A New Paradigm for Load Balancing in WMNs

Mohammad Shahverdy *shahverdy@iautb.ac.ir*
Department of Computer
Tafresh Branch, Islamic Azad University
Tafresh, 3951736874, IRAN

Misagh Behnami *misagh_behnami@yahoo.com*
Department of Computer
Arak Branch, Islamic Azad University
Arak, IRAN

Mahmood Fathy *mahfathy@iust.ac.ir*
Computer Engineering Faculty
Iran University of Science and Technology
Tehran, IRAN

Abstract

In this paper, we address the problem of load balancing in Wireless Mesh Networks. We consider a Cluster Based Wireless Mesh Architecture in which the WMN is divided into clusters that could minimize the updating overhead during topology change due to mobility of mesh nodes or congestion of load on a cluster. Each cluster contains a gateway that has complete knowledge about group memberships and link state information in the cluster. The gateway is often elected in the cluster formation process. We consider load of gateways and try to reduce it. As a matter of fact when a gateway undertakes to be an interface for connecting nodes of a wireless mesh network to other networks or internet, there would be some problems such as congestion and bottleneck, so we introduce a new paradigm for these problems. For solving bottleneck we use clustering to reduce load of gateways and after that by use of dividing cluster we prevent from bottleneck on gateways. We study how to detect congestion on a gateway and how can reduce loads of it that preventing from bottleneck on gateway and therefore increasing throughput of network to encountering many loads. So we propose an algorithm to detect bottleneck and remedies for load balancing in Wireless Mesh Networks. We also use Ns2-Emultion for implementing and testing the framework. Some qualitative results are provided to prove the correctness and the advantages of our framework.

Keywords: Wireless Mesh Networks, Load Balancing, Clustering, Bottleneck

1. INTRODUCTION

Wireless mesh networking is a new paradigm for next generation wireless networks. Wireless mesh networks (WMNs) consist of mesh clients and mesh routers, where the mesh routers form a wireless infrastructure/backbone and interwork with the wired networks to provide multi hop wireless Internet connectivity to the mesh clients. Wireless mesh networking has generated as a self-organizing and auto-configurable wireless networking to supply adaptive and flexible wireless Internet connectivity to mobile users. This idea can be used for different wireless access technologies such as IEEE 802.11, 802.15, 802.16-based wireless local area network (WLAN), wireless personal area network (WPAN), and wireless metropolitan area network (WMAN) technologies. WMNs Potential application can be used in home networks, enterprise networks, community networks, and intelligent transport system networks such as vehicular ad-hoc networks.
Wireless local area networks (WLANs) are used to serve mobile clients access to the fixed network within broadband network connectivity with the network coverage [1]. The clients in WLAN use of wireless access points that are interconnected by a wired backbone network to connect to the external networks. Thus, the wireless network has only a single hop of the path and the Clients need to be within a single hop to make

connectivity with wireless access point. Therefore to set up such networks need access points and suitable backbone. As result a Deployment of large-scale WLANs are too much cost and time consuming. However, The WMNs can provide wireless network coverage of large areas without depending on a wired backbone or dedicated access points [1, 2]. WMNs are the next generation of the wireless networks that to provide best services without any infrastructure. WMNs can diminish the limitations and to improve the performance of modern wireless networks such as ad hoc networks, wireless metropolitan area networks (WMANs), and vehicular ad hoc networks [2,3,4 and 5]. WMNs are multi-hop wireless network which provide internet everywhere to a large number of users. The WMNs are dynamically self-configured and all the nodes in the network are automatically established and maintain mesh connectivity among themselves in an ad hoc style. These networks are typically implemented at the network layer through the use of ad hoc routing protocols when routing path is changed. This character brings many advantages to WMNs such as low cost, easy network maintenance, more reliable service coverage.

Wireless mesh network has different members such as access points, desktops with wireless network interface cards (NICs), laptops, Pocket PCs, cell phones, etc. These members can be connected to each other via multiple hops. In the full mesh topology this feature brings many advantages to WMNs such as low cost, easy network maintenance and more reliable service coverage. In the mesh topology, one or multiple mesh routers can be connected to the Internet. These routers can serve as GWs and provide Internet connectivity for the entire mesh network. One of the most important challenges in these networks happens on GW, when number of nodes which connected to the internet via GW, suddenly increased. It means that GWs will be a bottleneck of network and performance of the network strongly decreases [4, 5, and 6].

