# Intrusion Detection and Secured Data Transmission using Software Hardware Codesign

S.Arul
Research Scholar
Dr.M.G.R. Educational and Research Institute, University, Chennai -95, Tamilnadu, India

S.Ravi, PhD.
Prof. and Head, ECE Dept.,
Dr.M.G.R. Educational and Research Institute, University, Chennai -95.

Sangappa S B
AP/ECE Dept.,
K.S. Institute of Tech.
Bangalore.

## ABSTRACT
Dependability of the nodes in a group network is very important for its successful applications in the engineering area. Conventionally, when a node has a failure, it (i.e. data from that node) is usually discarded and the network is reorganized with faultless nodes to continue with the normal operation without a tradeoff with the functional coverage of the networks. In this paper, it is planned that the sensor nodes designed with self-healing ability can dynamically change their node configurations to repair during hardware failures. The work once integrated with an existing standalone target group nodes or Host/Target group communicating nodes can improve the robustness of the group network and reduce the maintenance cost when deployed in real time applications.

## Keywords
Intrusion Detection, E-Mote System, Data Encryption Standard (DES)

## 1. INTRODUCTION
In this work, intrusion detection is focused via, software hardware co-design. Secured transmission insists schemes to bring into play the various technical aspect of cryptography in the transmission process. Apart from the concept too is also made use of the existing secured transmission nodes travel either toward the advancement, logics, techniques and algorithms of the software issues or towards the assuring aspects of the hardware but, proceeding towards security with the usage of the software hardware in advance makes the challenge here. Hence, data could be protected, via the software hardware co-design from the intruders.

## 1.1 Quality of Good Network
To meet the organization's information needs, a network must have the following qualities[2]

(i) Network should work together and operations should be transparent to users
(ii) They must provide remote access
(iii) They must maintain peak performance

While meeting all these demands for open access, a network must also meet the organization's security needs. The important ones are:

(i) Confidentiality
(ii) Reliability
(iii) Integrity

## 1.2 Objectives of this work
The objectives of this work include intertwining the worthy users amongst the eavesdroppers, to achieve an efficient and less complex cryptography scheme and to transmit the data in a secured fashion. The option to edit the list of valid host is also made available.
The Seven types of Network attacks[5] considered are

(i) Denial-of-service (DoS)
(ii) Information leakage
(iii) Regular file access
(iv) Misinformation
(v) Special file / database access
(vi) Remote arbitrary code executing
(vii) Elevation of Privileges

## 1.3 Benefits of the Research Work
(i) Assignment of functionality to the hardware/software domains can be performed using an ad-hoc approach based on the designer's experience.
(ii) The main problems solved are hardware and software synthesis co-simulation and interface generation.
(iii) It optimizes both the cost and the overall performance.
(iv) Based on the Decoupled information flow tracking, control/ data dependencies among information objects in network application are reduced. Also, it helps to associate application specific tags with input data and invokes the application specific processing on output data according to their tag values.
(v) Under the dynamic information flow, host address tracking based mobility analysis and host name tracking based live node detection is done. The individual nodes communicate with the server in collision free mode.

