Dynamic Channel Assignment for Wireless Mesh Networks using Clustering

Sadeq Ali Makram Mesut Güneş[†] Amine Kchiche Martin Krebs
Department of Computer Science, Informatik 4, RWTH Aachen University, Germany
{makram, guenes, kchiche, krebs}@nets.rwth-aachen.de
[†]Department of Computer Science, FU Berlin, Berlin, Germany
guenes@inf.fu-berlin.de

***Abstract*—The aggregate capacity of wireless mesh networks can be improved significantly by equipping each node with multiple interfaces and by using multiple channels in order to reduce the effect of interference. Since the number of available channels is limited, it is desired to allocate and reallocate channels on-demand. In this paper, a Cluster Channel Assignment (CCA) approach is proposed, to maximize the aggregate throughput by exploiting spatial reuse and local dynamic switching of the channels. A clustering approach is employed in order to maximize the network capacity while minimizing the interference and taking advantage of the possibility of reuse of channels among clusters.**

***Index Terms*—Channel Assignment, Clustering, Load Balancing, QoS and Wireless Mesh Networks.**

I. Introduction

Recently, wireless mesh networks (WMNs) are in the focus of academia and industry research. The reason is that WMNs have several interesting characteristics, such as self-organization, self-configuration, reliable services and Internet connectivity. WMNs consist of mesh-routers and mobile-clients nodes. A mesh-router is used to route the data to Internet connection and as well as an access point to the mobile-client nodes. To further improve the flexibility of mesh networking, a mesh-router (node) is equipped with multiple Wireless Network Interfaces (WNIC) built on either the same or different wireless access technologies. Today, the usage of multimedia applications and Internet connection is rapidly rising. The most important requirements for these applications are quality of service (QoS). To support high traffic load, we generally add more bandwidth by setting up additional channels. The interfering wireless links would then be able to operate on different channels enabling multiple parallel transmissions with a minimum interference.

The goal of channel assignment approaches is to allocate the available channels to network interfaces of nodes in a way that satisfies load balance and provides reasonable services to the users, i.e., the available bandwidth of the virtual wireless links should be proportional to the expected load, while taking into account ensuring network connectivity, and minimizing the overall interferences. To reduce the complexity of channel assignment, we have employed a clustering strategy. In fact clustering provides an effective way to allocate and reuse the wireless channels among different clusters.

The rest of this paper is structured as follows. In Section II, related work is discussed. In Section III, we present our contribution. Subsequently, in Section IV we describe the network model and introduce the terminology used throughout this paper. Section V presents our approach CCA in detail. Section VI evaluates the performance of CCA. Finally, Section VII concludes the paper.

II. Related Work

Bahl et al. [1] propose the dynamic switching of channels in such a way that the neighbors meet periodically on a common channel to communicate. The advantage of the approach is that it neither requires the modification of the MAC protocol nor multiple network interfaces. The drawback is the synchronization of the nodes, which is difficult to achieve. So et al. [2] propose that the nodes which have packets to transmit negotiate with the destination who sends in a specific time window. This approach assumes also that all nodes are synchronized. Wu et al. [3] suggest to divide the overall bandwidth in $n+1$ channels, one channel for control information and the other n to transmit data packets.

There are also approaches which assume multiple interfaces per node. Shin et al. [4] show that optimal channel assignment is NP-hard, and propose to assign as many distinct channels as possible to a node to improve the performance while satisfying the constraints of limited NICs and available channels. The channel selection to particular network interfaces is done randomly.

Raniwala et al. [5] propose a distributed channel assignment joined with routing. They represent a WMN as multiple spanning trees, and assume that a node can join multiple spanning trees to distribute the load among the trees. In their channel assignment approach nodes positioned higher in the tree hierarchy get a higher priority, since they are connected to the Internet. The nodes lower positioned in the tree hierarchy get lower priority in choosing channels and that may result in discriminating these nodes which can affect their communication performance negatively.

Ko et al. [6] propose a distributed channel assignment algorithm where each node can choose greedily a channel that minimizes its local objective function depending only on local information. Every node selects a channel that minimizes the sum of interference cost within its interference range. The

978-0-7695-3106-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICN.2008.96

advantage of their approach is that channel assignment can be achieved based on local information among nodes. However, they don't consider the number of interface cards per node and they don't deal with interface violation constraint mentioned earlier.

