0338770

Pic

The main usage on this log table is to enforce the dynamic router/session for putting central repository router which will monitor whole transmission of the casting nodes (destinations) from source node which is selected and controlled by central repository node. The above table shows the experimental result for latency calculation on generated bandwidth.

Dynamic generation of router/session: If we have the nodes {A, B, C, D, E} which are peers incrementally maintained under session builder (with log updation). It the sessions keeps growing for transmission the session tracker will keep tracking the success transmission for that particular transmission (***S****i* to ***d****j*) will be tracked for further session (id) comparison to assign dynamic router assignment. So once the session is updated in the log table with id that route will be evaluated based on source and destinations packet comparison. The nodes preauthorized according to success rate of transmission in decreasing model.



	Session id	Source	Destination	Success rate
1	104	Si	Di	89%
2	34	S(i+2)	D(j+3)	86%
3	34	S(i+2)	D(j+3)	86%

In the above case 2^{nd} and 3^{rd} cases are same session id's with same transmission and in this case if the transmission based on the same packets the session will be established but in fact this will be ignored for further transmission. The reason behind this is if all the sessions are unique (with out transmission paths) will be considered but with transmission paths are considered for ignorance of transmission.

The generated sequence resembles on the

```
//Initialization
Source= Vsrcε {v1,v2,v3….,vn}
Destination= Vdes ε {v1, v2, v3….,vn}
Ti= start time
Tf= final time
//Process
Start:
ti@Vsrc
tf@Vdes
latency = tf-ti
End;
```

The aggregated latency calculation: The above the table contains the latency for 2 sessions. But the all session's aggregated latency calculation is as follows:

If n is total number of sessions and if m is total duplication of sessions (ignored transmissions). All latencies exist in ***L***i so the latencies aggregation is depend on total number of sessions.

I = n-m and $\int_{0}^{i-1}$ (n-m)/n(math part for aggregation) // is the total aggregated latency.

IV. BANDWIDTH:

A Bandwidth is a chain of frequencies used for the transmission of telephone communication signal, radio, or computer network. It is the amount of data that can be sent in a decided amount of time.

In digital devices, the bandwidth is written in bits per second (bps) or bytes per second.

In analog devices, it is written in cycles /second, (or) Hertz

Bandwidth is mostly important for input/output devices.

For example, a fast disk drive can be slowed down by a bus with a low bandwidth.

The reason is that new buses like AGP have been developed for Computers.

Best Analogy is "Data is to available bandwidth as water is to the size of pipe"

V. WIRELESS AD HOC NETWORKS:

A wireless ad hoc network is a divided wireless network. The network is ad hoc as it will not depend on an already existing infrastructure. An ad hoc network inclines to feature a small group of devices all in very close in sense to each other.

Performance goes down as the number of devices comes up, and a large ad hoc network quickly becomes unmanageable.

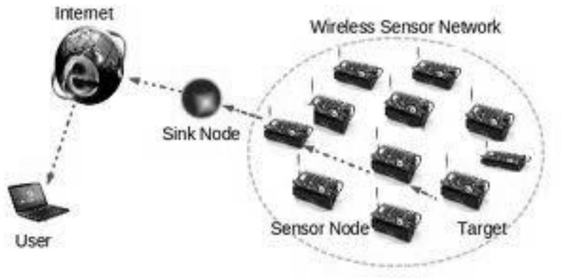
Ad hoc networks cannot get connected to wireless LANs or to the internet without installing a gateway.

Ad hoc networks will have a meaning when there is a need to build a small all-wireless LAN quickly and spend the minimum amount of money on equipment.

VI. Wireless sensor network:

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node/peer will be with several parts: a radio transceiver with an inbuilt antenna or connection to an attached externally built antenna, an electronic controller, and digital circuit for interfacing with the sensors and the resource of energy, usually a battery or an embedded form of energy consuming.

A sensor node/peer might differ in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created.