## 1.4 Proposed Methodology
Protection = Prevention + (Detection + Response)

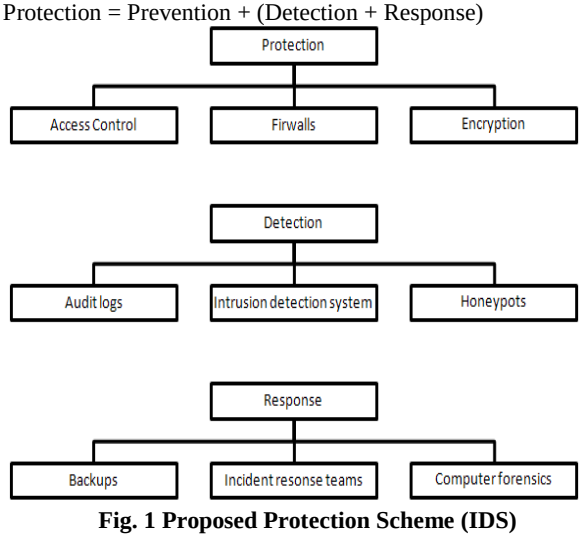



**Fig. 1 Proposed Protection Scheme (IDS)**

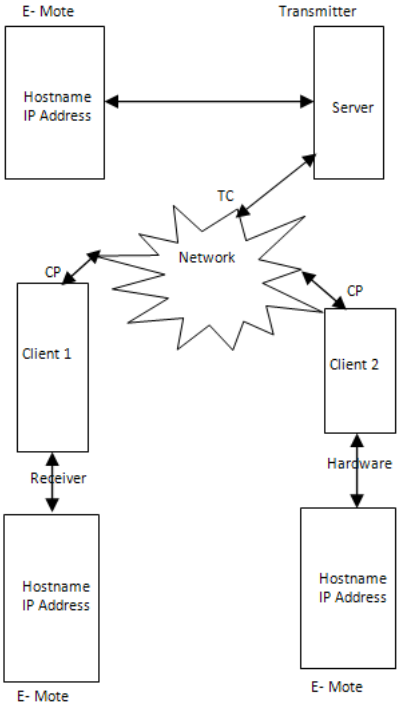
## 2. PREVIOUS WORK

A security approach for external memory in FPGA-based embedded systems that exploits FPGA configurability is presented. FPGA-based security core provides both confidentiality and integrity for data stored externally to an FPGA which is accessed by a processor on the FPGA chip. Each application requires a collection of tasks with varying memory security requirements. Security core is used in conjunction with a NIOS II soft processor running the Micro C/OS II operating system.

The offline authentication scheme is for IP modules. This scheme implements mutual authentication of the IP modules and the hardware platform, and enables us to provide authentication and integrity assurances to both the system developer and IP provider. This scheme requires a symmetric cipher and a Physically Unclonable Function (PUF).

## 3. ARCHITECTURE FOR INTRUSION DETECTION

The architecture for intrusion detection includes server, client, embedded mote, transmission medium and cryptographic algorithm. A particular system is the network, say, client, request for a confidential data. The server which already possesses a list of hostname validates the hostname and the ip-address of the client. It the server contains the details of the particular client and then the corresponding node is said to be trustworthy user. Once the client is identified to be a valid user, then the communication could be made and the data could be transferred. The data is not sent in the raw form, but encrypted using the data encryption standard [DES], is transmitted through the channel using the regulations of the protocol suite. The key used to encrypt the pain text too is transmitted access, via, Wi-Fi. The hostname and the IP Address of the genuine users are stored in a chip, called embedded mote. The embedded mote, heritages, referred as E-Mote is a hardware chip connected to each of the systems in the network.




DES = Encrypted Data via Channel

Note:

TC = Text → DES → Cipher Text

CP = Cipher Text → DES → Plain Text

**Fig. 2 Architecture of E-mote system**

## 4. HARDWARE CHIP FOR USER AUTHENTICATION

The Hardware Chip present along with the server in the transmission side possesses the hostname and the IP Address of the trustworthy clients. When a client demands for a data, the server sends a request via the Hardware E-mote, which checks for the validation of the particular IP Address. If the corresponding IP Address is present in the list of available IP Address in the e-mote then the corresponding hostname is said to be a trustworthy node and is guaranteed to initiate transmission. If the corresponding IP address is not available with the list of IP address present in the server e-mote, then that particular node is suspended to be an untrustworthy node. Then a key is made for its certificate. If the client sends a valid certificate, then the corresponding IP address is a validated and is decided to be a trustworthy user. Hence, the IP address and hostname of that particular node is also added as a trustworthy user. If the certificate is not received/received certificate is identified to be fake, then the corresponding node is identified to be an untrustworthy user and so initialization of communication is made.

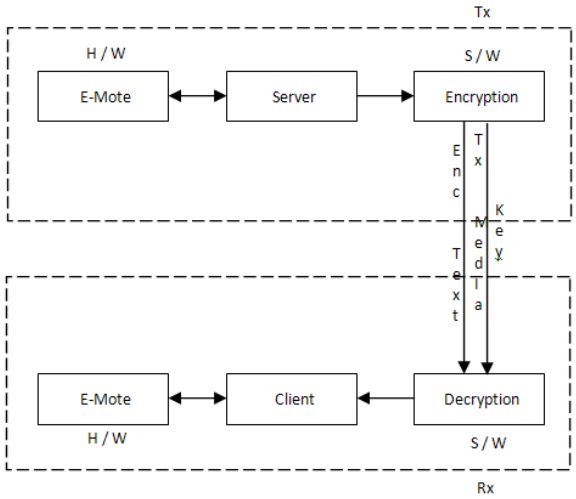



**Fig. 3 Block diagram for Hardware based authentication**

## 5. FIRST AND SECOND LEVEL SECURITY CONTROL

Table 1 illustrates how first level security works. It starts with a list of standard security techniques that represent the element of first level security