Another two approaches for channel assignment that are close to our presented approach are Tabu-based [7] and CLICA-SCE [8]. Subramanian et al. [7] designed a centralized Tabu-based algorithm and a Distributed Greedy algorithm. Both algorithms assign channels to communication links with the objective of minimizing network interference. Tabu-based algorithm consists of two phases. The first phase tries to find a good solution with minimum interference. However, this solution may violate interface constraint which is handled in the second phase. Furthermore, Tabu-based does not work well when the number of radio interfaces is limited.

Marina et al. [8] propose a polynomial-time heuristic algorithm (CLICA) for assigning channels to nodes radios. The algorithm assigns each node a given priority; and depending on this priority, the coloring decision is taken. Starting from the node with the highest priority, the algorithm tries to color all uncolored incident links from this node.

III. CONTRIBUTIONS

The main goal of channel assignment approaches is to allocate the available channels to network interfaces of nodes in a way that maximizes the average throughput, i.e., the available bandwidth of the virtual wireless links should be proportional to the expected load. To achieve this goal some requirements have to be fulfilled:

- Ensure network connectivity: The wireless mesh network must not be split due to channel assignment. This can happen, if a node does not share a common channel with any of its neighbors.
- Minimize the overall interferences: The interference generated by neighboring nodes (1-hop, 2-hop etc.) should be minimized to decrease the packet loss probability and improve thereby the overall performance of the network.
- Adaptive to the traffic load: The channel allocation has to be on-demand and based on the load of nodes.

In order to reach these objectives, a clustering strategy has been deployed. The purpose of clustering is to minimize the complexity of channel assignment into small local problems that are easier to handle. It permits us also to reuse channels among clusters in order to effectively utilize the bandwidth and minimize the interference. Furthermore, the channel assignment in a cluster and in between clusters can be locally controlled by the clusterhead.

IV. SYSTEM MODEL AND PROBLEM FORMULATION

A. Network Model

The considered wireless mesh network (WMN) constitutes a graph $G(V, E, K)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes, $K = \{k_1, k_2, \ldots, k_c\}$ the set of available channels, and $E = \{(v_i, u_j, k_r) | v_i, u_j \in V \wedge k_r \in K\}$ the set of virtual wireless links between the nodes v_i and its neighbors u_j on channel k_r. For the sake of simplicity we will denote $l^a_{i,j} = (v_i, u_j, a) \in E$ as the wireless link between node v_i and u_j on channel $a \in K$. The set $L_i = \{l^a_{i,j}\}$ describes all wireless links of the node i.

A node v_i with m_{v_i} wireless network interfaces may allocate up to m_{v_i} different channels, if available. The set of assigned channels to node v_i is denoted as $K_{v_i} = \{k_1, k_2, \ldots, k_n\}$, $n \leq m_{v_i}$. Furthermore, N^a_v denotes the one step neighbors of node v_i on channel a and all neighbors are given by $N_v = \bigcup_{a \in K_v} N^a_v$. Based on the previous terms, the channels of all neighbors of node v_i are given by $K_{N_v} = \bigcup_{u \in N_v} K_u$.

B. Clustering

Our approach requires a clustering at the beginning, wherein the router nodes are grouped into subsets of nearby nodes $C = \{C_1, C_2, \ldots, C_c\}$. We deploy the Highest Connectivity Cluster (HCC) algorithm [9], where a node is elected as a clusterhead (CH) if it is the most highly connected node (having the highest number of neighbor nodes). It is also possible to employ any clustering algorithm which realize a uniform clustering and where the clusterhead is in the center of the cluster.

C. Traffic Load Estimation

Additionally, the second phase of our approach requires the information about the current load on a wireless link $l^a_{v,u}$ and the quality of the link. The approach we deploy is based on the packet loss probability. For that purpose, each mesh router v_i counts all sent packets $s(l^a_{v,u}, t)$ and acknowledgment packets received $r(l^a_{v,u}, t)$ for the sent packets on the link $l^a_{v,u}$, where the channel is a, during a specified time interval t. The packet loss probability on link $l^a_{v,u}$ from the point view of node v_i is given by:

$$P(\text{loss on } l^a_{v,u}) = 1 - \left(\frac{r(l^a_{v,u}, t)}{s(l^a_{v,u}, t)} \right) \quad (1)$$

There are many reasons for packet loss. These reasons include the high usage of the channel, hidden terminal problem, and interference from nearby nodes.

