A Power Saving Scheme for Open Flow Network

Bhed Bahadur Bista, Masahiko Takanohashi, Toyoo Takata, and Danda B. Rawat

Abstract—**Until recently the main focus of researchers in energy efficient networks was how to extend the life time of battery operated networks such as wireless sensor networks and mobile wireless networks by reducing the consumption of energy by nodes. But as Internet traffic has increased, the power consumed by Internet nodes has also increased. Researchers now focus on energy efficient wired network also. Here we proposed a scheme to put nodes to sleep or wake sleeping nodes depending upon the traffic in the Open Flow network. We focused in Open Flow network because it is flexible to manage and easy to deploy new protocols in it and is being standardized/supported by big enterprises such as NEC, Google, Cisco, Microsoft etc.**

Index Terms—**Open Flow network, sleeping/awake network nodes, power consumption, centralized decision, network reconstruction.**

I. Introduction

Although a lot of researches and developments have been done in energy efficient end systems, such as PCs, and battery based networks, such as wireless sensor networks and mobile networks for a long time, only recently researchers and industries have started putting their attentions to energy efficient wired networks. The motivation for developing energy efficient end systems such as PCs/processors was to improve their performance (less heat better performance, less chance for damage due to heat) while in wireless networks, especially battery operated devices, the motivation was to increase the battery life. However, the motivation to develop energy efficient wired network is to actually save energy without degrading the performance of the network.

There are mainly two reasons why wired network operators have raised their interest in power aware networks. One is due to the environmental awareness; there is strong political and social pressure to industries to develop energy efficient systems. Another is that due to Internet traffic increase in recent years, the power consumed by the network has also increased. The Global e-Sustainability Initiative (GeSI)[1] estimated that in 2010, network energy requirement in European telecom operators about 21.4 TWh and forecasted a figure of 35.8 TWh in 2020 if no power saving initiatives are adopted.

It is well know that for robustness and fault tolerance, over provisioning of link bandwidth and massive link redundancies are adopted in networks. It is stated that the average utilization of backbone networks/links is less than 30 percent [2]. It has been found that the energy consumption by network equipment is almost the same whether it is in idle or fully utilized [3].That is, the network equipment's power consumption is independent of its utilization. From the findings in above works, we see that a large portion of energy is consumed by redundant links or idle network devices.

Observations have shown that, over a large timescale, the Internet traffic exhibits strong daily and weekly patterns which do not change over years [4]. Moreover, it has been shown in [5] that higher the link speed, more the energy consumption by the device/link. They have shown that a 1 Gbps Ethernet link consumes about 4W more than a 100 Mbps link.

Taking the advantages of the aforementioned characteristics of wired networks/devices —underutilization of network devices, regular network traffic patterns and lower the link speed lower the power consumption —researchers have proposed different methods to put some links (or whole device) to sleep or lower the link speed depending upon the traffic volume in the network. The approaches are either distributed or centralized. The basic approach is to measure traffic at each device and depending upon the traffic volume, actions for reducing the power consumption, such as shutting down some links/devices or reducing the link speed are taken such that the overall network performance does not degrade.

In this paper, we use Open Flow network thus our approach is centralized approach. Unlike in classical router/switch where data path and control path (routing protocols) occur on the same device, in Open Flow routers/switches, these are separated [6], [7]. The data path still resides on the routers/switches, while high-level routing decisions are moved to a separate controller, typically a standard server. The Open Flows witch and controller communicate via the Open Flow protocol. Basically, the controller configures routers/switches, manages flow table (routing) based on the network state and routers/switch state.

The Open Flow is being standardized and supported by Open Networking Foundation whose members are Google, Cisco, Microsoft, facebook etc. and is considered to be the future network architecture for flexibility and maintainability [8]. In our proposal, the controller periodically checks the traffic load in the network. If the traffic load is less than the capacity of the network, then it checks if any nodes can be put to sleep. If it can, it calculates the new route and put nodes to sleep. If the traffic load increases and the remaining active nodes cannot handle it, the controller wakes up the sleeping nodes and re-calculates the routing path.

The paper is organized as follows. In Sect. II, we present some related works. In Sect. III, we present the detail of our proposal followed by the evaluation of the proposal in Sect. IV. Finally, we conclude the paper in Sect. V.