Network generation:

Input←no of peers (n)

Output←wireless sensor network (WSN)

```
count←0
λ ←n
Sn← Server
For each k in n
Loop start
Monitor(l)(10) // function which will update the node in the log table and server table
Sn← Peer ID
WSN ε Sn
End loop
```

VII. SESSION BUILDER

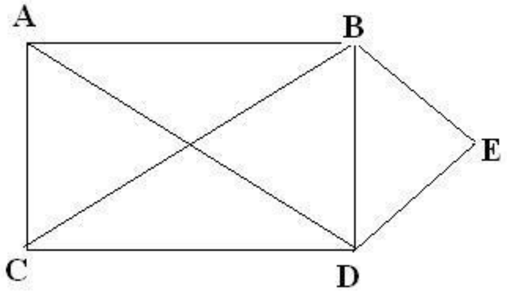
Algorithm:-

```
∑ S←0 // Total similar vector
Ψ ← 76 // Threshold for success rate in path for transmission
S n // source node
D n // destination node
Rn// router for monitoring
C ← cloning flag
∑Log ← log vector
```

```
        ∑ Ap ← Avalable path
        Rn ←router node

Available Path (Δ,d)

        Source node ← b
        Destination node ←d
        For each node   n in Ap
        Loop start
        Rn← Router(b,d)
        if (Rn)
        if (packer size >= Ψ)
        S ← Build session (b,d)
        End if
        End if
    End for

Router monitoring (Casting and duplication):-
            For each s in S
            Loop star
        If    SAME    (s,d)    &&
PACKETS(Δ)
            {
            C← CLONE(S)
            UPDATE (log)
            }
            Else
            S← Build session (Δ,d)
            End if
                    End for
```

Latency Algorithm

```
V1=latency
V2=bandwidth
  a                    b
∑Av   //All table contents with bandwidth and latency
Loop start
For each j in Av
    t1 = Av (V1)
   c
    t2 = Av (V2)               d
  abs (t1/t2) = Ψ
if  Ψ > ceil(t1/t2)
     flag=1   //  latency confirmed
end loop
```



Algorithm Description:

Firstly we consider a vector of source nodes. We also consider a destination node. Combined we pass them to the available path function. This function returns the nodes that are intermediate to the source and destination. In this function we also check if the nodes are routers or not. If it returns true then we make the packet size invisible to that particular router. If it returns false we build a new session using the intermediate node and destination. Another algorithm is for router monitoring. This will check the data that is being transmitted. If same data packets are found we simply use the old session that has been built previously. Else we update the info in log. This concept is known as cloning.

Example:

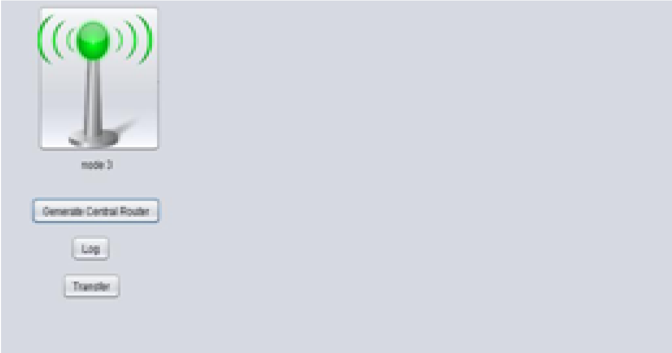
Let us assume a source nod from the above figure as node ‘A’ also destination ‘D’. Now passing these two nodes to the available path function that will return the paths available, let us suppose that to be ‘A-B-D’. In the function itself another task will be done i.e. checking router. If ‘B’ is determined as ***router*** then packets are nullified to that particular router.

Conclusion: In this work session will take important role to calculate the best latency in through put with **SESSION BUILDER** approach. The average flows are always depends on the various factors in sensor networks. In this we take mesh topology for sensor network. The reason behind the tree based sensor network is to have shortest path in short time. The shortest path discovery is in less time in wireless sensor network. In this approach we put some marker at the transmission side and log will be updated and once the shortest path found which will leads for broad casting. The broadcasting will be logged for the same path. For the same path depends on % of transmission average latency will be calculated with our **SESSION BUILDER** approach.

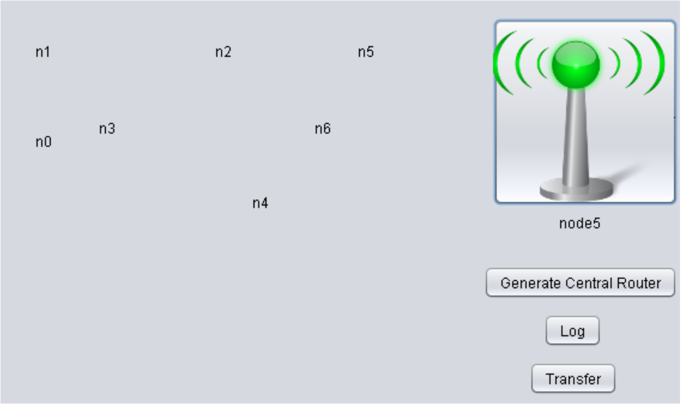
Here the router will be monitoring the log table continuously for the loss of aggregated flows and controls and ignores the regular over all latency paths. These paths will be totally vanished for better broad casting with low level latency. The main advantage for this approach is to adopt the current network to other Ad hoc and regular topologies.

