An Enhanced Architecture for an Infrastructure based Middle-ware using Adaptive Network Coding and Global Data Formatting to Achieve Efficient IoTs

Kayalvizhi Jayavel
Assistant Professor (Sr.G)
Dept. of Information Tech.
SRM University,Kancheepuram
Tamil Nadu – 603203 India

Nagarajan V
Professor
Dept. of Computer Science
SRM University,Kancheepuram
Tamil Nadu – 603203 India

Sheikh Jaaved Hassan
Student M. Tech
Dept. of Information Tech.
SRM University,Kancheepuram
Tamil Nadu – 603203 India

ABSTRACT
Among many Infrastructures based Middle-wares proposed Global Sensor Network (GSN) is one among them to mitigate the co-existence issue to interconnect IP based and legacy Wireless Sensor Networks. It is an open-source, Infrastructure based abstraction Middle-ware developed in Java. Though there are advantages like simplicity, adaptability, lightweight and scalability, there are still few areas for improvement. This paper attempts to identify those, propose an enhanced architecture and analyze with a case-study. Data-Acquisition module is one area to peek in. GSN represents every WSN as a Virtual Sensor (VS) using XML. Based on its requirements, each user application aggregates data from various Wireless Sensor Networks (Virtual Sensors). As the data format is Virtual Sensor specific, it is the responsibility of every heterogeneous user application to identify and resolve data discrepancy if any. Moreover, the data aggregation method employed is not explicitly described and when investigated, is no different than the lower level abstractions. Hence this paper proposes an enhanced architecture with an adaptable data aggregation achievable via integrated Network Coding and a global data format as part of the Middle-ware..

General Terms
Wireless Sensor Networks, Middleware, Internet of Things

Keywords
Global Sensor Networks (GSN), Data aggregation, Network Coding

1. INTRODUCTION
The world is progressing towards Internet of Things (IoT). IoT refers to uniquely identifiable objects (things) and their virtual representation in an Internet as a structure. RFID (Radio Frequency Identification) is often seen as prerequisite for Internet of Things. This concept became popular through the Auto-ID center/Lab which is a research group in the field of networked RFID and emerging sensing technologies. Before we talk about the current state of IoT, it is important to agree on a definition. According to the Cisco Internet Business Solutions Group (IBSG) [1], IoT is simply the point in time when more “things or objects” were connected to the Internet than people. In 2003, there were approximately 6.3 billion people living on the planet and 500 million devices connected to the Internet.3 By dividing the number of connected devices by the world population, we find that there was less than one (0.08) device for every person. Based on Cisco IBSG’s definition, IoT didn’t yet exist in 2003 because the number of connected things was relatively small given that ubiquitous devices such as smart phones were just being introduced. For example, Steve Jobs, Apple’s CEO, didn’t unveil the iPhone until January 9, 2007 at the Mac-world conference. IPv6 [2] paved way for such a reality to interconnect every smart object with IP based system. But the smart objects or WSN's have hardware/memory and power constraints. To overcome this some kind of adaptation between IPv6 and LoWPAN is needed. Hence 6LoWPAN IETF WG (Internet Engineering Task Force Work group) [3] has already come up with an adaptation layer. The function of adaptation layer is compression, fragmentation and address auto-configuration.

6LoWPAN is useful for developing applications where smart objects or Embedded devices need to communicate with Internet based services using Open-standards and able to scale across large network infrastructures with mobility. Many LoWPANs are connected to other IP networks through Edge routers (Routers at edge). Edge Routers [14] takes care of data transfer in and out of LoWPANs + 6LoWPAN compression + Neighbor discovery for LoWPAN. That is adaptation between full IPv6 and LoWPAN format is performed at edge router. The fundamental advantage of using IPv6 is Smart objects can use Web services.

In this direction a Application layer protocol to achieve web transfer CoAP (Constrained Application Protocol) is under progress. CoAP is efficient in all IP infrastructures. But the real scenario is coexistence of IP based solutions and legacy technology. Hence a viable alternate is high level Middle-ware which maintains an Internet Overlay Architecture in which network protocols are all inherited from Internet backbone. A Middle-ware comprises of Business logic and Storage. The former is capable of programming, configuring, monitoring and controlling system behavior. The later may be a supportive database with minimum historicity and limited processing or Archive database which is capable of much more processing.

