A Clustering-Based Channel Assignment Algorithm and Routing Metric for Multi-channel Wireless Mesh Networks

Chao Liu[1], Zhongyi Liu[1], Yongqiang Liu[2], Huizhou Zhao[1], Tong Zhao[1],
and Wei Yan[1]

[1] School of Electronics Engineering and Computer Science, Peking University, China
{liuchao,lzy,zhaohuizhou,zt,weiyan}@net.pku.edu.cn
[2] NEC Laboratory China
liuyongqiang@research.nec.com.cn

Abstract. Multiple non-overlapped channels are available in IEEE 802.11 but are rarely used today in wireless multi-hop networks. Wireless mesh network is a special type of multi-hop ad hoc network and is envisioned to provide high capacity and large coverage. In this paper, we propose a 2-hop clustering based multi-interface, multi-channel network architecture and design a novel channel assignment algorithm and routing metric. Channel assignment is composed of Inter-cluster Static Assignment and Intra-cluster Dynamic Assignment. Since traditional routing metrics, such as hop-count, may not perform well in multi-channel wireless networks, we propose the CDM routing metric, which combines hop-count, channel diversity and channel switching capability together. Simulation results show that our algorithms achieve up to 3.3 times higher end-to-end throughput.[1]

Keywords: wireless multi-hop networks, wireless mesh networks, multi-channel, multi-interface, channel assignment, routing metric.

1 Introduction

Despite significant improvement has been made in physical layer technologies, the bandwidth problem is still severe for multi-hop ad hoc networks due to interference from adjacent hops on the same path as well as from neighboring paths [1]. The IEEE 802.11b/g and 802.11a standard provide 3 and 12 non-overlapped channels respectively, which could be used simultaneously within a neighborhood. However, such bandwidth aggregation is rarely applied to 802.11-based multi-hop ad hoc networks. Most ad hoc network implementations use only a single frequency channel, wasting the rest of the spectrum.

Wireless mesh networks (WMN) is a promising technology emerged recently. The WMN backbone operates just like a network of fixed routers, except that they are connected only by wireless links. In such networks, most of the nodes are stationary

---

[1] This research has been supported by NSCF of china under Grant 60673181. Partially supported by NSCF Grant 60573166 and Network Key Lab Grant of Guang Dong Province.

I. Stojmenovic et al. (Eds.): ISPA 2007, LNCS 4742, pp. 832–843, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and do not rely on batteries. Hence, the focus of WMN design is on improving the network capacity, instead of coping with mobility or minimizing power usage. Providing each node with multiple wireless interfaces offers a promising approach for improving the capacity of WMNs.

A multi-interface-per-node wireless mesh network architecture raises two research questions: channel assignment and routing. Channel assignment deals with assigning a physical channel to a given interface when the number of interfaces per node is lower than the number of available channels. Channel assignment must meet the following two requirements: (1) Maintain the connectivity of the network; and (2) solve the *channel dependency problem* [12]. The second requirement is illustrated in Fig.1. Imagine that there are two interfaces per node and the initial result of channel assignment is labeled on the links. At some moment, the channel of link DE is changed from channel 3 to channel 7. This means that the channel of link EF must also do the same change, so as link DH and HI. This example shows that a change in local channel assignment could lead to a series of channel re-assignments across the network. The channel dependency problem makes it difficult to develop distributed dynamic channel assignment algorithms as local change may have global effects.

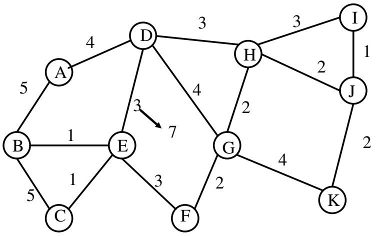

Fig. 1. The Channel Dependency

When nodes have multiple radios, the minimum-hop routing does not perform well [2]. New routing metrics must incorporate the channel diversity in the physical layer of the route. Hence, cross layer design is a key technology for the design of new routing algorithms.