Overall latency measurements are globalized to single one for various factors like throughput, bandwidth variation, and large data transmissions.

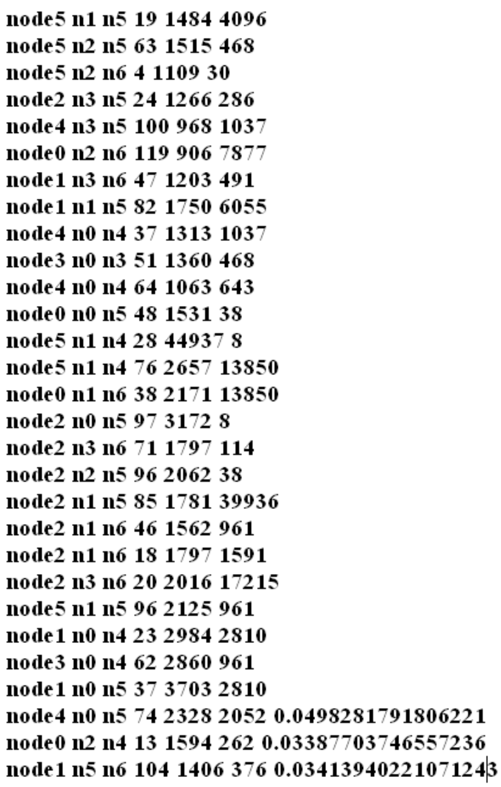
Results:



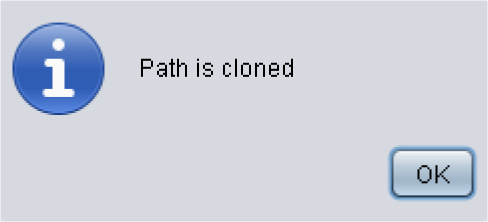
Picture 1



Picture 2



```
node5 n1 n5 19 1484 4096
node5 n2 n5 63 1515 468
node5 n2 n6 4 1109 30
node2 n3 n5 24 1266 286
node4 n3 n5 100 968 1037
node0 n2 n6 119 906 7877
node1 n3 n6 47 1203 491
node1 n1 n5 82 1750 6055
node4 n0 n4 37 1313 1037
node3 n0 n3 51 1360 468
node4 n0 n4 64 1063 643
node0 n0 n5 48 1531 38
node5 n1 n4 28 44937 8
node5 n1 n4 76 2657 13850
node0 n1 n6 38 2171 13850
node2 n0 n5 97 3172 8
node2 n3 n6 71 1797 114
node2 n2 n5 96 2062 38
node2 n1 n5 85 1781 39936
node2 n1 n6 46 1562 961
node2 n1 n6 18 1797 1591
node2 n3 n6 20 2016 17215
node5 n1 n5 96 2125 961
node1 n0 n4 23 2984 2810
node3 n0 n4 62 2860 961
node1 n0 n5 37 3703 2810
node4 n0 n5 74 2328 2052 0.0498281791806221
node0 n2 n4 13 1594 262 0.03387703746557236
node1 n5 n6 104 1406 376 0.03413940221071243
```



Path is cloned

Picture 3

Experimental table with results:

Router node/session	Source	destination	Data absolute path	Data size/kb	Sesion ID	Bandwidth	Latency
Node4	N0	N5	xxx.xx	1158	102	1547	0.10213316
Node4	N1	N6	xxx.xx	727	29	1438	0.10292072
Node5	N3	N6	xxx.xx	7876	86	1125	0.053333
Node5	N2	N6	xxx.xx	4096	51	1047	0.1548236

The above table shows consolidated experimental results for each session. The session id is not shown with casting duplications in this table. These results are experimentally proven for aggregated latency / session (math part for aggregation).

