Local Distributed Fault Detection in Wireless Mesh Network

LI Hong-jian, XU Ming

Department of Network Engineering, School of Computer Science, National University of Defense Technology Changsha, China

Email: hongjianli.nudt@gmail.com, Xuming-64@hotmail.com

Abstract: Wireless mesh networks (WMNs) offer an easy and economical alternative for providing broadband wireless Internet connectivity, but fault in WMN might lead to a sharp decline in network performance. Existing works on fault detection often face the following problems: they are either offline detection that can not detect the fault in real-time, introduce additional overload or cannot distinguish the causes of the fault. We proposed local distributed fault detection algorithms in wireless mesh network that does not have such drawbacks. Firstly, we analyzed faults of link congestion, hidden terminals and noise interference in depth. Then the local distributed fault detection algorithms were proposed in the paper according to the different characters of faults. Finally, the theoretical analysis, simulation and experiments in platform were made to evaluate the algorithms. The results of analysis and simulations and experiments in platform show that the proposed algorithms perform well for the fault detection in WMNs with low cost.

Keywords: Fault detection, Wireless mesh network, link congestion, hidden terminal, noise interference

1. Introduction

Wireless Mesh networks (WMNs) have become increasingly prevalent over the last few years. Compared with traditional networks, The WMN is a more economical choice for Internet access. WMN can provide greater coverage and higher throughput than WLAN. So far, there are more than 400 cities in the world where WMN has already been or plan to be deployed.

Three types of node should be included in a typical wireless mesh networks, mesh gateways, mesh routers and mesh access points. The data can be transmitted in the wireless mesh network with multi-hop communication. However, fault in WMN may lead to poor performance of the whole network, such as long delay, the instability of the signal and others. There are many reasons that will lead to fault. The quality of the wireless link may bedisturbed by the physical environment, such as weather, noise or heavy network load. Carrier sense multiple access with collision avoidance (CSMA / CA) in IEEE 802.11 MAC protocol may have the hidden terminal problem, which greatly affects the network performance. In addition, the wrong configuration of the equipment and protocol can also cause performance decline in overall network.

Three types of fault are considered in this paper:

— **Link congestion** can cause a sharp decline in network performance. When mass communication in WMN exceeds the capacity of the channel, link congestion occurs.

— **Hidden terminal** causes the decline in network performance. It refers that sender is out of range of other senders or a collection of senders. Since senders can not sense the carrier, CSMA / CA does not work and collisions occur. The back-off mechanism in 802.11 MAC protocol increase the impact on network performance.

— **Noise interference** comes from the RF devices that work in the some radio frequency. There are many other devices that share the 2.4GHz spectrum with IEEE 802.11 devices, such as indoor cordless phones and microwave ovens. The communication nodes around these devices with the same spectrum can detect noise or delay the transmission.

The rest of the paper is organized as follows. Section II introduces existing related works and motivation, fault detection algorithms were presented in section III presents, and section IV is analyses and evaluations. Finally， we give the conclusions about the work in this paper.

2. Related Work & Motivation

Fault detection in wireless network can be divided into active and passive detection. Active fault detection involves using active measurements for detecting faulty links ([5,6]). Passive detection is mainly through the sniffer nodes. It is to make decision according to results of sniffer nodes. Passive detection requires special sniffer nodes, and can usually provide off-line detection [3,4,8,9]. The author in [7] used signature-based fault detection, but the current embedded devices (Mesh Router / Gateway) can not provide enough performance for such operations yet [10].

Hidden terminal problem has been studied in [1,2], reference [1] distinguish the hidden terminal and capture effect by packet dropping rate, and the new protocol was

Project partially supported by Hunan Provincial Natural Science Foundation of China No.09ZZ4034

designed to avoid the hidden / exposed terminal in [2]. Noise detection threshold was given in [1] according to the experiments. In addition, we must resolve the problem that the node is too close to the source of noise to report the noise interference.

3. Fault Detection Algorithms

As fault means the poor conditions of the network, it is the basic principle to reduce data communication that is generated by fault detection algorithm. In this section, the local distributed fault detection algorithms are proposed to solve the fault in wireless mesh networks.

3.1 Link congestion detection algorithm

Link congestion is mainly generated by interference between nodes with different data flows. The major reason of link congestion is that the load excesses the capacity of network. In addition, load balance can also effect the link congestion. Since downlink traffic fraction is about 80% of the data in WMN [2], it is the major part to cause link congestion.