In this paper we make the following research contributions:

- We propose a 2-hop clustering-based multi-interface, multi-channel wireless mesh network architecture. 2-hop clustering largely ensures that non-neighbor clusters could not interfere with each other. The connectivity problem is break up into two aspects: inter-cluster connectivity and intra-cluster connectivity. The channel dependency problem is constricted to a local cluster as channel change of a link in one cluster will not lead to changes in other clusters.
- We develop a novel channel assignment algorithm based on 2-hop clustering. Channel assignment is composed of inter-cluster static assignment and intra-cluster dynamic assignment. In this way, the connectivity of the network is maintained and channels are distributed evenly among the network.

- We propose a new routing metric- Channel Distribution Metric (CDM), which combines hop-count, channel diversity and channel switching capability together. We also modify AODV routing protocol to support CDM based routing.
- The evaluation results show that our Clustering-based Channel Assignment algorithm combining with new CDM metric improve the end-to-end throughput up to 3.3 times.

The rest of the paper is organized as follows. Section 2 reviews past works related to this research. Section 3 describes the channel assignment and routing metric. Section 4 presents the results and analysis of the proposed algorithms. Section 5 concludes the paper with a summary of research contributions and future work.

2 Related Work

Multi-channel / Multi-radio research. Several researchers have proposed MAC protocols based on IEEE 802.11 for utilizing multiple channels [3, 4]. A couple of channel assignment algorithms are also proposed. In [5], authors use multiple 802.11 NICs per node in an ad hoc network by assuming an *identical channel assignment* to all nodes. NIC-1 is assigned channel-1, NIC-2 to channel-2, and so on. This approach can only yield a maximum of factor 2 of improvement using 2 NICs. Raniwala et al. [6] proposed a load-aware channel assignment algorithm. One important assumption of Raniwala's protocol is that traffic load between all nodes are known, which is usually not held in practice.

D-Clustering. For *d-clustering algorithms* such as those presented in [7, 8], each node is either a cluster-Head or is at most *d* hops away from a cluster-Head. The value of *d* is a design parameter of the algorithm. The algorithm proposed in [7] partitions the nodes according to their IDs. The algorithm presented in [8] constructs a multi-layer hierarchy of cluster-Heads (i.e., the cluster-Heads of layer 1 are the cluster members of layer 2). Upper bounds on communication overhead for d-clustering algorithms are investigated in [9].

Routing Metrics. Draves et al. [2] proposed WCETT, a new metric for routing in multi-channel networks. The metric is used with LQSR, a source routing protocol, and ensures "high-quality" routes are selected. WCETT is defined as follows:

$$\mathrm{WCETT} = (1-\beta) * \sum_{i=1}^{n} \mathrm{ETT}_i + \beta * \max_{1<=j<=k} X_j \;. \tag{1}$$

where ETT_i is the expected transmission time of link *i*, *k* is the number of channels, X_j is given as:

$$X_j = \sum_{\text{Hop i is on channel j}} \mathrm{ETT}_i \,, 1 <= j <= k \;. \tag{2}$$

As we can see, X_j is the sum of all the expected transmission time of links working on channel *j* along the route. However, if the path is so long that different links on the same channel do not interfere with each other, X_j will over estimate the cost of the route. In this paper, we propose the CDM metric, somewhat avoiding the drawback of WCETT. The description of CDM will be given in section 3.5.

3 Channel Assignment Algorithm and Routing Metric

3.1 Overview

Fig.2 is an example to explain our channel assignment algorithm. The network is partitioned into different clusters, each of which has a cluster *Head* and a *FIXED* channel, i.e. the common channel within this cluster. For example, the FIXED channel of cluster1 is channel1, and the FIXED channel of cluster2 is channel2. Different clusters are linked by *Gateway* nodes, which are the nodes on the periphery of clusters. Note that one gateway only belongs to one cluster, and there may be multiple Gateways between two clusters. All the links between Gateways are working on the same channel, named *DEFAULT* channel. Those nodes other than Gateways and Heads are called *Ordinary* nodes. After the initial assignment, the nodes may still have more idle interfaces. They set these idle interfaces to *DYNAMIC* status and switch channel according to the local traffic load. It is important to notice that we use clustering-based architecture only for channel assignment and clusters are transparent for routing algorithms and upper layer protocols.