The node v_i shares the link information with its neighbors u_j as well as with a clusterhead. Notice, that all these calculations are done locally on each node and only represent the view of the network from the point of view of node v_i, since wireless links show strong asymmetry. Therefore, each clusterhead gather all links information belonging to its cluster to control the dynamic channel switching for on demand. Furthermore a clusterhead can share these information with neighbors clusterheads for the purpose of QoS routing and load balancing.

V. CLUSTER CHANNEL ASSIGNMENT

The *Cluster Channel Assignment* consists of two stages. In the first stage the clustering algorithm mentioned in Section IV-B is applied to compute the clusters, see Figure 1(a).

TABLE I
NOTATION

Symbol	Definition
C_i	Cluster i
CH_i	A cluster head of cluster i
K	Set of available channels
K_{C_i}	Set of channels allocated to cluster i
NC_i	Set of neighbors cluster of cluster i
B_{C_i}	Set of border nodes of cluster i
v, u	A node in the network
N_v	Set of neighbors node of node v
m_v	The number of WNIC in v
k_x	A channel in K
K_v	Set of assigned channels to node v
$\|K_v\|$	The number of divers channels allocated to the WNICs of node v

Subsequently, the available channels in the network are equally distributed to the clusters in a way that two neighbored clusters get disjoint sets of channels. In the case that the clustering is not uniform, we distribute the available channels K to neighbored clusters as follows:

$$|K_{C_i}| = \begin{cases} \lfloor n_i \cdot \frac{K}{n} \rfloor, & \text{if } |K| > |NC_i| \\ 1, & \text{otherwise} \end{cases} \quad (2)$$

$$K_{C_i} \subset \begin{cases} K \setminus (K \cap K_{NC_i}), & \text{if } |K_{NC_i}| < |K| \\ min(K_{NC_i}), & \text{otherwise} \end{cases} \quad (3)$$

Equation 2 determines the number of channels $|K_{C_i}|$ that can be assigned to cluster C_i. $|K_{C_i}|$ is proportional to the number of nodes n_i in cluster C_i where the number of available channels $|K|$ is higher than the number of neighbors NC_i otherwise each cluster is assigned only one channel. Equation 3 determines the set of channels K_{C_i} to be assigned to cluster C_i. A cluster is marked as *assigned* when it gets a set of channels. In the case of small number of available channels, we reuse the least used channels from K_{NC_i} which are already assigned to some neighbor clusters. For example, if there are only 3 available channels $|K| = 3$ and $NC_i = 4$, then each cluster is assigned only one channel where each cluster gets a disjoint channel. In this case, we can reuse the least used channel from K_{NC_i} and then assign it to the rest of the neighbor clusters. This distribution of the channels is repeated for all unassigned clusters in the network. We allow by this distribution the other clusters to reuse the same channels as those used by cluster C_i and its neighbors. Based on these assumptions and using the terminology defined in Table I, we describe our algorithm called *Cluster Channel Assignment* (CCA) in detail as next.

Algorithm 1 Static CCA

Phase 1: Static CCA
1: Each cluster C_i
2: **while** node v in C_i **do**
3: {/* Allocate one channel from K_{C_i} to all nodes in the cluster C_i*/}
4: Given $k_x \in K_{C_i}$
5: Assign channel k_x to v
6: **if** $v \in B_{C_i}$ **then**
7: {/*If node v is a border node and has free WNIC, then assign the channel of the neighbor cluster to it*/}
8: **if** $(|K_v| < m_v) \wedge (\exists j \in NC_i)$ **then**
9: Given $k_y \in K_{NC_i}$
10: Assign channel k_y to v, $K_v = (k_y \cup K_v)$
11: **end if**
12: **end if**
13: **end while**