**Table 1. List of first level security controls**

| Administrator computer access control | Procedural / Technical |
|---|---|
| Backup data files and programs | Procedural / technical |
| Comply with laws and regulations | Procedural |
| Allow for contingency recovery-equipment replacement | Disaster Control |
| Create a disaster recovery plan | Disaster Control |
| Encrypt the password file * | Technical |
| Establish computer security management committee | Procedural |
| Establish passwords for networks | Procedural / Technical |

| | |
|---|---|
| access * | and physical |
| Isolate sensitive production jobs | Procedural / Technical |
| Log user trouble cells | Procedural / Technical |
| Minimize the no. of copies of sensitive data files and reports | Procedural |
| Minimize traffic and access to work areas * | Physical |
| Place employees identification on work products | Procedural |
| Place physical security of remote network node | Physical |
| Network activity records | Procedural |
| Restrict the display of sensitive information | Technical |
| Validation data input | Technical |

Table 2 illustrates how second level security works. There are standard tactics you can use when conditions indicate. The table includes some conditions under which you would normally consider them.

**Table 2. List of second level security controls**

| Item | Use when | Class |
|---|---|---|
| Appoint a computer security officer | You have enough computer resources to justify the position | Procedural / physical |
| Control access to loading locks | You want to maintain strict control overall access | Physical |
| Encrypt data * | You need the highest degree of protection | Technical |
| Generate passwords automatically * | There are many people with access to the network | Technical |
| Monitor computer use | You want to ensure that only employees have access | Procedural |
| Provide alternative | You need a high degree of reliability | Disaster control |
| Provide for dynamic password. Changes by users * | There may be frequent interruptions in the network node use | Technical |
| Provide for identification and trust worthiness of couriers * | You use couriers to carry sensitive information | Physical / Procedural |
| Provide terminal identifiers | You need a high level of security at individual | Technical |
| Separate test and production systems | You have a large system of linked networks | Technical |
| Sign agreements with remote users | Your want to control remote access | Procedural |

# 6. SECURITY MODEL

An important issue when designing the software that will operate and control, secure computer systems and networks is the security model that the system or network will be based upon[3]. The security model will implement the security policy that has been chosen and enforce those characteristics deemed most important by the system designers. For example, if confidentiality is considered paramount, the model should make certain that no data is disclosed to unauthorized individuals. A model enforcing confidentiality may allow unauthorized individuals to modify or delete data as this would not violate the tenets of the model since the true values for the data would still remain confidential. Of course, this model may not be appropriate for all environments. In some instances, the unauthorized modification of data may be considered a more serious issue than its unauthorized disclosure. In such cases, the model would be responsible for enforcing the design on its critical if you want to ensure that the resulting system accurately enforces the security policy desired.

Symmetric encryption algorithms (i) AES/ Rijndael, (ii) Blowfish, (iii) CAST5, (iv) DES, (v) IDEA, (vi) RC2, (vii) RC4, (viii) RC6, (ix) Serpent, (x) Triple DES, and (xi) Twofish.

There are five basic types of host-based IDS sensors

(i) Log analyzers
(ii) Signature-based sensors*
(iii) System call analyzers
(iv) Application behavior analyzers, and
(v) File integrity checkers

**Note: *** Indicates that these items exist in the proposed IDS

## 6.1 The Anomaly Vs Signature Detection

Intrusion detection system must be capable of distinguishing between normal (not security-critical) and abnormal user activities, to discover malicious attempt in time[1]. However, translating user behaviors (or a complete user session) in a consistent security- related decision in often not that simple- many behavior patterns are unpredictable and unclear.

# 7. E-MOTE TO SERVER INTERFACE

The E-Mote present in the server contains the hostname and the IP Address of the different client identified as trusted in the network. When a client request occurs on a server, it gets access to its E-Mote. The E-Mote validates and contains mechanism to prevent (E-Mote stores all the hostname of trustworthy marked nodes) a particular client if it is untrustworthy.

