International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(1): 84-96
The Society of Digital Information and Wireless Communications, 2011 (ISSN 2220-9085)

A Training Based Transmission Period Setting Protocol

Guangri Jin, Li Gong, Hiroshi Furukawa
Graduate School of Information Science and Electrical Engineering,
Kyushu University
744 Motooka, Nishi-ku, Fukuoka, 819-0395 Japan
jingr@mobcom.ait.kyushu-u.ac.jp,
kyoriki@mobcom.ait.kyushu-u.ac.jp,
furuhiro@ait.kyushu-u.ac.jp

ABSTRACT

We have proposed Intermittent Periodic Transmission (IPT forwarding) as an efficient packet relay method for wireless backhaul. In IPT forwarding, a source node sends packets to a destination node with a certain time interval (IPT duration) so that signal interference between relay nodes that send packets simultaneously are reduced and frequency reuse is realized, which brings about an improvement of system throughput. However, the optimum IPT duration setting for each node is a difficult problem which was not solved adequately yet. In this paper, we propose a new training based IPT duration setting protocol which employs some training packets to search for the optimum IPT durations for each node. We also extend the proposed protocol for wireless backhauls applying directional antennas. The proposed protocol is evaluated both with computer simulation and with experiments. Evaluation results show that the proposed protocol is not only very effective but also practical.

KEYWORDS

Multi-Hop, Ad-Hoc, Wireless Backhaul, IPT Forwarding, Training Packet, FDA.

1 Introduction

The high-speed data transmission with an order of 100Mbps envisioned for the next generation wireless communication system will restrict the cell radius to less than 100m (a class of pico- cell), which increases the number of cells needed to cover the service area. Deployment of many base nodes considerably raises infrastructure costs, thus cost reduction must be key success factor for future broadband systems.

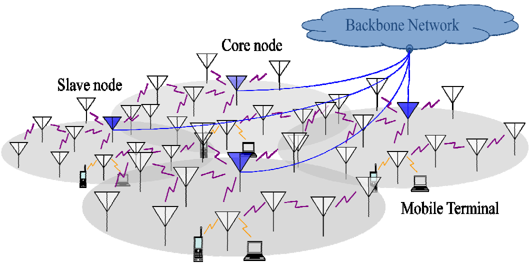

Fig. 1 Wireless backhaul system.

In recent years, wireless backhaul systems have drawn great interest as one of the key technologies to reduce infrastructure costs for next generation broadband systems [1]~[2]. In wireless backhaul, base nodes have the capability to relay packets by wireless, and a few of them, called core nodes, serve as gateways to connect the wireless backhaul with outside backbone network (i.e. Internet) by cables. Upward packets originnated from the mobile terminals (e.g. cell phone) which are associated to one of the base nodes and directed to the outside network are relayed by the intermediate relay nodes (slave nodes) until they reach the core nodes. Downward packets originated from the outside network and directed to a mobile terminal in the wireless backhaul are sent by the core nodes and relayed by slave nodes until they reach the final node to which the mobile terminal is associated (Fig. 1). By connecting base nodes by wireless links, flexibility of the base nodes deployment is realized and total infrastructure costs are reduced due to the reduction in cable constructions [2].

Wireless backhauls traditionally have been studied in the context of Spatial TDMA (STDMA) and the Ad Hoc network. STDMA can achieve collision free multi hop channel access by a well designed time slot assignment for each cell [3]~[5]. However, such planning is not feasible in real systems because of the irregular cell forms in real environments. Additionally, frame synchronization must be managed carefully in STDMA, which induces rather difficult optimization issues [9]. With regard to the Ad-Hoc network, many studies have contributed to improve its performance. In [6], Li et al. have indicated that an application of IEEE802.11 to wireless multihop network fails to achieve optimal packet forwarding due to severe packet loss. In [7], Zhai et al. have proposed a new packet scheduling algorithm called Optimum Packet scheduling for Each Traffic flow (OPET) which can achieve an optimum scheduling of packets by assigning high priority of channel access to the current receiver. However, the overhead imposed by the complicated hand-shake process decreases frequency reuse efficiency. In [8], Bansal et al. have indicated that the throughput of wireless multihop network decreases as the hop count increases.

On the other hand, we have proposed Intermittent Periodic Transmission (IPT forwarding, [9]) as an efficient packet relay method with which the system throughput can achieve a constant value. In IPT forwarding, source node intermittently sends packets with a certain time interval (IPT duration), and each intermediate relay node forwards the relay packet immediately after the reception of it. The frequency reuse space attained by the method is proportional to the given IPT duration. In [10], a series of experiments have been carried out to confirm the effectiveness of the method with real testbed. IPT forwarding is further enhanced with the combinations of MIMO transmission [11] and directional antenna [12].

IPT duration is the most important parameter for applying IPT forwarding method. In [13], a collision free IPT duration setting method was proposed and evaluated with computer simulations. However, the method is not feasible since it introduced some new MAC packets, which makes it difficult to be implemented with general WLAN modules without any modifications. Additionally, system throughput is not guaranteed to be maximized by the IPT durations attained by the method.

In this paper, we propose a new IPT duration setting protocol which employs training packets to search the optimum IPT durations for each slave node. With these IPT durations, the end to end throughputs for each slave node are maximized. A new metric for the training process is also presented. Then we extend the proposed protocol for wireless backhauls using directional antennas introduced in [12]. The proposed protocol is evaluated with both computer simulations and experiments on a real testbeds.