Our algorithms are based on the following three assumptions:

- Every node has at least two network interfaces; but we don't require every node has the same number of interface.
- The network topologies seen by different network interfaces of one node are identical.
- The number of interfaces per node is lower than the number of available channels.

The execution cycle of the algorithms consists of three phases: *Clustering* phase, *Inter-cluster Static Channel Assignment* phase, and *Intra-cluster Dynamic Channel Assignment* phase.

In the Clustering phase, the network is partitioned into clusters. Each cluster has a maximum radius of 2 hops, i.e. every node is at most 2 hops away from its cluster Head.

During the Inter-cluster Static Channel Assignment phase, each cluster is assigned a FIXED channel. The principle of the inter-cluster channel assignment is that neighboring clusters are assigned with different FIXED channels as long as there are enough available channels. After the inter-cluster channel assignment, channels are well distributed in the network and connectivity of the network is maintained.

The last phase is Intra-cluster Dynamic Channel Assignment phase. The cluster Head node continually measures the available bandwidth on each channel, and periodically notifies every member with *Top_Free_Channel* message, i.e. the list of freest channels in its area. Other member nodes switch the channels of their DYNAMIC interfaces based on local traffic load to get more bandwidth. Data packets are transmitted on FIXED, DEFAULT and DYNAMIC interfaces, while all the broadcast and control/management messages, including those flood messages from upper layer, are transmitted on the FIXED and DEFAULT interfaces to guarantee every node get the messages.

To explore the potential of our multi-channel network as much as possible, we propose a new routing metric named CDM and modify the AODV routing protocol to support CDM-based routing.

One distinct characteristic of our entire algorithm is that we assign the channels based on clustering. The motivation behind that includes:

- It makes channel dependence problem restricted within the cluster since different clusters use different FIXED channel and are only connected by DEFAULT channel.
- The neighboring clusters can use non-overlap FIXED channels to decrease the interference and improve the throughput.
- Since the nodes within one cluster are connected with FIXED channel and every cluster are connected with DEFAULT channel, so the entire network maintains exactly equivalent connectivity as the single channel network.
- Since all members within a cluster share one common FIXED channel, the broadcast and flooding, which is considered to be a complicated problem in multi-channel network, can be done efficiently in our network.

3.2 Clustering Algorithm

An amount of clustering algorithms are already proposed by researchers. In this paper, we adopt the Max-Min D-cluster algorithm proposed by [7]. Max-Min D-cluster is a distributed algorithm. It partition nodes into *d*-clusters based on their IDs, where *d* is an input parameter. After the algorithm converges, every node is either a cluster Head or at most *d* hops away from its cluster Head.

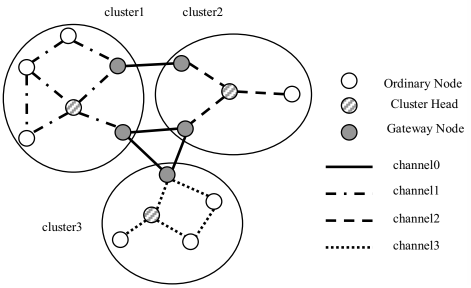

Fig. 2. Example for Channel Assignment

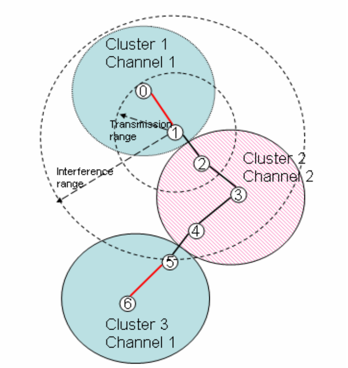

Fig. 3. Non-neighbor Cluster Interference in 1-cluser

One critical decision of our algorithm is how to choose the radius of clusters (i.e. the parameter *d*). We prefer 2-hop clustering due to the following reasons:

If the radius of clusters is set to 1, then two non-neighbor clusters may still interfere with each other. For example, In Fig. 3, cluster 1 and cluster 3 are non-neighbors and both choose channel 1 to be FIXED channel. Node's transmission range is 250, and interference range is 550. Node 1 and node 5 are within each other's interference range. In contrast, if the radius of the clusters is 2, then non-neighbor cluster interference can only occur when the intermediate cluster is small enough, which reduces the possibility of non-neighbor cluster interference to a very small extent.

