Session-oriented Adaptive Routing for Improving Path Stability in WLAN based Mesh Networks

Yu-Jin Son, Sung-Hee Lee and Young-Bae Ko

School of Information and Computer Engineering,
Ajou University, Suwon, Republic of Korea

yujin@uns.ajou.ac.kr, {sunghee, youngko}@ajou.ac.kr

Sang-Joon Park and Yoo-Seung Song

Electronics and Telecommunications Research Institute,
Republic of Korea

{yssong00, sangjoon}@etri.re.kr

Abstract—**The IEEE 802.11s standard is designed to support multi-hop mesh networking between WLAN devices. It features the HWMP (Hybrid Wireless Mesh Protocol) as its default path selection protocol which combines the proactive tree building mode and the on-demand path selection mode. The proactive mode can periodically maintain the lowest link cost route to the root node, but poses the path instability problem, where the path is unnecessarily changed when the path is updated. As a result, the performance of the network is degraded. To alleviate this problem, we propose the Session-oriented Adaptive Routing (SOAR) protocol, which recognizes each data session and prevents path switching while the data session is in progress. To achieve this, we do not switch the existing path even though the link cost may be higher than the new path, unless the potential throughput gain of the new path is considered better than the pre-determined threshold. Moreover, the new path with lower link cost is also stored and used when a new data session is initiated. We have conducted preliminary simulation studies via the ns-3 to evaluate the performance of the proposed scheme.**

Keywords—*Wireless mesh networks; Path instability; Adaptive routing*

I. Introduction

As the next core networking technology, wireless mesh networks (WMNs) have received considerable attention as it can provide high scalability and easy deployment methods with relatively low costs by utilizing wireless multi-hop routing [1]. As one of the guidelines for designing a mesh network, IEEE have organized the 802.11s task group for standardizing WMN, and have recently released the Draft 7.0 version, nearing its completion [2]. Utilizing the 802.11s standard can provide benefits in that it suggests a unified method of implementation which improves compatibility between mesh devices.

The IEEE 802.11s standard defines the *Hybrid Wireless Mesh Protocol* (HWMP) as its standard multi-hop routing protocol. The HWMP is distinctive compared to other routing protocols in that it uses the *Airtime link metric* by calculating the frame error rate and transmission bit rate as the important factors for selecting an optimal multi-hop path. The HWMP uses both the on-demand mode that is identical to the algorithms proposed in AODV [3] and the proactive tree building mode which periodically manages tree routing paths from the root node to each other mesh node. In the on-demand mode, a source broadcasts *Path Request* (PREQ) messages upon a data transmission request from the application layer. Nodes that receive the PREQ message update the lowest link cost path to the source node by utilizing the airtime link cost and also the sequence number in the PREQ. By transmitting back a *Path Reply* (PREP) message back to the source node, the destination node can reinforce a bi-directional multi-hop path.

The proactive tree building mode, which creates routes proactively before data transmission requests are made, is divided to the proactive PREQ mechanism and the proactive RANN mechanism. In the proactive PREQ mechanism, a two-way handshaking method is used where PREQ messages are periodically broadcasted by the root node and each node transmits PREP messages back to the root node to reinforce the route. In the proactive RANN mechanism, a three-way handshaking is used where RANN messages are advertised by the root node, PREQ unicasted in reply to the root node, and PREP transmitted back to each node for route reinforcement.

The proactive tree building modes can provide ready-made routes before any data transmission is actually triggered, providing speedy transmission which is required in variety of QoS related applications. However, the proactive building mode poses the path instability problem which means the path is frequently changed at the periodic update interval. The path instability causes the network to wrongly switch routes to the root node due to miscalculation in the current link cost. When applications are active and under transmission while a route switch occurs, this may cause degradation in throughput and end-to-end delay. Although the similar phenomenon in wired and wireless networks has been identified and stated in several works [4, 5], there is no proper solution to efficiently solve this problem in the standard based WMN.

