SWiFiNet : A Task Distributed System Architecture for WSN

A.W. Rohankar
Dept. of IT
Sinhgad College of Engineering,
Pune University, India.

Mrinal K. Naskar
Dept. of ETCE,
Jadavpur University
Kolkata 700 032, India

Amitava Mukherjee
IBM India Pvt. Ltd.
Salt Lake,
Kolkata 700 091, India

Abstract—**WSN is a technology, straddling many application areas of this millennium. The research and practices in WSN have not mended their ways due to many unjust requirement specifications for WSN architecture. The paradox is WSN characteristics and traditional approaches for network operation are diagonally opposite. Power optimization, resource constraints and small form factor are the main characteristics of the WSN node. Taking into consideration these main characteristics, the 'SWiFiNet': a task distributed reusable architecture for WSN has been proposed in this paper. The focus of this architecture developed is to identify and reuse system components while preserving the sensor node characteristics. The complex network functionality is pushed onto overlay second tier devices, leaving sensor free for application development. This work demonstrates the implementation of 'SWiFinet' on hardware platform and network simulator using complete portability of reusable system components. Simulation and hardware results have been presented which illustrate that 'SWiFiNet' performs better and are application independent generic framework for WSN application development.**

Keywords—Wsn; Reusable; Reconfigurable; Network Architecture

I. INTRODUCTION

WSN technology is holding many promises for the future short-range wireless communication enabled applications. The applications will be two fold, firstly replacement of existing wired systems and secondly many newer systems, which could not be realized due to wired technology constraints. The resulting systems would be agile in development and deployment yet economically viable. The unjust requirement specifications coupled with WSN characteristics have failed to address issues of integrated reusable generic WSN application framework that can be used across the WSN application domains. Research and development in wireless communication are an ongoing process due to dynamic and unpredictable behavior of wireless radios but most of the research work is directed towards specific isolated ideas and areas like scattered, random topology, Ad-hoc, nondeterministic network [1, 2]. Present Wireless communication research and development still follows the communication research and development still follows the footprints of existing communication standard and protocols coupled with inflated specifications of WSN. It is influenced by the traditional network characteristic like mesh topology, self organizing network, large scalable network etc. Integrated network architecture for the application domains at large are not thought of. Typical approach has been to develop powerful smart wireless interfaces, which supports the important features/requirements for a particular class of applications like military, environment sensing or more focused applications like fuel-level control in automobiles. The result is a plethora of wireless interfaces appropriate for a certain class of applications; but almost no interoperability between them [3]. This approach has resulted in horizontal system models leaving little space for reusability and slimness, which are most desirable requirements of WSN. The survey of WSN applications and their distribution shows that, despite many proposals, no common benchmarks or detailed, large-scaled experiments have been published. The research seems to focus either on node implementations or theoretical work on distinct aspects, such as routing algorithms, without a realistic relation to physical platforms [4].

The diverse application areas have inflated requirement specifications for WSNs. There is a need to synthesize the WSN application requirements, WSN characteristics and design the system accordingly. We believe that rational specifications could lead to a reusable WSN framework that would be useful for many of the WSN applications. We take bottom to top approach for the system design. The end node hardware and software architecture is designed meeting requirements of WSN characteristic, and then the rest of the system is built as a support system. This work presents 'SWiFinet' (Swift Wi-Fi Network); a reusable task distributed WSN architecture. The architecture is based on practical assumptions drawn from the WSN application requirements. SWiFiNet is an application independent task distributed architectural framework that is usable in a large number of various WSN applications.

Section 2 talks about the characteristics of a reusable and reconfigurable WSN in terms of node specifications, network type and size, network life time and energy optimization, network dynamics, self organization and scalability, data transmission and communication reliability and lastly overlay backbone support. Section 3 describes the earlier studies that have motivated Design of 'SWiFiNet'. Section 4 presents the features of 'SWiFinet', designed as per the requirements explained in earlier sections. Section 5 presents implementation on hardware as well on simulated platform. Section 6 evaluates the SWiFiNet with performance parameters and section 7 concludes the study.

II. CHARATECTERISTICS OF REUSABLE, RECONFIGURABLE WSN

Hierarchical network, network dynamics, data transfer models, scalability and energy optimization using scheduling are desirable features of the reusable WSN architecture.

