Traffic Flow Estimation based Channel Assignment for Wireless Mesh Networks

Wooguil Pak[1] and Saewoong Bahk[2]
[1] Jangwee Research Institute for National Defence
San 5, Woncheon-dong, Yeongtong-gu, Suwon 443-749, Korea
[e-mail: wgpak@ajou.ac.kr]
[2] Department of Electrical Engineering and Computer Science, INMC, Seoul National University
599 Gwanangno, Gwanak-gu, Seoul, 151-742, Korea
[e-mail: sbahk@snu.ac.kr]
*Corresponding author: Saewoong Bahk

Received October 25, 2010; revised November 24, 2010; accepted December 5, 2010;
published January 31, 2011

Abstract

Wireless mesh networks (WMNs) provide high-speed backbone networks without any wired cable. Many researchers have tried to increase network throughput by using multi-channel and multi-radio interfaces. A multi-radio multi-channel WMN requires channel assignment algorithm to decide the number of channels needed for each link. Since the channel assignment affects routing and interference directly, it is a critical component for enhancing network performance. However, the optimal channel assignment is known as a NP complete problem. For high performance, most of previous works assign channels in a centralized manner but they are limited in being applied for dynamic network environments. In this paper, we propose a simple flow estimation algorithm and a hybrid channel assignment algorithm. Our flow estimation algorithm obtains aggregated flow rate information between routers by packet sampling, thereby achieving high scalability. Our hybrid channel assignment algorithm initially assigns channels in a centralized manner first, and runs in a distributed manner to adjust channel assignment when notable traffic changes are detected. This approach provides high scalability and high performance compared with existing algorithms, and they are confirmed through extensive performance evaluations.

Keywords: Wireless mesh network, multi-channel, multi-radio, channel assignment, flow estimation

---

This work was partly supported by Mid-career Researcher Program through NRF grant funded by the MEST (No. 2010-0027750) and ITRC Support Program supervised by the NIPA (NIPA-2010-(C1090-1021-0011)) of the MKE, the Korean government.

DOI: 10.3837/tiis.2011.01.004

1. Introduction

Today, many countries have been delayed to transit into an information-oriented society due to a high cost of constructing information super-highway. Moreover, the cost for last mile is also a big burden even for countries that already have high speed networks. For example, it will cost one billion dollars and take 19 years to upgrade the last mile in the United States.

The wireless mesh network (WMN) is regarded as a promising solution for the last mile. It can provide internet connections with a very small installation cost, and increase the network performance by adding more mesh routers without significantly increasing costs because all routers except MPP (Mesh Portal Point) are connected through wireless links. Basically, WMN consists of mesh routers (Mesh Points, MPs) and mesh clients. MAP (Mesh Acess Point) is an MP which can provide access point functions to mesh clients. MPP is a special MP which connects with external networks. The wireless backbone network consists of MPs. Thanks to this feature, it can minimize the cost of network deployment compared with other network technologies. The WMN also supports self-configuration, self-healing and self-organization, which enable us to minimize the cost for network maintenance [1].

In terms of hardware aspects, technologies for WMN are already commercially available. However, researches for essential software parts such as channel assignment and topology construction are still in progress. For multi-radio multi-channel WMN, channel assignment is a very important component that mainly affects network performance [2][3], but it is very difficult to find an optimal channel assignment algorithm because the channel assignment is coupled with routing, traffic profile and channel interference. Due to their dependencies, it is not easy to make a distributed channel assignment algorithm with high performance.

Most of previous channel assignments perform in a centralized manner to achieve high performance but the centralized channel assignment can suffer from performance degradation. A central server gathers the whole network information from all MPs and distributes the channel assignment information to MPs. The channel assignment takes some time and causes many data packet drops because of temporal disconnections of links during the channel assignment. The situation becomes worse when the network experiences high dynamics.

In this paper, we propose a new estimation algorithm for flow rates between MAPs at first. It is simple, accurate and lightweight with very small control overhead. Then we propose a channel assignment algorithm that uses the flow information obtained from the flow estimation algorithm. After the initial centralized channel assignment, assigned channels can be changed in a distributed manner in order to sustain a high network performance regardless of changes in traffic flow. Since this hybrid channel assignment algorithm utilizes both of centralized and distributed algorithms, it can achieve high performance and high flexibility at the same time.

This paper is organized as follows. In Section 2, we briefly introduce the related work. In Section 3, we describe our flow estimation algorithm and hybrid channel assignment algorithm. We present the simulation results for performance evaluation in Section 4. Section 5 concludes our paper.

2. Related Work

In WMN, routing, channel assignment and traffic estimation are tightly coupled with each other. We briefly explain previous works for each of them in this section.

2.1 Flow Estimation

Flow estimation provides flow rate information between source and destination nodes. The information is essential in network planning and maintenance. Existing flow estimation methods investigate all packets in some selected routers for a sampling duration [4]. From these observations, they estimate characteristics of the whole network traffic. Their estimations become more precise with a longer sampling duration, but they consume higher computational power and require larger memory. Since they cannot consider packet drops, the estimation can fail if the network has a high packet drop rate.

2.2 Channel Assignment

Channel assignment algorithms can be divided into centralized [5][6][7] and distributed channel assignment [8] according to the requirement of a central server. In the centralized channel assignment, the central server calculates channel numbers for links of each MP by using network wide information. In the distributed channel assignment, each MP finds channel numbers for its own links by using partial network information. Due to the difference of network information, the centralized algorithm can achieve better performance than the distributed one. However, the distributed algorithm can reassign channels faster than the centralized algorithm according to changing traffic. Therefore, we should choose one of algorithms depending on the network stability.