One such Middle-ware is Global Sensor Network (GSN)[4]. It represents each Wireless Sensor Network as a Virtual Sensor. Each Virtual Sensor will need a new Wrapper implementation whenever user application requirement changes (say Energy management application, People Tracking, Environment monitoring etc…). It uses Semantic query mechanism to overcome this issue. Semantic web services are flexible mechanism for data retrieving and dynamic discovery services. It also has a centralized architecture where only point to point communication is allowed. This paper tries to propose an adoption of different scheme of device discovery and data aggregation technique to improve the performance of existing framework.

The rest of the paper is organized as follows. Section 2 presents the Overview of Middle-ware. Section 3 presents the related research works. Section 4 provides the proposed Architecture model. Section 5 provides an analysis of the proposal using a case study. Section 6 concludes the paper with future directions.

2. OVERVIEW OF MIDDLE-WARE

Sensor Networks are deployed and accessible in an ad-hoc fashion. Many of the researchers were fascinated by the success of Web and wanted to achieve the same for heterogeneous wireless sensor networks. The reasons for success of web were 1. Very few logical abstractions say URL, hyper-links, HTML and 2. Basic communication protocols (HTTP, Web services). These are the parameters helped to achieve universal access and linking among autonomously published data sources. Abstractions [5] can happen at three levels Node level, Network level and Infrastructure level as given [Fig.1].

2.1 Abstraction at Various Levels

2.1.1 Node level: Hardware and Communication protocols are abstracted at node level. Popular abstractions to list a few are TinyOS, Contiki and Mate. Application are developed using NesC or C. The application tasks include sensor data reading, processing, forwarding the same to others when needed and above all also capable of updating or distributing applications in the network. But all these are limited to node level only.

2.1.2 Network level: This level focuses on data access (queries) and processing (data aggregation) that needs cooperative behavior among distributed nodes in the Network. They provide distributed measurement sharing for Node applications through distributed memory abstraction. Few examples for this kind of abstraction are TinyDB, COUGAR, Agilla and TinyLime. To consolidate the main task of this abstraction is access to WSN measurement data. But the drawback is this measurement data alone is not sufficient for any end-user application. Hence there is an unavoidable need to incorporate some amount of meta-data say information about physical location and/or further processing, and/or combining data from other technologies. So it becomes very clear that Node level and Network level abstractions are not adequate to cater the requirements of an end-user application. Thus an Infrastructure level abstraction.

2.1.3 Infrastructure level: The function this layer carry through is similar to Network abstraction but at a larger scale. That is, Network abstraction works with heterogeneous nodes with the network whereas the Infrastructure abstraction works across heterogeneous Networks. But both are classified as WSN Middle-ware. This level extracts the details needed for the end-user application from the Node and Network abstractions. Every technology would have demanded a unique tailor made solution in the absence of a Infrastructure level abstraction. The infrastructure abstraction is fairly new research area with the growing end-user applications deployed over various technologies there by demanding seamless interoperability to achieve IoT. To summarize Infrastructure abstraction paves way to achieve fast, efficient technology independent WSNs to inter operate

Global Sensor Networks aims at achieving the same benefits of web for WSN saysimple data access, powerful and