A. Node Centric System Characteristics

WSN is defined as congregation of small tiny wireless sensor nodes. The sensor nodes are lightweight, power aware, small size, low cost and are in large numbers to be deployed in the field. The node description translated into hardware and software specifications of the end node imply the use of 8 bit/16 bit processors having less computing power, limited memory size, less peripheral resources, small size and low cost. The power source is often small battery, prompts for aggressive energy optimization at all levels.

B. Hierarchical Network, Network Size

Hierarchical networks are suitable to deploy and manage the network. The WSNs are deployed with specific tasks and aims, in most of the cases it is data gathering from the deployment field. Real field deployments of WSN demonstrate that the network size is limited often to 10s to 100s of nodes satisfying large share of WSN application domains [1,5]. The network design specifications like network throughput, reliability and lifetime would demand deterministic behavior of network devices in terms of energy consumption, reliability metrics.

C. Network Lifetime and Energy Optimization

WSN lifetime can be described as time until the last useful message is delivered from the network to the system. This implies maximum lifetime for the end node. Major energy consumption is in radio communication. The radio communication can be reduced by having an optimum sleep - wake up schedule satisfying requirements of the application

D. Network Dynamics, Self-Organization, Scalability

Advantage of wireless network is easy deployment and immediate startup of the network. The network topology can change over time or due to application demand. Scalability is often an issue in the networks. The network must support a sufficient number of nodes and their addition and deletion in the field. A resource constrained sensor node cannot handle these issues like traditional networks with resourceful and constraint less end device. Most of this functionality can be pushed onto backbone network devices.

E. Data Transmission Models and Communication Reliability

There are three types of data transfer which take place in WSN. Periodic data transfer, event driven data transfer and queried data transfer. The network layer should support all these transport models. The communication reliability is a measure of success of receiving the message at the destination.

F. Overlay backbone support

The hierarchical architecture has second tier devices having more resources than end node. We further expand this concept to overlay infrastructure that will act as message catching layer, which will be well equipped for handling the complexity of network and application functionality in distributed form. While overlay infrastructures are free from the constraints imposed by WSN characteristics, it will provide increased network and computational capacity enabling large-scale deployments

III. BACKGROUND

'SWiFiNet': a task distributed reusable architecture for WSN has been inspired by earlier researchers who worked on the concepts based on the parameters/keywords discussed above. Hierarchical task distributed WSN architectures for example have been proposed earlier also in the WSN research domain.

Theme of such proposals is providing an intermediate overlay layer in terms of system components of high resources to facilitate the system interface to the high end networks. Many of them have proposed 802.11 based dynamic or fixed backbone devices so that rest of the system can make use of existing system software and hardware to interface with internet cloud. Some of the researchers have proposed centralized network management. An infrastructure based deployment (i.e. Fixed wireless deployment) [3], Tenet architecture based on three important tenets [6], three layered hierarchical architecture presented in iCASS [7] are few examples. Leel et al. have presented ART wise gateway architecture [8]. The envisioned architecture is two-tiered, IEEE 802.15.4/ZigBee for Tier 1 and IEEE 802.11 for Tier-2.

According to Linfeng (2010) there are two challenges in WSN architecture design: the first is the message interactions among different modules; the second is how to reuse the communication protocols. The authors have presented Environment-Adaptive Architecture Model for Wireless Sensor Networks (EAWNA) The tier 2 is capable of the overlay WLAN must be able to deal with a large number of nodes so that its increase does not affect the behavior of the two-tiered architecture [9]. The ANGEL architecture is presented for health care application, where the authors claim that it can be reused for any other application as well [10]. In Reconfigurable wireless networks H. Ramamurthy has presented a reusable solution for industrial wireless sensor networks [11].

Triantafyllidis has proposed an open and reconfigurable Wireless Sensor Network (WSN) for pervasive health monitoring, with particular emphasis on its easy extension with additional sensors and functionality by incorporating embedded intelligence mechanisms [12]. ZUMA [13], is a centralized future smart-home platform that interconnects all kinds devices in the home environment, Mote herding uses a mix of many 8-bit sensor nodes (motes) and fewer but more powerful 32-bit sensor nodes (micro-servers) [14], EMMON a system architecture for large-scale, dense, real-time embedded monitoring[15] are few more works have some resemblance. EMMON provides hierarchical communication architecture together with integrated middleware, command, control software and follows the lead line of having overlay dynamic/fixed second tier resourceful devices.