The rest of this paper is organized as follows. Section 2 explains the principle of IPT forwarding and the collision free IPT duration setting protocol proposed in [13]. Section 3 explains the proposed protocol in detail. In section 4 we extend the proposed protocol for a wireless backhaul system applying directional antennas. In section 5, we evaluate the proposed protocol with both computer simulations and experiments. Section 6 concludes this paper.

2 Intermittent Periodic Transmission

In this section, we explain the principle of IPT forwarding along with the conventional packet relay method. The collision free IPT duration setting method proposed by [13] is also introduced.

2.1 Principle of IPT forwarding

In order to clearly explain the principle of IPT forwarding, we illustrated the packet relay mechanism of the conventional CSMA/CA based method and IPT forwarding in Fig. 2 and Fig. 3, respectively.

In the two figures, 9 nodes are linearly placed and instantaneous packet relays on the route are shown in accordance with time. All the packets to be sent are re-formatted in advance to have the same time length.

In the case of the conventional CSMA /CA based method, the source node sends packets with a random transmission period of P_CNV and each intermediate relay node forwards received packets from its preceding node with a random backoff period. In the case of IPT forwarding, the source node transmits packets intermittently with a certain transmission period of P_IPT and each intermediate relay node immediately forwards the received packets from the preceding node without any waiting period. No synchronization is required for both the conventional method and IPT forwarding method.

In the conventional method the co-transmission space, which is defined as the distance between relay nodes that transmit packets at the same time, is not fixed. In such situations, packet collision occurs due to co-channel interference if the co-transmission space is shorter than

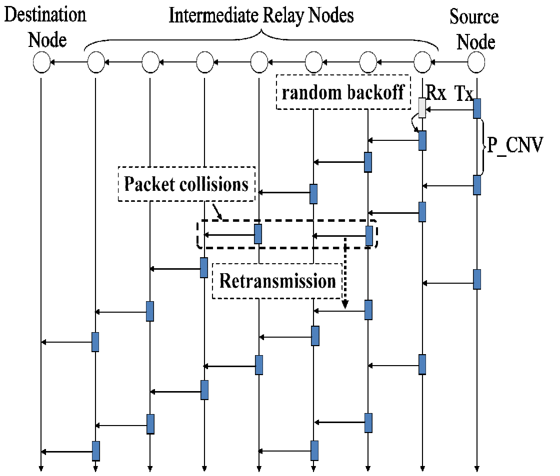

Fig. 2 Packet relay mechanism in conventional method.

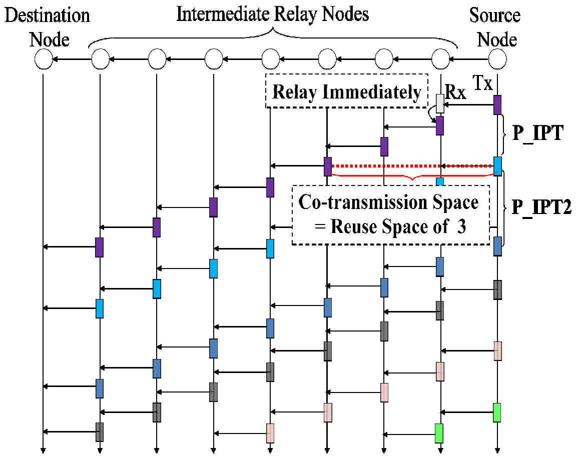

Fig. 3 Packet relay mechanism in IPT forwarding.

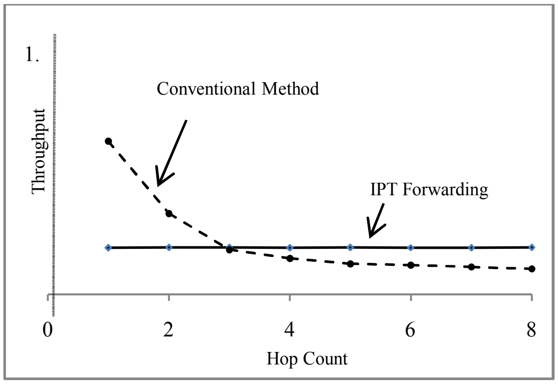

Fig. 4 Performance comparison of the conventional method and IPT forwarding

the required frequency reuse space, as shown in Fig. 2. On the other hand, in the case of IPT forwarding it can be readily understood that the co-transmission space could be controlled by the transmission period of P_IPT that is given to the source node, as shown in Fig. 3 in which reuse space is assume to be 3.

Reduction of the packet collisions will help to reduce retransmissions and will consequently help to improve the system performance. If an adequate IPT duration is set in the core node, it is possible to remove interference between co-channel relay nodes that send packets simultaneously. If the IPT duration is equal to the threshold, the resultant throughput observed at the destination node can be maximized.

Fig. 4 schematically shows the normalized throughput versus hop count feature of the conventional method and IPT forwarding for the systems in Fig. 2 and 3. In Fig. 4, constant IPT duration is applied for all slave nodes and thus the resultant throughputs are all the same [9].

