

Improving the Starvation of TCP Flows in Wireless Mesh Networks with Modifying Routing Path

Chuan-Gang Liu¹, Hung-Hsiou Hsu², Ai-Jia Hsieh³, Cheng-Han Lin⁴,
 Department of Information Technology, Chia Nan University of Pharmacy & Science^{1,2,3}
 Postdoctoral Fellow, National Kaohsiung University of Applied Sciences⁴
 60 ,Erh-Jen RD .,Sec.1 ,Jen-Te ,Tainan, Taiwan, R.O.C.^{1,2,3}
 Chien Kung Campus 415 Chien Kung Road, Kaohsiung 807, Taiwan, R.O.C. ⁴
 *email: chgliu@mail.chna.edu.tw¹
 hhhsu127@gmail.com²
 qajhsieh@mail.chna.edu.tw³
 jhlin5@hpds.ee.ncku.edu.tw⁴

Abstract — Wireless mesh networks are expected to provide wireless broadband access to Internet. Multitier wireless mesh architecture provides a large-scale deployment. However, the popular congestion-controlled protocol, TCP, performs badly in such networks. Many studies have revealed this performance drawback. Recent paper even discovers the starvation of TCP flows in a specified wireless scenario, called the basic scenario. Some TCP flows will starve even with a TCP flow coming into this basic scenario. This paper proposes a new way to improve this critical issue. Comparing to previous solution, it just uses a routing way, called extending routing path here, without modifying TCP and it also solves the starvation of TCP flows and achieves fairness among TCP flows.

Keywords : wireless mesh network, starvation of TCP flows, extending routing path.

1. INTRODUCTION

Wireless mesh networks are expected to provide wireless broadband access to Internet. Multitier wireless mesh architecture provides a large-scale deployment. Figure 1 shows the multitier wireless mesh architecture. Several researches develop well-designed solutions to overcome several possible issues in wireless mesh networks. Some important issues are just like interfaces and channels assignment problem, eg. multi-interfaces and multi-channels assignment[2][3][4], gateway placement[5][6], etc. However, a well-know transport protocol,

Transmission Control Protocol, usually performs

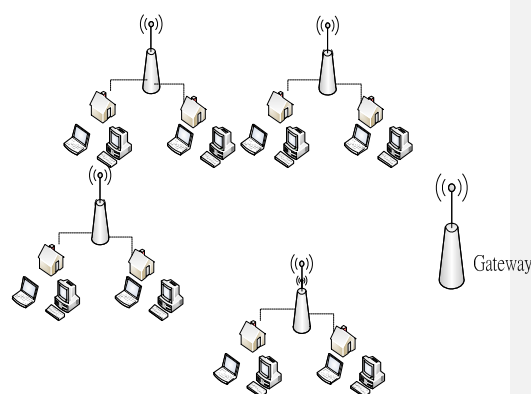


Fig. 1. The multitier wireless mesh architecture.

badly in such wireless networks. Although significant progress has been made for improving TCP performance[7][8], the starvation of TCP flows still occurs under some network scenario. Starvation of some TCP flows leads to severe performance degradation and unfairness.

Previous studies have been found that TCP and MAC mechanism are key points to lead to unfairness, eg. binary backoff and carrier sense. Hence, some proposals just try to tune backoff[9] and handshake[10] mechanisms. The parameters in 802.11 are also redesigned in order to achieve better performance of flows. However, these methods do not solve the starvation of TCP flows. Therefore, some researchers develop other ways from the view of TCP mechanism. By tuning TCP transmission window, and determining the causes of packet loss, TCP can adapt itself to wireless network much better than before. Variant TCP mechanisms are developed thereafter.

Recent study try to analyze the origins of starvation of TCP flows in wireless mesh

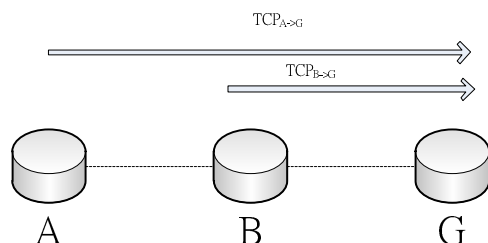


Fig. 2. Two flows in a basic network scenario

networks. It discovers the starvation of TCP flows in a specified wireless scenario, called the basic scenario and some TCP flows will starve even with a TCP flow coming into this basic scenario. Figure 2 shows this basic scenario. This basic scenario always exists in any mesh networks. This architecture just comprises three nodes and one of them is gateway (eg. Node G in Figure 2) and the other two nodes (eg. Node A and B in Figure 2) should transmit their TCP flow to gateway. In Figure 2, G and Node A are hidden mutually and Node A must transmits its TCP traffic via Node B. Node G transmits traffic to Node A via Node B similarly. Node B can communicate other two nodes and plays the role forwarding traffic from Node A and G. Due to the fact that Node A and G are hidden mutually, Node A and G usually interfere with each other. Recent study[1] uses an analytical model to discover the cause of starvation of TCP flows in such network scenario. It describes that to understand the interaction between TCP and MAC mechanisms is important to propose a way to solve this issue. The solution in [1] is to set Minimum contention window in TCP flows in node B much larger than original setting. However, to modify MAC mechanism is usually not practical.