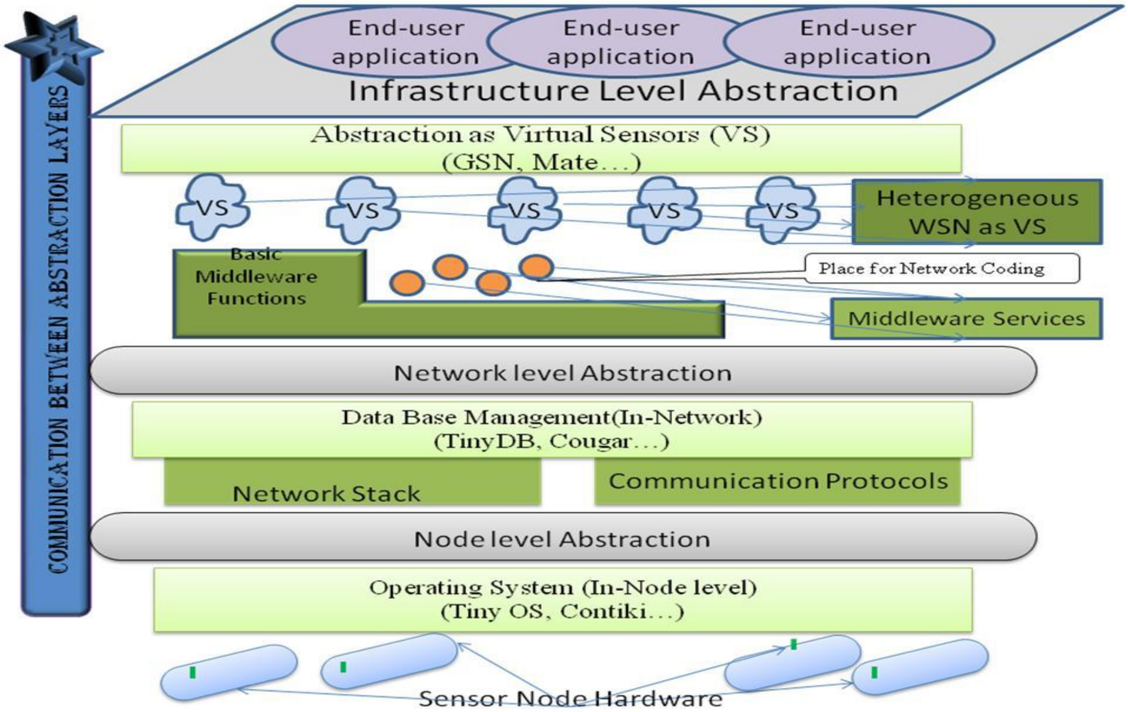

Fig. 1: Overview of Middle-ware

flexible architecture [6]. The 4 goals of GSN are 1. Simplicity is achieved through SQL and XML, SQL as a data manipulation language and XML as syntactic framework, 2. Adaptability through container based implementation allowing dynamic reconfiguration, 3. Scalability based on peer to peer architecture to achieve distributed query processing and distributed discovery of sensor networks 4. And it has a Light-weight implementation.

Sensor Networks provide data that can be streamed (audio/video) or event based (Temperature, RFID tag read). The sensor readings are different from other forms of data say multimedia, for the reasons as synchronization issues, erroneous messages, limited resources and streaming leads to packet loss. And the raw data received from various senor networks become useless if not available in a structured way. Hence ways to automatically process and manage the data became need for the hour. Context awareness, Data fusion, Information integration, Information merging of traditional networks is becoming a necessity in WSN too.

3. RELATED WORKS

Wireless Sensor Network Middle-ware is used to bridge the gap between the high level requirements of the application and underlying complexity of the WSN. A Middle-ware is software and a tool used to abstract the complexities and heterogeneity of the hardware and network platforms, ease the management of system resources, and increase the predictability of application execution. The application requirements include flexibility, re-usability and reliability. The complexities of WSN includes resource constrained hardware, dynamic network topology and low level Embedded OS API. In earlier days when the applications were very simple there existed no necessity for a Middle-ware. But now with growing Application needs the gap has almost created a handicapped situation.

The Middle-ware provides the developer or the user the flexibility to concentrate on his/her application logic ignoring the system level details. They have the capability to provide code re-usability and data filtering to achieve easy deployment and execution of application [7]. Apart from these it also provides Network Infrastructure Management, Adaptation (example: Power management), System Integration and Monitoring.

Middle-ware is well established in the domain of Distributed computer systems, but poses new challenges in WSN's. The requirements of traditional Middle-ware techniques are very much different than needed for WSN. For instance the Distributed computer systems aims at providing transparency by hiding the context information as against WSN were the whole focus is on context awareness. Moreover WSN being data centric networks demands the location and mobility of senor nodes to be dealt differently. Also WSN's use attribute based addressing [8] as against unique network wide node addressing. Data aggregation has much more importance in WSN's when compared to traditional Middle-wares for Distributed computing systems. Above all the WSN Middle-ware has to be light weight [9]with additional requirements towards Hardware, Operating Systems, Routing protocols as well as applications.

The basic components expected in a Middle-ware are abstraction layer for application programmers, set of functions to achieve the abstraction, extensions to use the underlying Operating System, features to achieve Quality of service. It is not mandatory that all Middle-wares should posses all of the above. And there is no rule that the Middle-ware should be centralized at sink node. There is possibility to decentralize the Middle-ware functions across end devices, sink nodes and at user terminals. These distributed features co-ordinate to achieve the goal.