Manuscript was received on October 29, 2012; revised December 30, 2012.

Bhed Bahadur Bista, Masahiko Takanohashi, and Toyoo Takata are with the Faculty of Software and Information Science, Iwate Prefectural University, 020-0193, Japan (e-mail: {bbb, takata}@iwate-pu.ac.jp).

Danda B. Rawat is with Eastern Kentucky University, Richmond, KY 40475, USA (e-mail: db.rawat@ieee.org).

DOI: 10.7763/JOCET.2013.V1.63

II. Related Works

There are various researches on energy saving on wired networks. Gupta *et al,*.[9], [10],[11]have first mentioned the energy efficiency issue in wired networks. They have developed several algorithms to find inactive period of links and put them to low power mode (or sleep) during that period. The point is when and how long to turn off the links. Without detailed knowledge of the incoming traffic pattern, this approach has to compromise with network performance (e.g. packet loss, delay etc.) on energy consumption reduction. Moreover, this approach cannot be applied to backbone routers. As it is already shown that low speed links consume less energy, an adaptive link rate (ALR) for Ethernet is proposed in [3]. When a link is underutilized, the speed of the link is reduced to save energy. However, the proposed method only works for Ethernet. More research is needed to extend it to other network links. Arai *et al.* [12] have proposed a distributed routing protocol for the network running OSPF (Open Shortest Path First) [13] routing protocol to put some nodes to sleep. Besides, OSPF's LSA (Link State Advertisement), and LSDB (Link State Database), their proposed protocol, floods NSA (Network State Advertisement) and maintains NSDB (Network State Database) and HLDB (Historical Link State Database). As the original OSPF cannot distinguish between sleeping and failure nodes, HLDB is used to distinguish such nodes in their proposed method. The routers calculate traffic trend from NSDB and change weight of links and then apply standard OSPF operation. Backbone nodes, which are not in the path of any one pair of access nodes, shift to sleep mode. If the traffic increases the sleeping nodes are put to awake. The main problem of the proposal is that routers have to maintain three large databases (LSDB, NSDB and HLDB). There will be large control packets flow in the network as the routers have to flood both NSA and LSA regularly. Computation to shift nodes to sleep and vice versa is also high for routers. More survey of energy efficiency in Internet can be found in [14].

III. Proposed Energy Saving Algorithm for Open Flow Network

A. Network Environment

As shown in Fig. 1, the Open Flow network we are considering consists of a controller and a number of routers/switches called nodes. All nodes are connected to the controller and are connected to each other in random manner. There are two types of nodes, access nodes and backbone nodes. Access nodes are connected to users' terminals or servers whereas backbone nodes are connected to other nodes only. We assume that all nodes are homogeneous, i.e., have the same power consumption rate and data transfer rate.

B. Outline of the Power Saving Algorithm

The controller puts a node to sleep or wakes it up and changes routing path according to the following steps.

Step 1. Measure the network traffic periodically.
Step 2. Depending upon the network traffic, calculate the number of required nodes to forward the traffic and also determine the root node.
Step 3. Use the sleep/wake algorithm, mentioned later, to put alive nodes to sleep or wake sleeping nodes as required.
Step 4.Change the routing path using the alive nodes. This step uses the shortest path first algorithm to calculate the routing path.

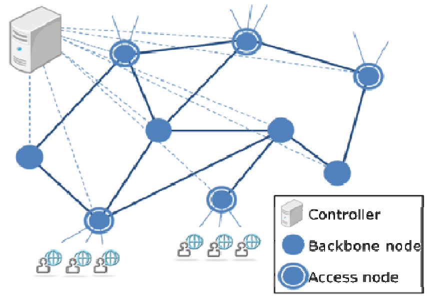


Fig. 1. Open Flow network architecture.

The number of required nodes is calculated as follows.

No. of required nodes =Total network traffic / transfer rate of a node.

For example, if the total network traffic is 700 Mbps and the maximum transfer rate of a node is 100 Mbps:

The required number of nodes = 700/100 = 7

C. Selection of Root Node

A root node is selected based on the network traffic and the traffic volume passing through each node. The root node is selected in such a way that the maximum number of nodes can be put on to sleep. Selection of the root node is as shown below.

