IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 9, Issue 5, Ver. V (Sep - Oct. 2014), PP 01-09
www.iosrjournals.org

Tactical Addressing in Autonomic Network Architecture for Wireless Sensor Networks: A New Paradigm

Sanjay K N[1], Shaila K[2], Venugopal K R[3]and L M Patnaik[4]
[1]*Asst. Prof., Dept. of ECE, Vivekananda Institute of Technology, Bangalore, India*
[2]*Professor and Head, Dept. of ECE, Vivekananda Institute of Technology, Bangalore, India*
[3]*Principal, University Visveswaraya College of Engineering, Bangalore, India*
[4]*Honorary Professor, Indian Institute of Science, Bangalore, India*

Abstract: *Wireless Sensor Networks has complex and time consuming configuration process. This involves the availability of skilled engineers with good networking knowledge whenever process is performed. One of the sub functionality is to configure the address of the interface. Before routing is enabled verify if the connectivity is established between the two nodes. The key requirement of tactical networks is quick recognition of nodes and fast movement of information. This configuration poses challenge to meet the requirement of tactical networks. We have developed a model that securely automates the process of configuration with minimum steps and assuring security by providing authentication in order to reduce the number of skilled engineers in handling the process in various levels. The outcome is the reduction in the total setup time by ¼ of the actual expected time. The software for enabling TCL scripting is developed which aids in securing configuration process and enable the Wireless Sensor Network nodes to interact with more feasibility.*

Keywords: *Autonomic Network Architecture, Sensor, Tactical Address Assignment, Wireless Sensor Network, Zebra*

I. Introduction

The central premises of sensor networks are the distributed collection and digitization of data from a physical space, providing an interface between the physical and digital domains. Sensor networks consist of a potentially large number of sensor modules that integrate memory, communication, processing and sensing capabilities. The sensor modules self-organise themselves to form a network in order to share the collected physical data and to provide this data to the network user or operator. Sensor networks have a wide range of applications, including medical, environmental, military, industrial and commercial applications [1].

Routers have to follow complex and time consuming configuration process that involves availability of skilled engineers with good networking knowledge every time the process is performed. This type of tedious configuration process poses challenge to meet the requirement of tactical networks [2-6]. The key requirement of tactical networks is quick recognition of nodes and fast movement of information. The high mobility of tactical forces results in rapid changes to the network topology [7]. Network topology typically changes when networked nodes appear online and offline intermittently. Depending on physical location, the nodes may appear offline if they go beyond the effective communication range [8].

The connectivity changes when one node disconnects from its current network and connects to another neighboring network due to energy restrictions and sudden changes in node status (e.g., failure) cause frequent and unpredictable topological changes. In such situations, network connectivity has to be reconfigured by self-organizing networks and the sent data are routed to the destination node even if the networked node switches to connect the neighboring network. At present, the connectivity of the network is done by the system administrator manually. The time required to perform this action is more and is prone to security breaches. The solution is to simplify and automate process of configuration into fewer steps, minimizing the requirement of skilled engineers to handle the process till various levels [9-10]. The process can be made more secure by developing external hardware authentication token that secures configuration process. This involves development of software based on Shell scripting and C programming.

1.1 Motivation

The military groups scattered around an area keep changing their positions. Every time a group shifts its node, link and position changes hence, completion of the process takes more time. To establish a network communication between systems in each group is established by providing IP address and links. The network engineer will be aware of all the credentials involved in the network so that he can deploy and configure the network [11-14]. This poses a security threat, as network administrator can be an internal adversary. In order to overcome these problems and reduce the time consumption the process has to be automated.

1.2 Contribution

Communication media models operating templates and data sending profiles. It is setup (i) to verify network design principles and theories (ii) to assess the data traffic loading effects and (iii) effective bandwidth utilization. Simulation is necessary for large-scale network design and validation as the actual network set-up in the operating environment is impractical and resource-intensive.

In this paper, we have developed and implemented an Automated IP Address Assignment for tactical networks. Autonomic networking enables the autonomous formation and parameterization of nodes and networks by letting protocols sense, so as to adapt to the networking environment at run time. Besides its dynamic aspects, a core requirement of autonomic networking is to define a structured framework and execution environment that enables algorithms to operate in a continuously changing environment [15-16].

1.3 Organization

In Section 2, various research works related to Tactical addressing and autonomic network architecture is discussed. Background work is discussed in Section 3. In Section 4, Problem definition is stated for the automation of self-addressing. Wireless Sensor Nodes within the network leads to adaptation of network and automates address assignment with TAA model is developed in Section 5. Algorithm and Implementation with Automating the networking process are explained in Section 6. The performance evaluation with respect to setup time for the network, need for skilled engineer and its security are elaborated in Section 7. The conclusions of the paper are discussed in Section 8.