This paper develops a new solution based on the modification of routing path in Node B. This opinion is very different from previous solutions. We replace common opinion, the modification of TCP or IEEE 802.11, with a new view, routing path modification. Hence, the main contributions of this paper are listed as follows.

1. Without the modification of TCP and IEEE 802.11 MAC, starvation of TCP flows in a basic scenario still can be solved. For improving TCP performance, it is really important advance. Furthermore, our solution is easy to implement

and hence it is feasible.

2. We provide a new view to improve TCP in wireless mesh network. The following study can keep looking for other ideas base on this work.

In this study, we show this newly-developed solution still can solve the starvation of TCP flows in a basic scenario. We provide extensive simulations to verify our proposed solution.

The remainder of this paper is organized as follows. Section II describes discussion including of various solutions to improve TCP performance in wireless mesh network and explanation of possible ways to improve TCP behavior in basic scenario. Section III gives details of our proposed solution for starvation problem of TCP flows in wireless mesh network. In Section IV, several simulations are provided. We give the future work in Section V. Finally, Section VI concludes the paper.

2. DISSCUSSION

TCP performance usually suffers from many uncertain factors in wireless mesh networks. Different from the wired network, the cause of packet loss is not just due to congestion. Wireless environment and MAC mechanism lead to several TCP packet losses. Many studies analyse the performance of TCP in wireless mesh networks to understand relation between TCP and IEEE 802.11. Furthermore, there are many studies to develop mechanisms to counteract the degradation of TCP performance in wireless mesh networks. These researches usually need to modify TCP including some new TCP-friendly protocols and enhanced TCP. The first one includes Split TCP and redesign TCP parameters. The other one usually includes variant proposals to enhance TCP, for instance, available bandwidth estimation TCP and loss cause prediction, etc. However, in wireless mesh network, these researches are not able to improve the starvation of TCP flows in a basic scenario completely. The main reason is due to the characteristic of TCP feedback loop. Unfortunately, this basic scenario must embed in wireless mesh network.

Recent study[1] tries to show interaction between TCP and IEEE 802.11 and find the origin of TCP starvation in a basic scenario, just like shown in Figure 2. It explains that the interaction between MAC and TCP leads to bistability. Hence, flows originating from node A starve even if only one flow from Node B to G. To achieve the goal that TCP flows in Node A

can survive, TCP flows in Node B should slow down its increasing rate. Therefore, it proposes a

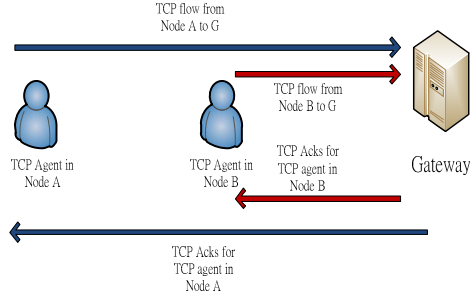


Fig. 3 Two TCP transmission loops.

solution to improve starvation of TCP flows in such scenario. In short, the idea of its solution is to set a large minimum contention window for flows whose source nodes are apart one hop from gateway. Here, we try to deeply discuss TCP behaviour and discover possible difficulty in previous work. In Figure 2, we assume that there are two TCP flows and they originates from node B and A, respectively. TCP flow in node B does not interference with hidden nodes but one in Node A does.