2.2 Collision free IPT duration setting

As discussed earlier, in order to achieve optimal performance the core node should set an adequate IPT duration for each slave node. However, the optimum IPT duration for each slave node depends on many environmental factors such as channel characteristics, node placements, antenna directions and so on. To make IPT forwarding method practical, an automatic IPT duration setting method is required. To this problem, a collision free protocol has been proposed to automatically find IPT durations for each node in [13].

In this subsection, we will first introduce the collision free protocol and then indicate its drawbacks.

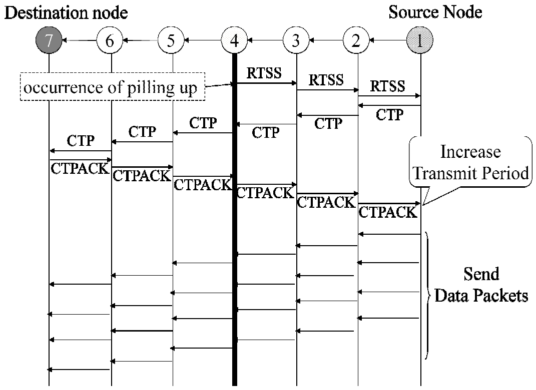

Fig. 5 Collision Free IPT duration setting.

1) Summary of collision free method

Three new MAC layer packets, RTSS (Request to Stop Sending) packet and CTP (Clear to Pilling UP) packet and CTPACK (CTP ACKnowledgement) packet, are defined in [13] and a hand shaking algorithm is employed to find the IPT duration for each node.

As shown in Fig. 5, when the IPT duration setting started the source node (node 1) continuously sends data packets to the destination node (node 7) with certain IPT duration. If a data packet transmission fails in an intermediate node (e.g. node 4 in Fig. 5) due to interference, the node sends a RTSS packet to the source node to stop sending data packets. The source node suspends the sending of data packets immediately after reception of the RTSS packet and sends a CTP packet to the destination node. The CTP packet is relayed in the same way as that for data packet and therefore the destination node can know that all the relaying data packets are cleared out from the system by reception of the CTP packet. The destination node immediately sends a CTPACK packet to the source node on reception of the CTP packet. After receiving the CTPACK packet, the source node increases the

IPT duration by one step and resumes the sending of data packets. This process repeats until no data packet forwarding failure occurs in the relaying route.

2) Drawbacks of the collision free method

Although the collision free method can obtain certain IPT durations for wireless backhaul, it has some severe drawbacks as described below.

1) Since new MAC layer packets are introduced, it is difficult to be implemented by general wireless interface modules.
2) The packet transmission state is confirmed by checking the MAC state of each node. However, existing MAC drivers (e.g. MAD WiFi Driver) do not provide such functions.
3) System throughput is not guaranteed to be maximized by applying the IPT durations attained by the method.

Any modification to existing standards will cause extra costs. Since one of the major advantages for wireless backhaul is the ability to reduce costs, a new IPT duration setting method which is not only practical but also exploits the optimum system performance is required.

3 Throughput maximization IPT duration setting protocol

In this section, we propose a new training based IPT duration setting protocol which maximizes the end to end throughput for each slave node. The proposed protocol employs some training packets and performs a series of training process to search the optimum IPT duration for each slave node. During the training process, core node continuously sends a number of training packets to each slave with an IPT duration which increases gradually until the end to end throughput from the core node to the slave node reaches the maximum value.

Throughout this paper we assume that the route of wireless backhaul is already decided before the IPT duration setting starts and will not change during the process of the protocol.

3.1 Variables and parameters

We defined the following variables and parameters in the new protocol.

1) Training packet
2) Number of training packets: *N*
3) Training time for each node: *T*
4) Training metric for each node: *TM*
5) IPT duration for each node: *D* (micro second)
6) Training Step in the process: Δ (micro second)

In these variables, the training packet is defined as OSI link layer's data packet with the length of 1450 Byte and identified by sequence number. The parameters *TM* and *D* are initialized whenever new training begins for a new slave node. The training metric *TM*, which is described latter in detail, is used as the criterion for the training process.

3.2 Details of the protocol

As shown in Fig. 1, wireless backhaul can be considered as the union of a few sub systems each of which is consisted of a core node and several slave nodes belonging to it (i.e. each slave node is connected to the outside network via the other slave nodes intermediately and finally through the core node by wireless multihop fashion). We call the sub systems a mesh clusters (Fig. 6) throughout this paper and the IPT duration setting is performed for each mesh cluster respectively in the same way.

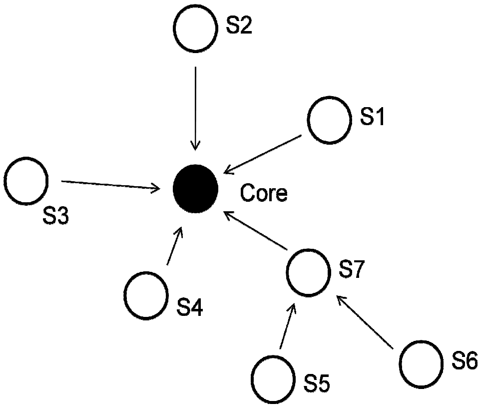

Fig. 6 Mesh cluster.

Now let us consider a mesh cluster with a core node *C* and a set of slave nodes $\{S_1, S_2, ..., S_n\}$. For each slave node $S \in \{S_1, S_2, ..., S_n\}$, the following process is executed .

