An Efficient Scheme for Solving Bandwidth Guaranteed Scheduling and Shortest path Routing Problem in Wireless Mesh Networks

Ziming Zeng†
School of Information Management
Wuhan University
Wuhan,430072, China

Yuanyuan Zeng††
School of Information & Engineering
Shanghai University of Finance & Economics
Shanghai, 200433, China

Summary
The bandwidth problem is one of the most serious problems for multi-hop wireless mesh networks (WMNs) due to the influence of link interference. In this paper, we study joint problem of interference-aware bandwidth guaranteed scheduling and shortest path routing in IEEE 802.11-based multi-channel wireless mesh networks with dynamic traffic. We present distributed bandwidth guaranteed TDMA scheduling for given flow requests. And we propose a bandwidth guaranteed shortest path routing algorithm based on k-shortest path approach with admission control. The simulation results show our algorithm achieves good performance, and it can effectively provide bandwidth guaranteed path for connection request comparing with minimal hop-count routing algorithm.

Key words:
bandwidth guaranteed, routing, wireless mesh networks

1. Introduction

Wireless mesh networks (WMNs) have emerged as a key technology for being used on the last mile for extending or enhancing Internet connectivity for mobile clients located on the edge of the wired network. [1,2,11]. WMN is dynamically self-organized and self-configured, with the nodes in the networks automatically establishing and maintaining mesh connectivity. WMN consists of mesh routers and mesh clients. Mesh routers in the backbone have minimal mobility and no constraint on power consumption, while mesh client nodes usually desire the support of mobility and power efficiency.

Many applications in WMNs need to support broadband multimedia communication. The bandwidth problem becomes very serious for multi-hop wireless mesh networks due to link interference. Conventional single-channel wireless network architecture cannot adequately support the bandwidth requirements applications. Using multiple channels instead of a single cannel has been shown to be able to improve the network throughput dramatically. In wireless mesh networks, mesh routers can be provided with multiple radios and multiple channels that can greatly alleviate network capacity reduction. With multi-channel, nodes can transmit and receive simultaneously or can transmit on multiple simultaneous transmissions. Using multiple heterogeneous channels offers trade can improve robustness, connectivity and performance. But due to the limited number of channels available, the interference cannot be completely eliminated and in addition algorithms must be designed to mitigate the effects of interference.

Such characteristics imply that scheduling and routing algorithms designed for ad hoc networks may not be appropriate for WMNs. New algorithms should support multi-channel factors, for routing algorithm needs to select a path in-between of different nodes with appropriate channel on the path. And cross-layer design becomes necessary because change of a routing path involves the channel switching in a mesh node. Moreover, different from ad hoc networks, most applications of WMNs are broadband services with various Qos requirements. Thus, performance such as bandwidth must be considered in WMN.

In the paper, we take cross-layer design method to study the joint problem of interference-aware bandwidth guaranteed scheduling and routing. We propose distributed algorithms for the problem. In our scheme, the scheduling uses TDMA to guarantee a schedulable flow. Based on it, we present a routing algorithm by k-shortest path approach and then verifying the paths with admission control to guarantee bandwidth requirement and shortest path in WMN.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related work in the literature. Section 3 is network model and assumption. Section 4 discusses our distributed interference-aware bandwidth guaranteed scheduling. Section 5 is our distributed routing algorithm. Section 6 is simulation results and analysis. Section 7 is the conclusion and future work.