1) A node which is handling maximum traffic volume (at the time of traffic measurement) is selected as a root node. The reason for selecting a node with the maximum traffic is to avoid putting such a node to sleep. (The root node will never be put into sleep).
2) If there are more than one node which are handling the same amount of maximum traffic volume, a node with higher number of connections is selected as the root node. The reason is that such a node will have more access to the network.

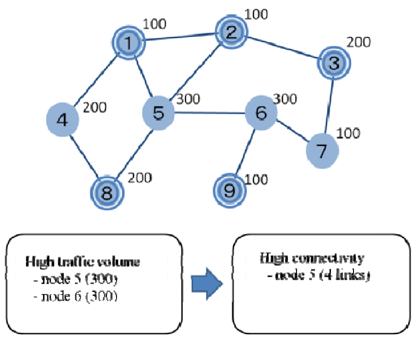


Fig. 2. Root node selection example.

For example in Fig. 2, double circle nodes are access nodes and single circle nodes are backbone nodes. The number above a node indicates traffic volume the node is handling at the time of measurement. According to the root node selection procedure, nodes 5 and 6 are first selected as candidate root nodes (they have the same traffic volume).

Next since node 5 has more connections than node 6, it is selected as the root node.

D. Sleep/Wakeup Decision Making Algorithm

After calculating required number of nodes and selecting a root node, we use the following algorithm either to put analive node to sleep or wakeup a sleeping node to alive. Whether we can put any awake node to sleep or wake any sleeping nodes depends upon the number of awake nodes and the number of required nodes to forward the traffic. As such we have to consider three cases:

Case 1. The number of alive nodes is greater than the number of required nodes to forward traffic.

Case 2. The number of alive nodes is less than the number of required nodes to forward traffic.

Case 3. The number of alive nodes is equal to the number of required nodes to forward traffic.

Case 1: Number **of alive nodes > Number of required nodes**

This case states that the traffic volume in the network is less than before or less than the network's capacity. There is a possibility that some nodes can be put to sleep.

As stated previously the root node is selected first. We consider the two situations.

1) The currently selected root node is different from the previously (previous run of the algorithm) selected root node.
2) The currently selected root node is the same as the previously selected root node.

E. Previous Root Node ≠ Current Root Node

Step 1. Construct an access tree with the shortest path from the root node to other nodes. This can be easily contracted using Dijkstra's Shortest Path Algorithm.

Step 2. Select candidates for sleep nodes. Note that the nodes in between the root node and the access nodes cannot be put to sleep. Rests are candidates for sleep nodes.

Step 3. Put the candidate nodes to sleep one at a time beginning with a node with minimum traffic volume until the number of alive nodes is equal to the required number of nodes or all candidate nodes are put to sleep.

Note that if the number of alive nodes becomes equal to the number of required nodes, no candidate nodes will be put to sleep.

We show the procedure with an example. Let's consider the network topology as shown in Fig. 2. According to the root node selection method, node 5 is selected as the root node. From the root node an access tree with minimum hop to all nodes is constructed as shown in Fig.3. According to the step 2, all nodes from access nodes to the root nodes are considered as alive nodes and the rest are candidate for sleep nodes. In Fig. 3, nodes 4 and 7 are candidate sleep nodes and the rests are alive nodes. If the candidate nodes are to put to sleep, the node 7 will be put to sleep and then node 4. If they are to be woken up the reverse order will follow. For example if the required number of nodes is 8, the node 7 will be put to sleep as shown Fig. 4. If the required number nodes are 7 or less, both nodes 4 and 7 will be put to sleep as shown in Fig. 5.

Suppose if node 6 is selected as the root node, the access tree will be as shown in Fig. 6 and the candidate sleep node will be node 4 only. No matter how many number of required nodes there are only node 4 can be put to sleep.

F. Previous Root Node = Current Root Node

Since the currently selected root node is the same as the previous root node, the access tree does not change. However, if there are candidate sleep nodes which were not put to sleep previously, the nodes may be put to sleep in the same manner as mentioned above. Otherwise, no other operation will be performed.