IV. DISTRIBUTED SYSTEM ARCHITECTURE: 'SWIFINET'

'SWiFinet' is designed as distributed task architecture for the WSN where the emphasis is on task distribution and reusability of the system components. A task-layered architecture is shown in figure 1. The network is envisaged as hierarchical architecture, master sink node, router node and sensor node. The routers are used for extending the physical distance connectivity and network management. The base layer is 802.15.4 MAC/PHY layer. Built on top of this layer is a common DLL task layer. This layer provides basic functionality for transmitting and receiving data and control packets. The DLL layer provides functionality for network communication handling and is transparent to the user. The network task stack grows as per the device hierarchy.

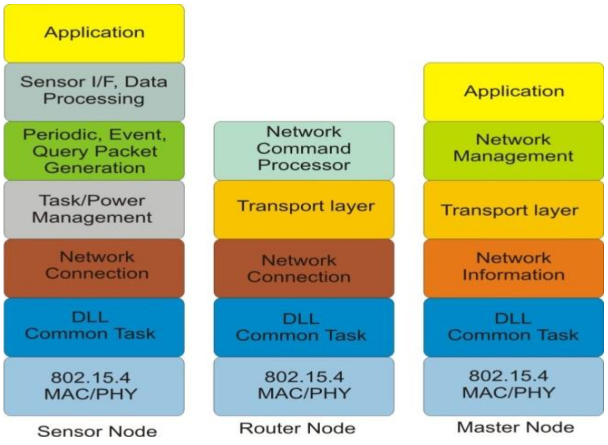


Fig.1. Distributed Task architecture for WSN

A. Sensor Node

The sensor node architecture carries the only network task restricted to connecting to the available neighboring overlay router nodes or to the master. The transport mode that is with 'ack' or without 'ack' and message priority is embedded in communication packet. Once this packet is handed over to the parent device, it is the responsibility of the parent device to transfer the message to the destination device.

The network layer maintains a routing table containing two parent device entries. The first entry is a primary parent address and the second entry is a secondary parent address. The network connection layer invokes connection functionality when there is no entry in the routing table and a data packet request is generated from upper layer. A hello packet is broadcasted in this case. The neighboring parent device issues join request. This join request is accepted, if the received packet meets LQI threshold criteria. If there are multiple join requests received by the sensor node then minimum hop criteria is applied. Task/Power management layer provides support for wakeup-sleep schedule for the node. This schedule is application dependant and can be configured runtime by the master sink node. The tasks are managed by task scheduler if available or can be a super loop structure. Sensor interface and data processing layer contain drivers for I2C, SPI, ADC, DAC devices and data processing routines. The application layer manages the application deployed with associated configuration parameters set at lower layers.

B. Router Node

The router node is used if a sensor node is not in the range of the master sink node. The router node can also be used if there is more number of devices and clustering is required in case of network management. The router joins to the network in a similar way as explained for the sensor node except here the router starts connection operation immediately instead waiting for the data transfer request to arrive from the upper layer. The additional task of generating a join request is added in this layer. The join request is generated if the device receives hello packet from the sensor node or neighbor router node. The router can respond to the hello packet only if he has joined to the network and secondly the requesting device meets the threshold LQI criteria. The router maintains a similar table like sensor node having parent device information leading to the master sink node. Another table maintains device addresses for the child routers down the network line. The router does not maintain any database for end node devices. Once the routing node receives the data packet, it is the routing nodes responsibility to transfer to the next destination node and so on. The command processor layer provides functionality for data aggregation, data fusion depending on the configuration parameters. A master node can set the configuration parameters at run time.

C. Master Node

The master node maintains complete topology information of the network. The network information layer is responsible for maintaining routing tables for each node in the network. Mainly this is required for sending command packets and querying the end node. The routers and end nodes will send their routing tables every time these tables are updated. The network and application parameters can be tuned by configuring routers and sensor nodes configuration tables.

D. SWiFiNet: Points of Comparison

SWiFiNet even though has similarities with the architectures explained in section 3.0 also has major differences in many ways. The second tier in the SWiFinet architecture uses the same hardware as sensor nodes except sensor part and is based on 802.15.4 complaint trans-receivers unlike many of the above architecture having higher bandwidth radios like 802.11 likes at the second tier. Sleep-wake schedule in SWiFinet is application dependant and is dictated by an end node. This solves two major issues, firstly the sensor node life is deterministic and application dependant and secondly no time synchronization overhead incurs saving program complexity and power consumption in communication. The overlay devices are always active when they need to hear the end node communication. The configuration parameters at each task layer provide facility to tune the network and application specific parameter. In SWiFiNet network operation is completely decoupled from the application functionality. SWiFiNet aims to provide an integrated reusable and reconfigurable architectural framework for a variety of applications.