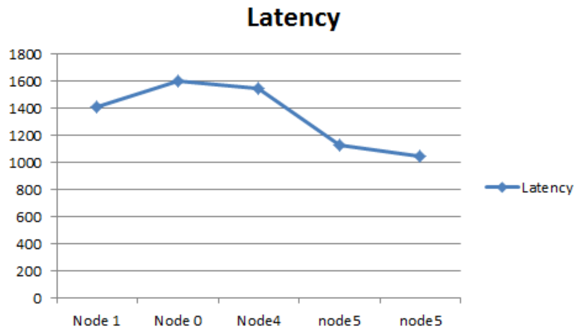
Ex: let's take the morphed set of the experimental set. R = 0.10292072

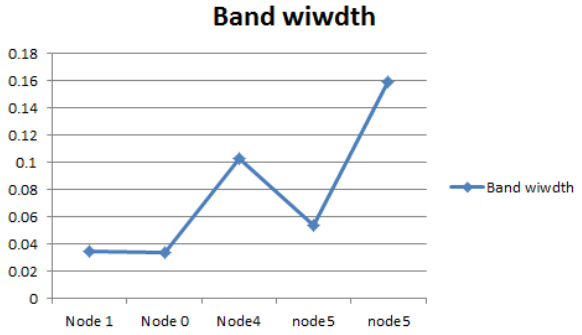
Lets the value (first factor) is the estimated value for simulation it t1 and t2 is bandwidth and latency / session is as follows.

Latency = t1/t2

And total aggregated latency will be as follows first need to remove the casting nodes sessions from consolidations(m) from total sessions n-m and aggregation is n-m/n which is aggregated latency and this can be extended for further network adaptation which is aggregated latency/network(f+1) with f as basic existing network.

Graph





Future work: This work can be extended to calculate aggregated latency for adoptive networks; the networks can be adopted with the current network who is in progress with aggregated latency calculation frame work. The adoptive feature for the current network is always in the perspective of cumulative latency calculation using gateway ip/adoptive network. In this case gateway IP is main source for the third party network's adoption.

8. *References:*

[1] Cisco, "Cisco extends highly secure, improved communications in rugged environments with new family of industrial Ethernet switches," 2007 [Online]. Available: http://newsroom.cisco.com/dlls/ 2007/prod_111407.html

[2].DuffieldandM.Grossglauser,"Trajectory samplingfordirecttraffic observation," in Proc. ACM SIGCOMM, Aug. 2000, pp. 271–282.

[3] N. Duffield, A. Gerber, and M. Grossglauser, "Trajectory engine: A backend for trajectory sampling," in Pro. IEEE Netw. Oper. Manage. Symp., Apr. 2002, pp. 437–450.

[3] N. Duffield, "Simple network performance tomography," in Proc. USENIX/ACM Internet Meas. Conf., Oct. 2003, pp. 210–215.

[4] Y.Lu,A.Montanari,B.Prabhakar,S.Dharmapurikar,andA.Kabbani, "Counter braids: A novel counter architecture for per-flow measure- ment," in Proc. ACM SIGMETRICS, Jun. 2008, pp. 121–132.

[5] S. Machiraju and D.Veitch, "A measurement-friendly network(MFN) architecture," in Proc. ACM SIGCOMM Workshop Internet Netw. Manage., Sep. 2006, pp. 53–58. [6] J.Mahdavi,V.Paxson,A.Adams,andM.Mathis, "Creatingascalable architecture for Internet measurement," in Proc. INET, Jul. 1998.

[7] R. Martin, "Wall street's quest to process data at the speed of light," InformationWeek 2007 [Online]. Available: http:// www.informationweek.com/news/infrastructure/showArticle. jhtml?articleID=199200297

[8] V. Misra, W.-B. Gong, and D. Towsley, "Stochastic differential equa- tionmodeling and analysis of TCP windowsizebehavior," in Proc. IFIP WG 7.3 Perform., Nov. 1999.

[9] M. Riska, D. Malik, and A. Kessler, "Trading flow architec- ture," Cisco Systems, Inc., San Jose, CA, Tech. Rep., 2008 [Online]. Available: http://www.cisco.com/en/US/docs/solutions/ Verticals/Trading_Floor_Architecture-E.pdf

[10] T. Szigeti and C. Hattingh, "Quality of service design overview," Cisco, San Jose, CA, Dec. 2004 [Online]. Available: http:// www.ciscopress.com/articles/article.asp?p= 357102&seqNum=2

Mr.Hemendra pursuing CSE in MVGR College. Done study on latencies on legacy networks to modern networks like and Wireless sensor networks. That leads to this work which is totally a new approach for latencies in network attacks.