II. Related Work

Ghazi Bouabene et al., [1] propose the Autonomic Network Architecture (ANA). ANA explores the novel ways of organizing and using networks beyond legacy Internet technology. It aims at designing and developing a network architecture that can demonstrate the feasibility and discusses the properties of autonomic networking. Clark et al., [2] discuss the main guiding principle behind the architectural work in ANA. The design at all levels of the architecture is performed in order to maximize the degree of flexibility to support functional scaling.

In Functional scaling a network is able to extend both horizontally (adding more functionality) as well as vertically (different ways of integrating abundant functionality) which is one of the advantage for the networks. Jennings et al., [3] develops a network architecture that can assemble itself, when a high level instructions is used. It reassemble itself as requirement changes, automatically discovers when something goes wrong and automatically fix a detected problem or explain why it cannot do the functional scaling.

In contrast to the available frameworks, ANA introduces an abstraction level and machinery for manipulating communication entities in a generic manner at the system level as described by Cheng et al., [4]. Kephart et al., [5] proposed that ANA could be used to develop management systems as it provides the abstraction layer and flexible machinery upon which such advanced autonomic services could operate.

Cerpa et al., [6] describes overall, ANA as a meta-architecture since it is a framework to host, interconnect and federate multiple heterogeneous network instances. But unlike the Internet, which relies on a unique and globally shared addressing scheme, ANA is not another one-size-fits-all network waist. According to Steere et al., [7] ANA differentiates itself by merging the concepts of packet forwarding with protocol binding. It uses a single forwarding table where a unique label that identifies the next operation (e.g. send packet to either output interface or next functional block) to be performed.

Lai et al., [8] describes Zebra as a routing software package that provides TCP/IP based routing services with routing protocols support such as RIP, OSPF and BGP. Estrin et al., [9][10] explains about Smaller feature size in chips. These chips have driven down the power consumption of the basic components of a sensor node to a level that the construction of WSNs can be contemplated. Chandrakasan et al., [11] defines that there is certainly not a single standard available nor would such a standard necessarily be able to support all application types. Cepra et al., [12] describes the goals of the knowledge plane seem pretty similar to autonomic networking. However, this position paper has remained a visionary exercise and has not materialized into a prototype.

Burrell et al., [13] proposed that building an Wireless Sensor Networks has only become possible with some fundamental advances in enabling technologies. First and foremost among these technologies is the miniaturization of hardware. This is particularly relevant to microcontrollers and memory chips as such, the radio modems, responsible for wireless communication, have become much more energy efficient. Boriello et al., [14] describes that choosing the hardware components for a wireless sensor node, is specific with regard to size, costs and energy consumption of the nodes – communication and computation facilities that are considered to be an acceptable quality, but the trade-offs between features and costs is crucial.

Burell et al., [15] and Asada et al., [16] presents that more realistic applications, the mere size of a node is not so important; rather, convenience, simple power supply and cost. Akyildz et al., [17] describes the

minimization of power and routing with efficient algorithms for WSN. Banerjee et al., [18] proposed the clustering scheme in which the efficient optimization of nodes with their parent node in the specific range is achieved.

Hill et al., [19] proposes the various embedding network technologies in WSN and routing the available WSN algorithms with the address agnostic. For example, ANA defines a set of primitives and related methods which are address agnostic and hence do not impose a common address type and format as described in Kahn et al., [20]. Mainwairing et al., [21] proposes the health monitoring of the nodes in the WSN with less power consumption and more energy efficient in corresponding with different routing networks.

Potie et al., [22] mentioned the use of different architectures for defining the wireless sensor nodes with change in MAC layer design and use of Hoc Ultra network for low power consumption of the nodes as described in Rabaey et al., [23]. Srivastava et al., [24] described the efficient use of the sensor based clusters and the change in medium access control layer for the autonomic networks with different routing in Linux. Steere et al., [25] proposed the deployment of the WSN for N nodes with energy minimization and power consumption.

Szewczyk et al., [26] proposes the Switching algorithms which are relatively simple; it is the same for most routing protocols. In most cases, a host determines that it must send a packet to another host. After acquiring a router's address by some means, the source host sends a packet address specifically to a router's physical (Media Access Control (MAC)-layer) address, with the protocol (network layer) address of the destination host as proposed by Zhao et al., [27].

Berrou et al., [28] describes the packet's destination protocol address. Here, the router determines that it either knows or does not know how to forward the packet to the next hop. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, it changes the destination physical address to that of the next hop and transmits the packet. Jung et al., [29] describes the distributed data compression where in the distributed nodes are held with protocol with less unevenness and Abdelhakim et al., [30] describes the architecture for the protocol with less unevenness and more efficiency with setup time of the network.