V. IMPLEMENTATION

'SWiFiNet' is implemented on hardware platform as well in simulator ns-2 platform. This gave an advantage to test all the logic in a simulation environment. Particularly difficult and complex situations that are difficult to realize during hardware testing were simulated and problems were resolved on simulation platform. The system software code was reused with hardware and in creating agent '*SWiFi'* in ns-2 simulator.

A. Hardware Implementation

SWiFiNet architecture is realized using a single chip CPU plus trans-receiver solution. The implementation is not processor specific and can be ported to any other hardware platform. The Single chip processor is 32-bit RISC processor operating at 16 MHz, with 2.4 GHz. 802.15.4 compliant trans-receiver. Receiver sensitivity is -97dB, consumes 17.5 mA in active mode. Transmission power is +0.5 to +2.5 dB with programmable stages. The processor has SPI, I2C, UART bus, along with timers and IO ports. Figure 2 Shows sensor node hardware. The same hardware can be used for router and master node realization.

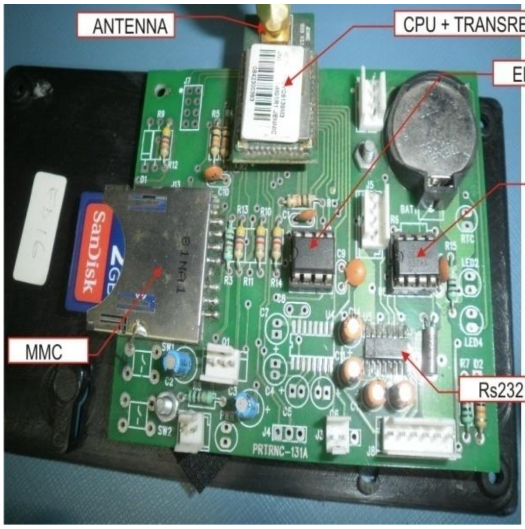


Fig.2. Hardware platform used for WSN nodes

B. ns-2 Simulation of SWiFinet

Simulation has its own advantage of checking all the operational logic and allows testing limiting conditions which otherwise could not be tested in real field situations. The ns-2 network simulator provides basic radio models and base functionality for packet communication. It allows building proprietary protocol model by creating network agent. This agent can be invoked in the TCL interface script to verify the behavior of the protocol in different scenarios.

C. Agent SWiFi

Agent '*SWiFi'* software architecture is shown in figure 3. The portability of the software was truly tested while porting the software blocks in ns-2 environment. There was 100% success in porting software architecture between ns-2 environment and hardware platform. Network common tasks are used as basic interface between device specific tasks.

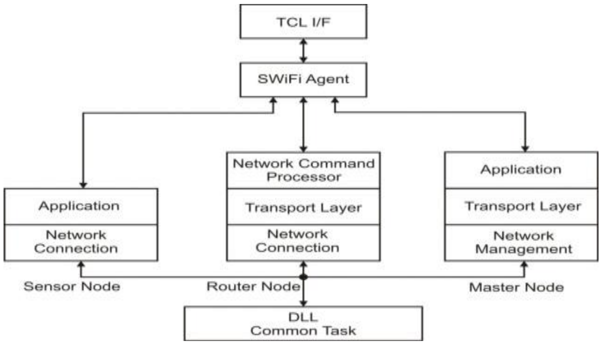


Fig.3. Block schematic of network component simulation in ns-2

C. Architecture and building interface

The devices were identified by the accessible variable passed from the TCL script while invoking the '*SWiFi'* agent. Value 1, 2, 3 were attached to master, router and sensor node sink node functionality respectively. An application layer at end node was used for generating packets at a desired sampling rate, packet size and data mode transfer. An application layer at the master sink node was tested by developing PING application.