2.2.1 Centralized Channel Assignment

The central server collects the information about traffic, topology and radio interfaces from all MPs, and then finds channels for links using a channel assignment algorithm. This result is broadcasted to the network. When each MP receives the channel assignment messages from the server, it switches channels. Since all the MPs switch channels at the same time, temporal disconnections of links and packet drops are unavoidable. This motivates the system not to run the channel assignment frequently. Consequently, the centralized approach is not suitable to support a network with high dynamics although it can find an optimal channel assignment with exact and up-to-date information. Well-known centralized channel assignment algorithms are interference based channel assignment and linear programming based channel assignment.

(1) Interference based channel assignment

It considers interference between neighbor nodes and assigns channels to minimize the interference [6]. It assumes uniformly distributed traffics in the network, so that it can estimate interference strength. Due to the feature, its algorithm can be simple. However, the performance can be deteriorated when the traffic distribution is biased.

(2) Linear programming based channel assignment

With topology and flow informations, it formulates link and node constraints and it finds an optimal channel assignment by *linear programming* [7]. This approach requires all information which affects the performance, i.e. network topology and traffic profile. However, it is impossible to predict the exact network topology and the traffic information in real world before a network deployment. For example, we can know the exact traffic characteristics only after the network starts and flows are generated. In real networks, we should use estimated and inaccurate information, and consequently, we cannot achieve high performance. Moreover, it can not reassign channels according to network dynamics because most of channels should be changed simultaneously. It can cause multiple link disconnections and network partitioning, which incur a high packet drop rate.

2.3 Distributed Channel Assignment

The distributed approach uses partial information about topology and traffics in channel assignment. Against the centralized approach, each MP can update channels of its own links whenever necessary, and experiences less packet drops thanks to asynchronous channel assignment. It has a merit to support incremental channel updating. However, the channel assignment result is not optimal because it is based on the partial information of the network, and thereby its performance can be poorer than the centralized one if the network is very stable. One of well-known algorithm is MPP-oriented channel assignment. In WMNs, traffics are generally concentrated into an MPP. This algorithm uses this characteristic and the assumption that traffics are distributed uniformly throughout the overall network [8]. In order to provide higher throughput, the link closer to the MPP will be assigned to a better channel, i.e. a less interfered channel. The strong merit of this approach is simplicity because it does not need any real traffic flow information. However, traffics are not distributed uniformly in reality and inter-MAP traffics can not be ignored because P2P services become popular. Due to this difference between reality and assumptions, the performance can be poor.

3. Hybrid Channel Assignment Algorithm

Proposed algorithm assigns channels in a hybrid manner with the traffic information obtained by its own flow estimation algorithm and thereby it can achieve high performance and flexibility.

There have been many hybrid channel assignment algorithms which combine features of static and dynamic channel assignments [2][9]. They assign some radio interfaces to channels permanently but others dynamically. However, our hybrid channel assignment algorithm merges merits of centralized and distributed algorithms instead of static and dynamic algorithms. It assigns optimal channels by using a centralized algorithm and then, updates channels in a distributed manner when traffic flow changes. The distributed channel assignment is restrictively used not to decrease throughput of existing flows. If distributed channel assignment cannot adapt channels efficiently, the centralized channel assignment is triggered. Therefore, our algorithm totally differs from existing hybrid channel assignment algorithms.

Our channel assignment algorithm belongs to a category of flow level algorithms which objective is to maximize flow level throughput [10][11]. The flow level algorithm requires flow information to assign channels and it is generally implemented with a routing protocol for cross-layer optimization. Since it uses the entire network and flow information, it can achieve better performance than other link level algorithms that exploit link level information such as strength of interference between nodes or links, and bandwidth of links. However, it is very difficult to obtain flow information in real networks.

For exact flow estimation, our flow estimation algorithm tries to find the incoming amount of traffic regardless of packet drops. Since it calculates the aggregated traffic size between incoming and outgoing routers, i.e. MAPs, instead of traffic size per flow or per session between end-to-end nodes, proposed flow estimation algorithm is very scalable. To avoid ambiguity, ‘flow’ means the aggregated flow between MAPs. For example, a flow rate from MAP A to MAP B means total aggregated traffic rates incoming from MAP A and outgoing to MAP B. We also use ‘traffic’ to generally designate delivered packets regardless of flows.