Abstraction layer for application programs has various programming paradigms based on the Data collection patterns and the type of application itself. The ways of data collection may be continuous, event-driven and query based. For example, Event driven uses publish/subscribe paradigm, query based uses database paradigm. In some cases database paradigm is also found suitable for event driven data collection applications. This layer is also responsible for providing appropriate interface to the outside world. They again classify in to Descriptive interface or Insistent interfaces. The Descriptive interface demands query interpretation which consumes more resources whereas the latter provides flexibility at the cost of burden on programmer to specify the programming logic. Hence it is the responsibility of the user to choose the right choice of paradigm and interface appropriate to his/her application.

Though there are numerous Middle-wares [7][10] available GSN became very apt when considering IoT. Global Sensor Network focuses on data centric event based applications as many of the WSN's fall in to this category. It uses Semantic web to achieve its data collection, data aggregation and data fusion. The Semantic web approach proves much more efficient than approaches adopted by other Middle-wares. It uses SQL like languages that allow one to define how data can be gathered from pervasive systems and the same can be processed efficiently. It uses Virtual sensor abstraction a key concept of GSN. And when there is a need to connect all IP based networks with legacy technology there is a definite need for a highly featured Middle-ware that should be able to maintain Internet overlay architecture. Internet overlay architecture inherits the networking protocols from the Internet backbone. The other merit which helps to scale GSN compared to others is its data in-dependency that is data dependency between sensor data and application is void. This is achieved through the concept called Wrapper classes. But this merit comes with a hidden knot, when comes to consolidation of data from various Virtual Sensors. The data (may be of different types and formats) collected from various sensor networks by a user application should be resolved of all its ambiguity before further processing. This demands resolving at user application level. And GSN still has some areas for improvement at data collection, data aggregation and dynamic discovery services which has lead to the reason of this paper. In this paper a new approach of network coding integration towards data collection/data aggregation is proposed which is dealt in forthcoming section with a case-study.

4. PROPOSED ARCHITECTURE MODEL

Network coding refers to a scheme where a node is allowed to generate output data by blending the data it has received. This is achieved by application of various encoding schemes available. In traditional routing Network coding is gaining momentum. The same technique is picking up in various forms in the fast growing wireless sensor networks too. In routing of output messages will only be copies of received messages whereas in application of Network coding each node in a network can perform some computation.

There are two different ways of data processing namely, Centralized and Distributed. As the name signifies in