III. Background Work

A. Routing in Network

Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the observer. The primary difference between the two is that bridging occurs at Layer 2 (the link layer) of the OSI reference model, whereas routing occurs at Layer 3 (the network layer). This distinction provides routing and bridging with different information to use in the process of moving information from source to destination, so the two functions accomplish their tasks in different ways.

Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internetwork. In the context of the routing process, the latter of these is referred to as packet switching. Although packet switching is relatively straight forward, path determination can be very complex.

B. Routing in Linux

Zebra support special BGP Route Reflector and Route Server behavior. In addition to traditional IPv4 routing protocols, Zebra also supports IPv6 routing protocols. With SNMP daemon which supports SMUX protocol, Zebra provides routing protocol management information bases [8].

Zebra uses advanced software architecture to provide high quality, multi-server routing engine. Zebra has an interactive user interface for each routing protocol and supports common client commands. Due to this design, new protocol daemons can be easily added. Zebra library can also be used as a program's client user interface. Zebra is distributed under the GNU General Public License.

Quagga daemons are each configurable via a network accessible CLI (called a 'vty'). The CLI follows a style similar to that of other routing software. There is an additional tool included with Quagga called vtysh, which acts as a single cohesive front-end to all the daemons, allowing one to administrator nearly all aspects of the various Quagga daemons in one place [16].

IV. Problem Definition

Consider a given Wireless Sensor Network consisting of 'x' networks with N nodes. Communication with each group of the network is established by the IP address. The network administrator deploying the network will be aware of all these details. There is a possibility of the network administrator being an adversary at any point of time. The main objective of this work is to automate the network address assignment in tactical networks.

Assumption: The assumptions of a tactical communication networks are: (i) Narrow effective Communication range (ii) Potentially hidden nodes (iii) Mobility of tactical nodes rearranges the network topology on an adhoc basis.

V. Tactical Address Assignment Architecture

The tactical communication network can be designed to achieve effective data communication and exchange. There is a need for a structured approach that comes with a clear understanding of the operating environment characteristics, communication media limitations and application data exchange profiles. These form the basic tactical network design considerations to define the relevant network parameters and values. The tactical network design is then verified through simulation. The key challenge is in addressing tactical communication network constraints such as limited bandwidth and latency, intermittent communication links and potentially hidden nodes in certain operational terrains. In addition, the mobility of tactical nodes causes rapid changes in network topology with the nodes leaving and joining the network on an ad hoc basis.

The automate network address assignment in tactical network involves solution provider to develop a software module for WSN router choosing an appropriate deployment platform. This involves the design and development of software/daemon that securely gets various simplified inputs from the network operators, process and translates these inputs; perform intelligent network queries [17]. After gathering sufficient network information it performs analytics to automate the complex WSN address assignment on the routers and nodes it has been deployed.

The requirement includes the system level and functional requirements of the Tactical Address Assignment Daemon. The system considered is a Pentium Processor of 1.6GHz or higher and minimum 512 RAM and 50MB available Hard disk space and also one or more Ethernet ports. The functional level is the LINUX GNU platform with Pentium builder and pre-installed with Quagga and Zebra routing daemons to achieve flexible shell scripting.

The block diagram of TAA architecture is shown in Figure 1 and consists of different modules. Man Machine Interface is the space where interaction between humans and machines occurs. The goal of this interaction is effective operation and control of the machine on the user's end and feedback from the machine, which aids the operator in making operational decisions [18]. The user interface includes hardware (physical) and software (logical) components. User interfaces of various systems are available and provide a means of input, allowing the users to manipulate a system and output, allowing the system to indicate the effects of the users manipulation.

The user provides the input by typing a command string with the computer keyboard and the system provides output by printing text on the computer monitor. Used by programmers and system administrators in engineering and scientific environment and by technically advanced personal computer users [19, 20]. Command line interface links all other modules.