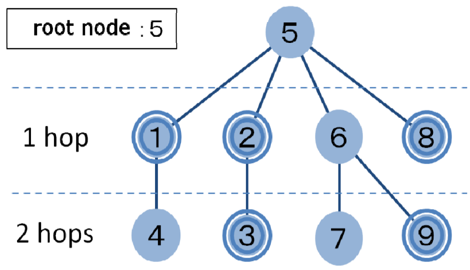


Fig. 3.Access tree when node 5 is root node.

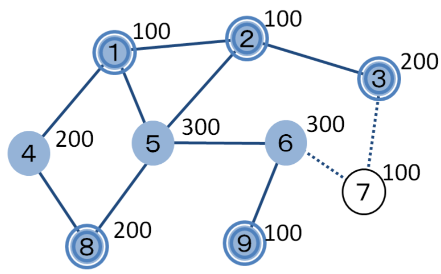


Fig. 4. Network topology when no. of required nodes is 8.

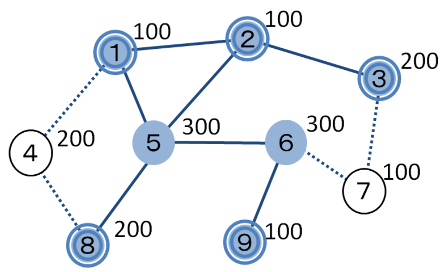


Fig. 5. Network topology when no. of required nodes is 7.

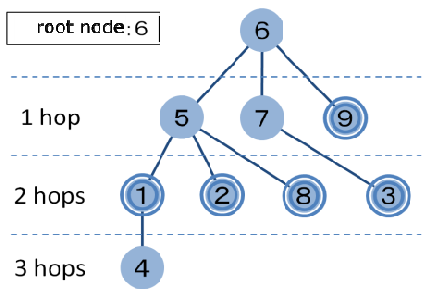


Fig.6.Access tree when node 6 is root node.

Case 2: Number of alive nodes < Number of required nodes

In this case the current alive nodes are not enough to achieve the required throughput of the network (some packets may have been dropped as a result). So sleeping nodes will be woken up to reach the required number of alive nodes.

G. Previous Root Node ≠ Current Root Node

Since it shows that the current root node is not the same as the previous root node, the concentration of traffic pattern in the network has been changed which will result the different access tree. In order to make the different access tree, we have to wake all the sleeping nodes and start from 1. of Case 1 above.

H. Previous Root Node = Current Root Node

Since the root node is the same as the previous root node the access tree will not change. The sleep nodes are awakened until required number of alive nodes is reached. The order of waking of sleeping nodes is in reverse to the order of putting alive nodes to sleep.

Case 3: Number of alive nodes = Number of required nodes

In this case, alive nodes are enough for forwarding the traffic and it is not possible to put any alive nodes to sleep. At the same time there is no need to wake up sleeping nodes.

IV. Evaluation

We evaluated our proposal with simulation. We used Open Flow VM for simulation tool and NOX [15] for controller software. As shown in Table I, we constructed 9 network topologies with 5, 10, 15 access nodes and 10, 20, 30 backbone nodes for the simulation. In each topology, nodes were randomly connected and traffic was randomly generated at each access nodes.

TABLE I: Simulation Environment

Topology(Topo.)	Access Nodes	Backbone Nodes	Total
1	5	10	15
2	5	20	25
3	5	30	35
4	10	10	20
5	10	20	30
6	10	30	40
7	15	10	25
8	15	20	35
9	15	30	45

As Arai *et al.* [12] (ECO-RP) is the most closely related to our proposal —putting nodes to sleep— we compare our proposal with ECO-RP using the following criterions.

1) Maximum numbers of possible sleep nodes (thus power consumption of the network).
2) Required number of control packets for network topology construction.
3) Time taken to change the topology, i.e., putting nodes to sleep or wake them up, and forming new routing paths.

A. Maximum Number of Possible Sleeping Nodes

From the simulation, we found that, in general, we can put more nodes to sleep compare to ECO-RP (Fig. 7). The number of sleeping nodes in the network directly relates to the number of units of power reduction in the network. However, in our proposal, we have to consider the power consumed by the controller also. It is believe that the power consumed by a controller is 2.5 times more than power consumed by a node in Open Flow network. Since we have assumed that the nodes in the network are homogeneous, for simplicity, we consider that a node consumes one unit of power/hour while alive and zero unit of power/hour while asleep. In reality, power consumed by a node depends upon its hardware and traffic profile[16].After including the power consumed by the controller in our proposal, the power consumed by our proposal and ECO-RP is shown in Fig. 8. Though the number of nodes that can be put into sleep are more in our proposal, the power consumed in some topologies is better in ECO-RP since they do not use a central controller.