Manuscript received December 5, 2007
Manuscript revised December 20, 2007

2. Related work

Recently, lots of research about interference, interference-aware scheduling and routing for improving network capacity in wireless networks is proposed. Gupta and Kumar in [3] propose that in a wireless network with n identical nodes, the per-node throughput is $\Theta(1/\sqrt{n \log n})$ by assuming random node placement and communication pattern. K. Jain et al in [10] present LP-formulations for max-flow and related problems in a wireless network. They formulate their constraints in terms of arbitrary conflict graphs which can incorporate any interference model. Kumar et al in [11] propose design algorithms for the jointly routing and scheduling to maximize network capacity in WMN. They present distributed algorithms for the problems incorporate fairness, energy and dilation requirements and provide a unified framework for utilizing the network close to maximum throughput capacity. In [12], the authors present a framework for multihop packet scheduling to achieve maximum throughput by considering the influence of intra-flow and inter-flow contention. Alicherry et al in [13] mathematically formulate the joint channel assignment, scheduling and routing problem, taking into account the interference constraints for wireless mesh networks to maximize the bandwidth allocated subject to fairness constraint.

Qos routing in wireless mesh networks is another challenging requirement due to interference among different transmission. Qos routing in single channel MANET has been well studied. Goff et al in [5] propose a Qos-aware routing protocol based on AODV using TDMA scheme. Xue et al in [14] introduce a resource reservation-based routing and signaling algorithm AQOR to provide end-to-end Qos support in terms of both bandwidth and end-to-end delay. In [15], the authors present a shortest widest path routing problem in ad hoc networks. They propose a distributed algorithm to address the problem. Yang et al in [16] compute feasible paths based on the knowledge of scheduling schemes and interference models. Tang et al in [4] present the first algorithm to address Qos provisioning in an IEEE 802.11-based multi-channel wireless mesh networks. It is the most relevant to our work. In the paper, we present a bandwidth guaranteed shortest path scheduling and routing algorithms in WMNs.

3. Model and assumptions

We use similar network architecture as described in [6]. The wireless router is equipped with a traffic aggregation access point that provides connectivity to end user mobile states within its coverage area. The wireless routers form a multi-hop wireless ad hoc network among themselves to relay the traffic to and from mobile stations. Each node in a wireless mesh network is equipped with multiple 802.11 compliant Network Interface cards (NICs). Each NIC is tuned to a channel and that any two NICs at the same node are tuned to different channels. There are totally C non-overlapping frequency channels in the system and each node equip Q NIC where $Q \le C$. The transmission range of each node is r, and the interference range is R (which is typically 2 to 3 times of r).

We model the backbone of an infrastructure of WMN as a directed graph G= (V, E). A channel assignment F assigns each node $v \in$ V a set of $F(v)$ of Q different channel : $F(v) \subseteq \{1,2,\ldots,C\}$. When a channel assignment algorithm is executed, corresponding network topology is constructed as G' (V, E'). It is a multi-graph. In G; there is an edge $e = (u, v; k)$ in G' when node u and v can communicate with each other by channel k.

For direct communication, two nodes need to be within communication range r of each other, and need to have a common channel assigned to their interface, e.g., node u can communicate with node v if and only if $d(u, v) \le r$ and have the common allocated channel $\lambda \in F(u) \cap F(v)$. We use $N(u)$ donate the set of communication neighbor and corresponding channel information, e.g., $N(u)=\{(v;1),(w,2)\}$, it means u can communicate v with channel 1 and communicate w with channel 2. A pair of nodes that use the same channel and are within interference range R may interfere with each other's communication, even if they cannot directly communicate. A transmission between u and v may block all transmissions within R away from either u or v. Then we use $I(u)$ donate the set of possible interference nodes within interference range of u. Then we define the link potentially interference model as: transmission pair of (u, v) *and* (x, y) potentially interfere with each other, if the distance of one of the four pairs of edges nodes $d(u, x)$, $d(u, y)$, $d(v, x)$, $d(v, y)$, is at most R apart. Let $I_p(e_1)$ denote the set of edges which potentially interfere with edge e_1, i.e., e_1 may not transmit successfully whenever an edge $e_2 \in I_p(e_1)$ is transmitting. The potential interference links can simultaneously transmit/receive data on a different channel. We define the co-channel interference model as: edges e_1 (u, v), e_2 $(x, y) \in$ E interference with each other, when they are potentially interference with each other and they use the same communication channel, i.e., communication with channel $\lambda \in F(u) \cap F(v) \cap F(x) \cap F(y)$. This definition of link interference also includes the cases where the two links share a common node and the case where e_1 and e_2 are identical. Simultaneous transmission along interference links will lead to collision. Let $I(e_1)$ denote the set of edges which interfere with e_1, i.e., e_1 and $e_2 \in$ E cannot transmit at the same time.