Algorithm 2 On demand CCA

Phase 2: On demand CCA
Each cluster-head CH_i
if $\big((|K_v| < m_v) \wedge (\exists u \in N_v, |K_u| < m_u)\big)$ **then**
{/*Both partners have an unused WNIC*/}
if $(\exists k_x \in K_{C_i})$ **then**
{/*The cluster has a free channel*/}
Allocate channel k_x to v and u
else
Send borrow-request to the neighbor cluster NC_i who offers $k_x = freechannel\{K_{NC_i}\} \vee \min\limits_{load}\{K_{C_i} \cup K_{NC_i}\}$
Allocate channel k_x to v and u
end if
else if $(|K_v| < m_v) \wedge (\nexists u \in N_v, |K_u| < m_u)$ **then**
{/*Only router v has a free WNIC*/}
$k_x = \min\limits_{load}\{K_{N_v}\}$
Allocate channel k_x to v
else
{/*No unused WNICs are available*/}
$\exists u \in N_v | k_x = \min\limits_{load}\{K_v \cap K_{N_v}\}$
Send switch-request to v and u to use channel k_x
end if

The algorithm consists of two phases: Static phase and on-demand phase. In the static phase, each clusterhead, using a minimal set of the available channels for its cluster, makes a first allocation of channels for its nodes In the second phase, each clusterhead tries to detect the nodes having a high load and tries then to assign them new channels. These new channels would be taken from the still unused channels of K_{C_i}, but could be also borrowed from neighbor clusters having enough free channels. In the following sections, we describe in detail these both phases.

A. Static CCA

At the beginning, each clusterhead assigns a single common channel to all nodes which belong to its cluster (Static CCA, lines 3 to 6). We assume that each node has a distinct number of WNICs and the common channel is allocated to one of its WNICs, thus the nodes can communicate with each other using the same channel. Lines 7 to 13 describe the case where a node is a border node being in the range of neighboring clusters. For those border nodes a common channel is agreed to allow an inter-cluster communication. These steps are repeated for all clusters in the network until a stable configuration is reached.

Based on that initial channel assignment, we assume that all routers have a static connectivity matrix of the network. Based on this matrix all routing within the wireless backbone is done. For the routing, a router creates a set of link-disjoint paths to its destinations. Initially, the shortest path is used. Regardless of the channel switching, the routing is always valid, hence there is no routing overhead incurred in a channel switch.

B. On-demand CCA

In this phase, the clusterhead tries to locally modify the channel assignment to minimize the experienced loss rate and thereby maximize the overall performance. After the initialization of static phase, each router in each cluster periodically estimates the load of all its communication links and records the probability of loss $P(\text{loss on } l^a_{v,u})$ on all links l_v of router v. These information is send to the clusterhead and neighboring nodes. There are two methods to exchange the information of the link status and channel usage. Either the router periodically sends out broadcast messages to its neighbors or it sends this information on demand as soon as one of its neighbors announces a channel switch. The first solution is more reliable and saves the overhead of the data collection prior to a channel switch but it creates a relatively high network overhead. We chose the second possibility since channel switches are not assumed to happen frequently. We assume also that a clusterhead has an overview of the load estimation and channel usage of the neighbor clusters, since it can exchange these informations with its neighbors. If the router v experiences a loss rate $P(\text{loss on } l^a_{v,u}) \geq \sigma$ on channel a, especially on a currently link $l^a_{v,u}$, it proceeds to the on demand phase.

Based on the connectivity matrix, the router calculates all node disjoint paths to its destination. If it discovers an additional unused path using neighbor $x \neq u$, it checks its local assignment table to find the channel b so that $l^b_{v,x} \in L_v$. If $a \neq b$ the router simply activates the newly found path in its routing table and starts using the multiple paths according to the route selection algorithm (see Section V-C). If there are additional paths but none of them uses a different channel, then v send a *CH_REQUEST* message to the clusterhead CH_i belong to it. The *CH_REQUEST* message contains the channels K_v currently used by router v and their loss values. The message also indicates whether an unused WNIC is available on v and the possible next hops N_v to its destination. CH_i checks whether a suitable $u \in N_v$ has an unused WNIC, then it looks for a free channel k_x and compiles a *CH_REPLY* message to inform v and u. In case of no free channel, *CH_BORROW* message is compiled to request a free channel from the neighbors clusterhead (NC_i). After that, k_x is allocated for the link $l_{v,u}$. In the worst case a available channel with minimum load is commanded. Lines 1 to 10 describe the case in which the requester and at least one neighbor have an unused WNIC available.