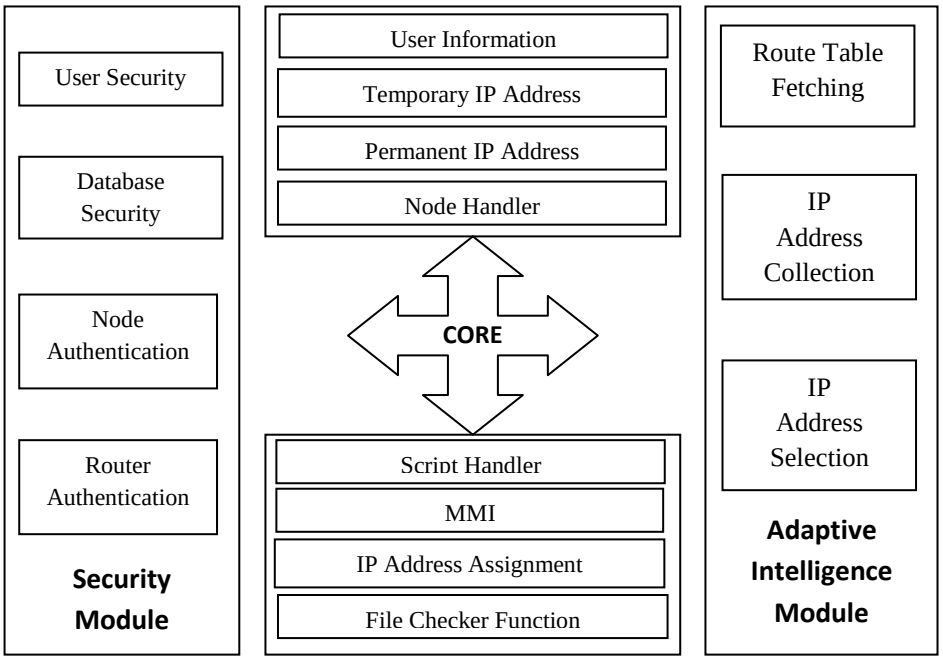

Figure 1: Block Diagram of TAA Architecture

A database is an organized collection of data that are typically organized to model relevant aspects of reality. It should be arranged such that it supports processes requiring this information. They are created using simple text file and few of the different databases like Username and Password, Node number and link number, Temporary IP Database, Permanent IP Database. The Node and link information is stored in the database module for further processing by the core and the Automation block processes the requisite data in the script handler and the multimedia interface allows the core to display the message with the corresponding address assignment stored in the IP address assignment. The Automation block checks the required data packets with File checker function if data packets are unavailable an error will be reported through MMI to the core. The adaptive intelligence module sets the necessary routing to the route table. If required they can be fetched for the selection of the IP address in IP address selection. The selected address are collected from the core in IP address collection block. With the security module the user security and database security provides the necessary authentication for the node and router respectively with necessary lookup data stored in node and router authentication blocks [21, 22].

VI. Algorithm and Implementation

Automating the networking process makes deployment faster and it is not essential for the soldier at the node to have the knowledge of networking. The software used to automate the process takes care of connecting the systems in the network by providing IP address (by itself after appropriate authentication process) depending on the node position and link through which it has to be connected. The software language used for automating the networking process is scripting language.

In this, a prototype of Tactical Address Assignment Daemon was implemented. The aspect of the design is to implement software that gives an advantage to the user. If the end user is not a skilled network engineer, he can configure and communicate with the desired node using this daemon [23]. If the user is interested in saving time during configuration of the network components he can implement using software daemon. Algorithm for the establishment of automated network assignment is shown in Table I.

Table I: Tactical ANA Algorithm

Step 1:	Load the program window which contains the title and credits.
Step 2:	Verify the complete availability of modules and files. Then, display the result.
Step 3:	If verification is successful in step 1 and step 2, continue the task else display error message.
Step 4:	Login with the defined Username and Password. Verify username and password using database security module. If verification is successful login access is provided else display error message.
Step 5:	Authentication decides whether the user is client or admin. If Admin user then he can access the databases, add/delete users from the database, change the users passwords and modify the modules to specific requirements. If client then requests to establish a physical link. Then enter node number and link number.
Step 6:	Check the validation of node and link number and display the result as in Step 5. If it is not available, display error message.
Step 7:	During Script execution, Ping the router, Telnet to router and Assign temporary IP. If successful ensure the program is running on the other end via socket, then Ping all available subnets and Check whether the node is remote or backbone node.
Step 8:	Check for the availability of routes. Exchange the routing tables on both the sides via socket. Select and assign appropriate Permanent IP from the database. Once operation is completed update the database and the Routing tables.

The modular design of the daemon allows for addition of new types of hardware and the user of other network technologies. The use of C / C ++ programming language also makes the daemon portable to other flavors of LINUX platforms. In light of the requirement specifications, the design and implementation of the daemon provides the necessary functionality to make it operable according to intentions [14]. A thorough evaluation of the functionality largely depends on subjective criteria.

A scripting language or script language is a programming language that supports the writing of scripts. The programs are written for a special runtime environment that can interpret and automate the execution of tasks which could alternatively be executed one-by-one by a human operator [12]. Environments that can be automated through scripting include software applications, web pages within a web browser, the shells of operating systems (OS), several general purpose and domain-specific languages used for embedded systems. Router configuration is performed using shell script to Ping the router, Telnet to router and assign temporary IP. If successful, ensure the program is running on the other end via socket. Then, Ping all available subnets, and check whether the node is remote or backbone node. Check the validation/correctness of all the files of each module. If any files are missing then display file check error.

VII. Performance Evaluation

The performance of the TAA Daemon can be measured by investigating the quantitative deviations introduced by the tactical network as it is not well defined standard network. The location connection, attributes and operators will be constantly changing.

The test description, the expected output and the actual outcome is tabulated in Table II.