VI. RESULTS AND DISCUSSIONS

SWiFiNet shares a common characteristic with Dynamic Source Routing (DSR) protocol as they both initiate routing activity on an on demand basis. The reactive nature of this protocol is a significant departure from proactive traditional network protocol. The motivation behind this is the reduction of the routing load. Since the routing load is an important parameter in WSN in perspective of power consumption SWiFiNet adopted reactive approach to set up the network. Although DSR is meant for mobile ad-hoc network, it will be interesting to compare them in static topology. SWiFiNet also supports mobility of the devices within a network. Simulation in ns-2 is carried with propagation model: Two Ray Ground model, net-interface: Phy/WirelessPhyExt and mac layer MAC/802_11Ext. Field size is set to 1000 m x 1000 m and bandwidth is set at 1 Mbps. The results are averaged over three sets of experimental values. Hardware results are presented in comparison with simulation. Standalone hardware results are presented with a variation in packet interval.

I. Performance Parameters

The normalized routing load is number of routing packets transmitted per data packets sent to the destination and also each forwarded transmission is calculated as one transmission. Average route acquisition time is the time taken by the network devices to join the network and become functional. An average transmission delay is an end-to-end delay for the message to reach destination averaged over total message transmitted.

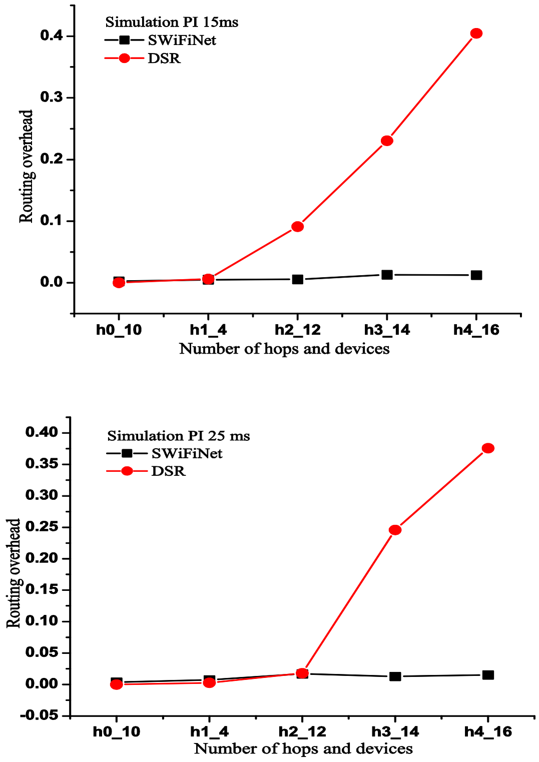
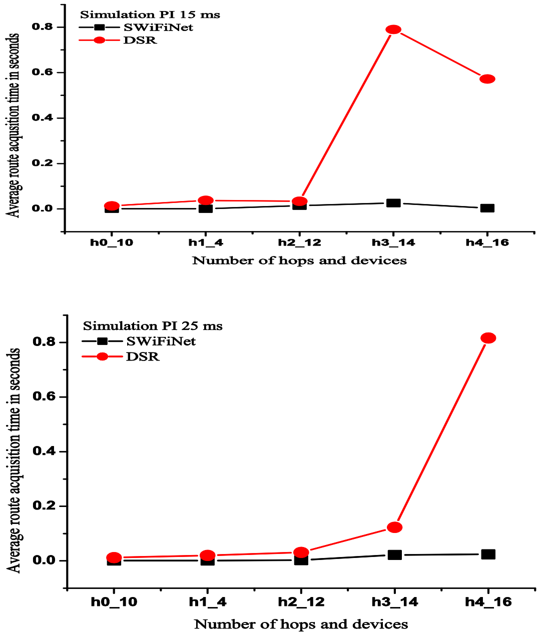
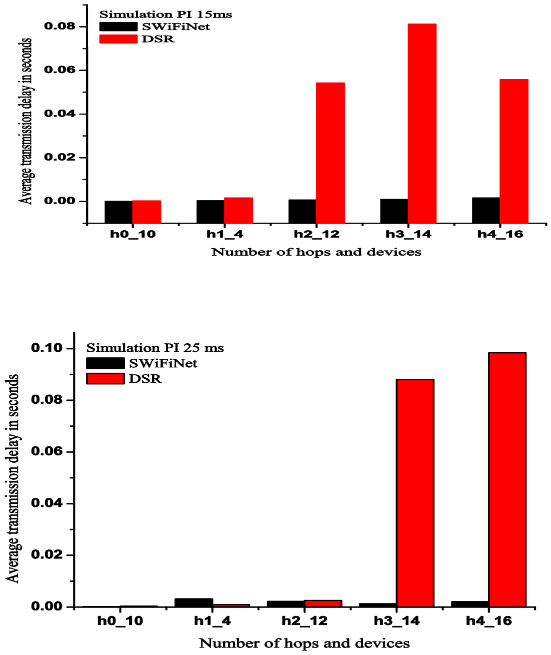