In case that a previously unused channel is selected and assigned to the unused WNICs. If CH_i experiences an average loss rate $P(\text{loss on } l^a_{v,u}) \geq \sigma$ of all links using the same channel a within a cluster. Therefore, it looks for the active nodes within a cluster, which currently using the same channel a. lines 2 to 9, check whether one of them has an used WNIC. Then, it will add the new channel to the suitable node. Thus, will reduce the highly usage of channel a which reduces also the heavy load occur in the requester node and improve the performance.

If only the requester has a free WNIC (lines 11 to 14), then the selected channel is the one with the lowest loss probability within the neighborhood. The selected channel is assigned to the unused WNIC at the requester. Probably the most common case that no additional WNICs are available is handled in lines 15-18. The set of available paths is constructed as an intersection of the channels K_v and K_{N_v}, meaning the channels the router v and its neighbors use. From this set the channel with the lowest loss probability is selected. The construction of the channel selection algorithm guarantees that existing network links are never invalidated.

After the algorithm has selected a suitable channel and neighbor, it informs the neighbor of the channel to switch using a *CH_SWITCH* message and executes the channel switch. It then awaits a *CH_ACK* message from the neighbor acknowledging the switch.

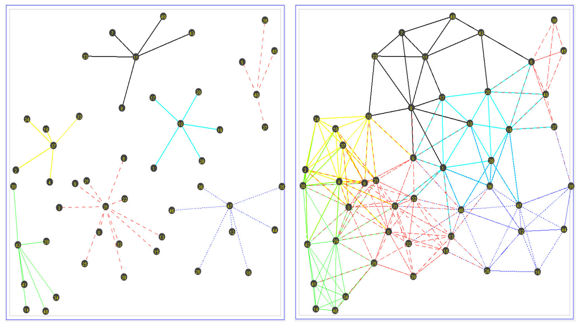
As a last step of the on-demand CCA phase the router updates its routing tables. Since, it now has multiple paths to its destination the router has to perform a route selection algorithm. This is detailed in the following section. It has to be mentioned, that the routing is source routing based on the locally available connectivity matrix no network overhead is generated by route searches or the activation of routes.

C. Route Selection

In our approach we considered two possible mechanisms for the route selection: round robin and single path. Round robin uses each of the multiple paths one after the other where single path uses only one path to the destination. Directly after a channel switch the newly created path is preferred. After a certain damping time $t_{wait} = 10s$ another path might be chosen according to the loss rate of the first links, the path length or any weighted combination of both. The waiting time t_{wait} is introduced to prevent alternating channel assignments and route changes.

VI. PERFORMANCE EVALUATION

We have performed simulations using ns-2 simulator [10] to evaluate the performance of the proposed approach. To illustrate the simulation we refer to Figure 1. The 50 nodes routers are randomly placed in an area of $1000 \times 1000m^2$. The physical and MAC layers of ns-2 are set up to simulate IEEE 802.11a with a maximum bit rate of 24Mbps and



(a) Clustering (b) Static CCA

Fig. 1. Channel Assignment using clustering

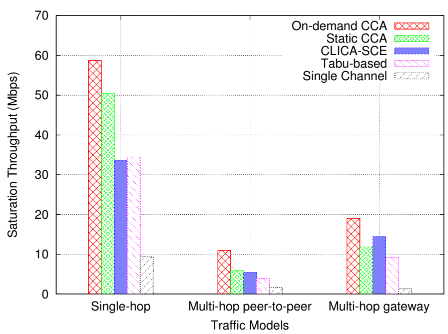
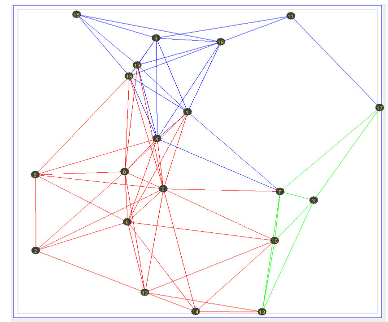
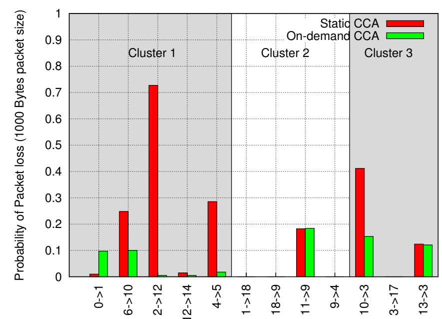