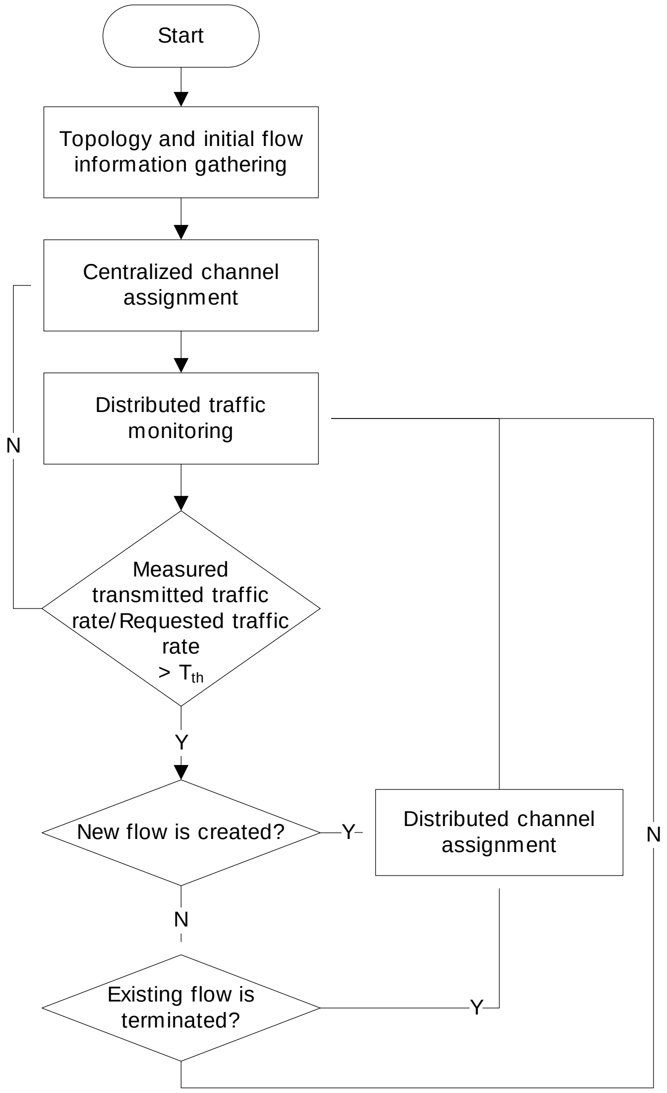


Fig. 1. Hybrid channel assignment procedure.

When a network starts, MPP becomes a central server for the centralized channel assignment and gathers topology information such as MPs and links, and initial flow if available. Based on these informations, MPP can find channels for links of the network. Each MP monitors traffic flows and maintains a flow information table. To decide which channel assignments are used, several conditions can be used. For example, if any MP recognizes that it suffers from a severely poor performance or a significant change in the network topology; it can request the MPP for channel re-assignment in a centralized manner. When MP detects flow creation and termination, it can adjust assigned channels for its links in a distributed manner. It chooses a new channel carefully not to decrease the performance of existing flows, and the new channel information will be delivered to other MPs and MPP. If the MPP counts total number of changed links and it exceeds a predefined threshold value, it can reassign the entire network channels for a better performance by using centralized channel assignment. In this paper, we consider only the ratio of measured transmitted traffic rate to requested traffic rate. If the ratio is below the predefined threshold value, T_{th} , the network has a poor performance. In this case, the MP requests MPP for centralized channel assignment. T_{th} is set by a network administrator according to QoS requirement of the network. In this paper, we set T_{th} to 0.3. Brief procedures of hybrid channel assignment are shown in **Fig. 1**.

3.1 Flow Estimation Algorithm

It finds aggregated flow rates between MAPs or between MAPs and MPP instead of session level flow rates. MAPs and MPP measure the total traffic rate incoming from mesh clients or external networks, denoted by R_{in}^{m}, where m is the MAP or MPP ID (Identification number). Every T_s, router m samples an incoming packet randomly and creates a new packet named FIP (Flow Information Packet). The FIP header has the same source and destination IP addresses as the sampled packet. We assign a protocol number to FIP_PROTO[1]. The FIP payload contains the reception router ID, i.e. m, and the incoming traffic rate, i.e. R_{in}^{m}. Intermediate routers process the FIP with a high priority to avoid its drop even if the network traffics are congested. The packet processing diagram in the transmission MP is shown in **Fig. 2**.

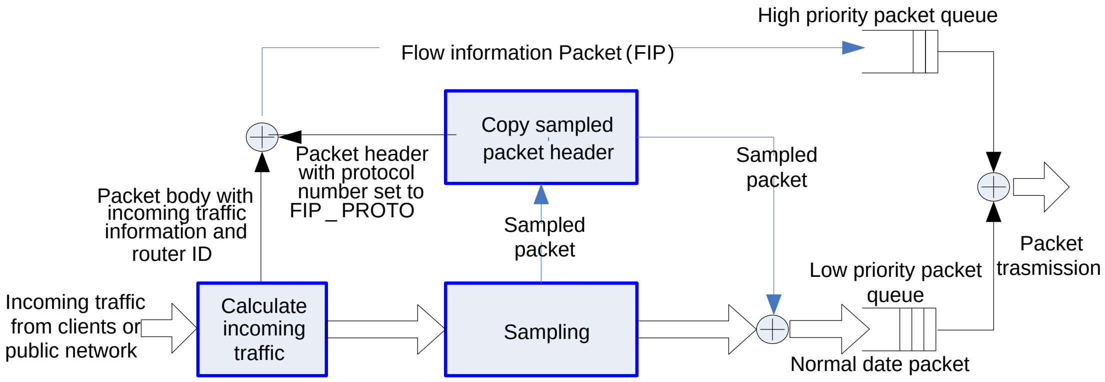


Fig. 2. Flow information packet creation and transmission in MP.

Whenever the destination MAP or MPP, i.e. MP n, receives the FIP, it counts the number of FIPs received from MP m, denoted by N_{n}^{m}, during T_W where T_W is a sliding window size and T_W □ T_S. MP n extracts m and R_{in}^{m} from the FIP, and estimates the total flow rate R_{in}^{m} from MP m to n as follows.

$$R_{n}^{m} = \frac{T_S}{T_W} \cdot N_{n}^{m} \cdot R_{in}^{m}. \tag{1}$$

If we have smaller T_S, we can measure more exact traffic rate but control overheads increase. However, if we have smaller T_W, our estimation can be wrong but we can adjust network performance without delay. Therefore, they should be selected by administrators

---

[1] FIP_PROTO is set to 244 to avoid confusion with existing protocols.

according to the network characteristics. In this paper, we assume that T_S and T_W are set to 10 sec and 2 min, respectively. The processing diagram for FIP in MP is shown in **Fig. 3**.