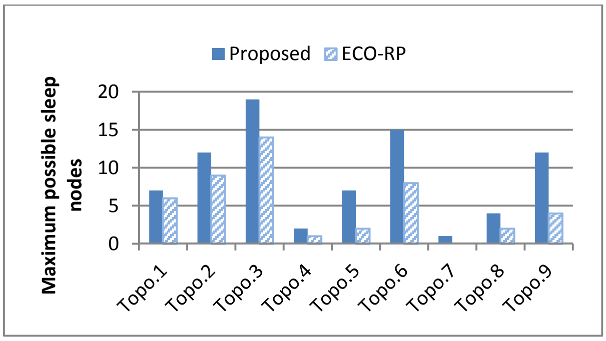


Fig. 7. Maximum possible candidate sleep nodes.

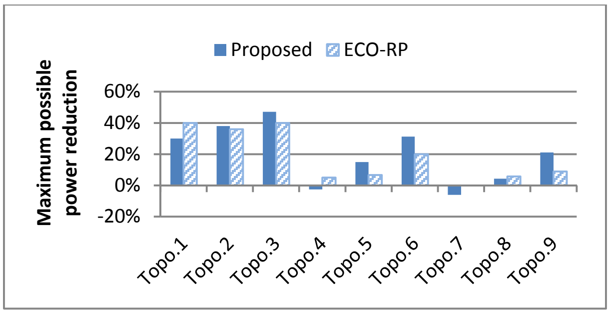


Fig. 8. Maximum percentage of possible power reduction.

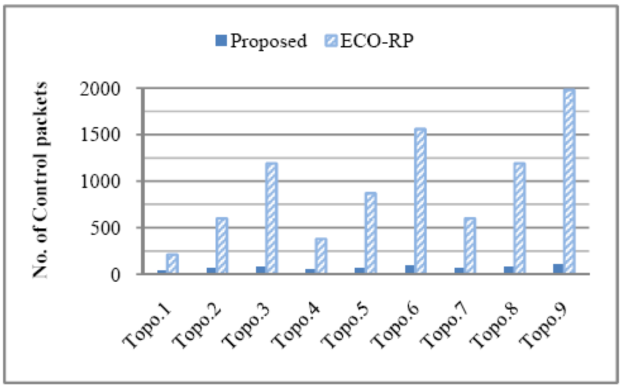


Fig. 9. Number of control packets.

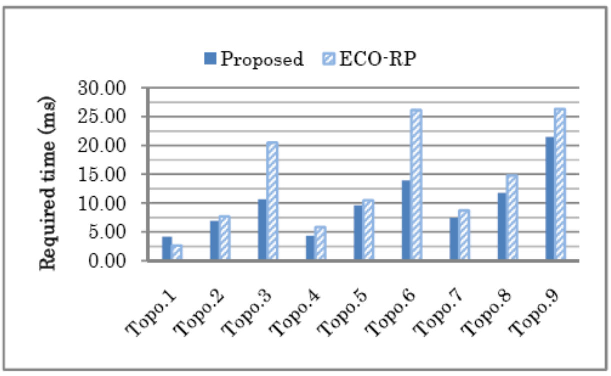


Fig. 10. Time to change the network topology.

B. Number of Control Packets

Control packets are different from the data packets and used for gathering network information such as link costs, hop counts etc., in order to calculate data delivery path from one node to other nodes and other network maintenance.

In our proposed method the controller needs two packets (request and reply) to get the information from a node. So the number of required packets is $2n$ where n is the number of nodes. However, in Arai *et al.* case, each node sends the control packets to all nodes. Thus the number of control packets are $2(n(n-1))$ (for LSD and NSD). In our case, we need control packets to put nodes to sleep or awake them. They are equal to number of sleeping (or number of sleep to waking) nodes. Arai *et al.* method does not need control packets for this case. The control packets required for topology construction is as shown in Fig. 9.