In Figure 2, while TCP flows in Node A, B transmit their TCP packets to G and G transmits ACKs to node A and B, respectively, two TCP transmission loops are formed. Figure 3 shows two TCP transmission loops. TCP transmission loop formed by Node B, G, defined as $L_{B,G}$, captures more network resource than $L_{A,G}$. The reason is Node A and G interfere with each other. Hence, TCP packets(ACKs) in Node A(G) are usually dropped due to interference from G(A). Successful transmission in Node A(G) can win wireless media usage for a period of time. Node A and G occupy wireless media alternately, called bistability in [1]. Reader can refer to the details of analysis in [1]. Node B can sense the transmission from Node A and G and hence TCP in Node B can transfer normally. Based on this observation, TCP flow in Node B should obtain more network resource than one in Node A. Those flows in Node A even starves under this imbalance resource distribution. [1] proposes a solution that adjusts minimum contention window of node B. The idea behind this solution is to constrain the increasing rate of TCP transmission in Node B and TCP flow in node A has more opportunity to transmit its traffic. This solution indeed improves unfairness between TCP flows in Node A and B. However, to adjust

MAC parameter, CW_{min} , needs to modify IEEE 802.11b and it is not usually a practical way. In Table 1: Definition of parameters used in this paper

Items	Definition
$L_{B \rightarrow G}$	The link of node B to G
$L_{A \rightarrow G}$	The link of node A to G
$TCP_{A \rightarrow G}$	TCP flows from Node A to G
$TCP_{B \rightarrow G}$	TCP flows from Node B to G
$N_{(1,G)}$	The nodes apart one hop from Node G
$N_{(>1,G)}$	The nodes apart more than one hop from Node G
N_t	The nodes apart two hop from Node G

this paper, a new solution is given and it also improves the fairness issue and starvation of TCP flows in basic scenario. This new solution is developed based on modifying routing path of some node and is much more practical.

3. PROPOSED SOLUTION

2.2 First, we describe the initial idea of our solution. Previous works [1] often focus on modification of TCP or MAC protocol to improve TCP in wireless mesh network, which are not possibly practical. Hence, this paper develops a new solution in the view of routing path. It does not modify any network protocols and just changes the routing path of some flows. Routing protocol is also not our concern here and hence any routing protocols are still allowed to be employed in wireless mesh networks. However, AODV routing protocol is employed here. Then, we use some analysis to explain our opinion and thereafter this paper focuses on TCP flows in a basic scenario, just like in Figure 2. Table 1 show the network parameters defined in this study.

Based on discussion in Section 2, $TCP_{B \rightarrow G}$ flows can operate normally due to the fact that Node B can sense the communication of G. Hence, TCP transmission window can increase fast. For $TCP_{A \rightarrow G}$ flows, its transmission window usually decreases and keeps low because of that A and G are hidden mutually. Soon, $L_{B \rightarrow G}$ is full of TCP packets and ACKs of TCP flows in Node B and TCP flows in Node A starve. It shows that $TCP_{A \rightarrow G}$ flows can survive only if $TCP_{B \rightarrow G}$ slow down its increasing rate. To achieve this goal, this paper use different way that lets TCP flow in Node B route the similar routing path of TCP flow in Node A. This reason is as follows. As

格式化: 項目符號及編號

validation of [1], $TCP_{A \rightarrow G}$ flows and traffic from G usually get network resource alternatively, which they call it as bistability. We observe all flows passing through Node A must experience this circumstance of bistability exception to Node B. The proposed solution in this paper is to change the routes of $N_{(1,g)}$. The procedure of our mechanism is listed as following.

1. All nodes run their routing protocol to discover routing path.
2. To indicate $N_{(1,g)}$, the way is to look up the next hop item in routing tables in those candidate nodes. It should be G in the next hop item.
3. Then, to find the nodes, called as transit nodes and denoted them as N_t , apart two hops from G, the routing tables in $N_{(1,g)}$ give the solution. Because those nodes must be apart one hop from $N_{(1,g)}$ in downstream direction.
4. Re-Setting the next hop of the route to Node G in the routing table of $N_{(1,g)}$ is one of transit nodes in step 3.
5. Transit nodes forward traffic from corresponding $N_{(1,g)}$ to the gateway with the use of routing table in transit nodes.

We provide an example to explain. First, Node A and B discover their routing path with the use of routing protocol. The simplified routing tables in Nodes A, B are listed as Table 2. As shown in Table 1, the next hop item of routing table in node B is gateway for the path to gateway, which means Node B is one of $N_{(1,g)}$. Furthermore, exception to the path to gateway, nodes in next hop item are transit nodes for some paths. Next, it changes the path to gateway, just replacing the original next hop item with corresponding one of N_t and then that chosen transit node forwards traffic based on its routing path to gateway. All things in our mechanism have been explained. However, there is a possible issue in our descriptions, which is there should be a loop route existing between one of $N_{(1,g)}$ and its corresponding transit node. To see the Figure 2, the route of the flow to gateway must be a loop, which the flow originating from node B must arrives at original node B again. Node B should drop it to avoid traffic looping between B and A. Then no traffic can arrive at destination, G, finally. Hence, node B should let TCP traffic go if it is from B's corresponding transit node and its source is B. And at this moment, the route should use original one. We use flow chart to describe this procedure.