Setup Time: The time required to deploy a base and arrange a communicating node is twenty to thirty minutes and this daemon helps in reducing the time required for setup. The time required for manual configuration of the network and assigning IP addresses to the systems between two distinct nodes is 36 minutes, whereas the time taken using the daemon is 4 minutes 25 seconds. This difference shows that there is effective reduction in the usage of time for configuring the network.
Skilled engineer: It is not required for the person present at the node to be a skilled engineer for configuring the node. Whereas earlier works it involves manual assignment of IP addresses and configuration of routers.
Security: Using this daemon, the need for security issues are overcome since the tactical network is automated [24-30].

Table II: Results

Test No	Test Description	Output Expected	Actual
1	Start the TAA Daemon with default configuration	No Errors and all modules should be loaded.	Successfully loaded all the modules
2	Start TAA with default Configuration and without appropriate files	Error : TAA CLI to output missing files	Error : TAA CLI to output missing files
3	Enter the login details for authorization of the user	User is authorized and distinguished whether admin or local user	User authorized successfully and the corresponding menu depending on the user type are displayed
4	login details for authorization of the user is incorrect	Provide a threshold of 3 times to enter the valid credentials for login then terminate the daemon	User authorization is terminated after threshold of 3 times and the daemon is terminated
5	Check for physical link establishment by the user	If yes proceed further or else display message to connect and terminate the program	If yes the daemon proceeds to the next step if no displays the message and terminates the program
6	Enter the Node and desired Link to be used to connect	if valid the details of the node and link should be displayed	The details of the node and link are displayed
7	Enter the Node and desired Link to be used to connect is incorrect	Error message must be displayed saying that the node and link details are invalid	Error message displayed
8	Temporary IP address assignment function is loaded to assign temporary IP's from the database appropriately	Random Temporary IP should be assigned from the database checking whether the IP is not assigned in order to prevent IP clashes in the network	Random Temporary IP generated by the function from the database and assigned
9	Router configuration function is loaded	Identify the router present at the node and configure the router	Router identified and configured
10	Router configuration function is loaded and if router is not identified	Error message must be displayed saying that the Router identification failed	Router identification fail and configured

Figure 2 to Figure 6 shows the screen shots related with the different testing conditions and their error messages. Figure 2 shows the node number and link number details and displays the information about node and link, once the link is established. Later, physical link clarification is done, then the daemon proceeds further and ask to enter node and link number. Once, a valid node number and link number are given, the information about node and link selected are displayed. Next, the temporary IP address is assigned that is selected randomly from the database as shown in Figure 3. Figure 4 shows how the program entering into router configuration module requests the user whether to gain the root access and then assigns the selected temporary IP. The configuration of routing with the valid IP address and the lookup table for routing is done as shown in Figure 5. In case the node and link number are entered incorrect then the error message will be displayed as shown in Figure 6.