Fig.4. (a). Normalized routing load



(b). Average route acquisition time



(c). Average transmission delay

Figure 4 (a) shows normalized routing overhead. The routing overhead increases sharply with the increase in number of hops and devices with DSR. The routing overhead more or less remains constant in SWiFiNet. This is because SWiFiNet is based on deterministic communication protocol. The joining process is simple and involves minimum message communication. Secondly, the parent devices respond to neighboring device only when it is part of the network. This deterministic behavior is supplemented with results of route acquisition time shown in figure 4 (b). The SWiFinet route acquisition time shows marginal variations with an increase in hops and devices as compared with DSR. DSR shows varying and steep increase in time to increase in the hops and number of devices. Average transmission delay is plotted in Figure 4 (c). The SWiFinet shows almost constant delay value since routing paths are fixed. Large variations in average transmission delay are observed with DSR due to non-deterministic routes formed by the DSR protocol as expected.

II. Hardware Results

Hardware experimentation is carried out within an open air field in line of sight devices. A set of 50 ms and 250 ms packet interval are compared considering the 802.15.4 radio 250 kbps bandwidth. Figure 5 (a) shows normalized routing overhead for 25 ms packet interval simulation results compared with 250 ms packet interval for hardware experiment result set. The plot shows an elevated overhead graph for hardware results incurring more command packets per data packet as compared with simulation results. This elevation is attributed to difference in simulation radio model and field radio.

The field radio requires more command packet transmission retries owing to field environment. Figure 5 (b) and figure 5 (c) shows average route acquisition time and average transmission delay respectively. The route acquisition time increases with decrease in packet interval rate. This increase in time is due to the mobility mechanism introduced into the protocol.

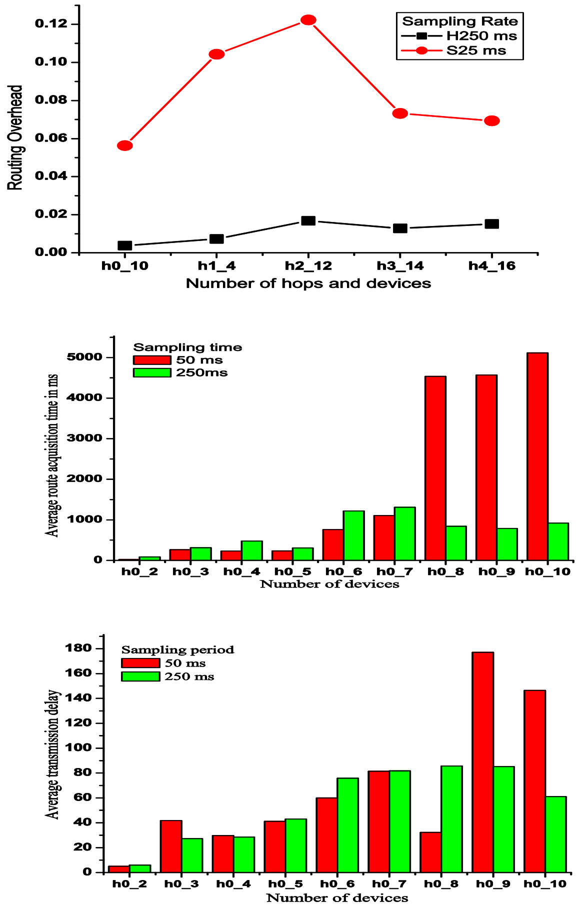


Fig.5. (a). Normalized Routing Load Figure 5 (b). Average route acquisition time Figure 5 (c). Average transmission delay

If a threshold repeats failure count crosses the set threshold value, the transmitting device assumes that the path is broken and starts a fresh joining process. The average transmission delay time increases as the communication congestion increases in the network due to higher packet interval rate and the number of devices.

VII. Conclusion

SWiFiNet is task distributed generic reusable architecture for WSN. The sensor node has been spared from wireless network management complexity, keeping network component required for joining the network. The network complexity of reliable data transfer, network management and other network related operation has been pushed on to overlay second tier devices having more power resources. This will make the node; small size, low cost and power optimized as envisioned by the WSN characteristics. The results show that SWiFiNet performs better on many network parameter accounts. The deterministic behavior of the network operating at end node will allow the application designer to calculate network lifetime to accurate degree. The reusable modules will pave the way for building and deploying the application quicker. The design is scalable and self-organizing after deployment. The configuration parameter at various levels allows the designer to tune the network as per the application requirement.