For the bandwidth guaranteed scheduling and routing problem discussed in this paper, we assume that the system operates synchronously in a TDMA time slotted mode. According to TDMA, bandwidth can be allocated to the network links using a schedule of interference-free slots. During every slot, several links are activated for transmission such that no conflicts occur. It enables a simple end-to-end set up of periodically allocated time slots over a multi-hopping path and that provides maintenance means to inform oblivious nodes of a reserved transmission with a bandwidth requirement. Our goal is to make a Qos connection request with bandwidth requirement for serving mesh clients, due to interference among different transmissions. Given the network and existing traffic, a call setup request (s, d, B_w), we try to find a shortest path from source s to destination d with available bandwidth B_w. And the interference links cannot use the same timeslot. If such a feasible flow allocation can be found, the connection request is admitted, and corresponding path routing is established. Otherwise, the connection request will be blocked.

4. Link scheduling: feasibile condition and Algorithm

4.1 Problem Formulation

In this section we will focus on bandwidth guaranteed feasible periodic schedules. We consider each edge in G'= (V, E') has a capacity $c(e(i))$ bits/sec for channel i and denotes the maximum data that can be carried on e in a second. A network flow that associates with each edge $e=(u, v; i)$ values $f(e(i))$, $1\leq i\leq$ C where $f(e(i))$ is the rate at which traffic is transmitted by node u for node v on channel i. The links can be scheduled to transmit in the same time slot only if they do not interfere. The set of edge channel pair $(u, v; i)$ (edge between u and v using channel i) will be scheduled at time slot t, t=1, 2….T where T is the number (integer) of time slots in a schedule cycle. For easy of exposition, we assume that each time slot is a unit second. An interference free feasible schedule is defined as: if the edges of no two edge pairs $e_1\,(u, v; i)$, $e_2\,(x, y; i)$ scheduled in the same time slot for a common channel i interfere with each other, then it is a feasible schedule. Our bandwidth guaranteed feasible scheduling (BGFS) should satisfy that: 1) The scheduling is interference free, i.e., no interference links transmit at the same time slot. 2) The scheduled flow on $e=(u, v; i)$ is larger than required bandwidth, i.e., $f(e(i)\geq B_w$.

4.2 Feasible Link Flow Scheduling: Necessary and Sufficient Conditions

Next we analysis the necessary and sufficient condition for bandwidth guaranteed scheduling (BGFS). The system uses TDMA scheduling, which demands each slot the links scheduled for transmission do not interfere with each other and flow is satisfied with bandwidth requirement. Previous work in [7] presents the necessary and sufficient conditions for an interference free feasible schedule. We add bandwidth constraint into it to solve our BGFS schedule.

Recall that the interference is incurred within interference range R, which is often 2 to 3 times of transmission range r. We assume that R is q times of r, which is a fixed value. An edge $e' \in I(e)$ is the interference link of e.

LEMMA 1. (A necessary condition) Any valid bandwidth guaranteed and interference free link flow schedule on communication channel i, must satisfy the link flow constraint:

$$\frac{f(e(i))}{c(e(i))}+\sum_{e'\in I(e)}\frac{f(e'(i))}{c(e'(i))}\leq c(q)\quad AND\; f(e(i))\geq B_w \qquad (1)$$

Where $c(q)$ is a constant that depends only on q. For example $c(q)$= 4,8,12 for q=1,2,2.5 respectively [7].

Lemma1 is the necessary condition for our BGFS scheduling. The proof of lemma 1 is given out by feasible schedule problem in [7], so we omit it.