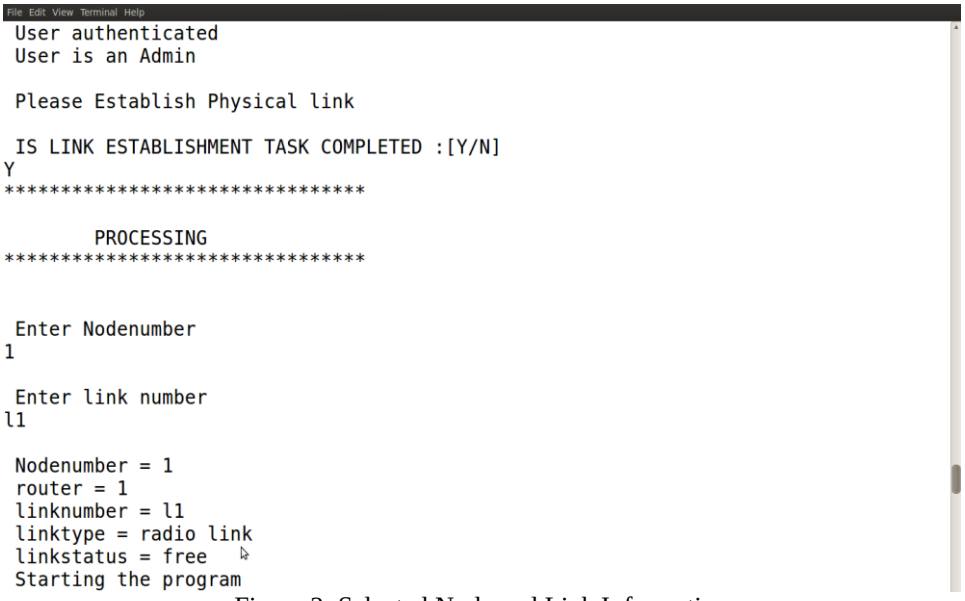

Figure 2: Selected Node and Link Information

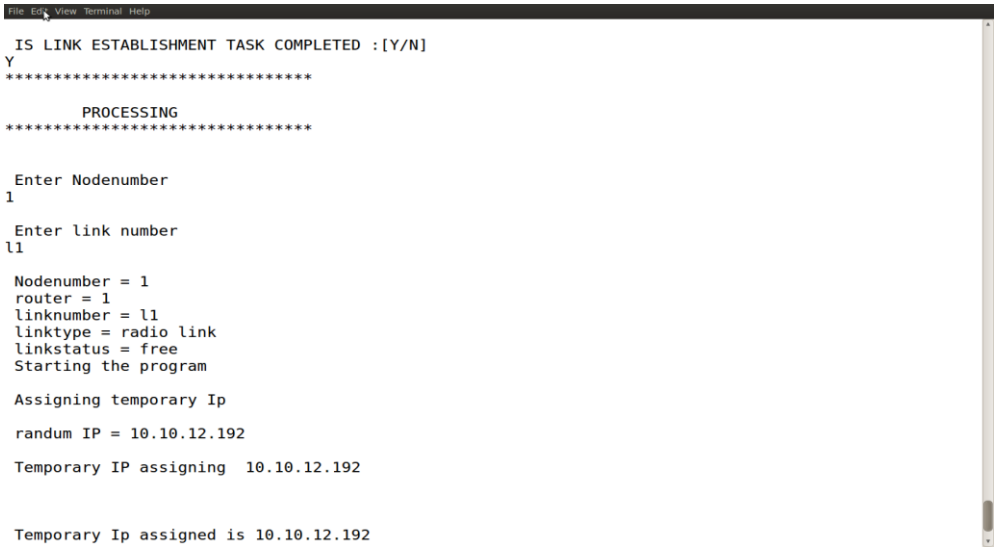

Figure 3: Temporary IP Assignment

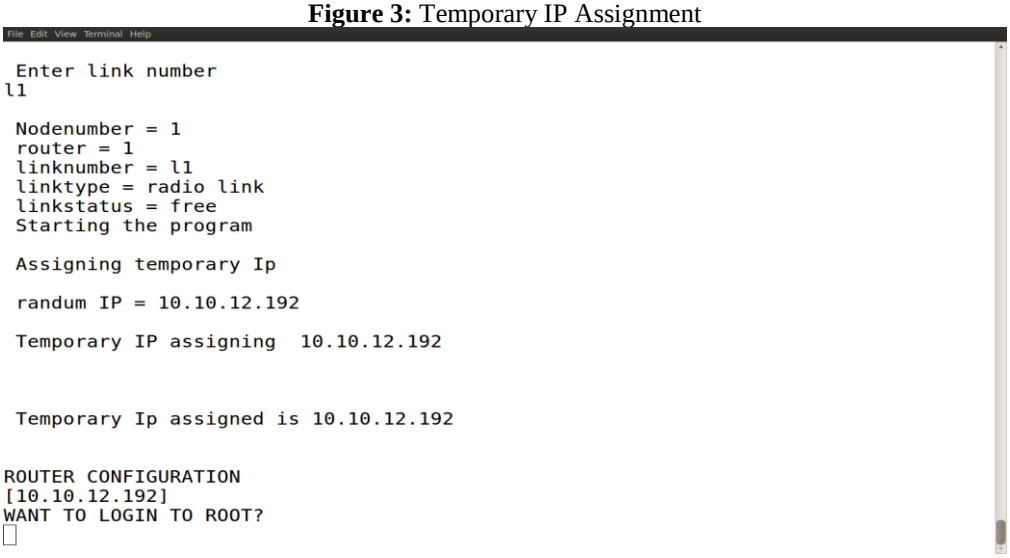

Figure 4: Routing Configuration Module with Temporary IP

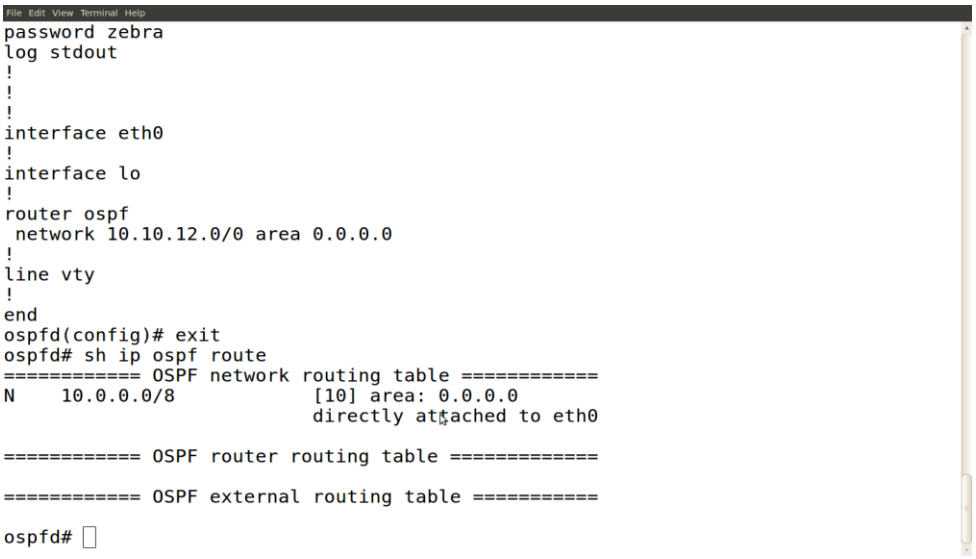

Figure 5: Configuration of Router and Displays the Routing Tables.

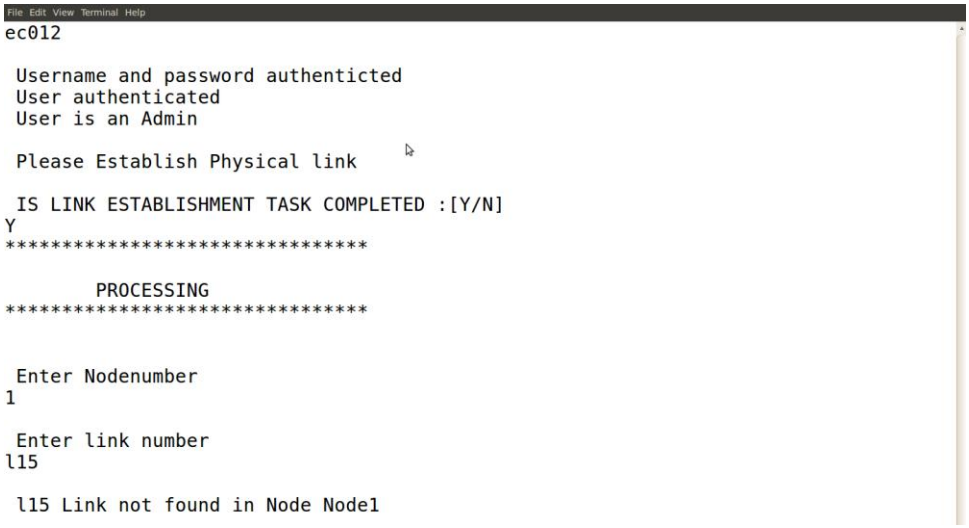

Figure 6: Error message for Invalid Link and Node Assignment

VIII. Conclusions

The design of a fully functional TAA automatically stabilizes the address assignment reducing human error and complex tasks. With design of secure login process, address assignment process results in flexible database enabling implementation of additional modules.

The modular and flexible design allows making use of third party application for TAA processing, establishing framework for easy addition of software module, protocols and solutions are used as optimized approach for migration of other routers like CISCO, JUNIPER etc. and existing router infrastructure with minimal changes in architecture. There are different routing algorithms in WSN which lags the requirement of energy efficiency and power throughput