VIII. Acknowledgments

The authors wish to acknowledge the support given by M/S Partonics & Sinhgad Technical Education Society.

References

[1] K. C. Bhaskaran Raman, "Censor Networks: A Critique of “Sensor Networks” from a Systems Perspective," *in proceeding of ACM SIGCOMM Computer Communication Review* vol. 38, pp. 75-78, July 2008

[2] P. D. David Culler, Cheng Tien Ee, Rodrigo Fonseca, Jonathan Hui, Philip Levis, Joseph Polastre, Scott Shenker, Ion Stoica, Gillman Tolle, and Jerry Zhao, "Towards a Sensor Network Architecture: Lowering the Waistline," pp. 1-5.

[3] B. S. P. Harish Ramamurthy, Rajit Gadh, "Reconfigurable Wireless Interface for Networking Sensors (ReWINS)," *in proceeding of the 9th IFIP International Conference on Personal Wireless Communications (PWC 2004),* 2004.

[4] M. H. a. a. Mauri Kuorilehto, Timo D. H¨am¨ al ¨ainen, "A Survey of Application Distribution in Wireless Sensor Networks," *EURASIP Journal on Wireless Communications and Networking* vol. 5, pp. 774–788, 2005.

[5] M. K. N. A. W. Rohankar, Amitava Mukherjee "A STEP TOWARDS REUSABLE WSN ARCHITECTURE," *International Journal of Research and Reviews in Wireless Sensor Networks (IJRRWSN)* vol. Vol. 2, 2012.

[6] E. K. D. E. R. Govindan, F. Bian, K. Chintalapudi, O. Gnawali, S. Rangwala, R. Gummadi, and T. Stathopoulos, "Tenet: An Architecture for Tiered Embedded Networks," *in Proceedings of Tech. Rep,* 2005, November 10

[7] M. Cinque, Di Martino, C. And Testa, A, "iCAAS: Interoperable and Configurable Architecture for Accessing Sensor Networks," *In Proceedings of the 3rd International Workshop on Adaptive and DependAble Mobile Ubiquitous Systems - ADAMUS 2009,* pp. 19-24, 2009, July 13-17.

[8] J. C. Leal, A. Alves, M. Koubaa, A., "On a IEEE 802.15.4/ZigBee to IEEE 802.11 gateway for the ART-WiSe architecture," in *Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on*, 2007, pp. 1388-1391.

[9] L. Linfeng, "Research on Environment-Adaptive Architecture Model of Wireless Sensor Networks," in *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, 2010, pp. 130-133.

[10] A. H. Willig, J. H. Karowski, N. Baldus, H. Huebner, A., "The ANGEL WSN Architecture," in *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on*, 2007, pp. 633-636.

[11] B. S. P. H. Ramamurthy, and R. Gadh, "Reconfigurable Wireless Interface for Networking Sensors (ReWINS)," *In Proceedings of IFIP TC6, 9th International Conference, PWC 2004,* 2004, September

[12] A. K. Triantafyllidis, V. Chouvarda, I. Maglaveras, N., "An open and reconfigurable Wireless Sensor Network for pervasive health monitoring," in *Pervasive Computing Technologies for Healthcare, 2008. Pervasive Health 2008. Second International Conference on*, 2008, pp. 112-115.

[13] M. N. K. V. G. Soini, J. Rabaey, J. M. Sydanheimo, L. T., "Beyond Sensor Networks: ZUMA Middleware," in *Wireless Communications*

and Networking Conference, 2007.WCNC 2007. IEEE, 2007, pp. 4318-4323.

[14] L. G. T. Stathopoulos, J. Heidemann, and D. Estrin, "Mote herding for tiered wireless sensor networks," *in proceeding of Technical Report 58, CENS*, 2005, Dec. 7.

[15] M. e. B. Stefano Tennina, Pedro Braga, Ricardo Gomes, Mario Alves, Farrukh Mirza, Vincenzo Ciriello, Gabriella Carrozza, Pedro Oliveira, Vinny Cahill, "EMMON: A WSN System Architecture for Large Scale and Dense Real-Time Embedded Monitoring," presented at the Ninth IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2011.