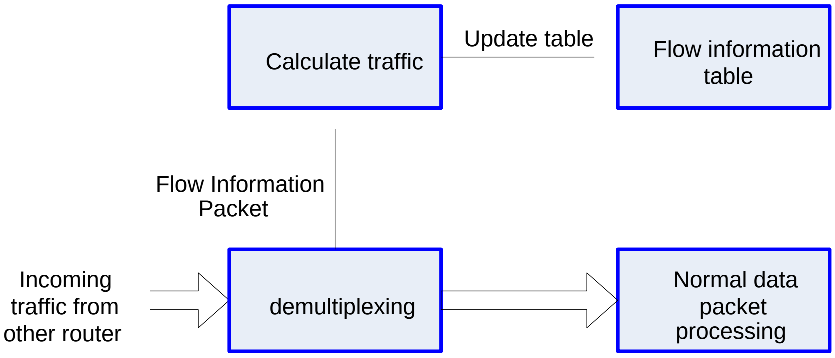


Fig. 3. Flow information table update with FIP in MP.

3.2 Channel Assignment Algorithm

When a network is deployed, all channels of MPs are initially set to a common channel. All the MPs send their neighbor lists and flow informations[2] to a central server, i.e. MPP for central channel assignment. The centralized channel assignment can also be performed whenever any MP requests. This happens when MP detects severe change in flow characteristics, channel quality or topology. When MPP receives all informations from all the MPs, it finds new routing paths for active flows, and assigns channels according to traffic load of each link.

If the residual bandwidth of link l is less than $p \cdot W_l$, where $0 < p < 1$ and W_l is maximum bandwidth of link l, link l is assigned to a common channel and it is classified as a *Re-allocatable link*. During distributed channel assignment, only channels for re-allocatable links can be reassigned. This approach guarantees that bandwidth for existing flows are not affected, and serious link disconnections or packet drops are not occurred during the distributed channel assignment. Now, we will describe our centralized and distributed channel assigment algorithms.

3.3 Centralized Channel Assignment

MPP collects network-wide flow information for all MPs. It also has the information about topology and MPs such as the number and type of radio interfaces of each MP, each channel status and neighbor MPs.

3.3.1 Flow Normalization

The total incoming and outgoing flow rates at node n can not be greater than the total bandwidth of links of node n. We call this limit *physical interface constraint*. If the required total rate is bigger than the constraint, we linearly scale it down to meet the constraint. This flow normalization is repeated until all flows satisfy the constraints for all MAPs and MPP

3.3.2 Topology Generation

After flow normalization, MPP finds a minimum cost path for each flow. We define the cost of link l, i.e. C_l as

---

[2] If the information is not available, it can be skipped and estimated flow information is used.

$$C_l = \begin{cases} 1+M & \text{if } w_l - f_i < -M, \\ 1-(\rho \cdot w_l - f_i) & \text{if } -M \le \rho \cdot w_l - f_i \le 0, \\ 1 & \text{if } w_l - f_i > 0, \end{cases} \tag{2}$$

where w_l is the residual bandwidth before the ith flow is assigned, f_i is the normalized bandwidth of the flow, M is a positive constant and ρ is set to 1 if one radio interface is dedicated to link l, and 2 if link l shares a radio interface with other link(s)[3]. **Fig. 4** shows an example of the cost. The path cost is defined by the total sum of costs for all links which belong to the path.

We assign each flow into the network in descending order of the sizes of flows. Depending a path cost, it is possible to share one radio with other links to increase connectivity or to assign one link into one radio to increase link bandwidth. Consequently, it usually generates more meshed topology instead of simple tree topology of existing algorithms. This feature enables us to support inter-MAP flows as well as MAP-MPP flows.

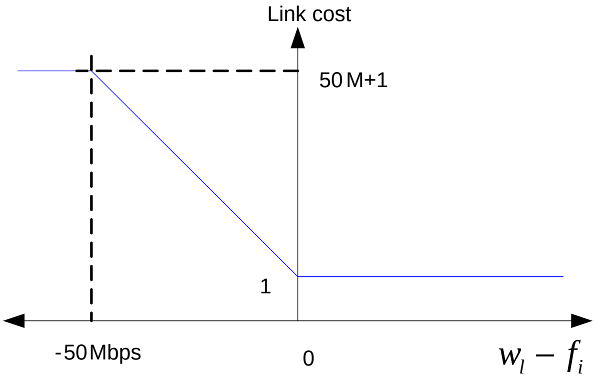


Fig. 4. Link cost according to residual bandwidth when M =50Mbps and ρ =1.

After assigning all flows, we assign a less interfered channel to a link with a higher load.

3.4 Distributed Channel Assignment

When a new flow is generated, its throughput can be poor. Then the end MAP of the flow, i.e. destination router, tries to find/make a new path by changing channels of re-allocatable links for better performance. Since it relies on proactive link state routing protocol, it finds the path based on all possible connectivities using channel switching and it chooses channels for re-allocable links by considering required flow rates, inter and intra flow interferece. These procedures are similar with the centralized channel assignment but only re-allocable links can be considered in distributed channel assignment. Therefore, it results in very fast channel assignment compared with centralized channel assignment, and existing flows can be free of performance degradation. We present a flowchart for the distributed channel assignment in **Fig. 5**.

---

[3] It is for the case that the total number of required links is bigger than that of interfaces.