for each and every nodes present in the network. This is overcome in our autonomic address assignment by 4 minutes 25 seconds. In future, we are planning to embed the autonomic networking with increased number of nodes and with different carrier in the network.

References

[1] Ghazi Bouabene, Christophe Jelger, Christian Tschudin, Stefan Schmid, Ariane Keller and Martin May, The Autonomic Network Architecture (ANA), in IEEE Journal On Selected Areas in Communications, vol. 28, no. 1, pp. 33-39, January 2010.

[2] D Clark, C Partridge, J Ramming and J Wroclawski, A Knowledge Plane for the Internet, in Proceeding of ACM SIGCOMM 2003, pp. 3-10, August 2003, Karlsruhe, Germany.

[3] B Jennings, S Van der Meer, S Balasubramaniam, D Botvich, M Foghlu, W Donnelly and J Strassner, Towards Autonomic Management of Communications Networks, IEEE Communication Magazine, pp. 112– 121, October 2007.

[4] Y Cheng, R Farha, M S Kim, A Leon-Garcia and J W K Hong, A Generic Architecture for Autonomic Service and Network Management, Computer Communications, vol. 29, no. 18, pp. 3691–3709, November 2006.

[5] Kephart and D Chess, The Vision of Autonomic Computing, IEEE Computer, vol. 36, no. 1, pp. 41–50, January 2003.