Routing table in Node B			Routing table in Node A		
Entry number	Destination	Next hop	Entry number	Destination	Next hop
1	Node G	G	1	Node G	B
2	Node A	A	2	Node B	B

Table 2: routing tables in Node A, B

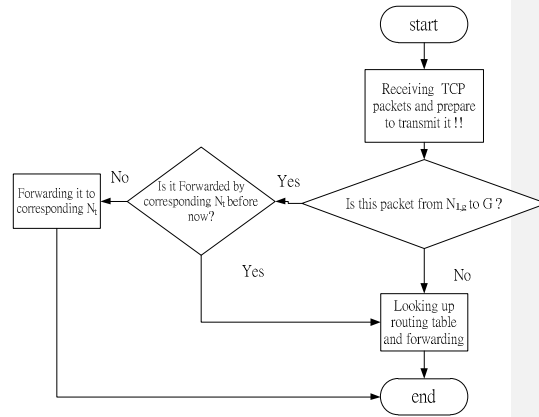


Fig. 4. The flow chart of forwarding TCP packets in $N_{(1,g)}$

4. SIMULATIONS

In this section, the performance of two TCP flows in a basic scenario is investigated. Here, we use ns2 simulator to simulate network scenario. Our proposed solution is compared with the solution in [1]. The following simulations are made in basic scenario depicted in Figure 2. Node A and B transmit packets to the gateway node. We make four simulations.

Simulation1 : First, we show the starvation of TCP flows in basic scenario. The parameters of TCP and 802.11 MAC are set as default value in ns2 simulator. Figure 5 shows the result which $TCP_{A \rightarrow G}$ flow starves and $TCP_{B \rightarrow G}$ flow gets high network throughput. For $N_{(>1,g)}$, their performance must suffer from this starvation issue.

Simulation 2: In this simulation, we perform the simulation with our solution. Figure 6 show the results. We can find the throughput of $TCP_{A \rightarrow G}$ flow increases. $TCP_{B \rightarrow G}$ flow still can transfer its TCP traffic steadily. Furthermore, A and B

share bandwidth equally. The results of solution achieves fairness between two flows and avoids the starvation of TCP flows.

Simulation 3: Then, we continue the simulations with proposed solution in [1]. In simulation 3, MAC mechanism in node B adopts $CW_{min}=128$. From Figure 7, we can see that this simulation shows that the throughput of Node B can increase than before and is similar to the result of Node B in simulation 2. But it still exist unfairness relation between $TCP_{A \rightarrow G}$ and $TCP_{B \rightarrow G}$ flows. Although the sum of their throughputs is higher, being not fair enough and modifying CW_{min} parameter in 802.11 make its solution doubtful.

Simulation 4: In Simulation 4, we set CW_{min} to 512 and the result is shown in Figure 8. We can find the sum of throughput of both flows become low and it is still unfair.

Based on these simulation, our solution and proposed one in [1] can improve the starvation of TCP flows efficiently. However, our solution can simultaneously achieve fairness between two flows.

5. FUTURE WORKS

This study observes the starvation issue of TCP flows in wireless mesh network and provides a new way to improve it. Section 3 give our solution and make several simulations to verify its performance. Based on our solution, it exactly improves the starvation issue. However, other network scenarios like more $N_{(>1,g)}$ nodes should also be simulated. Furthermore, the reason behind this solution should be validated with compact theory. A mathematical study is indeed needed to prove it. We leave it as the future work and try to find more evidence to confirm our solution.

Furthermore, in addition to the view of routing path modification, we also try to solve the starvation issue with other ways. Active Queue Management may play an important role to manage flows from $N_{(1,g)}$ and $N_{(>1,g)}$ respectively. The possible way may be like to distinguish protected flows from flows in $N_{(1,g)}$ and design schemes to improve the starvation issue. We will keep studying in this field in the future.

simulation demonstrate that our proposed

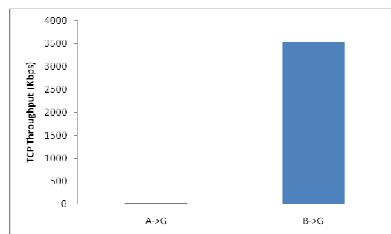


Fig. 5. The starvation of TCP lows in a basic secnario in Fig.2.