If the radius of clusters is larger than 2, the efficiency of clustering algorithm will be reduced sharply. And the intra-cluster interference will get severe, despite the Intra-cluster Dynamic Channel Assignment.

3.3 Inter-cluster Static Channel Assignment

At first, every node set one of its interfaces to work on the DEFAULT channel (this interface is called DEFAULT interface), e.g. channel 1. Then the network performs Max-Min D-cluster on the DEFAULT channel and nodes are partitioned into clusters. After that, the inter-cluster channel assignment phase begins. It includes three steps:

1. In the first step, every cluster Head performs a conflict-avoiding algorithm to find the FIXED channel of this cluster.
2. Next, Head nodes flood this FIXED channel to its cluster members.
3. At the last step, Ordinary nodes change their DEFAULT interface to FIXED channel, and Gateway nodes set one idle interface to the FIXED channel.

After all the steps are completed, the connectivity within cluster is maintained by the FIXED channel, and neighboring clusters are linked with Gateways using the DEFAULT channel. Hence the connectivity of our network is maintained, and equivalent to the single-radio single-channel network.

The conflict-avoiding algorithm is the key design issue of the inter-cluster channel assignment. This problem can be considered as a classical graph k-coloring problem: each cluster is a vertex in the graph; two clusters are neighbors means that there is one edge connecting these two vertices in the graph; then color all vertices with k colors (k is the number of the available channels), so that neighboring vertices have different colors, or the number of conflicts is as small as possible.

To solve this problem, we design a distributed greedy algorithm. Every node runs following steps for one time:

1. Wait all the neighboring Heads that have higher IDs than its own to tell it about their FIXED channels, and store their FIXED channels in a *Neighbor_Channel* list.
2. Choose a free channel which is not included in *Neighbor_Channel*, or choose a channel which causes least conflicts if there is no free one
3. Notify all the neighboring Heads that have lower IDs than its own about the FIXED channel it chose.

To determine the minimum number of colors needed to color a given graph is NP-complete [11]. Our algorithm does not intend to color the graph with minimum colors, but gets a suboptimal solution and causes few color conflicts with a greedy method. In worst case, where all clusters are connected as a complete graph, the time complexity of the algorithm is $O(n^2)$ (n is the number of clusters). In most case, the time complexity is $O(mn)$, where n is the number of clusters, m is the maximum degree of vertices in the graph.

3.4 Intra-cluster Dynamic Channel Assignment

After all clusters get FIXED channels, the nodes may still have some idle interfaces (also called DYNAMIC interfaces), especially on those non-Gateway nodes which only use one interface on the FIXED channels. This section is about how to utilize these DYNAMIC interfaces.

It is well recognized that channel assignment and routing can be efficiently executed only if the traffic patterns are taken into account [10]. Considering that the global traffic load information is sometimes impossible to get, we will use some local information in our dynamic channel switching.

In the intra-cluster channel assignment phase, each Head node selects a particular node, called *Measurer*. If this Head is not a Gateway, it selects itself to be Measurer, else it selects the one that is non-Gateway and has largest ID. The selected Measurer sets one idle interface to hop on the available channels (i.e. all channels except those already used by this cluster). On each time slot, the Measurer node listens on that channel and gets an estimation of free time percentage. The estimation can be simply done by subtracting the channel busy time from the entire time slot. Thus the Measurer node will always get an updated list about which channels are the freest in its local area. This list is called *Top_Free_Channels*, and is periodically flooded to the cluster members.

Then cluster members can use the ranking in the *Top_Free_Channels* to adaptively switch their DYNAMIC interfaces. There are two different strategies of this dynamic switching, and we call it *Greedy Switching* and *On-demand Switching*.

The Greedy Switching is straightforward: if a node has n idle interfaces, it sets these interfaces on the first n channels in the Top_Free_Channels list. Every node use hello message to discover neighboring nodes on each interfaces and use ETT [2] metric to choose the interface to transmit data packets when there are more than one common channels between itself and another neighboring node.