[6] FIND–Future Internet Design, http://www.nsf.gov/pubs/2006/nsf06516/nsf06516.html.
[7] Autonomic Network Architecture, http://www.ana-project.org/
[8] Lai Ying Cheung, Chia Wan Yin, Designing Tactical Networks – Perspectives from a Practitioner,http://www.dsta.gov.sg/docs/publicationsdocuments/designing-tacticalnetworks---perspectives from-a practitioner.pdf?sfvrsn=
[9] D Estrin, L Girod, G Pottie and M Srivastava, Instrumenting the World with Wireless Sensor Networks, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing(ICASSP 2001), Salt Lake City, UT-99, pp. 210-214, May 2001.
[10] D Estrin, R Govindan, J Heidemann and S Kumar, Next Century Challenges: Scalable Coordination in Sensor Networks, in Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks (MobiCom 1999), pp. 312-319, Seattle, Washington, DC, 1999.
[11] A P Chandrakasan, R Min, M Bhardwaj, S H Cho and A Wang, Power Aware Wireless Microsensor Systems, in Proceedings of the ESSCIRC 2002, pp. 345-360, Florence, Italy, September 2002.
[12] A Cerpa, J Elson, D Estrin, L Girod, M Hamilton and J Zhao, Habitat Monitoring: Application Driver for Wireless Communications Technology, in Proceedings of the ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, San Jose, Costa Rica, pp. 321-326, 2001.
[13] J Burrell, T Brooke and R Beckwith, Vineyard Computing: Sensor Networks in Agricultural Production, in IEEE Pervasive Computing Journal, vol. 12, no. 11, pp. 3–7, 2004.
[14] G Boriello and R Want, Embedded Computation Meets the World Wide Web, in Proceedings of Communications of the ACM, pp. 43-47, 2000.
[15] P Bonnet, J E Gehrke and P Seshadri, Querying the Physical World, IEEE Personal Communications, pp. 7–11, 2000.
[16] G Asada, M Dong, T S Lin, F Newberg, G Pottie and W J Kaiser, Wireless Integrated Network Sensors: Low Power Systems on a chip, in Proceedings of the 1998 European Solid State Circuits Conference, The Hague, Netherlands, pp. 98-104, 1998.
[17] I F Akyildiz, W Su, Y Sankasubramaniam and E Cayirci, Wireless Sensor Networks: A Survey, IEEE Transactions on Computer Networks, 38, pp. 393-422, 2002.
[18] S Bannerjee and S Khuller, A Clustering Scheme for Hierarchical Control in Wireless Networks, in Proceedings of the IEEE INFOCOM 2001, Anchorage, AK, pp. 391-396, April 2001.
[19] J Hill and D Culler, MICA: A Wireless Platform for Deeply Embedded Networks, IEEE Journal on Micro, pp. 22–27, 2002.
[20] J M Kahn, R H Katz, and K S J Pister, Next Century Challenges: Mobile Networking for "SmartDust", in Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom 99), Seattle, WA, pp. 99-105, August 1999.
[21] A Mainwaring, J Polastre, R Szewczyk, D Culler and J Anderson, Wireless Sensor Networks for Habitat Monitoring, in Proceedings of the 1st ACM Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, pp. 250-254, September 2002.
[22] G J Pottie and W J Kaiser, Embedding the Internet: Wireless Integrated Network Sensors, in Proceedings of Communications of the ACM, pp. 43-47, 2000.
[23] J M Rabaey, M J Ammer, J L Da Silva, D Patel and S Roundy, PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking, Journal on Computer, pp. 33–39, 2000.
[24] M Srivastava, R Muntz and M Potkonjak. Smart Kindergarten: Sensor-based Wireless Networks for Smart Developmental Problem-Solving Environments (Challenge Paper), Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, ACM Press, Rome, Italy, pp. 132–138, July 2001.
[25] D C Steere, A Baptista, D McNamee, C Pu and J Walpole, Research Challenges in Environmental Observation and Forecasting Systems, In Proceedings of the 6th International Conference on Mobile Computing and Networking (ACM Mobicom), Boston, MA, pp. 67-72, 2000.
[26] R Szewczyk, E Osterweil, J Polastre, M Hamilton, A Mainwaring and D Estrin, Habitat Monitoring with Sensor Networks, in Proceedings of Communication of the ACM, pp. 47–52, 2004.
[27] F Zhao, J Shin and J Reich, Information-Driven Dynamic Sensor Collaboration for Tracking Applications, IEEE Signal Processing Magazine, pp. 19–23, 2002.
[28] C Berrou and A Glavieux, Near Optimum Error Correcting Coding and Decoding: Turbo-Codes, IEEE Transactions on Communications, vol. 3, no.1, pp. 44–53, 1996.
[29] E S Jung and N H Vaidya, A Power Control MAC Protocol for Ad Hoc Networks, in Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking 2002 (MobiCom), Atlanta, Georgia, vol. 5, no. 2, pp. 212-217, September 2002.
[30] Abdelhakim M, Lightfoot LE, Jian Ren, Tongtong Li, Architecture Design of Mobile Access Coordinated Wireless Sensor Networks, in Proceedings 2013 IEEE International Conference on Communications (ICC), pp. 172 – 177, 2013.