LEMMA 2. (A sufficient condition) If the link flows on communication channel i, satisfy the following link scheduling constraint, then a valid bandwidth guaranteed and interference free edge schedule can be found.

$$\frac{f(e(i))}{c(e(i))}+\sum_{e'\in I(e)}\frac{f(e'(i))}{c(e'(i))}\leq 1\quad AND\quad f(e(i))\geq B_w \qquad (2)$$

Lemma 2 is the sufficient condition for our BGFS scheduling. The proof is very similar in [7], since we only add the bandwidth requirement to it. Note that we have $f(e(i)=a(e(i)\,c(e(i))$. $a(e(i)$ is the active fraction of time for edge $(u, v ;i) \in$ E'. The schedule of each edge $e\,(u, v; i)$ should allocate a certain number of slots donated as $S(e(i))$ with:

$$S(e(i)) = T\alpha c(e(i)) \qquad (3)$$

From lemma2, we can get the time slots relationship constraints of edge e and its interference edge $e' \in I(e)$, for achieving a BGFS schedule:

$$S(e(i)) + S(e'(i)) \leq T \quad AND \quad S(e(i)) \geq \frac{B_w}{c(e(i))} T \quad (4)$$

4.3 Distributed Link Flow Scheduling Algorithm

Here, we present a distributed scheduling scheme. Our distributed algorithm is based on [8]. The distributed algorithm is a variant of the distributed edge coloring problem. In the algorithm, each edge is allocated with the number of slots according to the bandwidth requirement. The pseudo-code is presented in algorithm 1.

Each node maintains several sets: communication neighbor set (containing corresponding communication channel) $N(u)$, possible interference neighbor set $I(u)$ and neighbors' interference neighbor set $I(N(u))$. $N(u)$ is achieved by HELLO message exchange with nodes within communication range r for neighbor discovery. To get possible interference neighbors, the sender sends out TEST message within interference range R. The neighbor's interference neighbors can be obtained by HELLO message too. When a node is elected as a candidate for time slot allocation, it broadcasts a CANDIDATE message to its possible interference neighbors. After a node allocates the time slots for its adjacent link, the corresponding slot information is recorded. From formula (4), the time slots allocation of a link e is restricted by its interference link e'. Node u can send out an Ad-Req message to its possible interference neighbor in $I(u)$ to collect the time slots for its interference link.

Among the above process, some messages need to research the interference range of R. There're two ways to get the information of possible interference neighbors. 1) We use a higher transmission power to reach the interference range R, and collect the information of possible interference neighbors. However, this method consumes much more power and it may bring more interference. 2) We use hop relay to disseminate messages to possible interference neighbors. We make $t=R/r$. So our messages can be relay to t+1 hop to collect the information of possible interference neighbors. Using this method to gather the interference neighbors' information may be imprecise. But this kind of inaccuracy is tolerable in use. In the paper, we use the second method to collect local inference information to prevent incur more contention.

In algorithm 1, step 1 is used to collect local interference information for each node. Step 2 is initialization of time slots for local edges. The edges adjacent nodes in $I(u)$ and $I(N(u))$ are potentially interference with edges adjacent to node u. From step 3 to 9 is the periodic schedule. Potentially interference links are avoided to be allocated at the same time. So each node has a probability to be a candidate for time slot allocation. The number of allocated time slots is determined by the flow and bandwidth requirement and should guarantee our BGFS schedulable. Step 5, 6 is according to formula (4), which is a variant of sufficient condition for BGFS. If such allocation can be found, it is admitted. Otherwise the transmission is deferred.