As shown in **Figure 1**, our link congestion detection algorithm works as follows:

The information involved in the algorithm are those in routing layer and transportation layer. As shown in Figure 1, downlink traffic is initiated by gateway nodes. Therefore the information about data flow f_i and routing path $P = \{v_1, v_2, ..., v_j\}$ is stored and calculated in the GW.

Algorithm maintains three sets. Data flow set $B(v)$ is the set of the data flows that flow through node v. Neighbor set $N(v)$ is the set of neighbors of node v. Interference set I is the set of the edges where whose the different flows flow through those two end nodes and the two flows interference each other.

When the data flow f_i starts or ends, the set $B(v)$ and I should be updated. If there is only one data flow inset $B(v)$ and there is not less than one data flows in set $B(p)$, where $p \in N(v)$, add the (v, p) to the set I. On the contrary, if there is no data flow in the set $B(v)$, and there is not less than one data flows in set $B(p)$, where $p \in N(v)$, delete the (v, p) or (p, v) from the set I. (p, v).

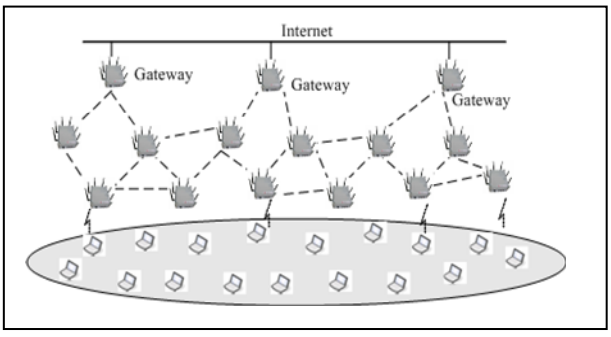

Figure 1. Structure of WMN

1 Initialize GW, $\forall v \in V$, the set of data flow $B(v) = \phi$, counter $C_{congestion} = 0$, find the neighbor set $N(v)$, Interference set $I = \phi$;

2 For any data flow f_i, $P = \{v_1, v_2, ..., v_j\}$,

a. for $k(1:1:j) < j$, do

b. {

c. if (f_i start) then {

d. $B(v_k) = B(v_k) \cup \{f_i\}$

e. for($\forall v_p \in N(v_k)$)

f. if(($|B(v_k)| = 1$) &&($|B(v_p)| > 0$))then { /*if$|B(v_p)| = 1$ $B(v_k) \neq \{f_i\}$ */

g. $C_{congestion} = C_{congestion} + 1$

h. $I = I \cup \{(v_k, v_p)\}$ }}

i. if(f_i end) then {

j. $B(v_k) = B(v_k) - \{f_i\}$

k. for($\forall v_p \in N(v_k)$)

l. if(($|B(v_k)| = 0$) &&($|B(v_p)| > 0$))then { //if$|B(v_p)| = 1$ $B(v_k) \neq \{f_i\}$

$C_{congestion} = C_{congestion} - 1$

n. $I = I - \{(v_k, v_p) | (v_p, v_k)\}$ }}

o. }

3 Calculate τ, update $B(v), C_{congestion}, I$ to other GWs.

According to the detection algorithm, we can get the information about local link congestion. Congestion rate is Calculated by $\tau = k \cdot \frac{C_{congestion}}{\|I\|}$, Where k is $2/D_{network}$, as correction factor, $D_{network}$ is the average neighbor nodes. The operation $\|I\|$ is to take the node number in set I. Then congestion ratio is interpreted as the ratio of the number of interference nodes to the number of neighbors. Congestion ratio belongs to [0,1]. Value 0 means no congestion, and value 1 denotes the worst con-

978-1-935068-23-5 © 2010 SciRes.

gestion.

In addition, the congestion ratio reflects the status of the local network. For the special node, we should calculate the number of the interference edge from set I to make decision whether link congestion happened near the node.

3.2 Hidden terminal detection algorithm

As shown in **Figure 2**, we can get the information about hidden terminal from the neighbor node. For WMN $G(V, E)$, $N(v)$ is the neighbor set of node v, and then the hidden terminal detection algorithm is as follows:

1. $\forall v \in V$, Find the neighbor set $N(v)$, and sent to all neighbor nodes within two-hop;
2. For the received $N(v)$, find $|N(u) \cap N(v)| > 0 \,\&\, u \notin N(v) \,\&\, v \notin N(u)$ and $u \neq v$;
3. Then node u and v is hidden terminals each other, store the information.