Step1: The core node *C* initializes the training metric *TM* as -1.0 and initialize *D* as D_0 for the slave node *S*, in which D_0 is a small non-negative value.

Step2: The core node *C* sends *N* training packets which have the sequence number of 1, 2, …, *N* to the slave node *S* continuously with the IPT duration *D*.

Step3: Whenever the slave node *S* receives a training packet which is destined to it, *S* records the sequence number and the packet reception time.

Step4: If the reception of training packets destined to itself is finished, the slave node *S* sends a report packet to the core node *C* which contains the sequence number and reception time (Seq1, T_1) of the first training packet it received and the sequence number and reception time (Seq2, T_2) of the last training packet it received. The number of training packets received without duplication, *Num*, is also included in the report packet.

Step5: When the core node *C* receives report packet from the slave node *S*, it estimates actual training time spent for *S* as below.

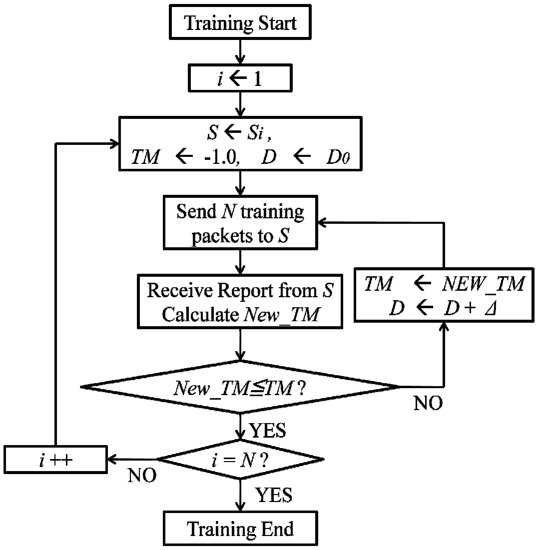

Fig. 7 Core node behavior during training.

$$\begin{aligned} &\sigma \stackrel{\text{def}}{=} (T_2 - T_1) \,/\, (Seq2 - Seq1) \\ &T_{start} = T_1 - \delta * (Seq1 - 1) \\ &T_{end} = T_2 + \delta * (N - Seq2) \qquad (1) \\ &T = T_{end} - T_{start} \end{aligned}$$

According to the estimated training time *T*, a new training metric is calculated as below.

$$New\_TM = \frac{Num}{T}$$

After the computation of new training metric, the training process branches into two cases based on the value of *New_TM.*

a) If $New\_TM \leq TM$, the core node *C* finishes the training for *S* and set the IPT duration of *S* as $(D - \Delta)$ and move to the training of next slave node.

b) If $New\_TM > TM$, the core node *C* increases the IPT duration *D* by Δ, replace the training metric *TM* with New_TM and repeats the above ***Step2~Step5***.

Step6: The core node *C* repeats the above Step2～Step5 until the training for all the slave nodes is finished.

In Fig.7, we showed the behavior of the core node during training process.

3.3 Features of the proposed protocol

We make some explanations on the new proposed protocol in this subsection.

1) Throughout the training process, we assume that the system does not provide packet relay service so that only training related packets exist in the system.
2) During the training process, if some packet losses occur, the first and last packet received by the slave node *S* could be different with the ones sent by the core node *C*. For this reason, the actual training time *T* must be re-estimated as in Step5. In formula (1), δ is the average training packet transmission time and the training start time and end time are complemented by δ, which consequently complements *T*.
3) The protocol repeats the same training process by gradually increasing the IPT duration for each slave node until its training metric reaches the maximum value. Since for each slave node the training finishes at the moment when *TM* begins to decrease, we set the IPT duration as the value of one preceding step as shown in a) of Step5.
4) It can also be easily realized that training metric *TM* is closely connected to the end to end throughput from the core node *C* to the slave node *S*. Since the protocol searches the IPT duration for each slave node which maximizes its training metric, the end to end throughput of each slave node is also maximized by the calculated IPT durations.

4. Extension of the proposed protocol

In this section we extend the proposed protocol to a wireless backhaul system which employs directional antennas and specified as FDA system [12].

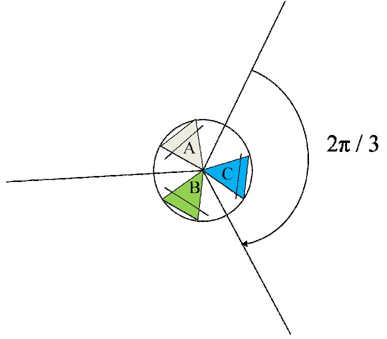

Fig. 8 An *F-3* node with 3 interfaces.

4.1 Introduction of FDA system

FDA system was proposed to simplify the application of directional antennas in wireless backhaul. In the following, we first briefly introduce the principles of FDA system.

In *F-n* FDA system (Fixed Directional Antennas), each node contains *n* wireless interfaces and each interface is equipped with one directional antenna. The *n* directional antennas are symmetrically spaced and each antenna is oriented to one of the *n* uniform directions which cover the whole space together. It is also assumed that the beamwidth of each directional antenna is equal to or less than $2\pi/n$. For illustration, Fig. 8 shows an *F-3* node which contains 3 wireless interfaces and 3 directional antennas.