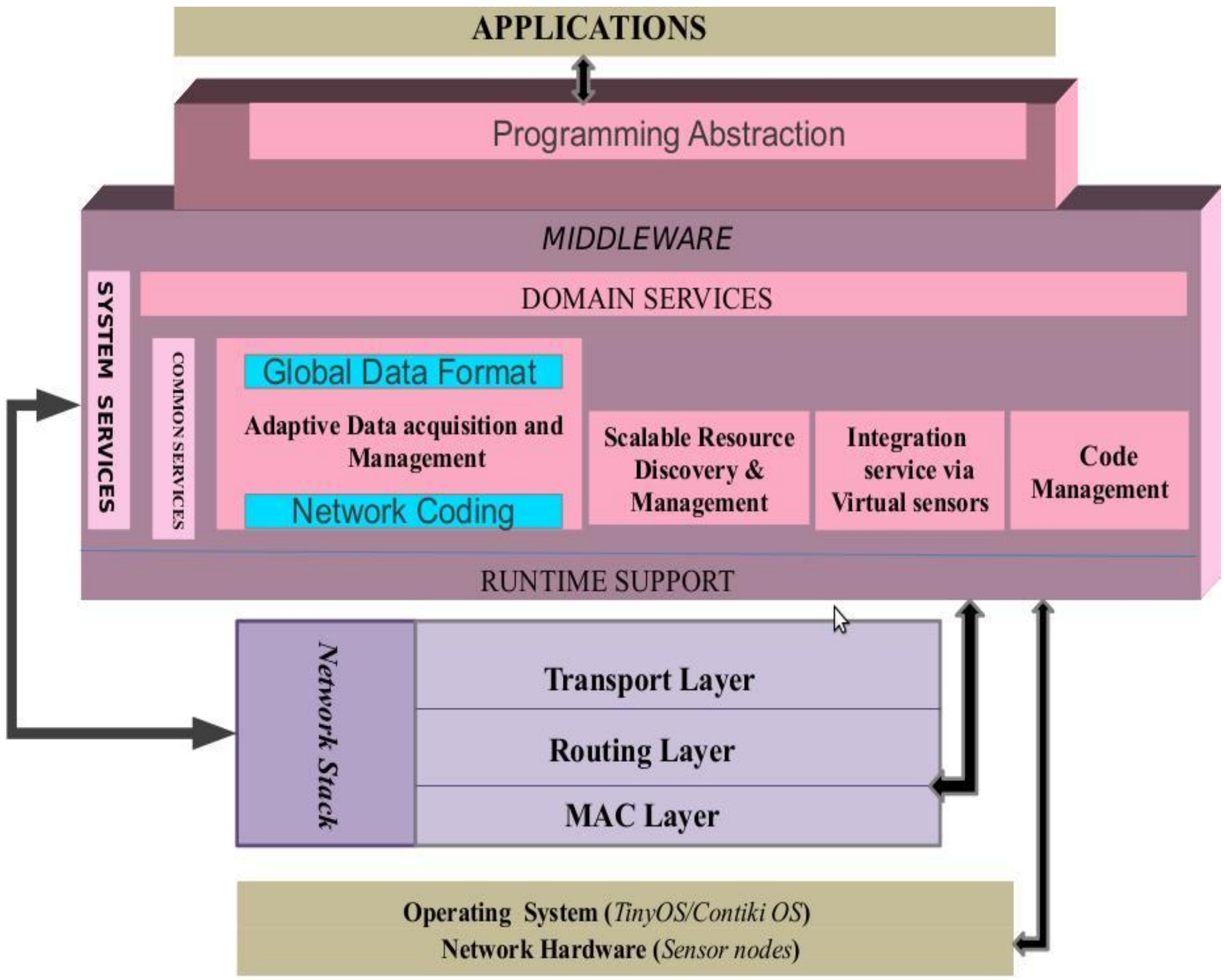

Fig. 2: Enhanced Middle-ware Architecture model

the former all the data are collected and sent to the central node for processing. But there is a possibility of over power utilization of that node. In the latter there are two subdivisions namely, Node level and Network level. In Node level distributed data processing every node does some minimum processing before the final processing at the sink node. The Network level distributed data processing incorporates node level as well as mutual information exchange between nodes hence there is 100% distributed processing possibility, as each node is aware of the final decision. Network coding is capable of reducing the total number of packets transmitted.

It is well known fact that communication cost is many folds compared to the computing cost. Hence some kind of In-network distributed computing through data aggregation or data fusion will definitely add up to the performance. This paves a way for the possible adoption of Network coding in to the existing architecture of GSN [Fig. 2].

As we have already discussed the prime components of a Middle-ware are Programming abstraction, System services, Run time services and QoS (Quality of Service). And it is the application programmer who has to choose the appropriate programming paradigm and data collection models based on his/her application. If the application user is also allowed to make his/her choice of Network coding this will definitely add up to the efficiency and performance enhancement without doubt. Here there is a question on flexibility available to the user about choice of Network coding or none. Thus this paper proposes a model of either default Network coding scheme or the choice of Network coding by user. So this is included as one of the functionality under the System services. As we remember System services is the module which comprises of all the actual functional implementation required for the application programmer. All these System services are available with the help of lightweight API to the user.

The static Network coding scheme can be implemented with a simple Linear Network coding scheme [Fig. 3] which is explained as follows. It needs a few linear operations and several bytes storage to reduce the power consumption to a greater extent and thereby enhancing the life time of sensor networks. Adapcode [11] one of the popular practical Network coding can be deployed for this service. But Adapcode has limitation that it cannot find all actual neighboring nodes which is critical in determining new coding schemes. Thus people have already proposed a power efficient neighbor discovery protocol to find out all its neighbors [12]. This integration could add up to the performance of the proposed system.