C. Time Taken to Change Topology (due to sleep/awake)

Time taken to change the topology, i.e., to put nodes to sleep or wake them up is shown in Fig 10. From the figure, we can see that our proposal takes less time than ECO-RP. In ECO-RP, each node makes decision whether to go to sleep or wakeup one at a time. Thus it takes longer time to change the topology as the number of nodes increase. Therefore in a topology where number of nodes is less, e.g. topology1, time taken to change the topology is less in case of ECO-RP.

V. Conclusion

Environmental issues and increase in Internet traffic have encourages the researchers to focus on energy efficient Internet. We have shown how we could put some nodes in OpenFlow network to sleep to save energy. We notice that our centralized decision making approach to put nodes to sleep is less complex, readily deployable and energy efficient than the distributed approach (ECO-RP).In our present approach, we can put a node to sleep only if we can shut down its all links. In some cases, it might not be possible to shut all links down. If our proposal is extended to put not only whole node to sleep but a part of it (some links only), we would be able to save more energy and as well as route the traffic more efficiently.

References

[1] GeSI. Enabling the low carbon economy in the information age. [Online]. Available: http://www.smart2020.org/_assets/files/02_Smart2020Report.pdf.

[2] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proceeding NSDI'08, Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 323-336, 2008.

[3] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR)," *IEEE Transactions On Computers*, vol. 57, no. 4, pp. 448-461, April 2008.

[4] C. Fraleign, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. ROcell, T. Seely, and S. C. Diot, "Packet-Level Traffic Measurements from the Sprint IP Backbone," *IEEE Network*, vol. 17, no. 6, pp. 6-16, Nov.-Dec.2003.

[5] C. Gunaratne and K. Christensen, "Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed," *International Journal of Network Management,* vol. 15, no. 5, pp. 297-310, Sept./Oct. 2005.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69-74, 2008.

[7] OpenFLow Switch Specification. [Online]. Available: http://www.openflow.org/wp/documents/.

[8] Open Net working Foundation. [Online]. Available: https://www.opennetworking.org/.

[9] M. Gupta and S. Singh , "Dynamic Ethernet Link Shutdown for Energy Conservation on Ethernet Links," *IEEE ICC 2007,* pp. 6151-6161, 24-28 June, 2007.

[10] M. Gupta, S. Grover and S. Singh, "A Feasibility Study for Power Management in LAN Switches," in *Proc. IEEE ICNP 2004*, pp. 361-371, 2004.

[11] M. Gupta and S. Singh, "Using Low-Power Modes for Energy Conservation in Ethernet LANs," *INFOCOM 2007*, pp. 2451-2455, 2007.

[12] D. Arai and K. Yoshihara, "ECO-friendly Distributed Routing Protocol for Reducing Network Energy Consumption," in *Proc. of CNSM 2010,* pp. 104-111, 25-29 October 2010.

[13] *RFC2328 OSPF Version 2*, 1998.

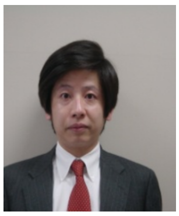
[14] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," *Communications Surveys & Tutorials, IEEE,* vol. 13, no. 2, pp. 223-244, Second Quater 2011.

[15] NOX. [Online]. Available: http://noxrepo.org/wp/.

[16] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power Awareness in Network Design and Routing," *INFOCOM'08,* pp. 457-465, 2008.

Bhed Bahadur Bista received his Ph.D. degree in Information Science from Tohoku University, Japan in 1997. He is currently working as an Associate Professor at Iwate Prefectural University, Japan. His research interests include energy efficient networks, mobile and wireless networks

Masahiko Takanohashi was born in Japan in 1989. He graduated from Iwate Prefectural University, Japan, in 2012. Currently he is working in a company in Japan. He is interested in power aware computer networks. The above research was done while he was in IPU.



Toyoo Takata received his Ph.D. degree in information and computer sciences from Osaka University, Japan,in 1989.Since 1998, he has been a Professor of Software and Information Science, Iwate Prefectural University. His research interests include network security, user authentication and information theory.

Danda B. Rawat is currently working as an Assistant Professor at Eastern Kentucky University, USA. He received his Ph.D. in Electrical and Computer Engineering from Old Dominion University, USA. His research interests include areas of wireless communications and network, and network security.