Beside the Greedy Switching, another strategy of dynamic channel assignment is On-demand Switching. Normally, all nodes use FIXED channel for data packets transmitting, and Gateway nodes also use DEFAULT channel to transmit inter-cluster traffic. At the same time, nodes continually measure the free time percentage on the FIXED channel using method of promiscuous listening. If the free time percentage drops to a threshold (30% in our implementation), this node will select a channel from Top_Free_Channles list, enable an unused DYNAMIC interface to the selected channel, and then broadcast a message on the FIXED interface to notify all its neighbor nodes about the DYNAMIC channel it just enabled. The broadcasted message is called *Dynamic_Change*, and includes the ID of the node and the ID of the selected channel. If the neighbor node has one DYNAMIC interface that is not used yet, it enables the interface to the selected channel, and replies a *Dynamic_Change_ACK* message to the original node. If no reply is received, the original node gives up this channel switch. After a while, if the congestion is relieved and free time percentage of FIXED channel rises to another threshold (75% in our implementation), the original node recalls this channel switch.

When a node chooses the suitable channel from the Top_Free_Channles list, it excludes those channels already selected by its own and its neighbor nodes from the list, and chooses the first channel that remains in the list.

The Greedy Switching and On-demand Switching are different in their ways of utilizing Free Channels. In Greedy Switching, every node makes use of the available channels as much as possible to achieve higher throughput. But they probably cause more interference. In On-demand Switching, nodes use the channels in a modest way, and they enable the DYNAMIC interface only when necessary. So when congestion happens in an area of the network, the free channel can be quickly enabled to relieve the heavy traffic load.

There is an assumption behind our algorithm: the measurement on the Measurer node can also reflect the channel usage situation on the positions of other member nodes. This is not always held in the wireless network. But considering that the overhearing range is much larger than the transmitting range, the Measurer node can mostly overhear the transmission of other cluster members. Besides, we only need a rough rank of the free channels, and the measurement on Measurer node is good enough. There is no need to make every node do the measurement itself, because per-channel hopping of the measurement would dedicate an interface and seems needless.

3.5 CDM Routing Metric

In this section, we propose a new routing metric – *Channel Distribution Metric* (*CDM*) and adapt AODV routing protocol to support CDM.

There are three goals for the CDM designing. First, it should take the channel diversity of a path into account, because the interference along the path is a main factor that decreases the throughput.

Second, the metric should be increasing while adding new route into the path. Because a longer path will obviously consume more resource in the network, a node should try to choose shorter path in the consideration of global optimal. A longer path would also cause longer roundtrip time and hence decrease the end-to-end TCP throughput. Last but not least, many routing protocols (e.g. DSR, AODV, DSDV and etc.) demand the metric to have non-decreasing property.

Third, the dynamic channel switching capability of nodes along the path should be explicitly considered. Intra-cluster dynamic channel assignment is a distinct characteristic of our multi-channel network. It can reduce the interference within the cluster and thus improve the throughput. The metric should make good use of this characteristic.

Our CDM metric is defined as follows:

$$\text{CDM} = \alpha \times \text{MLC} + \beta \times \text{HopCount} - \gamma \times \text{VCM} \ . \tag{3}$$

where α, β, γ are parameters subject to $0 < \alpha < 1$, $0 < \beta < 1$, $0 < \gamma < 1$, and $\gamma <= \beta$. A smaller CDM value indicates the route is better.

MLC donates the Maximum interfered-Link Count and is defined as

$$\text{MLC} = \max_{i=1,2 \ldots n-1} \left\{ \sum_{j=i+1}^{\min(i+InterferenceLen,n)} \text{I}(\text{C}(i) == \text{C}(j)) \right\} \ . \tag{4}$$

where *n* is the hop count of the route and *I(C(i) = C(j))* is an indicator function that is equal to one when channels being used by link *i* and link *j* are the same, otherwise set to zero. The *InterferenceLen* is the interference range measured in hop count; it can be a constant value (equal to 3 in our system). So MLC accounts for the interference along the path, and furthermore reflects the path's channel diversity.