In FDA, packet relays only occur between the two interfaces of transmitter and receiver nodes which direct their antennas to each other, thereby no need for switching antennas during run-time. During packet relay service, each node receives packets from one interface and forwards the received packets through one of the *n* interfaces. This feature significantly simplifies the application of directional antennas in wireless backhaul.

The authors have proposed a new routing protocol for FDA system which specifies not only the uplink and down-link interfaces used in each node, but also the corresponding interfaces in the destination nodes. The authors have also extended traditional IPT forwarding method for FDA system as below.

(a) Core node applies IPT forwarding in each of its wireless interface.
(b) Core node waits for a certain time (referred to as inter-path IPT duration in the paper) when it switches wireless interface during packet relay.

It was shown that by (a) and (b), packet collisions in FDA system can be avoided. However, the optimization problems for IPT duration and inter-path IPT duration in (a) and (b) remained to be solved.

4.2 IPT and inter-path IPT duration setting for FDA system

We extend the training based protocol for FDA system. Throughout this section, we assume that system route is already decided before the training process.

Since in FDA system antenna switching is not required during run time, the IPT duration setting method proposed in section 3 can be directly applied. To decide the optimum IPT durations in (a), for each slave node (e.g. *S*), core node only needs to execute the protocol using the interface corresponding to the slave node *S*.

Now we consider the inter-path IPT duration setting problem in (b), in which the inter-path IPT durations should be decided for each pair of slave nodes to which core node uses different interfaces to relay packets. We consider two slave nodes *S1* and *S2* to which core node uses interface *M1* and *M2* to relay packets respectively. The inter-path IPT duration between *S1* and *S2* is denoted as *Di* and we assume that core node switches interfaces from *M1* to *M2* as shown in Fig. 9. We also assume that the IPT durations for S1 and S2 are already decided and denoted as *D1* and *D2*.

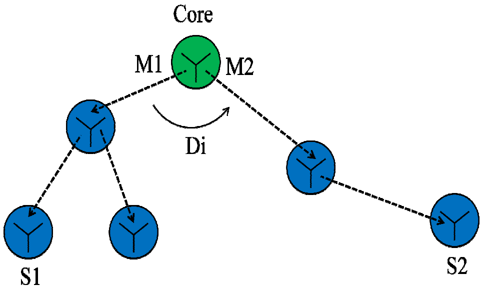

Fig. 9 Inter path IPT duration setting.

The inter-path IPT duration *Di* must be large enough, so that traffic from core node to slave node *S2* does not interfere with the traffic from core node to *S1*. *Di* should also be small enough to ensure the optimum system performance. Based on this consideration, we propose a new training based inter-path IPT duration setting method.

We defined the following variables and parameters for the method:

1) Training packet
2) Number of training packets: *N*
3) Training time for each node: *T*
4) Training metric for each node: *TM, TM1, TM2*.
5) Training Step in the process: Δ

All of these variables have the same meanings as defined in section 3.1. For the pair of slave nodes *S1* and *S2*, core node repeats the following process.

Step1: Core node initializes the training metric *TM1* and *TM2* as -1.0 and initialize *Di* as Di_0.

Step2: Core node sends *N* training packets continuously with the sequence number of 1, 2, …, *N* to slave node *S1* with the IPT duration of *D1*. Then Core node waits for *Di* time. After that Core node sends *N* training packets continuously with the sequence number of 1, 2, …, *N* to slave node *S2* with the IPT duration of *D2*.

Step3: Whenever slave nodes *S1* or *S*2 receive a training packet, they record the sequence number and the packet reception time.

Step4: When the reception of training packet is finished, slave node *S1* and *S2* send report packets to core node with the sequence number and reception time (Seq1, T_1) of the first training packet and the sequence number and reception time (Seq2, T_2) of the last training packet they received. The number of training packets received without duplication, *Num1* and *Num2*, are also included in the report packets.

Step5: When core node received report packets from *S1* and *S2*, it calculates *TM1* and *TM2* in the same way as shown in Step5 of section 3.2, and then calculates *New_TM* = *TM1*+*TM2*. After that the training process branches into two cases based on *New_ TM*.

a) If $New\_TM \leq TM$, then core node finishes the training and set the inter-path IPT duration as $(Di - \Delta)$.
b) Else If $New\_TM > TM$, core node increases the inter-path IPT duration Di by Δ, replace the training metric TM with New_TM and repeats the above ***Step2~Step5***.

It can be easily understood that the above method is an extension of the training based protocol discussed in section 3, which searches the optimum inter-path IPT duration for the pair of *S1* and *S2*. With the attained inter-path IPT duration, the sum of the end to end throughput for *S1* and *S2* is maximized.

5 Evaluation

In this section, we evaluate the proposed protocol with both computer simulations and experiments on a real test-bed under indoor environment.

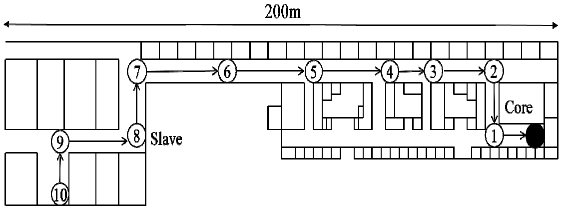

Fig. 10 Simulation site 1, string topology system.