Fig. 2. Saturation throughput vs. various traffic models



(a) Static CCA



(b) Probability of packet loss versus links

Fig. 3. 2D 600 × 600m^2 20-Nodes Random Topology Using CCA, An example 3 clusters and 3 channels

a transmission range of 250m. Each router has two radio interfaces and 12 channels available.

To generate traffic loads, we use Constant Bit Rate (CBR) sources, with a packet size of 1000 bytes. We run the simulation for three different traffic models like in [7]:

- Single-hop traffic model: This model distributes traffic equally in all communication links. It is used to evaluate the performance when all links in the network carry the same load.
- Multi-hop peer-to-peer traffic model: In this model, 25 randomly selected nodes are used as source nodes and the remaining 25 are used as destination nodes. The nodes communicate using multi-hop routes. The routes are computed statically using the shortest path as the metric, and don't change for the lifetime of the simulation.
- Multi-hop gateway traffic model: In this model, 4 random nodes are selected as gateways, and 25 nodes are selected as source nodes. The source nodes send traffic to the nearest gateway. Routes are determined as in previous model. Multi-hop traffic model is a common model when the mesh network is used for Internet gateway connectivity.

To evaluate the performance of our approach compared with other related work like Tabu [7] and CLICA [8] algorithms, we used a metric called *Saturation throughput*. Saturation throughput can be defined as the limit reached by the system throughput as the offered load increases, and it represents the maximum load that the system can carry in stable conditions [11]. We obtain saturation throughput as follows. We run a number of simulations starting with a low traffic. Then, we continue increasing the traffic till the throughput reaches a stable state, this means, the throughout will not change even if we increase the traffic.

Figure 2 shows the saturation throughput for each algorithm and for different traffic models. As it can be noticed from this figure, our algorithm delivers very good results compared to the other algorithms especially On-demand phase.

Additionally, to illustrate the two phases presented in our approach, and to show the performance for both 'Static CCA' and 'On-demand CCA' phases respectively, we run the simulation for a different topology refer to Figure 3(a). The 20 nodes (routers) are randomly placed in an area of $600 \times 600m^2$. The physical and MAC layers of ns-2 are set up to simulate IEEE 802.11b with a maximum bit rate of 11Mbps and a transmission range of 250m. To generate traffic loads, we randomly generate 5 constant bit rate (CBR) sources, with a packet size of 1000 bytes. There are 4 channels available.

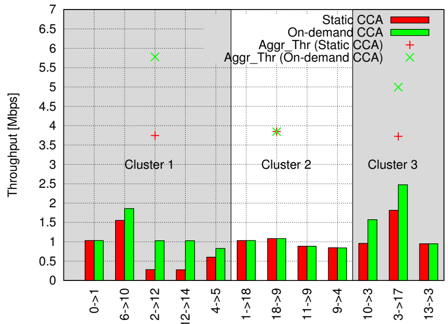


Fig. 4. Throughput distribution of active links within a cluster

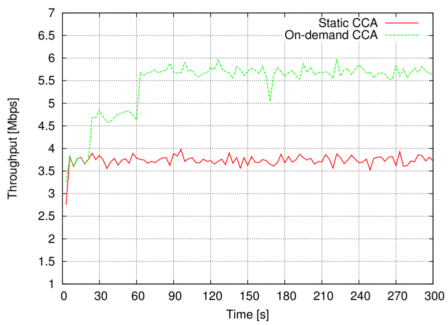


Fig. 5. Aggregate Throughput

The simulation duration is 300 seconds.