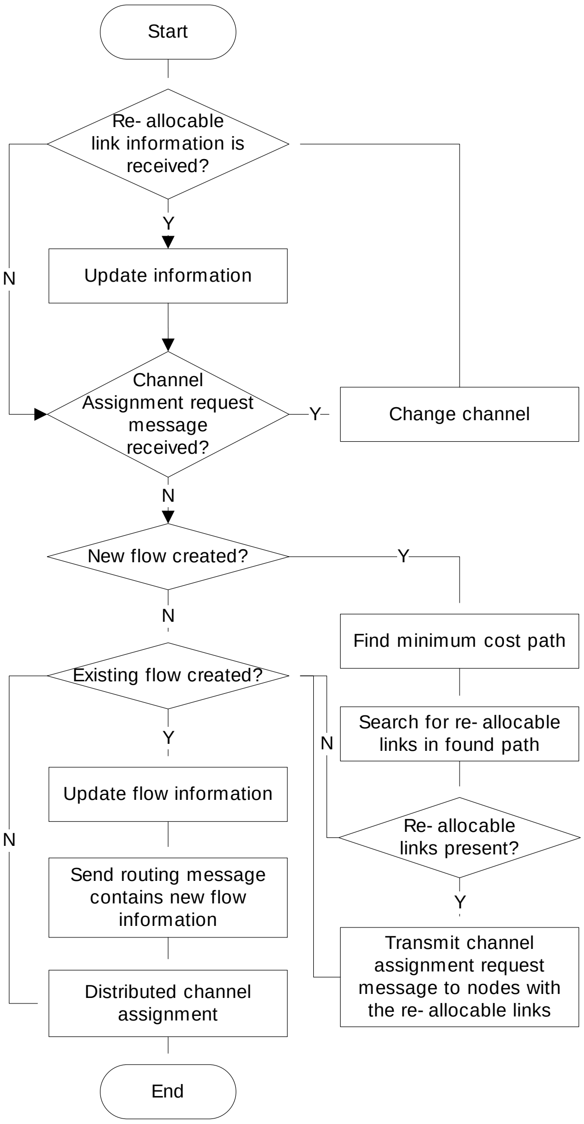


Fig. 5. Distributed channel assignment.

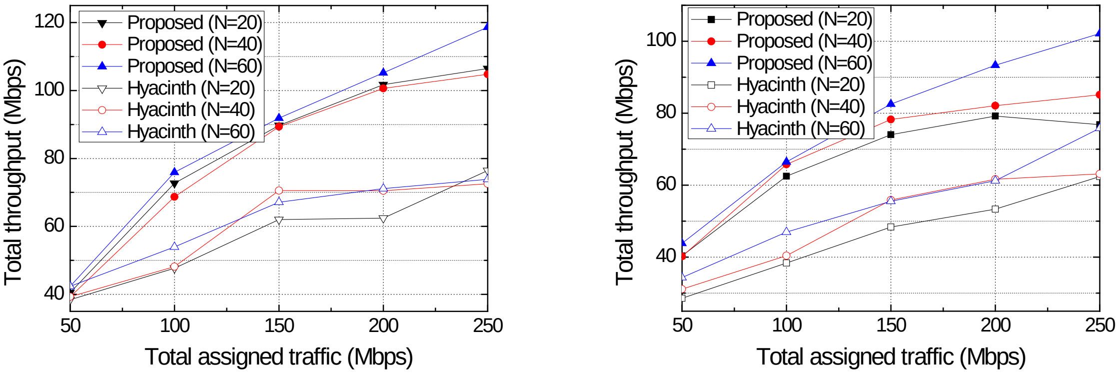
4. Performance Evaluation

The performance of our algorithm was compared with that of a well-known centralized channel assignment algorithm named *Hyacinth* [5], which performs the best from currently availble protocols including centralized and distributed protocols [8]. We used NS-2 simulator which version 2.29 for the performace evaluation. We used a two-ray path loss model to measure received signal powers in physical layer. We also used the same model to obtain total interference power by calculating received signal powers from concurrent transmitting nodes.

In simulations, we use a 7×7 grid topology where the outer-most routers are MAPs, the center router is a MPP, and others are plain MPs. The minimum distance between MPs is 100 m, the transmission range is 110m and the interference range is 220 m. Each MP has three 802.11a radio interface cards and each card has 12 disjoint channels. We used UDP and TCP streams and set the packet size to 1000 bytes. We set p and M to 0.1 and 50Mbps, respectively. We also assumed that channel switching takes 20 msec. The total simulation time is 70 secs. For *chnl* program in Hyacinth, *remove-edges* is disabled and the threshold value is set to 1.0 for all cases.

4.1 Comparison of Performance without Flow Update

We generated N flows simultaneously and measured the total throughput for last 20 secs to show the converged performance. We assume that the flow information is available before channel assignment. All flows start simultaneously after centralized channel assignment is finshed. Routing paths for flows are obtained during channel assignment. All the flows remain until the simulation finishes. We chose each flow rate according to uniform distribution from 0 to $2S/N$ where S is the total assigned traffic rate. As shown in **Fig. 6-(a)**, our proposed algorithm improves the performance by at least 40 % compared with Hyacinth when UDP streams are used. With the total flow rate increasing, the performance gap also increases. Since our proposed algorithm creates a higher degree of connectivity, the network can avoid traffic concentration and increase the performance. Although the difference is small, we have more chance to achieve higher performance with bigger N. Therefore, we can achieve the best performance when N=60.