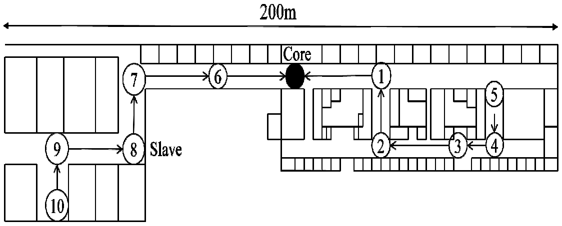

Fig. 11 Simulation site 2, tree topology system.

Table 1. Simulation Parameters.

MAC Model	IEEE802.11a, Basic Mode. Retry Count = 3.
PHY Model	Packet reception fails when SINR level becomes lower than 10dB.
Propagation Model	2 Ray Ground Reflection Model. No Fading Effect. 12dB Attenuation by a Wall.
Routing Method	Minimum Path Loss Routing.
Data Packet Length	1500 Byte.

5.1 Evaluation by Simulations

We assume IEEE802.11a as the wireless interface of each node and deployed two simulation scenarios with string topology and tree topology respectively as shown in Fig.10 (Scenario 1) and Fig.11 (Scenario 2). The simulation sites are models of West Building of ITO Campus, Kyushu-University, Japan.

Each system in the Scenarios is consisted of only one mesh cluster. The simulation parameters are shown in Table 1 and IPT forwarding is applied.

5.1.1 Simulation scenarios

In the first, we measured the end to end throughput from core node to each slave node with different IPT durations in the two simulation scenarios using the following formula (2).

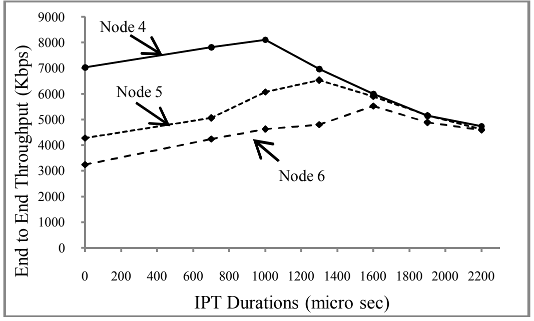

Fig. 12 IPT durations and end to end throughput for node 4, 5, 6 in simulation scenario 1.

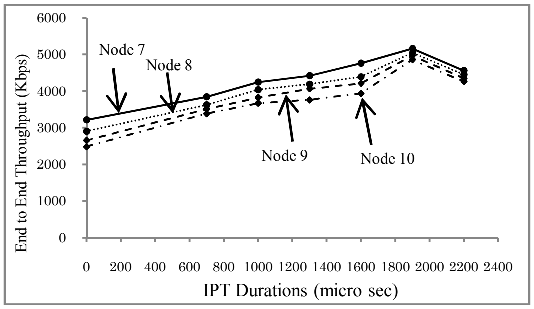

Fig. 13 IPT durations and end to end throughput for node 7, 8, 9, 10 in simulation scenario 1.

Table 2 Optimum IPT durations in simulation scenario 1 (µsec).

Node 4	Node 5	Node 6	Node 7
1000	1300	1600	1900
Node 8	Node 9	Node 10	
1900	1900	1900	

$$Th = \frac{Nr \times PL}{Tm} \qquad (2)$$

In the above formula *Th* is throughput, *Nr* is the number of received packets without duplication, *PL* is packet length and *Tm* is transmission time.

In this first evaluation, IPT durations are set manually for the purpose of searching an optimum IPT duration for each slave node. Manually taken optimum IPT durations are compared with the ones to be found by the proposed protocol, afterward.

We assume that no extra traffic occurs during the measurement and the number of transmitted packets in the above formula is 2000.

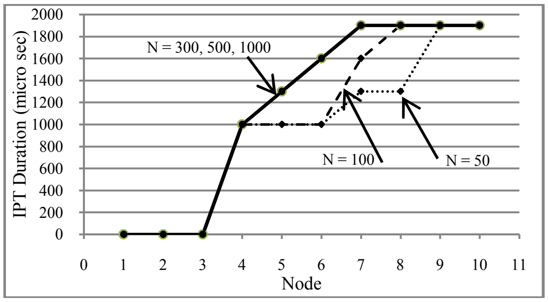

Fig. 14 Automatically calculated IPT durations in simulation scenario 1.

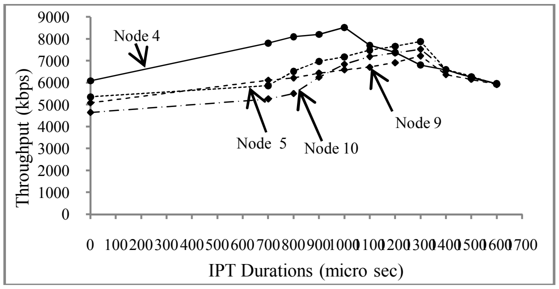

Fig. 15 IPT durations and end to end throughput for node 4, 5, 9, 10 in simulation scenario 2.

Table 3 Optimum IPT durations in simulation scenario 2 (µsec).

Node 4	Node 5	Node 9	Node 10
1000	1300	1300	1300

After that, we performed the proposed protocol to calculate the IPT durations for each slave node with the initial value D_0 to be 0, Δ to be 100 µsec.