In section 2 we first introduce related works. In section 3 system model and assumptions are discussed. In section 4 we present a new method for load balancing via GW. Section 5 evaluates the performance of the proposed scheme by means of simulation. Finally we conclude the paper in Section 6.

2. RELATED WORK

The problem of bottleneck in wireless mesh networks is an ongoing research problem although much of the literature [7, 8, 9, 10] available, addresses the problem without an introducing method for removing bottleneck and/or a well-defined way to prevent congestion. In [11], the authors proposed the MeshCache system for exploiting the locality in client request patterns in a wireless mesh network .The MeshCache system alleviates the congestion bottleneck that commonly exists at the GW node in WMNs while providing better client throughput by enabling content downloads from closer high-throughput mesh routers. There is some papers related to optimization problems on dynamic and static load balancing across meshes [11].Optimal load balancing across meshes is known to be a hard problem. Akyildiz et al. [12] exhaustively survey the research issues associated with wireless mesh networks and discusses the requirement to explore multipath routing for load balancing in these networks. However, maximum throughput scheduling and load balancing in wireless mesh networks is an unexplored problem. In this paper we present for the first time, a load balancing scheme in wireless mesh networks by using clustering and finding the best cluster head for new cluster by a new formula and evaluate its performance.

3. MODELS AND ASSUMPTION

We consider an area with wireless nodes (figure 1). These nodes can connect together and they can connect to internet or external networks via GW. GWs are wireless nodes that can route internal traffic to external networks when number of requests to GW increase then GW can’t service all requests punctually. Thus a load balancing method is needed to decrease workload of GW. We use the following metrics for evaluation our scheme.

Service ratio: service ratio is defined as the ratio of the number of nodes which receive service to the total number of nodes that send request to GW. A good load balancing method should service as many requests as possible in time confine.

Delay: delay is defined time distance between node request`s until node receive responses by GW. Delay is a good parameter to present a suitable scheme to have a good performance.

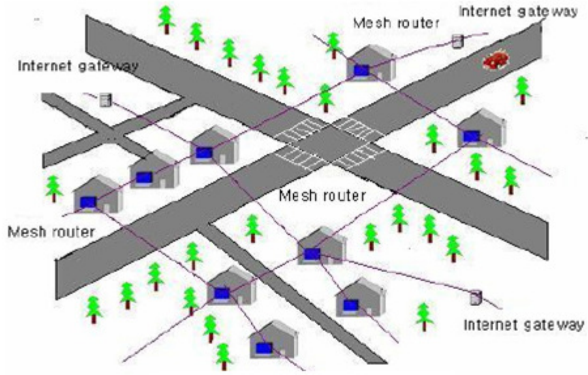

FIGURE 1: A wireless mesh network area

4. LOAD BALANCING VIA GATEWAY

The previous load balancing algorithms [7] do not spot fast traffic undulation. For instance, when the numbers of nodes (that are transferring data with a gateway) are increased suddenly then workload on transmission nodes and routers will be increased. This problem decreases performance of network because packets are aggregated in transmission nodes and cannot reach to destination on time. In first scheme for better control of workload on nodes we cluster nodes in specific groups. it is done thereby we put active nodes that are close together and can transmit data together directly in one cluster. Each cluster has a master that is called Gateway (GW). Each node in one cluster is aware of other nodes in this cluster. when a node wants to send data to a destination there will be two states: 1) if destination node is in the same cluster, transmission is occurred with one hop 2) if destination is not in the same cluster, at first data must be send to GW of its cluster and then this GW sends data to the destination cluster which the destination node exists in. The GW inside destination cluster sends data to destination node with one hop.

In a first look essence of a GW that all data transmit via them is bottleneck. In other word, when number of nodes that is sent data to outside of the cluster increased then workload of the GW is increased too. We implement node clustering for control of active node that worked with a GW. With this work we can control how many nodes are sending data via GW in specific time and when the number of these nodes is increased, we must decrease workload with suitable solution.