In the first step, we run the simulation for the static phase in which each cluster uses a different set of channels and assigns in a first phase the same channel to all links within it. For example, channel 1 for cluster1, channel 2 for cluster 2 and so on. Figure 3(b) shows the packet loss of the links in each cluster. This figure shows the bottleneck on Cluster 1 especially on link $2 \rightarrow 12$. Due to the high usage of channel 1 in cluster 1, the packet loss on this cluster is very high in comparison to other clusters.

The second on-demand phase comes now in application. In this phase, the clusterhead CH_1 of cluster 1 detects the heavy load on the router 2 and a high usage of channel 1 on the whole cluster, therefore it allocates the unused channel 4 to that router. Figure 5 shows at $t = 20s$ the effect of the new allocation of channel 4 to node 2 on the whole aggregate throughput. As effect, the aggregate throughput of all flows get increased of about 30% approximately. At $t = 60s$, the clusterhead CH_3 detects a heavy load on router 10. Since all channels have been already allocated, the clusterhead selects the lowest unused channel, channel 4 in this case, and allocates it to router 10. It is obvious, that after the second allocation, the aggregate throughput is increased again with approximately 25%. The total improvement of the aggregate throughput by deploying the on demand phase is approximately about 55%. The effects of this dynamic channel assignment locally are also clearly showed in Figure 3(b) and Figure 4. The packet loss have nearly disappeared on link $2 \rightarrow 12$ and also reduced on link $10 \rightarrow 3$. In general, the improvement has been not just registered for these links but also for other links within this cluster especially in cluster 1. As an expected consequence of the reduction of packet loss on the links, we can notice the increased throughput of other links too.

Furthermore the evolution of the aggregated throughput of each cluster (Agg-Thr) shows that the on-demand approach has good effects not only on the links for which there have been assigned but also in the overall cluster (See Figure 4).

VII. Conclusions

In this paper, we have presented a novel approach for dynamic channel assignment which is adaptive to the traffic load. We have seen that through this on demand strategy of allocation we can successfully respond to high packet losses occurring due to heavy loads on some nodes. The approach assigns new free channels for those heavy loaded nodes. The registered improvement could not have been realized using only a static assignment strategy, since it can not predict in any way the expected load on the different links of the network. We have also used a clustering strategy in order to simplify and reduce the complexity of dynamic channel assignment into local problems handled within clusters. The simulation results show that by deploying our approach a significant improvement in the aggregated throughput have been achieved.

References

[1] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking (MOBICOM)*, 2004, pp. 216–230.

[2] J. So and N. H. Vaidya, "Multi-channel mac for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proceedings of the ACM int. symposium on Mobile ad hoc networking and computing (MOBIHOC)*, 2004, pp. 222–233.

[3] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "A new multi-channel mac protocol with on-demand channel assignment for multi-hop mobile ad hoc networks," in *Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00)*. Washington, DC, USA: IEEE Computer Society, 2000, p. 232.

[4] M. Shin, S. Lee, and Y. ah Kim, "Distributed channel assignment for multi-radio wireless networks," in *Proceedings of the IEEE Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2006, pp. 417–426.

[5] R.Raniwala and T.Chiueh, "Architechture and algorithms for an IEEE 802.11- based multi-channel wireless mesh network," in *Proceedings of the Conference on Computer Communications (INFOCOM)*, 2005.

[6] B. J. Ko, V. Misra, J. Padhye, and D. Rubenstein, "Distributed channel assignment in multi-radio 802.11 mesh networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[7] A. P. Subramanian, H. Gupta, and S. R. Das, "Minimum Interference Channel Assignment in Multi-Radio Wireless Mesh Networks," in *Fourth Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2007)*, San Diego, California, USA, June 2007.

[8] M.K.Marina and S.R.Das., "A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks," in *Proceedings of IEEE BroadNets*, Boston, 2005, pp. 381–390.

[9] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wirel. Netw.*, vol. 1, no. 3, pp. 255–265, 1995.

[10] "The ns-2 network simulator." [Online]. Available: http://www.isi.edu/nsnam/ns/

[11] G. Bianchi, "Ieee 802.11saturation throughput analysis," in *IEEE COMMUNICATIONS LETTERS*, vol. 2, no. 12, December 1998.