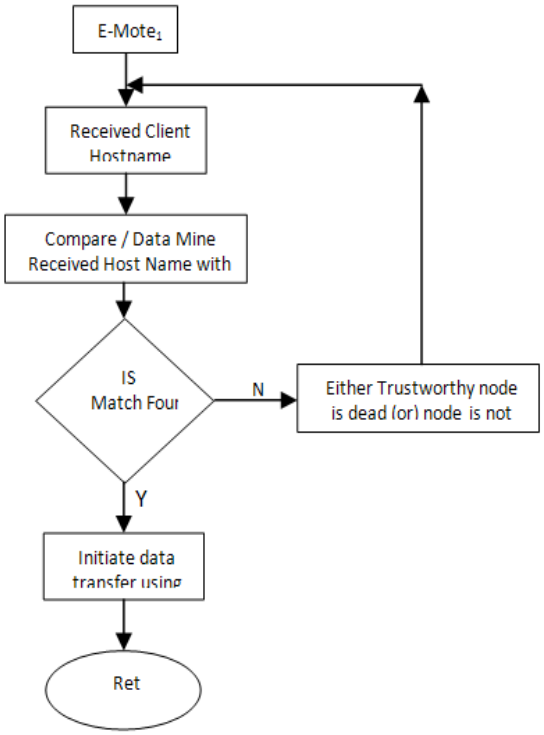



**Fig. 4 E-mote to server interface**

## 8. HARDWARE NODE DETECTION

Live Hardware node scanning is a process of discovering active hardware node in a specified range of hardware node addresses. To accomplish this, the live hardware node scanner sends ICMP messages to all hardware node addresses in the specified range and waits for the reply. Those addresses from which replies are returned within given timeframe are alive and all others are assumed to be dead. In order to avoid network overload, ICMP flood is controlled by specifying how many concurrent pings there can be and when the number is reached, the scanner won't send any more ICMP messages until the previous ping requests are completed.

Start method accepts hardware node address range as its parameter[4]. Range is represented by hardware node class. Constructors accept either hardware node range [the first and the last hardware node address] or subnet [hardware node address of the network and subnet mask]. This class also provides additional services, like calculating number of addresses in the range, comparing hardware node addresses, getting successive address and calculating distance between the addresses. On Scan Progress Update event is raised each time after the scanner finishes with and hardware node address which provides method of tracking progress to users.

List of found hardware node are available through alive hardware nodes uses property and when a live hardware node is discovered the scanner raises On Alive hardware node found event to notify stations. Each hardware node is represented by hardware Scan Hardware node State class. The class stores various information and statistics about hardware node that are discovered during the scanning process. Hardware node Scan Hardware node State class also provides methods for testing current state: Is Alive, Is Dead and Is Testing. There is On State Change event which is raised when the state of the hardware is changed.

## 9. OVERALL DATA TRANSFER ALGORITHM

The data to be transmitted is encrypted before it is transmitted. The algorithm chosen to encrypt the text is data encryption standard [DES]. DES is a symmetric block cipher published by the National Institute of Standards and Technology (NIST). For DES, data are encrypted in 64-bit block using a 56-bit key. The algorithm transforms a 64 bit input in a series of steps into a 64-bit output. The same steps with the same key are used to reverse the encryption. The encryption function has two inputs, plain text and key. It includes permutations, expansions, shifts, substitutions etc. DES is a standard algorithm and in this work, the focus is not on the cryptography but on the intrusion detection.

## 10. DATA ENCRYPTION STANDARD

DES is adopted in 1977 by the National Bureau of Standards and data are encrypted in 64-bit block using a 56-bit key[6]. The DES Encryption is depicted in Figure 5. As with any encryption algorithm, there are two input plaint text to be encrypted and the key. The processing of the plain text proceeds in three phases.

(i) The 64-bit plain text passes through an initial permutation [IP] that rearranges the bits to produce the permuted output.

(ii) This is followed by a phase consisting of 16 rounds of the same function which involves both the permutation and substitution functions.

(iii) The output of the last round consists of 64 bits that are a function of the input plain text and the key. The left and right halves of the output are swapped to produce the pre output. Finally the pre output is passed through a permutation ($IP^{-1}$) that is the inverse of IP function, to produce the 64-bit cipher text.

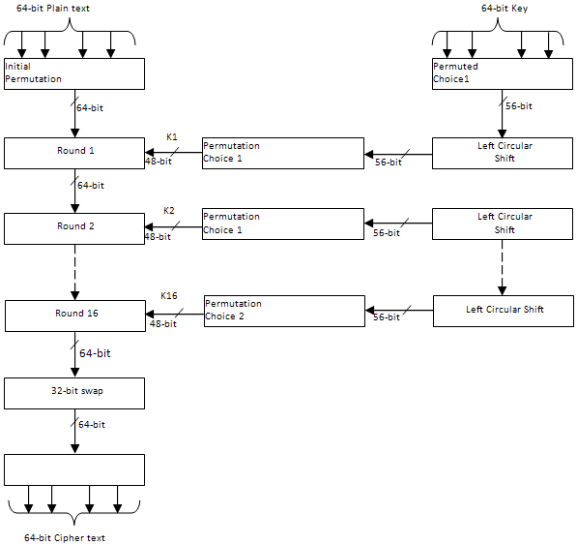



**Fig. 5 General Description of DES Encryption Algorithm**

Decryption uses the same algorithm as encryption, except that the application of the subkey is reversed.