4.1 Breaking a Cluster

For controlling workload of a GW, we must control number of the nodes that transmit data to the GW until the amount of nodes do not overreach more than specified limitation. To attain to load balancing, GW should know its power and throughput. Namely GW must know how many nodes can transmit data by itself simultaneously. After that if the number of nodes that connected to the GW increases more than GW capacity, we have to decrease the load of GW with a suitable method. Assume that a GW is working with maximum capacity, the problem occurs when another node wants to get some services from GW. Now the GW can’t respond to this request and the GW is converted to a bottleneck point. To solve congestion in GW we offer breaking cluster to two equal sections and then looking for a new GW for a new cluster (figure 2). After selecting new GW, it must connect to other GWs and connect to new cluster nodes. Therefore new cluster can operate such as the other clusters. Note that the nodes of a cluster must to be closed together geographically because of power saving and simple routing. A disadvantage of this breaking is time wasting for selecting a new suitable GW and making routes between it and other nodes. On the other hand, ordinary nodes that are registering in new cluster must reconstruct the routing table which will waste time again. For these reasons performance will decrement. In the following we present a new formula for selecting a new suitable GW for new cluster and then we propose a solution for above problems.

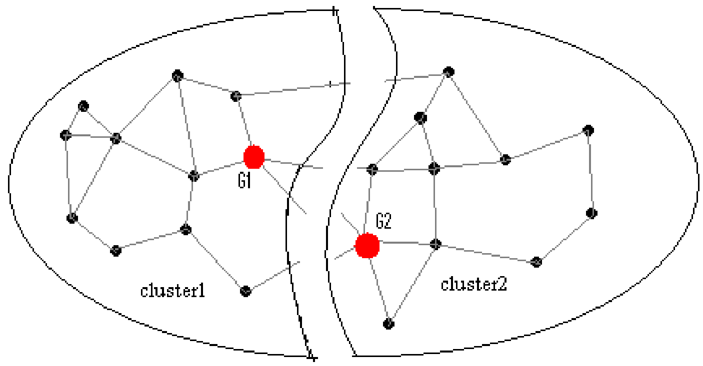

FIGURE 2: Breaking a cluster

4.2 Selecting a Suitable GW

When a cluster splits into two clusters, one of them has a GW but another one (new one) has no GW. So we need to find a suitable GW for the new cluster. Selecting Wrong GW can have effect on network performance. For example assume that if an inappropriate node is selected to be a GW then this GW may fail in its run time. So the cluster must select a new GW and establish a route table which is time wasting and this new GW may fail too. Therefore, random selecting GW causes low network performance because a failing GW causes accumulation data in a cluster and then increasing workload ascendant.

We survey parameters that effect on GW failures and then select a GW that has the best conditions via these parameters. The most effective parameters in stability of GW are: a) power supply, b) velocity of node, c) node constancy, d) distance to center of cluster and e) processing power of node.

The effect of power supply is that if each node that has high energy or has perennial power supply then it is more stable. Therefore the node that has this parameter is more suitable for being a GW because in the future it may be alive. The node with low velocity has less probability to go out from cluster. Therefore a node with low movement and low velocity is more suitable for being GW. In other words, if the node that has high velocity is accepted to be a GW then the GW may go out from the cluster and the cluster has to find and select a new GW again. Node constancy includes the time that a node exists in the cluster. For estimating this parameter each node can monitor a history of its lifetime in specific cluster and then each node that has longer lifetime is more constancy and more suitable for being GW. Central nodes have heavy workload rather than boundary nodes [1], as all nodes select shortest paths for optimal routing and these paths commonly pass from center of cluster. Hence traffic in central of cluster is very heavy. If a GW is selected in the center of a cluster then internal workload will add to external workload thus it is better that a GW is selected from the boundary of cluster. This parameter can be taken from GPS data. At last final parameter that we express is power of processing. A node with high processing power is more suitable for being a GW because it can do computation quickly. We integrated these parameters to a formula that is shown below:

$$G\_Value = \frac{Fower_{Supply} * Power_{CPU} * Constancy * Distance_{from\_center}}{Velocity}$$

In the above formula, we can calculate G_Value for each node in a cluster and then each node that has larger G_Value is more suitable for being a GW.