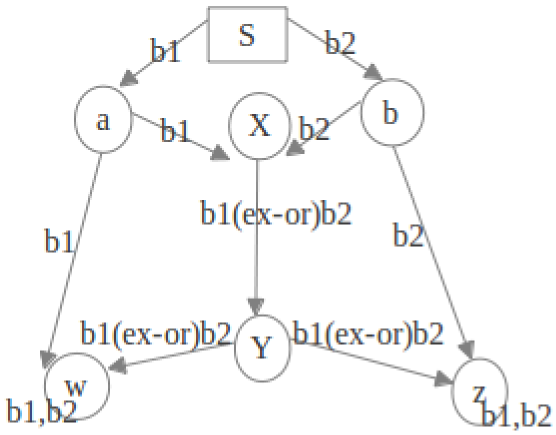

Fig. 3: Linear Network Coding with ex-or

Considering Fig. 3 all the links have unity capacity. Hence link X-Y can send either b1 or b2 in a given unit time thereby nodes w and z receive b1 and b2 in two consecutive cycles in traditional systems. But application of Network Coding has facilitated nodes w and z, receive b1 and b2 both. This significantly reduces the communication overhead. Network coding has been well studied at node level and network level. But problem again arises when there is a need to inter-operate with heterogeneous networks. Hence it would be an apt proposal to incorporate Network coding at the Infrastructure level of abstraction. Network coding suitable for Multi-cast Networks are Linear and Algebraic and for Non Multi-cast any non-linear codes can be chosen. This paper is not intended to discuss the various Network Coding implementations in detail as it will deviate from goal. Interested users can review the references given [13]. Having all these is mind the Infrastructure abstraction should adaptively choose appropriate Network coding relieving the user of the burden and confusion.

5. ANALYSIS OF THE PROPOSED MODEL: A CASE STUDY

GSN consists of several parts: a data acquisition module, a database module, a web-based query module and an external web services module. GSN is a stable model as it is continually being updated by an open source community. This paper tries to analyze the possible areas of improvement in the existing framework of GSN with a case study as illustrated below. We would like to present holistic algorithm which not only proves our proposal but also consolidates the whole model. The algorithm is given stepwise below:

1. Parts of GSN (data acquisition module, a database module, a web-based query module and an external web services module.)

i) Abstracts each WSN as Virtual Sensor (VS) to achieve Infrastructure abstraction.

ii) Each Virtual Sensor represented using XML for easy configuration and adoption for user needs

iii) Each VS takes many inputs (real or from other VS)

(1) Retrieval of data through Data level abstraction (SQL like languages) (part of N/W level abstraction)

(2) Meta-data to enhance the information quality of data (say Service discovery and Identification) from the nodes (dependent on Node level abstraction)

*(*iv) Drawbacks:

(1) Data format is described as part of Virtual sensors (vivid from sample XML below) which there by demands end-user applications to be aware of all possible structures available

1 Example: VS1 temperature as int; VS2 temperature as float; Problem: Burden on the end-user application to resolve.

i VS1: ...<virtual-sensorname=”temperature”priority =”10”><outputstructure><fieldname=TEMPERATURE”type =”int**”**/></output-structure>

ii VS2 :..<virtual-sensorname=”temperature” priority =”10”><outputstructure><fieldname=TEMPERATURE” type = ”double”/> </output-structure>

2 Solution: Data Format abstraction through Type casting approach can be made a global module at the System services level.

(2) It uses usual data aggregation methods like Averaging which are already available at Network abstraction level. More advanced techniques can be proposed

1 Example: Data aggregation of GSN …. <source alias=”source1” sampling rate=”1” storage size=”1”> <addresswrapper=”temperature”><predicatekey=”sampling rate”></predicate> </address> <query>select TEMPERATURE from wrapper</query> </source> <query> select avg**[**TEMPERATURE**]** from source1</query>...

2 Solution: To propose adaptive Network coding as part of the Data acquisition module of GSN. This module can be activated based on the needs of end-user application and once activated, appropriate coding scheme can also be selected accordingly if needed or switch back to default mode. This will improve the performance by reducing the communication cost much more than void it.

(3) There is also a possibility to incorporate the Network coding at Network level abstraction but the problem is again there will be discrepancy of various schemes which again travels back as burden on the end-user application. Hence it is more appropriate at the Infrastructure level of the Middle-ware.