Algorithm1 Distributed BGFS schedule

1: Each node u obtains local transmission neighbor in $N(u)$ and interference neighbor information $I(u)$ by sending out original power HELLO and TEST messages. And obtain neighbor's interference neighbor set $I(N(u))$.
2: **for** all edges adjacent to node u **do**
    $S(e)= \phi$
3: **for** each period j **do**
4: Each node u is elected to become a candidate with probability 1/d, where d=$|I(u)|+|I(N(u))|$.
5: If u elects to be a candidate, it sends out CANDIDATE message to its possible interference neighbors, and set a *timeout* delay. During the *timeout*, if u receive CANDIDATE message from other node, it will not participate in this cycle.
6: If node u participates in the cycle and chooses slots for edge $(u, w; k)$ or $(v, u; k)$. Node v and w sends out an Ad-Req message to its possible interference neighbors to collect the number of allocated time slots of its interference links: $S(e'(k))$.
7: Node u chooses slots greedily for the edge adjacent to node u as e $(u, v; k)$ or $(v, u; k)$, if $S(e(k))= \phi$. $S(e(k))$ needed to be allocated in $T\backslash S(e'(k))$ $(e' \in I(e))$, and $S(e(k))=B_wT/c(e(k))$.
8: If it can be allocated, the schedule is admitted. Otherwise not admitted and defer the transmission.
9: **end for**

Theorem 1: Algorithm 1 produces a bandwidth guaranteed and interference-free schedule.

Proof: Assume e_1 *and* e_2 (with common communication channel k) interfere with each other and the time slot allocation of e_1 is processed before e_2. Since e_1 *and* e_2 interfere with each other, so $e_1 \in \mathrm{I}(e_2)$, and hence $S(e_1(k)) \subseteq \bigcup_{e \in \mathrm{I}(e_2)} S(e(k))=S(e_2'(k))$, $e_2'(k) \in \mathrm{I}(e_2(k))$.So $S(e_2(k)) \subseteq T \setminus S(e_2'(k))$, i.e., for any two interference edges e_1 *and* e_2 (with common communication channel k) in the network graph, $S(e_1(k)) \cap S(e_2(k))=\Phi$. For each link, we allocate time slots as $S(e(k))=B_wT/c(e(k))$. So the bandwidth on link e is $S(e(k))/T*c(e(k))= B_w$.

5. Routing

5.1 Problem Formulation

The joint problem of bandwidth guaranteed scheduling and shortest path routing (BGSR) is to map a flow request (s, d, B_w) to a flow vector f by computing shortest feasible path p, where s, $d \in V$ are the source and destination nodes, and B_w is the bandwidth requirement. The flow f on each link and end-to-end bandwidth of the path should be B_w, and f on other links of network graph will be equal to zero. Even the interference free link scheduling sub-problem given the link flows is NP-hard. And BGSR is also referred to as the "integral flow with bundles" problem which is NP-complete [17]. We will present an approximation algorithm for the overall joint BGSR problem.

BGSR violates the principle of optimality, which means optimum in a myopic sense may not the optimal strategy for long-term performance. The shortest hop-count path may not satisfy bandwidth requirement. The k-shortest path is widely applied in multi-constrain Qos routing to address violation of the principle of optimality. So we take k-shortest path approach to computer several candidate paths and choose the best (shortest) among them. We try to find k-shortest paths from s to d and then verify those paths if they can satisfy bandwidth B_w. Note k is a number given by user. The solutions of BGSR answer the question that whether could find a shortest path which satisfy bandwidth requirement delivered between the given source-destination pair and thus guarantee to admit BGFS connection requests. Each node u maintains a set of k-shortest paths toward the source. Each item in the set forms a record, denoted by $r(u)=(s, p(u), len(p(u)), bw(p(u)))$ that contains the source node s, the current path from the s to u, and the path length in terms of hop-count and the path bandwidth. The k items of record donated as $r_1(u)$, $r_2(u)$ … $r_k(u)$ are sorted in order with high priority to shorter path, i.e., if $r_1(u) > r_2(u)$, it denotes $len_1(p(u)) < len_2(p(u))$ or $len_1(p(u)) = len_2(p(u))$ & $bw_1(p(u)) > bw_2(p(u))$. Each item in the record of each node except s is initialized as $r_j(u) = (s, \Phi, \infty, 0)$, $1 \le j \le k$, while Source is initialized as $r_j(u) = (s, s, 0, 0)$, $1 \le j \le k$.