4.3 Breaking Cluster With Use of Threshold

In previous section we expressed that each GW knows the number of requests in service. If requests amount surpasses the ability of GW then the cluster will be broken. Selecting a new GW plus creating route table for GW and other nodes are time consuming, therefore performance decreases. We present a solution for this problem. We suggest selecting a new GW and creating a routing table before breaking the cluster. To denouement it attends to an example. Assume that a GW can service requests up to 10. It means that if 11th request is sent to the GW, it can`t respond to a new request, thus cluster is broken and then a new GW must be selected and routing table must be established.

In this new method each cluster has two thresholds. One threshold is for selecting a new GW that is called TS_GW and another threshold for established routing table that is called TS_routetable. For example assume that TS_GW is 5 and TS_routetable is 8. It means that if number of nodes which send requests are larger than 5 then cluster must select a new GW. The current GW can do it hereby current GW gets G_Value of all other nodes and each node that has high G_Value is selected for being new GW. With this method the new cluster does not waste any time for selecting a new GW because it is done before breaking cluster.

Also if number of nodes that send requests larger than 8, then pre routing is occurred and route table for GW and nodes of new cluster is made. When number of requests reaches up to 10 then current cluster will be broken. Therefore with this method cluster does not waste any time for selecting a GW and building route table. Simulation results show that breaking the cluster with threshold conquest other algorithms.

4.4 Incorporating Two Clusters

We express that when workload of a GW is increased inordinately, the cluster will be broken into two clusters. What will happen if we assume that workload of a cluster is decreased to zero. In this situation we envisage to several clusters that have low workload. Thus, it is necessary that clusters with low workload joint together. When two clusters are merged together we have to select a suitable GW between two GWs of two old clusters. So we choose the one which has more heavier workload and is more suitable to be the final cluster because the number of nodes that are routed from this GW is larger than the other GW, thereupon we can change previous formula to gain new formula to selecting a new GW in this section. There is a formula as following as below:

$$\text{G value join} = \frac{\text{n\_routetable}^2 * \text{Power}_{\text{Supply}} * \text{Power}_{\text{CPU}} * \text{Constancy} * \text{Distance}_{\text{From \_ center}}}{\text{Velocity}}$$

Where the power supply, power CPU, node constancy, node distance and node velocity parameters are like section 4.2 and n_routetable is number of nodes which are routed via GW, power 2 is to emphasis of this parameter, lastly each node that has high G_value_join is selected for GW of final cluster.

5. PERFORMANCE EVALUATION

We have performed several preliminary quantitative experiments. To this end, the performance of our proposed schemes was evaluated by using NS2 [13], [14].

5.1 Experimental Setup

In order to keep the results closest to real experiment, we used NS2. The simulation area is a 400*400 square as 200 nodes randomly positioned on it. Some of nodes move a little on the simulation area, other simulation parameters listed in table 1.

TABLE 1: Parameters of NS2 Simulation

Parameter	Value
Simulation time	100 sec
Transmission rate	64 Kbps
Node velocity	0-10 Kmph
Wireless coverage	50 meter
Packet size	1000 byte
Routing protocol	DSR
Ratio propagation model	Two ray. Ground
Antenna model	Omni antenna
Mac type	IEEE 802.11

5.2 Service Ratio Evaluation

A Major goal of load balancing is decreasing workload of GW and preventing from bottlenecks. Figure 3 shows the service ratio for different load balancing schemes. The x axis is the simulation time. In figure 3 three schemes are compared as follow: 1) load balancing without clustering, 2) load balancing with simple threshold clustering and 3) load balancing with hysteresis threshold clustering.