(a) UDP stream **(b)** TCP stream

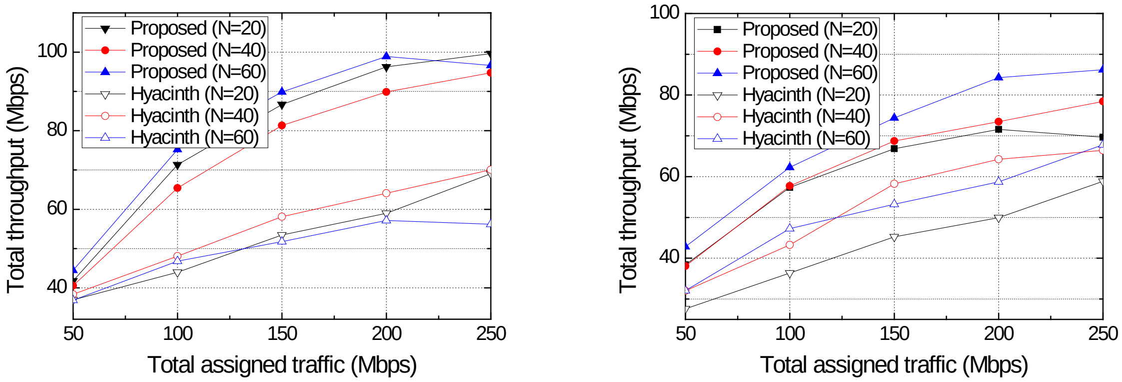
Fig. 6. Total average throughput without flow update (N: Number of flows).

For TCP streams, our algorithm shows about 30 % higher performance than Hyacinth in **Fig. 6-(b)**. Although the improvement ratio is decreased slightly, it shows almost the same tendency with the simulation results with UDP streams. From these results, we can say that our algorithms can improve the total throughput regardless of protocol types. It also shows the best performance when N=60.

4.2 Comparison of Performance with Flow Update

We also compared the performance when flows are updated. We generated N flows simultaneously when the simulation time is 5 secs, and one flow is selected randomly to be terminated and a new flow is created with the same traffic rate of the terminated flow but with randomly selected source and destination nodes every second. If the new flow is created, Hyacinth finds the routing path based on proactive routing protocol. Our proposed algorithm also relies on proactive routing protocol at first but the path can be changed according to distributed channel assignment algorithm. When the simulation time is 65 secs, only newly created flows remain. We assume that only initial flow information for N flows is available before initial centralized channel assignment.

In **Fig. 7**, we can see the average performance for last 20 secs of the simulation. Thanks to the distributed channel assignment of the proposed algorithm, the performance ratios of proposed algorithm to Hyacinth are slightly increased when UDP streams are used as shown in **Fig. 7-(a)**. However, the ratios decreased by about 20 % when TCP streams are used as shown in **Fig. 7-(b)**. In simulations with proposed algorithm, some packets can be delayed or dropped during channel switching of links belong to paths of packets. It can cause performance degradation more seriously for TCP than UDP due to packet retransmission.



(a) UDP stream **(b)** TCP stream

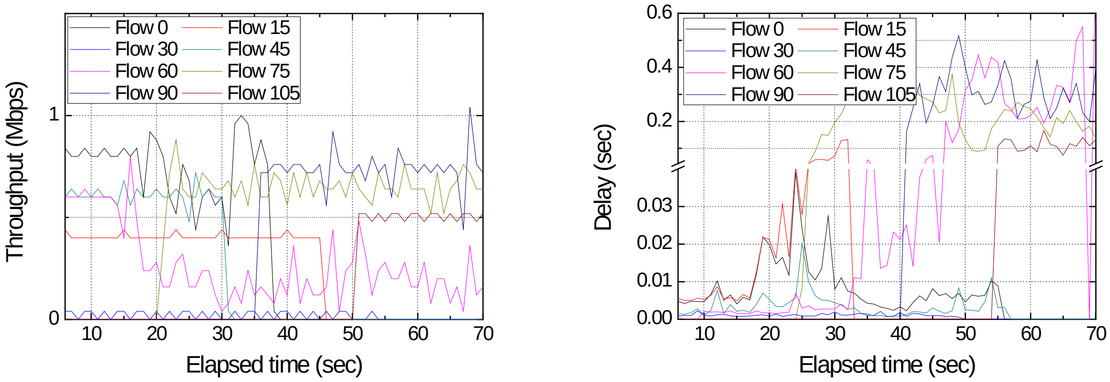
Fig. 7. Total average throughput with flow update (N: Number of flows).

4.3 Comparison of Performance for Individual Flows with Flow Update

To find how much performance of existing flows can be affected according to the total number of updated flows, we measured the total delivered traffic rates and delay of flows every 1 secs. Since simulation results show almost the same tendency independently with total flow numbers and traffic rates, we show the simulation results only when N=60 and total traffic rate is 50 Mbps. To make the graph simple for better understanding, we choose every 15th created flow among 60 initial flows and 60 new created flows. Only selected flows are shown in **Fig. 8** to **11**.