5.1.2 Simulation results

The throughput vs. IPT duration for each slave node is shown in Fig. 12, 13 for scenario 1 and in Fig. 15 for scenario 2. The optimum IPT durations with which the end to end throughput is maximized are shown in Table 2 for scenario 1 and in Table 3 for scenario 2. The IPT durations obtained by the protocol for each slave node are shown in Fig. 14 for scenario 1 and Fig. 16 for scenario 2.

As shown in Fig. 14 and. 16, the IPT duration calculated by the protocol is zero for node 1, 2, 3 in scenario 1 and for node 1, 2, 3, 6, 7, 8 in scenario 2.

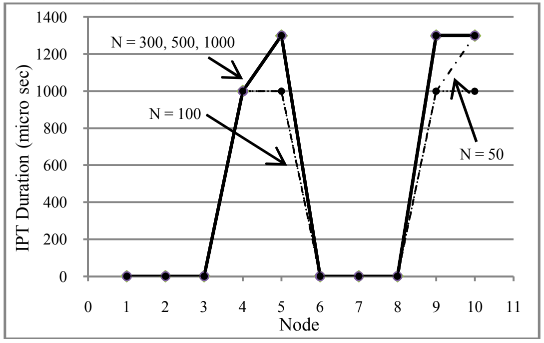

Fig. 16 Automatically calculated IPT durations in simulation scenario 2.

According to the feature of IPT forwarding, it can be easily understood that the IPT durations for the slave nodes which are located within the CSMA range of the core node could be set to zero because for these nodes no hidden terminal problem occurs and thus no need to purposely adjust the packet transmission time in the core node. For this reason, we deleted the throughput measurement results of these nodes in Fig. 12, 13 and 15.

In Fig. 14 and 16, the automatically calculated IPT durations match to the optimum ones in Table 2 and 3 with $N =$ 300, 500, 1000. However, with relatively small values of N (50, 100 in this case), the calculated IPT durations do not match to the optimum ones. This is because with such small N, the ratio of received training packets number and N varies intensively each time and the estimation of training time is not precise enough.

However, with the increment of N (larger than 300 in this case), these variations are suppressed and consequently the calculated IPT durations converge to the optimum values for each slave node.

The simulation results show that with adequate parameter settings, the proposed protocol can find the optimum IPT durations for each slave node with which the end to end throughput is maximized.

Fig. 17 Picomesh LunchBox

Table 4 Specification of LB.

CPU	AMD Geode LX800
Memory	DDR 512MB
Backhall Wireless IF	IEEE802.11b/g/a ×2
Access Wireless IF	IEEE802.11b/g/a ×1
OS	Linux kernel 2.6

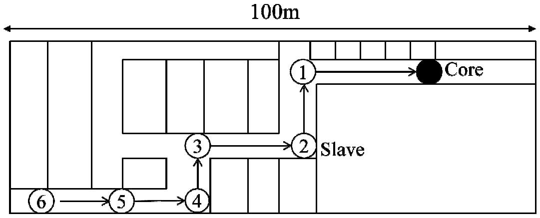

Fig. 18 Experimental site.

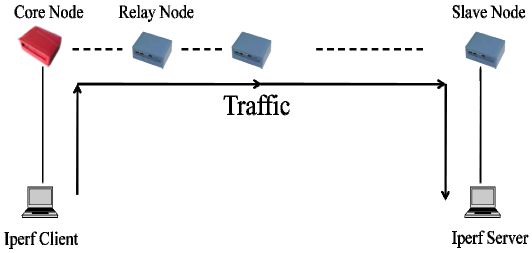

Fig. 19 Throughput measuring system.

5.2 Evaluation by Experiments

In order to further confirm its performance, we implement the proposed protocol into a testbed and evaluate its performance under real indoor environment.

5.2.1 Testbed and Throughput Measuring Tool

The testbed is called “PicoMesh LunchBox” (LB, Fig. 17). LB is the first product of “MIMO MESH Project”, which the authors are working on [14].

LB is equipped with three IEEE802.11 modules, two of which are used for relaying packets between base nodes and the other one is used for mobile terminal access.

Each module of LB is assigned with different spectrum so that the interference between these modules could be avoided. The hardware specification of LB is shown in Table 4.

In this experiment, we use IPerf to measure the throughputs [15]. IPerf is free software which can measure the end to end throughput in various networks with a pair of server and client. Additionally, we adopt UDP mode of its two operational modes (TCP mode and UDP mode) and measure the throughput from client to server.

5.2.2 Experimental Scenario

We deployed a wireless backhaul system with one core node and six slave nodes in West Building of ITO Campus, Kyushu-University, Japan (Fig. 18).

At first, we measured the end to end throughput of each slave node with different IPT durations by IPerf. Specifically, we set up the IPerf client in a PC which is connected to the core node and the IPerf server in a PC which is connected to the slave node (Fig. 19). During the measurement, traffic flows from IPerf client to sever and each measuring continues 30 seconds. We also assume that no extra traffic occurs during the measurement. In this first experiment, IPT durations are set manually for the purpose of searching an optimum IPT duration for each slave node. The optimum IPT durations are compared with the ones to be found by the proposed protocol, afterward.

In the next, we performed the proposed protocol to calculate the IPT durations for each slave node with the training packet number N to be 1000 and the initial value D_0 to be 1000, Δ to be 100 μsec.