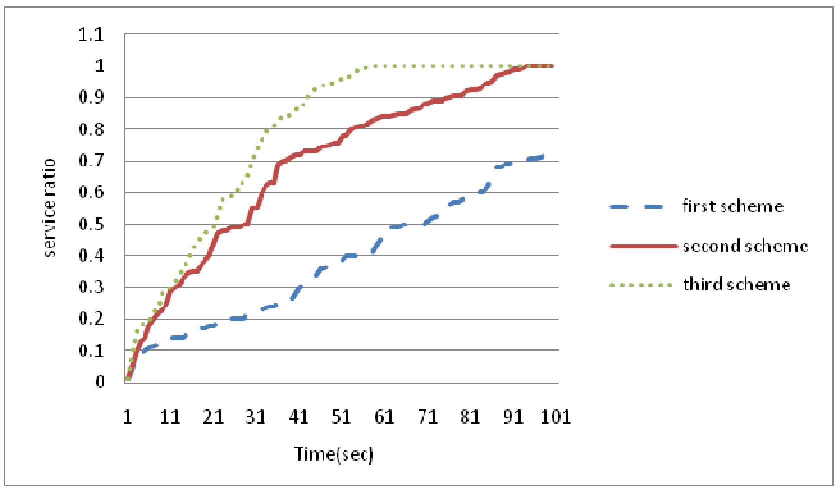

FIGURE 3: comparing service ratio for 3 schemes

As it follows from the figure 3 the third scheme leads to serving more requests in comparison to the other two schemes. The first scheme can't serve all requests because some request aggregate in queue of GW. The second scheme has low service ratio in each time in comparison to third scheme because delay of selecting GW and delay of making route table affect on service ratio.

5.3 The Effect of Workload

Figure 4 shows the effect of number of requests on load balancing performance for three schemes discussed in this paper. As shown in the figure 4 when the number of requests increases the service ratio of third scheme descends with a lower slope in comparison to the other two schemes. It follows figure 4 that in which the first scheme shows very poor performance.

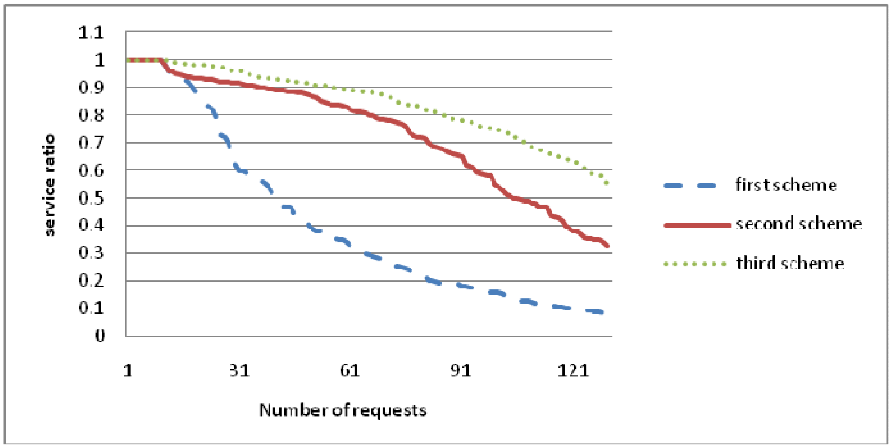

FIGURE 4: comparing service ratio with entered number of requests

5.4 Delay Evaluation

As shown in figure 5 when number of requests is increased then sum of delays in first scheme is increased with a high slope. This delay is the waiting time of requests in the GW queue. We dissemble other delays in network because they are similar in all schemes. Third scheme has least delay.