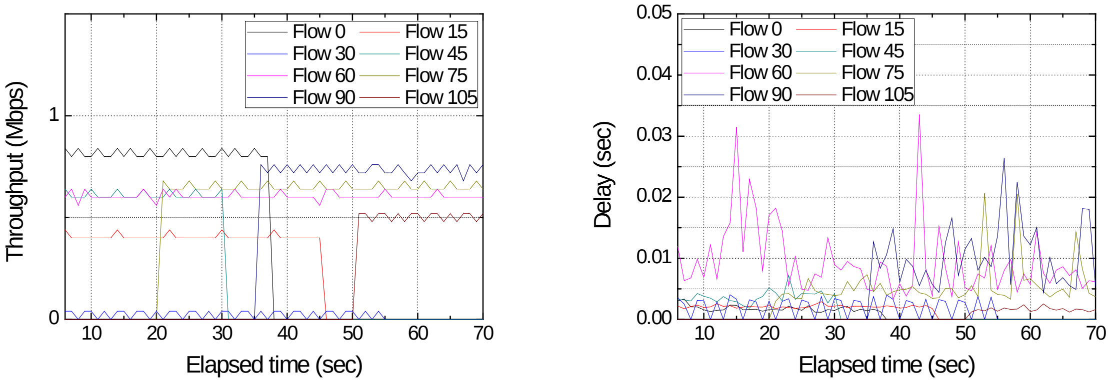
When Hyacinth algorithm is used with UDP streams, existing flows show stable throughput and end-to-end delay in the early simulation time. As the total number of update flows is increased, throughputs of flows are fluctuated or decreased significantly as shown in **Fig. 8**. For example, the throughput of flow 60 decreases by 70 % compared with the initial throughput. Newly created flows show very higher end-to-end delay than that of initial flows. It is obvious that they tend to have long paths since Hyacinth cannot apply channel switching dynamically according to updated traffics. The longer path incurs more resource consumption and higher interference, so that the overall network performance are degraded more and more with the number of updated flows increasing.

When our algorithm is used, both of throughput and end-to-end delay become very stable as shown in **Fig. 9**. Although distributed channel assignment causes addition delay from channel switching, we can find that average end-to-end delay of proposed algorithm is smaller than that of Hyacinth. The reason is that overall, an advantage of short routing paths obtained by dynamic channel assignment compensate for delay caused by channel switching. However, it cannot be applied for all dynamic channel assignment algorithms because our distributed channel assignment algorithm is used only under the very restricted condition to minimize its side effects.



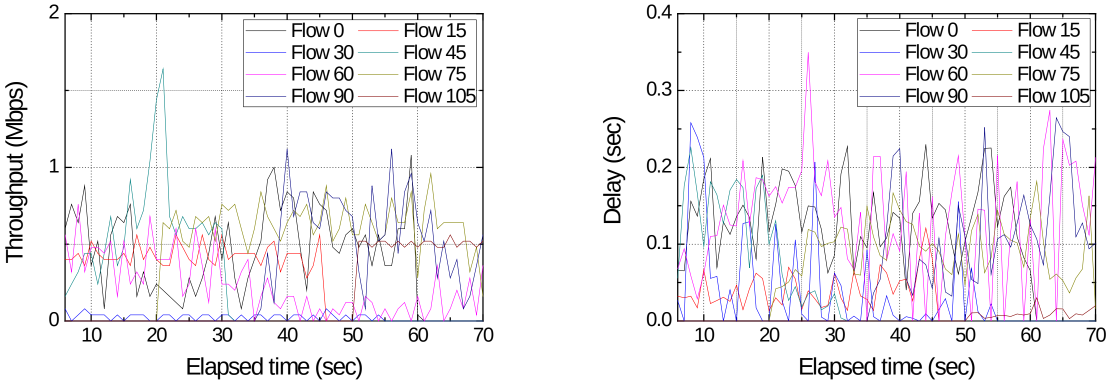
(a) Throughput **(b)** End-to-end delay

Fig. 8. Performance of selected UDP flows according to elapsed time when Hyacinth is used where *N*=60 and total traffic rate is 50Mbps.



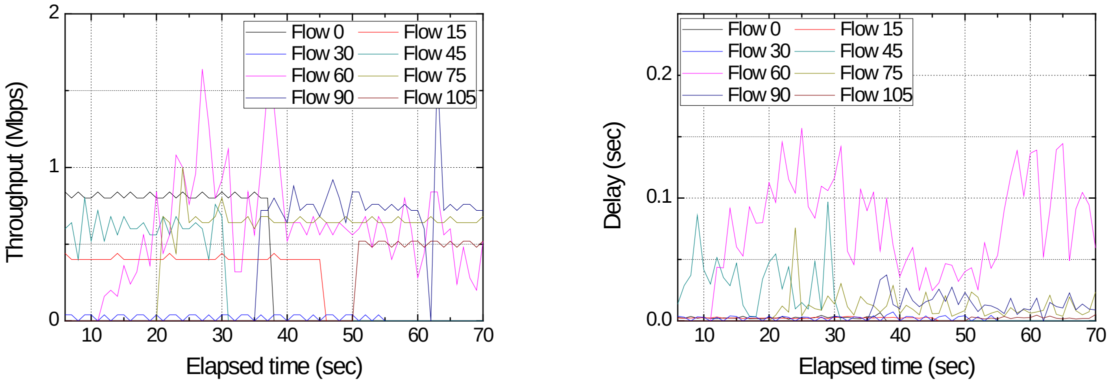
(a) Throughput **(b)** End-to-end delay

Fig. 9. Performance of selected UDP flows according to elapsed time when proposed algorithm is used where *N*=60 and total traffic rate is 50Mbps.



(a) Throughput **(b)** End-to-end delay

Fig. 10. Performance of selected TCP flows according to elapsed time when Hyacinth is used where *N*=60 and total traffic rate is 50Mbps.



(a) Throughput **(b)** End-to-end delay

Fig. 11. Performance of selected TCP flows according to elapsed time when proposed algorithm is used where N=60 and total traffic rate is 50Mbps.