Our algorithm should perform routing with interference-aware admission control to guarantee bandwidth requirement, which is carried out distributedly at each hop in the path. An incoming connection request is admitted if the additional induced demand flow allocation on the network links is realizable by a TDMA BGFS schedule. The routing process contains route discovery, route reply and route maintenance. In the phases, we take partial admission control and full admission control in route discovery and reply respectively. In routing discovery phase, the total route is still unknown at that time, so it's hard to determine all the contentions of the path. Partial admission is used to preliminarily eliminate routes without enough bandwidth. Since we only get partial route, the partial admission is undetermined. Full admission control makes a soft reservation of bandwidth with the full route from source to destination. In the routing process, each node on the path checks if the flow is admissible by its interference constraints. So it needs to know the total flow at each link only in its interference neighborhood. Route maintenance is for recovery of link failure.

5.2 Route Discovery and partial Admission Control

The aim of route discovery is to find a route between the sender and the receiver that has enough resources for the flow. We use an on-demand route discovery with source routing, similar to DSR [9]. The source routing-based approach allow us to specify directly which route that has been admitted by the admission control and has enough bandwidth for the flow. It can also provides easy traffic splitting at the source node so that two flows with the same destination can follow different routes to avoid creating hot spots in the network.

Firstly, source node s invokes the routing discovery process by broadcasts routing request messages Rt-Req. A Rt-Req message contains the required bandwidth of B_w, and an item in the record of current k-shortest partial routes from source. The partial route is a record of the sequence of hops and link communication channels. Each node performs partial admission control during the route discovery process and preliminarily eliminates routes without enough bandwidth when it receives a Rt-Req message. A node u receive a route request message from its neighbor v by channel i will perform partial admission control by extending path $p(v)$ to node u. The admission control is based on partial route information of $p(v)$. A flow allocation on link $e(v, u; i)$ must satisfy sufficient condition in formula (2), i.e., we can use algorithm 1 to allocate flow for link e according to interference in partial route. If the flow can be scheduled, a new item can be formed as $r'(u) = (s, p'(u), len(p'(u)), bw(p(u)))$ in node u. $p'(u)$ records the new partial route as $p'(u) = (p(v), u; i)$, i.e., by extending $p(v)$ to link $(v, u; i)$. $len(p'(u) = len(p_j(u)) + 1$, among which $len(p_j(u))$ is the length of the j^{th} path in k-shortest path record, and the j^{th} path is carried in the Rt-Req message. The formed new partial path end-to-end bandwidth can be calculated by the minimal value of bandwidth of path $p(v)$ and bandwidth of link $(v, u; i)$. The calculation of new path's bandwidth is presented as formula (5):

$$bw(p'(u)) = \min(bw(p(v)), \min c(e(i))(1 - \sum_{e' \in I(e)} \frac{f(e'(i))}{c(e(i)})) \quad (5)$$

The computed new item (a new partial path) $r'(u)$ will be compared with the existing k^{th} shortest path record maintained in the receiver node u. Since the k items of shortest paths are sorted in order, we compare the new item with the smallest one in the record, i.e., compare $r'(u)$ with $r_k(u)$. If $r'(u) < r_k(u)$, it means the new item $r'(u)$ will be a selected as a k-shortest path item of node u. Then we make $r_k(u) = r'(u)$. And then sort current k items in the shortest length propriety order. In this way, we always record the k items of shortest paths for each node.

For each item in k-shortest path record in each node u, if the flow can be schedulable and $bw(p(u) \geq B_w$, the admission control succeeds and the route request Rt-Req message can be forwarded out. Otherwise the admission control fails and the Rt-Req message is dropped.