[4] directly defines and analyses the path instability problem and its effect on the network by evaluating the problem in UCSB MeshNet and the MIT Roofnet test-beds. Even though the problem is thoroughly analysed and the problem is well defined, the authors do not provide a proper solution to the problem and only suggests methods that may solve the problem. [5] analyses the path instability problem of the OSPF protocol in the wired network, and proposes the modified OSPF-TE which can distribute network traffic flow to each link in a balanced way. The OSPF-TE protocol can re-search a path after route failure report, and maintain stability by successfully recovering these routes. However, in case of when frequent failure occurs in the network, processing complexity for each node can increase exponentially, affecting the performance of the network. Moreover, it is not well

applicable to the wireless network due to inherent difference between them.

This paper tries to alleviate the path instability problem by defining a threshold for limiting the frequency of route switching. First, in contrast to existing related works, we analyse the path instability problem that occurs in the HWMP. Also, the algorithm recognizes each data session and prevents path switching while the data session is active and under transmission. Even though the route may not change, alternative routes are stored and later used when new data sessions are initiated. Via simulation studies through ns-3 [6], the performance of the proposed scheme is evaluated and compared with the HWMP to show that our scheme can achieve better network performance.

II. Preliminary Analysis: Path instability Problem of the HWMP

This section analyses the path instability problem that occurs in the HWMP via ns-3 simulation studies. 7 nodes were deployed in the network to create a multi-hop topology, as shown in Figure 1.

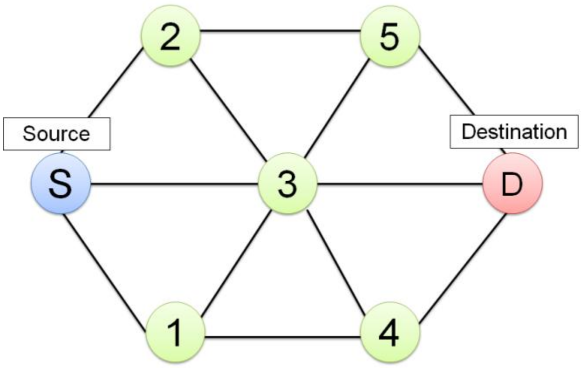


Figure 1. Test topology

Each node is installed with 802.11a radio and capable of transmitting data with rate of 54Mbps. The source node initiates TCP data at 5 seconds and continues transmission until 40 second interval. The result of the simulation is shown in Figure 2.

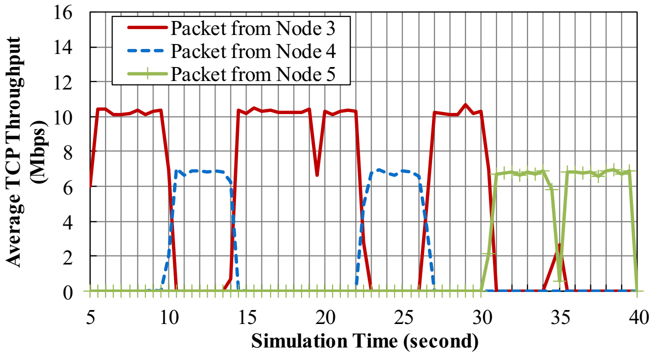


Figure 2. Path instability of the HWMP

Three samples are achieved from the simulation. As seen from Figure 1, the destination node receives data from three different relaying mesh nodes, which are node 3, 4, and 5. Data that are transmitted through node 4 and 5 are three-hop transmission, while data transmitted through node 3 is only a two-hop transmission. Therefore, generally we can consider that the two-hop path through node 3 can provide higher throughput than the longer three hop routes through node 4 and 5. Indeed, as seen in the graph, transmission through node 3 guarantees more than 10Mbps rate, while the other routes show less than 7Mbps transmission rate. Although data transmission through node 3 is ideal and most efficient, the simulation shows that routes through node 4 and node 5 are also used, actually degrading the performance of the network.