HopCount, the second part of equation (3), is derived from the traditional minimum-hop routing. It reflects the end-to-end delay of the path and the overall transmission time it consumes.

VCM is the abbreviation of *Variable Channel Metric*, and is given as follows.

$$VCM = \sum_{i=1}^{n} HasUnusedInterface(i) \ . \tag{5}$$

Where n is hop count, *HasUnusedInterface(i)* is an indicator function which is equal to one when node i has one or more unused DYNAMIC interface, otherwise equal to zero. Thus VCM reflects the channel switching capability of a route. The reason why the *HasUnusedInterface(i)* function is not equal to the number of the idle interfaces is because the CDM metric would lose the non-decreasing property if we allow the value of *HasUnusedInterface(i)* to be more than one. Consider adding one node that has a big number of idle interfaces into the route, and this node would probably lower the entire CDM metric, which is not the situation we want. The last constraint of the parameters, $\gamma <= \beta$, is also for the assurance of the non-decreasing property, because VCM should get lower weight than HopCount.

3.6 Discussion

Our channel assignment algorithm can eliminate most interference between neighboring clusters. But the intra-cluster interference still exists, especially on the FIXED channel. The Intra-cluster Dynamic Channel Assignment is designed to mitigate the interference within cluster. But due to the limited number of non-overlapping channels (3 orthogonal channels in 802.11b/g and 12 in 802.11a), the interference is inevitable.

Max-Min D-cluster is a simple clustering algorithm, whose result depends mainly on the distribution of IDs in the network. Max-Min D-cluster may lead to some small clusters (i.e. clusters with radius equal to one or clusters with small number of members). Small cluster may cause non-neighboring interference. Fortunately, the algorithm of clustering is substitutable, and one of our future works is designing a topology-based clustering algorithm and replacing Max-Min D-cluster.

Once Clustering and Inter-channel Static Channel Assignment are done, the clustering result and FIXED channels are fixed for a long period of time, such as hours or days. When a new node appears in the network, it just scans on every channel, looks for a neighboring cluster, joins in it, and set one interface on the FIXED channel of that cluster. Then depending on whether or not this new node has other neighboring clusters, it would become a Gateway node and set another interface on DEFAULT channel. When a node in the network fails, if this node is a cluster member, nothing needs to do. If the failing node is a cluster Head, other members in this cluster will sense this event through a continual absence of Top_Free_Channel message, and elect a new cluster Head that with the largest ID.

4 Evaluation

In the evaluation, we will compare our algorithm with one algorithm proposed by [6]. [6] designed two centralized channel assignment algorithms, *neighbor partitioning scheme* and *load-aware scheme*. The load-aware scheme outperforms the other one in their evaluation, but requires global traffic profile and needs to compute routes for all flows in the beginning of the experiment, which is not always feasible in real network. So we choose the neighbor partitioning scheme for comparing.

We modify NS2 to support multiple wireless interfaces per mesh node. The data rate is fixed to 1M bits/s, and transmission range and interference range are set to 250m and 550m respectively. The number of orthogonal channels is 12, which is the situation of IEEE 802.11a. Two types of network topologies are evaluated: grid topology and random topology. Specifically, we use an 8 * 8 grid topology, and the IDs are randomly distributed and different in each run. The random topology is 50 nodes distributed randomly in a 1500m * 1500m area.CBR UDP traffic with 512 byte packet size and 0.006s interval is used. *Throughput* is defined as the correct byte received by the destination node of each CBR flow. The *aggregated throughput* is the sum of throughput of all flows in the whole network.

In the implementation of our algorithm, we use On-demand Switching scheme to do the Intra-cluster Dynamic Channel Assignment. Currently, we have not implemented the Greedy Switching scheme yet. After we complete the Greedy Switching part, comparison of these two schemes will be given. For the CDM metric, in equation (3) and (4), the parameters are set as: $\alpha = \beta = \gamma = 1/3$, $InterferenceRange = 3$.