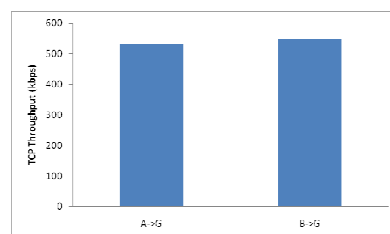


Fig. 6. The throughput of two TCP flows with our proposed solution.

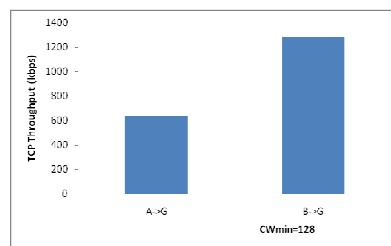


Fig. 7. The throughput of two TCP flows with $CW_{min}=128$.

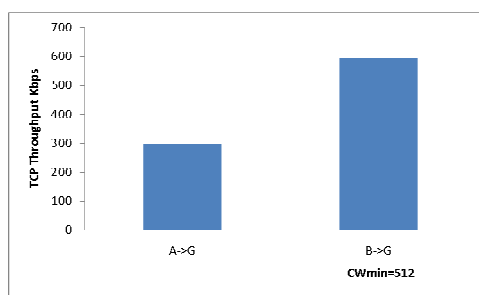


Fig. 8. The throughput of two TCP flows with $CW_{min}=512$.

6. CONCLUSION

In this paper, we discuss well-known drawbacks of TCP flows in wireless mesh networks. Related researches focus on modification of TCP and MAC protocol for improving TCP performance. However, it is usually not practical and cannot solve the starvation of TCP flows in basic scenario due to TCP's nature, control loop. Recent study tries to tune minimum contention window and it achieves better performance than before. But it needs to modify the parameter of MAC. This paper provides a new view to improve extreme unfairness among TCP flows in a basic scenario. The proposed scheme tries to change routing paths of $N_{(1,g)}$ to avoid the modification of original network setting. Extensive simulations confirm feasibility of our proposed scheme. Further, the most important thing is that we provide another view to improve TCP performance.

REFERENCES

- [1] Omer Gurewitz, Vincenzo Mancuso, Jingpu Shi, Edward W. Knightly, "Measurement and modeling of the origins of starvation of congestion-controlled flows in wireless mesh networks.", *IEEE/ACM Transactions on Networking*, Vol. 17 Issue 6, Dec. 2009.
- [2] Kyasanur, P., Vaidya, N.H., "Routing and interface assignment in multi-channel multi-interface wireless networks," *Wireless Communications and Networking Conference, IEEE*, 13-17 March 2005
- [3] Jun Wang, Huan Li, Weijia Jia, Liusheng Huang, Jingyuan Li, "Interface assignment and bandwidth allocation for multi-channel wireless mesh networks," *Computer Communications*, Vol. 31, Issue 17, 20, pp. 3995-4004, Nov. 2008
- [4] A. Hamed Mohsenian Rad, Vincent W.S. Wong, "Congestion-aware channel assignment for multi-channel wireless mesh networks," *Computer Networks*, Vol. 53, Issue 14, 18, pp. 2502-2516 Sep. 2009
- [5] Benyamina, D., Hafid, A., Gendreau, M., "Optimal placement of gateways in multi-hop Wireless Mesh Networks: A clustering-based approach," *IEEE 34th Conference on Local Computer Networks*, pp. 625 - 632 Oct. 2009,
- [6] Smadi, M.N., Ghosh, S.C., Farid, A.A., Todd, T.D., Hranilovic S., "Free-Space Optical Gateway Placement in Hybrid Wireless Mesh Networks," *Lightwave Technology, Journal of*, Vol. 27 Issue 14, pp. 2688 – 2697, July15, 2009
- [7] Karlsson, J., Kassler, A., Brunstrom, A., "Impact of packet aggregation on TCP performance in Wireless Mesh Networks," *IEEE International Symposium on WoWMoM* pp. 1-7, 15-19 June 2009.
- [8] Rung-Shiang Cheng, Hui-Tang Lin, "A cross-layer design for TCP end-to-end performance improvement in multi-hop wireless networks," *Computer Communications*, Vol. 31, Issue 14, September 2008, pp. 3145-3152
- [9] Y. Wang and J. J. Garcia-Luna-Aceves, "Channel sharing of competing flows in ad hoc networks," in *Proceeding of IEEE ISCC*, Kemer-Anatolia, Turkey, pp. 189-196, Jun. 2003
- [10] V. Bharghavan, A. Demers, S. Shenker, and L. Zang, "MACAW: A media access protocol for wireless LANs," in *Proceeding of ACM SIGCOMM*, London, U.K., pp. 212-225, 1994