This is due to the periodical advertising of the RANN or PREQ message in the proactive tree building mode of HWMP. While the TCP session is active, HWMP triggers a route advertisement that is propagated through the network to calculate the airtime link cost and update each route. Even though the two-hop route through node 3 has the lowest link cost route, the transmission of the TCP session actually causes the link to degrade. Therefore, the calculation of the airtime link cost of the route through node 3 actually increases beyond the cost of multi-hop routes through node 4 and node 5. As a result, less optimal routes through node 4 and node 5 are selected and the overall network throughput decreases. Once the transmission through node 4 or node 5 initiates, the link cost of the route through node 3 becomes low again and selected at the next root advertisement. However, node 3 is not maintained and other routes are selected, showing the path instability problem defined in [3]. To solve this problem, we provide several methods to constrain the fluctuation in the selection of routes and further provide solution for efficiently transmitting data by distinguishing data sessions.

III. Proposed Scheme

In this section, we propose the adaptive routing scheme to solve the path instability problem of the HWMP. Before the explanation of the scheme, we need to be aware of whether the path listed inside the routing table is currently in use or not. This information for the paths that have been created via on-demand route discovery can be easily recognized by checking whether the path is included in the routing table. Since on-demand routes are deleted from the route table entry once they are not used anymore, acquiring this fact is not a big problem.

However, the routes created from proactive route discovery cannot be determined whether they are currently in use or not. This is because the routes are updated periodically even if they are not in use. Therefore, to calculate this information, *Last Access Time* (LAT) is added to the routing table to record the most recent timestamp of the usage of the route. The LAT can be used to determine whether the proactive routes inside the route table are currently being used or not. If the difference between the current time and the LAT is lower than the pre-defined threshold, the current route is considered still used because not enough time has exceed to consider that this route is no longer in use.

Also, each session needs to be distinguished by some means. This is because we want to maintain the current active path for the existing sessions while providing new paths for the newly incoming sessions. To do this, we calculate the session ID for each data session by acquiring the hash value from the MAC addresses of multi-hop source node and destination node. Our method of calculating the session ID cannot distinguish multiple sessions having identical source-destination pair. This can actually be done by accessing the transport layer and acquiring the port number, but this would increase the process overhead. Therefore, we will use our simpler method to reduce the complexity of the scheme.

When a root advertisement message is transmitted by the root node, each node will receive this message and calculate its airtime link cost. This message is forwarded to all the nodes in the network, and each node will maintain the best multi-hop route to the root node. In the next advertisement interval, each node will receive the advertisement, and then follow the formula shown in (1) and (2).

$$Througput_{old} = \frac{FrameSize}{Cost_{old}}, Througput_{new} = \frac{FrameSize}{Cost_{new}} \quad (1)$$

$$Throughput_{gain} = \frac{Throughput_{new} - Throughput_{old}}{Throughput_{old}} \quad (2)$$

When a PREQ message received from the different previous hop from the current path table, the proposed SOAR checks the following conditions to prevent the change of the currently utilizing path. In equation (1), $Cost_{old}$ means the path cost of the existing path in the path table, and $Cost_{new}$ is the path cost recorded in the newly received PREQ message. We calculate the possible throughput of the old path and the new path as shown in the equation (1). Equation (2) calculates the possible throughput gain when the old path is changes to the new path. We switch the old path to the new one only when the throughput gain is larger than the predetermined threshold.

In the simulation study, we use the threshold value of 0.2, which means the path with 20% more potential throughput gain will be accepted. If the threshold value is too small, the frequent path switching is hardly prevented, and then the path instability problem may still happen. Otherwise, if the threshold value is too large, paths are rarely changed even when the actual condition of the existing path is really bad. It might lead another performance problem. Therefore, to determine the proper threshold value is very important and it is still an open issue.

In addition, if we do not accept the new path with the low throughput gain, this path is also recorded in the path table for the future use. When the new session with different session ID is detected, we assign this unused path for the new session. With this mechanism, the traffic of different session uses the alternative path, and traffic is distributed to the other forwarding nodes.