In below sections, we will compare the aggregated throughput of three kinds of networks:

- I-NET- Every node has only one interface and the network has only one channel.
- II-NET- Every node has two interfaces. The neighbor partitioning channel assignment proposed by [6] is used. The routing protocol is unmodified AODV.
- III-NET- Every node has two interfaces and our clustering-based channel assignment algorithm is used. The routing protocol is CDM-based AODV.

3 Long flows (i.e. longer than 4 hops) and 5 short flows (i.e. equal or shorter than 4 hops) exist simultaneously in each scene. The simulation time is 400s. Every scene is run for 50 times, and the final results are the average of the 50 runs. In each run, the source and destination nodes of each CBR UDP flow are randomly generated.

We present the aggregated throughput of short flows and long flows respectively in Table 1, group by the topologies and the type of flows.

Table 1. The average aggregated throughput (in Mbps) of three networks, group by topology and flow type

	Grid topology			Random topology		
	I-NET	II-NET	III-NET	I-NET	II-NET	III-NET
Short flows	1.4	1.8	2.5	0.7	1.1	1.4
Long flows	0.3	0.6	1.0	0.1	0.1	0.3

Fig. 4 and Fig. 5 are the aggregated throughput for 5 short flows. In Grids topology, the average throughput of I-NET is about 1.4Mbps, while II-NET is 1.8Mbps (28.6% higher than I-NET) and III-NET is 2.5Mbps (78.6% higher than I-NET). In random topology, the average throughputs are 0.7Mbps, 1.1Mbps and 1.4Mbps. II-NET and III-NET improve 57.1% and 100% upon I-NET respectively. The reason why III-NET outperforms II-NET is that: in III-NET, channels are distributed more reasonably and CDM-base AODV can utilize the multi-channel better than the traditional AODV.

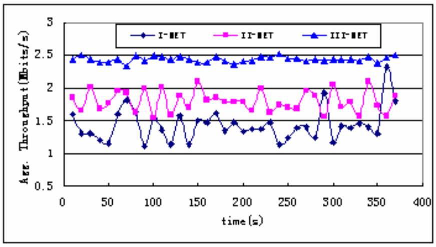

Fig. 4. Aggregated Throughput of Short Flows (Grid Topology)

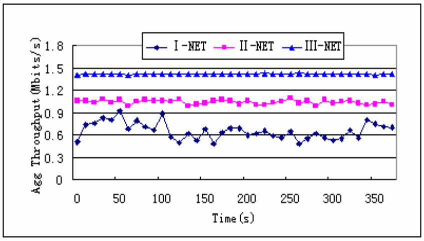

Fig. 5. Aggregated Throughput of Short Flows (Random Topology)

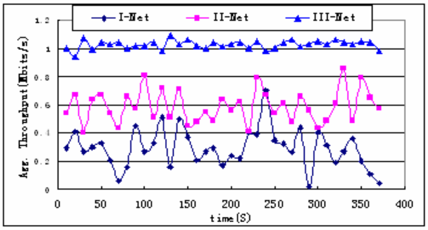

Fig. 6. Aggregated Throughput of Long Flows (Grid Topology)

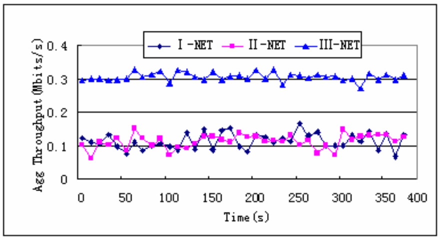

Fig. 7. Aggregated Throughput of Long Flows (Random Topology)

Fig. 6 and Fig. 7 are results for 3 long flows. As the figures indicate, in grid topology the aggregated throughput of III-NET is 3.3 times higher than I-NET while the aggregated throughput of II-NET is 2 times higher than that of I-NET. In the random topology, the results are similar for III-NET. But one notable phenomenon is that the aggregated throughput of II-NET is only slightly higher than I-NET. This is probably because the Neighbor Partitioning algorithm only considers the interference between adjacent links. Actually, the interference range is much larger than one hop, and links may interfere with each other even though they are not directly adjacent. This kind of interference gains more importance along the longer routes, and thus degrades the improvement caused by multi-channel in II-NET.