3.3 Noise interference detection algorithm

The interference generated by non-IEEE 802.11 devices can be divided into the following:

1) The interference source is located near the sender. The sender will not sense the channel free within a long time.

2) The interference source is located near the receiver. The receiver can not receive the data frame successful. There may be non-IEEE 802.11 equipment interference.

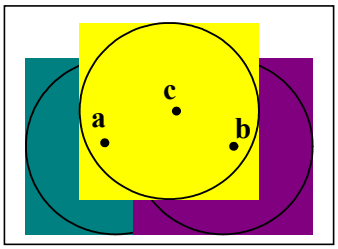

Figure 2. Hidden Terminal

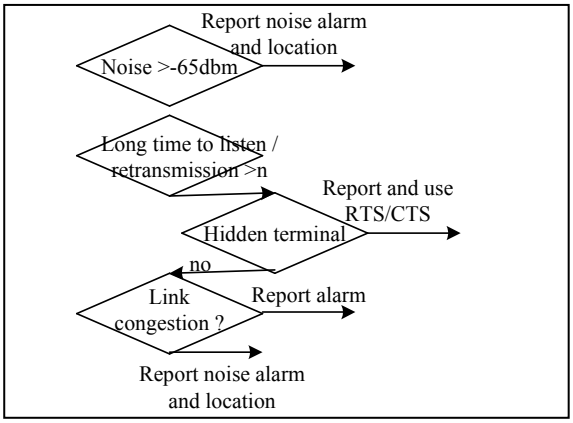

Figure 3. noise interference detection algorithm

First noise threshold is set to -65 dbm[1]. When node detects the noise above the threshold, alarm will be reported to the server. If the interference is too strong to report the alarm, we need the approach to detect the fault and report it.

4. Analyse and Evaluation

According to link congestion detection algorithm, computational complexity is $O(k \cdot f \cdot D_{network})$, where k is average length of the routing path, f is the average data flows in one GW node and $D_{network}$ is node degree. A well planning WMN guarantees that there is a GW for any mesh node within 4 hops. Here hidden terminal detection algorithm introduces some wireless communication costs, and the others scarcely introduce any wireless communication costs.

We evaluate the link congestion detection algorithm in the platform (**Table 1.**). The iperf application was used to test the performance.

As shown **Figure 4**, the CPU usage of link congestion detection algorithm in the 680M platform is not more than 2%, but the memory of the process has grown rapidly with the parameter D and k, up to the 8MB. Figure 5 shows that congestion rate reflects the status of network.

Additional communication overhead is $O(n \cdot D_{density}^2)$ in the hidden terminal detection algorithm, where n is thenumber of nodes. The algorithm is based on the to-

Table 1. Test platform

Platform	RouterStation PRO
CPU	MIPS 680MHz
Routing protocol	OLSR
OS	OpenWRT(Linux)
Number	9+n(virtual)
Radio	AR9002
Driver	Madwifi