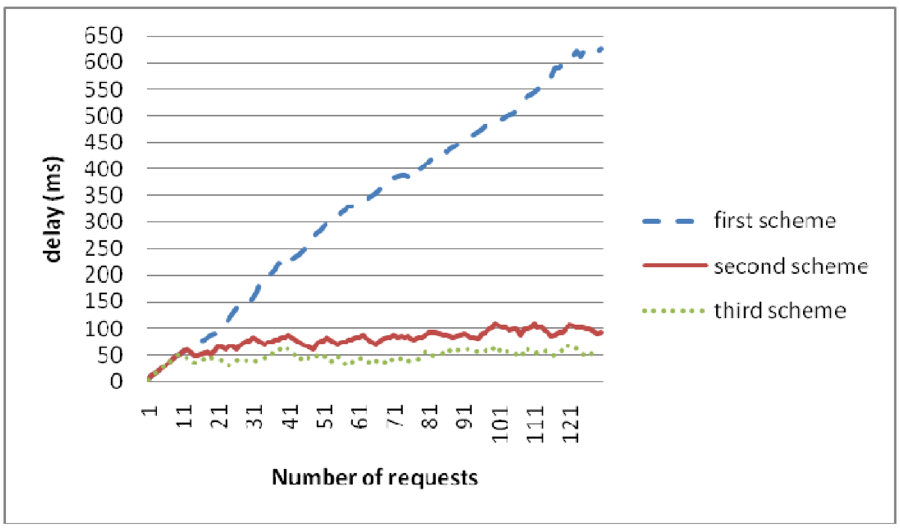

FIGURE 5: Effect of workload on delay

6. CONCLUSION AND FUTURE WORKS

In this paper we proposed load balancing schemes for WMNs. In first step we clustered all nodes to control the workload of them. If workload on a GW is increased up to maximum ability of the GW then the cluster is broken. Because selecting a new GW and establish a route table is time consuming, thus we propose a third scheme in which GW selection and creating rout table is done before breaking the cluster. Simulation results show that the proposed approach offers desirable performance and scalability. Although the paper considers most of the design aspects of the proposed infrastructure, it leaves some open issues and questions. For instance, surveying load balancing of multi channel GWs in clustering wireless mesh networks, finding maximum throughput of nodes in cluster based wireless mesh networks and how to find G_value for selecting a new GW with minimum overhead. Another open issue is using fuzzy logic for breaking the clusters.

7. REFERENCES

[1] J. Bicket, D. Aguayo, S. Biswas, R. Morris, Architecture and evaluation of an unplanned 802.11b mesh network, in: Proceedings of the 11th ACM Annual International Conference on Mobile Computing and Networking (MobiCom), ACM Press, Cologne, Germany, 2005, pp. 31–42.

[2] B. Aoun, R. Boutaba, Y. Iraqi, and G. Kenward, “Gateway Placement Optimization in Wireless Mesh Networks with QoS Constraints,” IEEE Journal on Selected Areas in Communications, vol. 24, Nov 2006.

[3] A.K.Hasan, A. A. Zaidan, A. Majeed, B. B. Zaidan, R. Salleh, O. Zakaria, and A. Zuheir, “Enhancement Throughput of Unplanned Wireless Mesh Networks Deployment Using Partitioning Hierarchical Cluster (PHC)”, World Academy of Science, Engineering and Technology 54 2009

[4] I.F.Akyildiz, X.Wang, W.Wang,” Wireless mesh networks: a survey”, Elsevier ,Computer Networks 47 (2005) 445–487

[5] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, “Impact of interference on multi-hop wireless network performance,” in Proceeding of ACM MobiCom, 2003.

[6] I. Akyildiz and X. Wang, "A survey on wireless mesh networks," IEEE Communication Magazine, vol. 43, no.9, pp.s23-s30,Sep. 2005.

[7] B.S. MANOJ AND RAMESH R. RAO ,” WIRELESS MESH NETWORKING”, Chapter 8 : Load Balancing in Wireless Mesh Networks, page 263

[8] Saumitra M. Das, Himabindu Pucha and Y. Charlie Hu School of Electrical and Computer Engineering Purdue University, West Lafayette, “Mitigating the Gateway Bottleneck via Transparent Cooperative Caching in Wireless Mesh Networks” NSF grants CNS-0338856 and CNS-0626703.

[9] Jangeun Jun and Mihail L. Sichitiu Department of Electrical and Computer Engineering
North Carolina State University Raleigh,”The Nominal Capacity of Wireless Mesh Networks” NC 27695-7911

[10] Abu (Sayeem) Reaz1, Vishwanath Ramamurthi1, Dipak Ghosal1, John Benko2, Wei Li2, Sudhir Dixit3, and Biswanath Mukherjee1 ,”Enhancing Multi-hop Wireless Mesh Networks with a Ring Overlay”

[11] G. Horton, "A multi-level diffusion method for dynamic load balancing", Parallel Computing. 19 (1993), pp. 209-229

[12] I. Akyildiz, X. Wang, W. Wang, "Wireless Mesh Networks: A Survey", Computer Networks Journal 47, (Elsevier), March 2005. pp. 445-487.

[13] “The network simulator - ns2”,http://www.isi.edu/nsnam/ns/.

[14] Daniel Mahrenholz and Svilen Ivanov, "Real-Time Network Emulation with ns-2," Proceedings of The 8-th IEEE International Symposium on Distributed Simulation and Real Time Applications, Budapest Hungary, October 21-23, 2004.

[15] M. Shahverdy, M. Fathy, S. Yousefi, “Scheduling Algorithm for Vehicle to Road-Side Data Distribution”, ICHCC-ICTMF 2009, Berlin Heidelberg, CCIS 66, pp. 22–30, 2010.