When comparing III-NET with I-NET, the improvement on long flows is much higher than that of short flows. We believe that this is caused by the following reason: the interference along the hops of one long route, which is also called *intra-route interference*, is the main factor affecting the end-to-end throughput. In III-NET, channel diversity between neighboring clusters along the route can greatly mitigate the intra-route interference, thus sharply increase the throughput. This meets our goal of enhancing the throughput of long flows to a higher extent than short flows.

5 Conclusions and Future Work

To utilize the non-overlapping channels available in IEEE 802.11a/b/g, and enhance the network capacity, we propose a channel assignment and routing algorithm based on 2-hop clustering. The channel assignment algorithm is a hybrid of static assignment and dynamic assignment: we use static channel assignment for inter-cluster assignment and dynamic switching for intra-cluster channel assignment. A new routing metric, named CDM, is proposed to enable the routing algorithm to select routes that can achieve higher end-to-end throughput. We also modify the AODV routing protocol to implement CDM based routing. Simulation results show that our algorithms can improve the end-to-end throughput up to 3.3 times.

At present, we use a simple but efficient clustering algorithm Max-Min D-cluster. However, the clustering result of Max-Min D-cluster depends on the distribution of node IDs on the plane. In the future, we may replace Max-Min D-cluster with other clustering schemes, e.g. connectivity based clustering. Our CDM metric combines hop-count, channel diversity and channel switching capability together, but the impact of weights on these three parts (i.e. the parameters in equation (3)) has not been explored. Hence the routing metric still needs to be improved. We have already developed a prototype of multi-channel wireless mesh network (called MeshNet) with 7 nodes; extensive experimental test will also be the future work.

References

1. Jain, K., Padhye, J., Padmanabhan, V.N., Qiu, L.: Impact of interference on multi-hop wireless network performance. In: Proc. IEEE/ACM MOBICOM (2003)
2. Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop Wireless Mesh Networks. In: Proc. ACM MOBICOM (2004)
3. Nasipuri, A., Zhuang, J., Das, S.R.: A multichannel CSMA MAC protocol for multihop wireless networks. In: Proc. IEEE WCNC (1999)
4. Jain, N., Das, S., Nasipuri, A.: A Multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In: Proc. IEEE International Conference on Computer Communications and Networks (IC3N). IEEE Computer Society Press, Los Alamitos (2001)
5. Kyasanur, P., Vaidya, N.H.: Routing and interface assignment in multi-channel multi-interface Wireless Networks. In: Proc. IEEE WCNC. IEEE Computer Society Press, Los Alamitos (2005)
6. Raniwala, A., Gopalan, K., Chiueh, T.: Centralized channel assignment and touting algorithms for multi-channel Wireless Mesh Networks. In: ACM SIGMOBILE Mobile Computing and Communications Review, vol. 8(2), pp. 50–65 (April 2004)
7. Amis, A., Prakash, R., Vuong, T., Huynh, D.: Max-Min D-Cluster formation in wireless ad hoc networks. In: Proc. IEEE INFOCOM. IEEE Computer Society Press, Los Alamitos (2000)
8. Banerjee, S., Khuller, S.: A clustering scheme for hierarchical control in multi-hop wireless networks. In: Proc. IEEE INFOCOM, pp. 1028–1037. IEEE Computer Society Press, Los Alamitos (2001)
9. Sucec, J., Marsic, I.: Clustering overhead for hierarchical routing in mobile ad hoc networks. In: Proc. IEEE INFOCOM, vol. 3, pp. 1698–1706 (2002)
10. Alicherry, M., Bhatia, R., Li, L.E.: Joint channel assignment and routing for throughput optimization in multi-radio Wireless Mesh Networks. In: Proc. ACM/IEEE MOBICOM (2005)
11. Cheeseman, P., Kenefsky, B., Taylor, W.: Where the really hard problems are. In: Proc. 12th International Joint Conference on AI (IJCAI-91), vol. 1, pp. 331–337 (1991)
12. Raniwala, A., Chiueh, T.: Architecture and algorithms for an IEEE 802.11-based multi-channel Wireless Mesh Network. In: Proc. IEEE INFOCOM. IEEE Computer Society Press, Los Alamitos (2005)