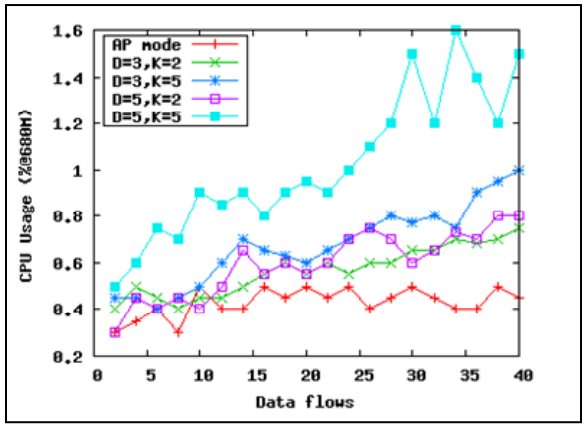

Figure 4. CPU Usage of link congestion detection

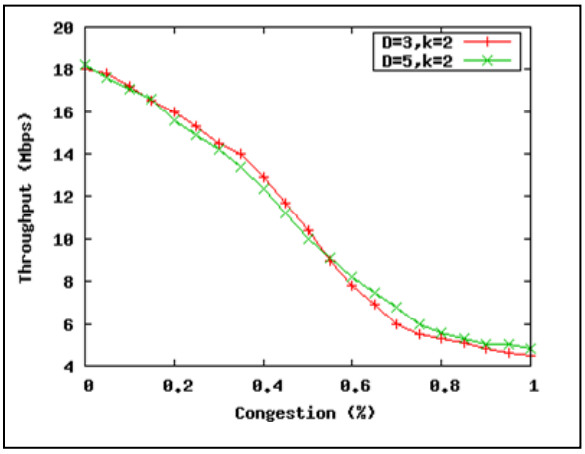

Figure 5. Throughput

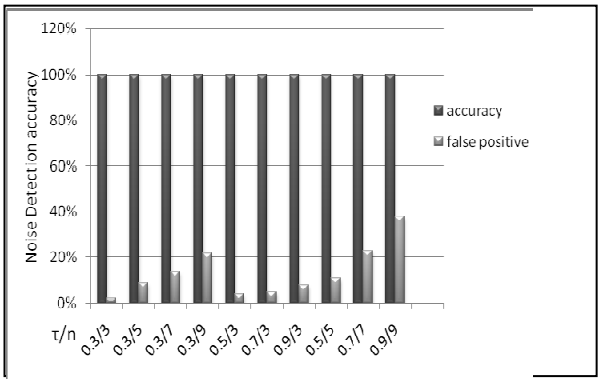

Figure 6. Accuracy of noise detection algorithm

pological relations. Therefore, detection accuracy was 100%.

Finally, noise detection algorithm relies on the accuracy of the link congestion detection. So the parameters τ and n can affect the accuracy and false positive of noise detection. Following figure is the test results.

5. Conclusions

In this paper, we proposed a local distributed detection algorithm to detect the link congestion, hidden terminal and noise interference. Compared with existing works, in our approach, the sniffer nodes are not needed and low workload and communication overload is introduced. Moreover, our algorithm can detect fault in real-time. Our algorithms introduce low computation and message overhead to the network, which is quite suitable for WMN.

Acknowledgements

This research program is partially supported by Hunan Provincial Natural Science Foundation of China No. 09ZZ4034.

References

[1] Anmol Sheth, Christian Doerr, Dirk Grunwald, Richard Han, and Dougla Sicker. “Mojo: A Distributed Physical Layer Anamoly Detection System for 802.11 WLANs”. MobiSys’06, June 19-22, 2006, Uppsala, Sweden.

[2] Vivek Shrivastava, Nabeel Ahmed, Shravan Rayanchu Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, Arunesh Mishra. “CENTAUR: Realizing the Full Potential of Centralized WLANs through a Hybrid Data Path”. MobiCom’09, September 20-25, 2009, Beijing, China.

[3] Lili Qiu, Paramvir Bahl, Ananth Rao, and Lidong Zhou. “Troubleshooting Wireless Mesh Networks”. SIGCOMM,’06, September 11-15,2006, Plsa, Italy.

[4] Atul Adya, Paramvir Bahl, Ranveer Chandra, and Lilli Qiu. “Architecture and Techniques for Diagnosing Faults in IEEE 802.11 Infrastructure Networks”. In MOBICOM, 2004.

[5] K. Naidu, D. Panigrahi, and R. Rastogi, “Detecting anomalies using end-to-end path measurements,” in 27th IEEE International Conference on Computer Communications (INFOCOM), April 2008.

[6] B. Wang, W. Wei, W. Zeng, and K. Pattipati, “Fault localization using passive end-to-end measurement and sequential testing for wireless sensor networks,” in 6th Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON), 2009.

[7] Dhruv Gupta, Prasant Mohapatra, Chen-Nee Chuah. “Diagnosing Failures in Wireless Networks using Fault Signatures”, In ICC2010.

[8] Yu-Chung Cheng, John ellardo, and Peter Benko. “Jigsaw:Solving the Puzzle of Enterprise 802.11 Networks”. SIGCOMM,’06, September 11-15,2006, Plsa, Italy.

[9] Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. “Analyzing the MAC Level Behavior of Wireless Networks in the Wild”. SIGCOMM,’06, September 11-15,2006, Plsa, Italy.

[10] Fabian Hugelshofer, Paul Smith, David Hutchison, Nicholas J. P. Race. “OpenLIDS: A Lightweight Intrusion Detection System for Wireless Mesh Networks”. MobiCom’09, September 20-25, 2009, Beijing, China.