6. CONCLUSION AND FUTURE WORK

Thus this paper aims at proposing an improved architecture for Infrastructure level Middle-ware like Global Sensor Network (GSN), a Middle-ware capable of interconnecting heterogeneous wireless sensor networks with the concept of Virtual Sensor abstraction. Though there are various Middle-wares available like Magnet, Mate, Cougar, SINA, DSWare, Impala, Milan, Envirotrack, GSN etc.. Very few qualify for heterogeneity along with GSN namely Mate, Magnet, and Envirotrack. But each one of them has more drawbacks than GSN. Magnet uses JVM (Java Virtual machine) which poses overhead at instruction level making not an appropriate candidate for WSN. Mate also uses VM (Virtual Machine) to achieve abstraction but brings in energy issues which make it suitable only towards sleepy applications and more details of application development missing. Envirotrack is worth considering Middle-ware for Data centric applications. Though it performs well for small scale networks more detail required towards self-organization and being autonomic. Hence the choice for our paper narrows down to GSN. And this paper attempts to address the minimal areas of improvement in GSN and proposes to solve using integration of adaptive Network coding at System services level of the Middle-ware architecture. But still there are so much research activity happening in the field of Adaptive Network coding. One such work is Adapcode++ which might be a favorable architecture to consider in future.

7. REFERENCES

[1] Berlin Erin Anzelmo, Alex Bassi, Dan Caprio, Sean Dodson, Rob van Kranenburg, Matt Ratto.:Discussion Paper on the Internet of Things commissioned by the Institute for Internet and Society, (2011)

[2] Joseph Davies.: Understanding IPv6 book

[3] Montenegro G, Kushalnagar N, Hui J, Culler D.: Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Internet Engineering Task Force, Request for Comments 4944, (2007).

[4] K. Aberer, M. Hauswirth, A. Salehi: Global Sensor Networks. Technical report LSIR-REPORT-2006-001, Lausanne, Switzerland (2006)

[5] Teemu Laukkarinen, Jukka Suhonen, and Marko H¨nnik¨ inen :A Survey of Wireless Sensor Network Abstraction for Application Development , Department of Computer Systems, Tampere University of Technology, Research Article, International Journal of Distributed Sensor Networks ,(2012)

[6] K. Aberer, M. Hauswirth, A. Salehi: A Middle-ware for fast and flexible sensor network Deployment. Proc. Of International Conference on Very Large Data Bases (VLDB), Seoul, Korea (2006)

[7] Miao-Miao Wang, Jian-Nong Cao2, Jing Li1, and Sajal K. Das: Middle-ware for wireless sensor networks: a survey : Journal of Computer Science and Technology 23(3): 305–32, (2008)

[8] Cecilia Mascolo, Stephen Hailes.: Survey of Middle-ware for networked embedded systems. Technical Report for Project: Reconfigurable Ubiquitous Networked Embedded Systems, University College London, (2005).

[9] Kay R¨omer , Oliver Kasten, Friedemann Mattern.: Middle- ware challenges for wireless sensor networks. Mobile Computing and Communications Review, (2002).

[10] Salem Hadim and Nader Mohamed : Middle-ware for Wireless Sensor Networks: A Survey : Department of Electrical and Computer Engineering , Stevens Institute of Technology , Hoboken, NJ 07030, USA , IEEE, (2006)

[11] Hnin Yu Shwe and Xiaohong Jiang : Efficient power savings in wireless sensor networks with network coding and overhearing avoidance :Electrical and Communication Engineering, Graduate School of Engineering, Tohoku University, 6-6-05 Aza-Aoba, Aramaki, Aoba-ku, Sendai, 980-8579 Japan., International Journal of Computer Engineering Research Vol. 2(2), pp. 34-44, (2011)

[12] Hnin Yu Shwe and Fumiyuki Adachi : Power Efficient Adaptive Network Coding in Wireless Sensor Networks , Department of Electrical and Communication Engineering Graduate School of Engineering, Tohoku University , Sendai, Japan, IEEE ICC proceedings (2011)

[13] Takahiro MATSUDA, Taku NOGUCHI, and Tetsuya TAKINE: Survey of Network Coding and Its Applications :INVITED SURVEY PAPER ,IEICE TRANS. COMMUN., VOL.E94–B, NO.3 ,(2011)

[14] Luca Mainetti, Luigi Patrono, and Antonio Vilei: Evolution of Wireless Sensor Networks towards the Internet of Things: a Survey: 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), (2011)