Figure 3 shows an example of the operation of data session distinguish algorithm. In the example shown in Figure 3, S1 is the first node that transmits its data via the path through node 3, and node D is the destination node. During the transmission, a periodical advertisement is triggered. While node 3 maintains its main path to the destination via relaying towards node 6, it will reserve an alternative path that reaches the destination via node 5. Therefore, the session created by S1 is maintained and transmitted through node 6. However, when S2 initiates its transmission towards the destination, node 3 will recognize that hash value of the data from S1 and S2 are different. Since S2 is recognized as a different data session, this data is transmitted to the destination through node 5 instead, reducing the congestion level that might occur at node 6.

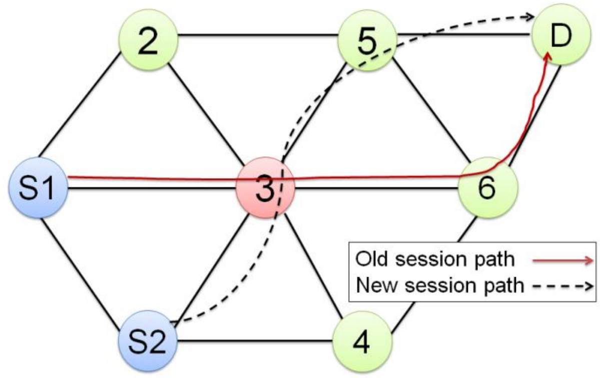
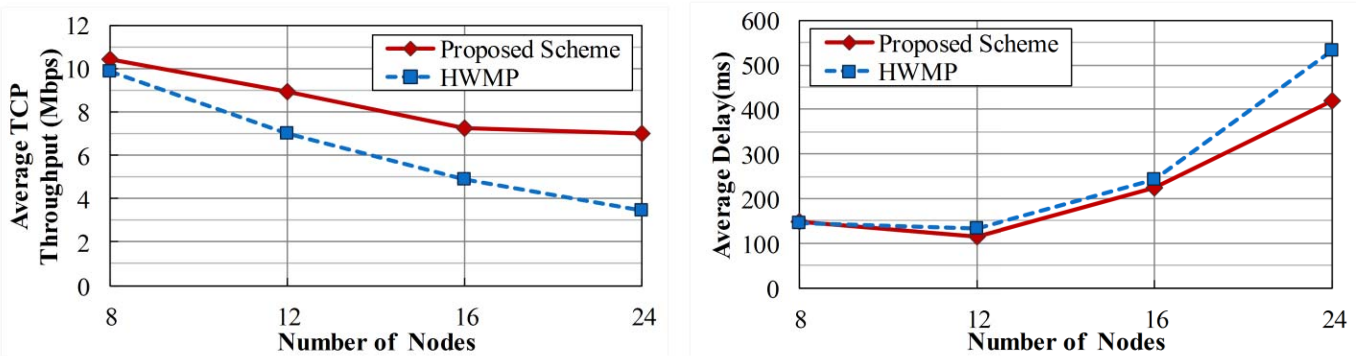


Figure 3. An example of the proposed scheme

IV. Preliminary Performance Evaluation

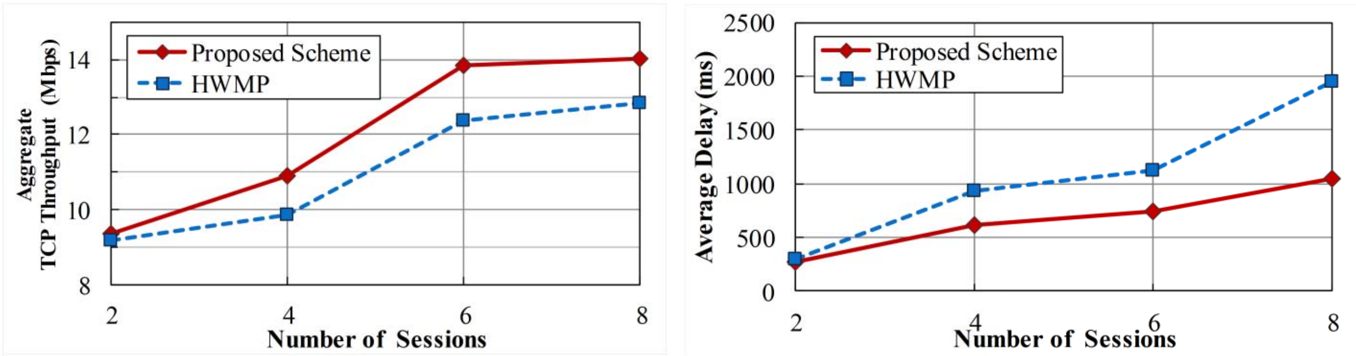
To performance the preliminary performance evaluation, we implement the proposed SOAR in the ns-3 simulator. The simulation model is almost similar in Section II, which is a multi-hop topology randomly placed. Each node is set to support 802.11a radio with 54Mbps. Transmission range without error is about 34m. We measure TCP throughput and average delay to evaluate the performance of the proposed scheme and the HWMP. During 100 seconds of the simulation time, multiple TCP sessions are created at random time with randomly selected destinations. The scenario above is simulated on 5 different topologies using random seed numbers.