The most important result is that new flows do not decrease the performance of existing flows. It means our distributed algorithm effectively decreases inter-flow interference. Also, the stable throughput of each flow shows that the algorithm successfully allocates required bandwidth for each flow. Therefore, we can estimate that it assigns channels to avoid intra-flow interference increasing.

From these results, we can verify that our proposed algorithm can adapt channel assignment and topology very efficiently in a distributed manner even though we use the distributed channel assignment very restrictively.

For TCP streams, each flow experiences a longer end-to-end delay and shows more fluctuated performance results compared the UDP stream case as shown in **Fig. 10** to **11**. As expected, these results causes from TCP retransmission. However, the throughput for proposed algorithm is relatively stable more than that for Hyacinth algorithm. The delay for proposed algorithm is also very small compared with Hyacinth.

From these simulations, we can confirm that our proposed algorithm achieves better throughput and delay performances for stable or dynamic networks. Also, we can verify that it finds routing paths and adjust channels distributedly to obtain high performance for new flows while minimizing the damage to performances of existing flows. This feature is very important to support various QoS in WMNs.

5. Conclusions

In this paper, we proposed a flow estimation method and a hybrid channel assignment algorithm for multi-radio multi-channel wireless mesh network (WMN) environments. In general, previous researches assumed that each flow rate is given in advance, and tried to balance between throughput and flexibility. To overcome this problem, our proposal measures aggregated flow rates on-line in a simple manner and these results are used in the channel assignment. Moreover, our hybrid channel assignment algorithm works in a centralized way at the beginning for high performance and uses a distributed approach for high flexibility. Thanks to the distributed channel assignment, it is able to minimize packet loss that usually occurs during channel switching. It can also improve the throughput and delay performance. Moreover, it can support various QoS in WMN very efficiently and works well even under heavily congested environments with high scalability.

References

[1] I. F. Akyildiz, X. Wang and W. Wang, “Wireless mesh networks: a survey,” *Elsevier Computer Networks*, vol.47, iss.4, pp. 445-487, March 2005. Article (CrossRef Link).

[2] P. Kyasanur and N. H. Vaidya, “Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks,” *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, vol.10, no.1, pp. 31-43, January 2006. Article (CrossRef Link).

[3] P. Bahl, R. Chandra and J. Dunagan, “SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks,” in *Proc. of ACM International Conference on Mobile Computing and Networking (Mobicom)*, pp. 216-230, September 2004. Article (CrossRef Link).

[4] J. Jedwab, P. Phaal and B. Pinna, “Traffic estimation for the largest sources on a network, using packet sampling with limited storage,” *Hewlett-Packard Company technical report*, March 1999.

[5] A. Raniwala, K. Gopalan and T. –C. Chiueh, “Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks,” *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, vol.8, no.2, pp. 50-65, April 2004. Article (CrossRef Link).

[6] K. N. Ramachandran, E. M. Belding-Royer, K. C. Almeroth and M. M. Buddhikot, “Interference-aware channel assignment in multi-radio wireless mesh networks,” in *Proc. of IEEE International Conference on Computer Communications (Infocom)*, pp. 1-12, April 2006. Article (CrossRef Link).

[7] M. Alicherry, R. Bhatia and L. Li, “Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks,” in *Proc. of ACM International Conference on Mobile Computing and Networking (Mobicom)*, pp. 58-72, August 2005. Article (CrossRef Link).

[8] A. Raniwala and T. Chiueh, “Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network,” in *Proc. of IEEE International Conference on Computer Communications (Infocom)*, vol.3, pp. 2223-2234, March 2005. Article (CrossRef Link) .

[9] V. Raman and N. H. Vaidya, “SHORT: A static-hybrid approach for routing real time applications over multichannel, multihop wireless networks,” *Springer Wired/Wireless Internet Communications, Lecture Notes in Computer Science*, vol.6074, pp. 77-94, 2010. Article (CrossRef Link).

[10] F. Ye, S. Roy and Z. Niu, “Flow oriented channel assignment for multi-radio wireless mesh networks,” *EURASIP Journal on Wireless Communications and Networking*, vol.2010, Article ID 930414, 2010. Article (CrossRef Link).

[11] J. Tang, G. Xue, and W. Zhang, “Cross-layer optimization for end-to-end rate allocation in multi-radio wireless mesh networks,” *Springer Wireless Networks*, vol.15, no.1, pp. 53-64, 2009. Article (CrossRef Link).

Wooguil Pak received B.S and M.S. degrees in Electrical Engineering from Seoul National University in 1999 and 2001, respectively, and the Ph.D. degree in Electrical Engineering and Computer Science from Seoul National University in 2009. From 2001 through 2009 he was with Samsung SECUi.COM as a software engineer. In 2010, he joined the Jangwee Research Institute for National Defence as a research professor. His current research interests include MAC and routing protocol design and QoS provisioning for wireless sensor network and wireless mesh network, and network security for high speed networks.

Saewoong Bahk received B.S. and M.S. degrees in Electrical Engineering from Seoul National University in 1984 and 1986, respectively, and the Ph.D. degree from the University of Pennsylvania in 1991. From 1991 through 1994 he was with AT&T Bell Laboratories as a member of technical staff where he worked for AT&T network management. In 1994, he joined the school of electrical engineering at Seoul National University and currently serves as a professor. He has been serving as TPC members for various conferences including ICC, GLOBECOM, INFOCOM, PIMRC, WCNC, etc. He is on the editorial board of Journal of Communications and Networks (JCN). His areas of interests include performance analysis of communication networks and network security. He is an IEEE senior member and a member of Whos Who Professional in Science and Engineering.