## 11. RESULTS AND DISCUSSION

### 11.1 Asynchronous System

In this work, a self-healing security architecture is designed to reduce interprocess communication overhead, increase speedup and minimize the seek time. This work focuses on

schemes to recover data from the transmission error. Pipeline and layout of a general-purpose core are the main features proposed, with simplified design and verification. Additionally, security policies with data at the lowest level in the system are also supported. A generic tagged memory architecture that implements a set of features required by a whole suite of dynamic analyses also exists. The block diagram of the self-healing (error detection and control) architecture using semaphores is shown in Figure 6. It consists of the Mutex tracking algorithm and tracks the information flow at the atomicity level.

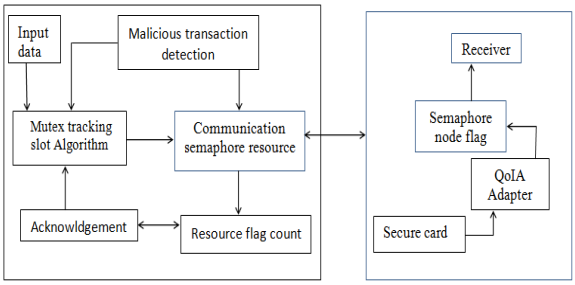



**Fig. 6 Overview Of Self Healing Architecture Using Semaphore**

The design environment is listed in table 3. The DES algorithm results are presented in Figure 7 to Figure 9.

**Table 3. Specification of the Emulator**

| S. No. | Specification | Implementation Type |
|---|---|---|
| 1. | Emulator | Netwinz |
| 2. | Data communication process | Asynchronous [Datalink layer/Network layer] |
| 3. | Error Detection and recovery Logic | Physical layer |
| 4. | Type of control | Semaphore switched |