In the first scenario, 3 TCP sessions are created in various network sizes. As shown in Figure 4(a), the average TCP throughput of the proposed SOAR outperforms the original HWMP with any number of nodes. Moreover, the difference of the TCP throughput increases as the number of nodes increases. Even when 24 nodes are deployed, the SOAR produces more than twice throughput than the HWMP. This is because, as the network size enlarges and the more number of nodes are included in the network, the path instability problem of the HWMP is exacerbated. Since the proposed SOAR successfully prevents this problem, the TCP throughput less degrades than the HWMP.



(a) Average TCP Throughput **(b) Average Delay**

Figure 4. TCP performance varying the number of nodes



(a) Aggregate TCP Throughput **(b) Average Delay**

Figure 5. TCP performance varying the number of sessions

Figure 4(b) depicts the average end-to-end delay of TCP sessions. The proposed SOAR shows less delay then the HWMP in all the scenarios, and the maximum reduction of the delay is over 100*ms*. This result confirms that the proposed SOAR holds more efficient paths during sessions are active. On the other hand, the frequent path switching in the HWMP causes packet reordering, and then high delay results from retransmissions in the TCP layer.

The second scenario shows the impacts of increasing the number of sessions on the performance. Figure 5 illustrates the aggregate TCP throughput and the average delay as the number of sessions increases. The proposed scheme provides 10% more TCP throughput in average compared to the HWMP. In case of the small number of sessions, average delays of two schemes are similar. However, as the number of sessions increases, the average delay of the proposed SOAR is much smaller than the HWMP. It means that the proposed scheme is effectively operated even in the large amount of traffic established.

V. Conclusion

This paper analyzes the path instability problem that occurs in the HWMP defined in IEEE 802.11s WLAN based mesh networks. The path instability can degrade the network by interfering with the current active data sessions and selecting sub-optimal multi-hop routes in the network. Our proposed scheme alleviates this problem by defining a threshold to constrain the frequency of route fluctuation. Also, data sessions are distinguished to prevent concentration of data traffic at specific paths and distribute data traffic to multiple routes. Future works include more precise selection of threshold levels and enhancements on the session distinguish module.

Acknowledgment

This work was supported by the IT R&D program of MKE/KEIT [10033886, Core technology development of large-scale, intelligent and cooperative surveillance system]

References

[1] M.-J. Lee, J. Zheng, Y.-B. Ko and D. M. Shrestha, "Emerging Standards For Wireless Mesh Technology," *IEEE Wireless Communications*, Vol. 13, Issue 2, Apr. 2006, pp.56-63.

[2] IEEE P802.11s™/D7.0, Amendment 10: Mesh Networking, July. 2010.

[3] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF RFC 3561, Jul. 2003.

[4] K. Ramachandran, I. Sheriff, E. Belding, and K. Almeroth, "Routing Stability in Static Wireless Mesh Networks," in Passive and Active Measurement Conference, 2007

[5] A. Basu, and J. Riecke, "Stability issues in OSPF routing," in proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, 2001.

[6] The ns-3 network simulator, http://www.nsnam.org/