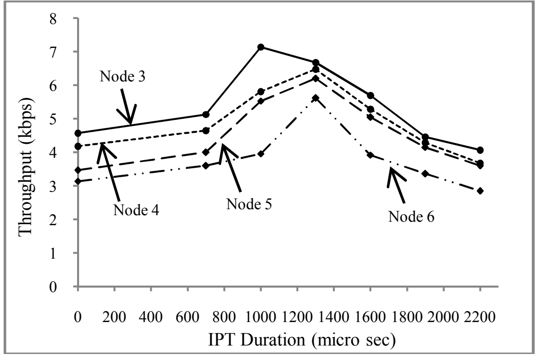

Fig. 20 IPT durations and end to end throughput for Node 3, 4, 5, 6 in experiment.

Table 5 IPT durations searched by the protocol (μsec) and its run time.

Node 1	Node 2	Node 3	Node 4
0	0	1000	1300
Node 5	Node 6	Run Time	
1300	1300	11 (sec)	

5.2.3 Result of the Experiments

The throughput vs. IPT duration for each slave node is shown in Fig. 20 and the IPT durations calculated by the proposed protocol and the protocol’s run time are shown in Table 5.

The calculated IPT durations of node 1 and 2 are zero in Table 5, which means that the two nodes are located within the CSMA range of the core node and thus we deleted the corresponding throughputs of the two nodes in Fig. 20.

As we can see from Fig. 20 and Table 5, the calculated IPT durations match to the optimum ones measured by IPerf with which the end to end throughputs reach the maximum values. With 6 slave nodes and 1000 training packets, the protocol spent 11 seconds to finish, which makes it practical enough in real applications.

5 Conclusion

In this paper we proposed a new IPT duration setting protocol which can calculate the optimum IPT duration for each slave node automatically. The proposed protocol is evaluated both with computer simulations and experiments on a real testbed in an indoor environment.

Evaluation results show that with the calculated IPT durations the end to end throughput of each slave node is maximized. Since the protocol does not demand modification of the existing standards, it could be easily implemented with general WLAN modules.

Additionally, we left the evaluation of inter-path IPT duration setting method as future work.

6 ACKNOWLEDGMENT

This work was supported by MIMO-MESH PROJECT under Grant of Knowledge Cluster Initiative implemented by Ministry of Education, Culture, Sports, Science and Technology, Japan.

7 REFERENCES

[1] G. Narlikar, G. Wilfong, L. Zhang, “Designing Multi-hop Wireless Backhaul Networks with Delay Guarantees,” in proc INFOCOM 2006. 25th IEEE International Conference on Computer Communications, pp.1-12, 2006.

[2] R. Pabst et al, “Relay-Based Deployment Concepts for Wireless and Mobile Broadband Radio,” IEEE Communication Magazine, pp.80-89, September 2004.

[3] R. Nelson and L. Kleinrock, “Spatial TDMA: A Collision Free Multihop Channel Access Protocol,” IEEE Trans. Comm., Vol. 33, No.9, pp. 934-944, September 1985.

[4] J. Gronkvist, J. Nilsson and D. Yuan, “Throughput of Optimal Spatial Reuse TDMA for Wireless Ad-Hoc Networks,” Proc. VTC’04 Spring, 11F-3, May 2004.

[5] H. Li, D. Yu and Y. Gao, “Spatial Synchronous TDMA in Multihop Radio Network,” Proc. VTC’04 Spring, 8F-1, May 2004.

[6] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee and R. Morris, “Capacity of Ad Hoc Wireless Network,” in Proc. ACM MobiCom 2001, July 2001.

[7] H. Zhai, J. Wang, Y. Fang, and D. Wu, "A Dual-channel MAC Protocol for Mobile Ad Hoc Networks, " in Proc. IEEE Workshop on Wireless Ad Hoc and Sensor Networks, in conjunction with IEEE GlobeCom, pp.27 -32, 2004.

[8] S.Bansal, R.Shorey, A.Misra, “Energy efficiency and throughput for TCP traffic in multi-hop wireless networks,” Proc INFOCOM2002, 23-27, pp. 210-219, 2002.

[9] H. Furukawa, “Hop Count Independent Throughput Realization by A New Wireless Multihop Relay,” Proc. VTC’04 fall, pp.-2999-3003, September 2004.

[10] Y. HIGA, H. Furukawa, “Experimental Evaluation of Wireless Multihop Networks Associated with Intermittent Periodic Transmit,” IEICE Trans. Comm., vol. E90-B, no.11, November 2007.

[11] E. M. Mohamed, D. Kinoshita, K. Mitsunaga, Y. HIGA and H. Furukawa, "An Efficient Wireless Backhaul Utilizing MIMO Transmission and IPT Forwarding", International Journal of Computer Networks, IJCN, vol. 2, No. 1, pp.34-46, March 2010.

[12] Guangri Jin, Ximing Huang, Hiroshi Furukawa, “Performance Enhancement of IPT forwarding for Wireless Backhaul by Fixed Directional Antennas”, Proc. IEEE APWCS 2011, Aug. 2011, Singapore.

[13] Y. HIGA and H. Furukawa, “Time Interval Adjustment Protocol for the New Wireless multihop Relay with Intermittent Periodic Transmit.” IEICE 2004, B-5-180, September 2004.

[14] http://mimo-mesh.com/en/.

[15] http://iperf.sourceforge.net/.