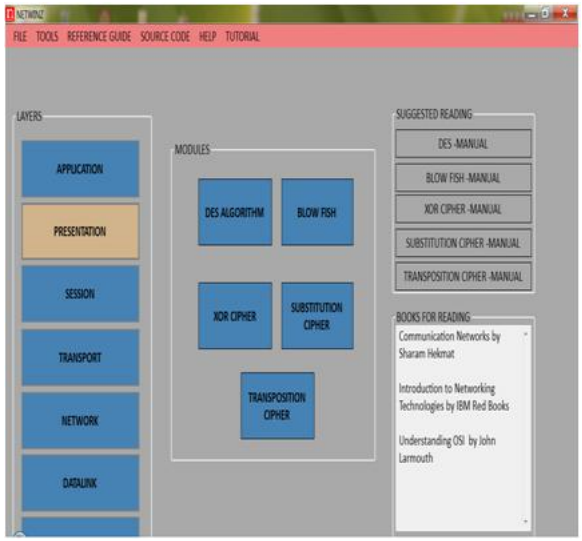



**Fig. 7 Presentation layer with DES encryption**

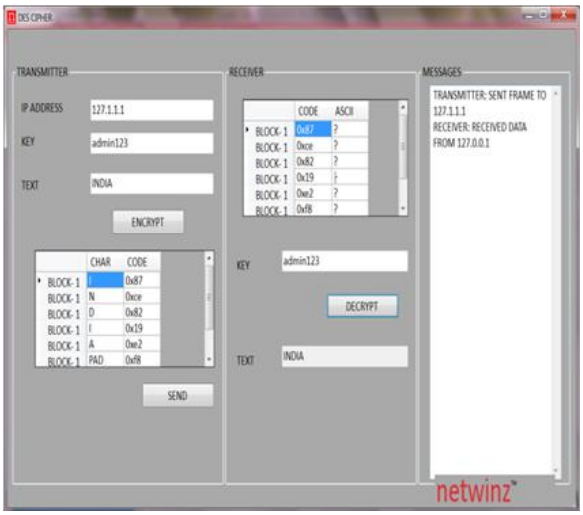



**Fig. 8 DES encrypted data transfer**

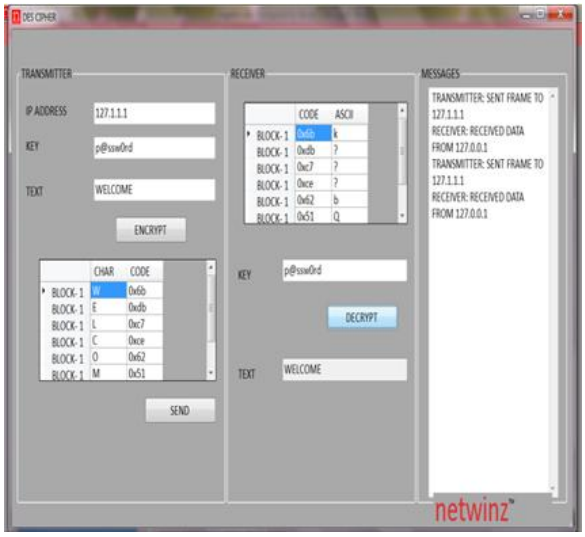



**Fig. 9 Decrypted output**

## 12. CONCLUSION

Typical constraints studied in this work included intrinsic error control, power control, energy control, multihop optimal routing, live node detection, node authentication and initial signature analysis.

## 13. ACKNOWLEDGEMENT

The authors heartly thank M/s. MicroLogic Systems, Chennai-600017, for the infrastructure to carry out this work successfully.

## 13. REFERENCES

[1] A.Elbirt, "An FPGA Implementation and Performance Evaluation of the CAST-256 Block Cipher", Technical Report, Cryptography and Information Security Group, Electrical and Computer Engineering Department, Worcester, MA, May 1999.

[2] Rebaudengo M, Sonza Reorda M, Torchiano M and Violante M, "Soft-error detection through software fault-tolerance techniques", International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 210-218, Nov 1999.

[3] Y. Xiong and L. G. Mason. "Restoration strategies and spare capacity requirements in self-healing ATM networks", proceedings of IEEE/ACM Transactions in Networking, Vol. 7, No. 1, pp. 98–110, 1999.

[4] Jae Min Lee, "Physical Layer Redundancy Method for Fault-Tolerant Networks in Picnet", Proceedings on the 13th CISL Winter Workshop, Sul-Ak Mountain, February 2000.

[5] P. K. Lala, "Self-Checking and Fault-Tolerant Digital Design", San Francisco, CA: Morgan Kaufman Publishers, 2001.

[6] N. Matloff, "Cyclic Redundancy Checking", in the 10th symposium of Department of Computer Science, University of California, 2001.

[7] Michaël Hauspie, "Localized Algorithms for Detection of Critical Nodes and Links for Connectivity in Ad hoc Networks", proceeding in 3rd IFIP Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET 2004), Bodrum, Turkey", 2004.

[8] Andrew R. Cormier, Carl B. Dietrich, Jeremy Price, Jeffrey H. Reed, "Dynamic reconfiguration of software defined radios using standard architectures", Physical Communication 3, pp. 73-80, 2010.

[9] Anh Thien Tran, Dean Nguyen Truong and Bevan Baas, "A Reconfigurable Source-Synchronous On-Chip Network for GALS Many-Core Platforms", IEEE Transactions on Computer-Aided Design Of Integrated Circuits and Systems, Vol. 29, No. 6, June 2010.

[10] E. Beigne, F. Clermidy, H. Lhermet, S. Miermont, Y. Thonnart, X.-T. Tran, A. Valentian, D. Varreau, P. Vivet, X. Popon, and H. Lebreton, "An asynchronous power aware and adaptive NoC based circuit", IEEE J. Solid-State Circuits, vol. 44, no. 4, pp. 1167–1177, Apr. 2009.

[11] E. Beigne, et al., [2009] presented the processors hardware to support dynamic voltage and frequency scaling (DVFS) through software or a local DVFS controller.

[12] P. Teehan, G. G. F. Lemieux and M. R. Greenstreet, "Estimating reliability and throughput of source-synchronous wave-pipelined interconnect", in Proc. ACM/IEEE International Symposium Network Chip (NOCS), pp. 234–243, May 2009.

[13] X. Xiang, X. Wang and Y. Yang, "Supporting Efficient and Scalable Multicasting over Mobile Ad Hoc Networks", IEEE Transactions On Mobile Computing, Vol. 10, No. 5, April 2011.

[14] J. H. Sarker, M. Hassan, S. Halme, "Power level selection schemes to improve throughput and stability of slotted ALOHA under heavy load", Computer Communication 25, 2002.