5.3 Route Reply and Full Admission Control

When the intended destination node receives a Rt-Req message, the item of partial path route carried in the request message becomes a full route. The destination then reverses the k shortest full routes and sends a route reply message Rt-Reply back to the source along the route. When the k shortest path routes arrive at the destination, and the destination will sends the Rt-Reply message along the shortest paths (i.e., path in $r_1(d)$) to verify it by full admission control. If the route can not satisfy end-to-end bandwidth requirement, other items of route are cached in the k-shortest path record are verified, until we find a shortest bandwidth guaranteed route or none of the k-shortest paths can meet requirement.

The route reply with full admission control process is invoked by destination d to send out a Rt-Reply message. A node v receives the Rt-Reply from u by channel i will perform full admission control. Since the full route is know, the link interference can be calculated exactly within node's interference range R. Node u should allocate flow on a link $e\ (v, u; i)$ by algorithm 1 according to full route. If admission control succeeds at a node, a soft reservation of bandwidth on the link is setup and a Rt-Reply is forwarded to the last hop node (in reverse direction of routing). Otherwise, if cannot accommodate the flow on e, it sends an admission rejection message Ad-Rej back its routing next hop node u. Then node u will make this j^{th} route path unusable as $r_j(u) = (s, \Phi, \infty, 0)$, $1 \leq j \leq k$. And u sends out Rt-Reply message according to path routes in $r_{j+1}(u)$ reversely. If there are no cached routes any more, node u will forward an Ad-Rej message toward its next hop. If d receives an Ad-Rej message and has no cached routes any more, it implies that none of the k-shortest path could satisfy bandwidth requirement. So the connection request is blocked. Otherwise, when s receives a Rt-Reply message successfully, enough end-to-end bandwidth has been reserved for the flow and communication can start.

5.4 Route Maintenance

When there exists a link breakage, the re-routing process is invoked. If a node does not receive periodic HELLO packets from its downstream node for a pre-defined interval, it marks the route as invalid. Upon discovering the violation, the node sends out a corresponding Rt-Error message to its upstream node in this failed route. The upstream node receives the Rt-Error will re-initiates a routing reply procedure by sending out Rt-Reply message by another alternative k-shortest path maintained by this node. Then source will select the best path by doing the same admission control operation in Section C again. In the process, the Rt-Reply message is forwarded back to the source by choosing another proper candidate path cached in k-shortest path record. Not all failures can be handled locally, if a failed node cannot find any usable candidate paths in its local upstream neighbor, and then a Rt-Error message is forwarded upstream recursively until destination. If such backup route can be found, then this new route will be used for data transmission. Otherwise, the connection request is blocked.

Theorem 2: The BGSR algorithm is a $c(q)$ approximation algorithm for joint bandwidth guaranteed routing problem with interference free link scheduling problem, where $c(q)$ is a constant defined in Section IV.

Proof: Note that given a link flow on a path calculated by BGSR, it must satisfy the necessary condition in formula (1). Otherwise it is not schedulable. Thus we scaling this given link flow by a factor of $c(q)$, i.e., $f'(e(i)) = c(q)\ f(e(i))$. Then the scaling flow $f'(e(i))$ will also satisfy the link schedule sufficient condition in formula (2). So our BGSR algorithm is able to find interference free and bandwidth guaranteed link schedule and routing. The scaling link flow $f'(e(i))$ is at least 1/ $c(q)$ fraction of the optimum link flow is routed. So the approximation bound is $c(q)$.

6. Simulations

In this section, we evaluate the performance of our joint bandwidth guaranteed scheduling and routing algorithms via simulations. We consider static wireless mesh networks with n nodes randomly located in a 900X900m^2 region. Each node has a fixed transmission range of 250m and interference range of 500m. We inject 1000 connection requests in each simulation run, and each connection request is generated with a randomly chosen source-destination pair and a random bandwidth requirement B_w which is no more than a given maximal bandwidth requirement B_{max}. Each connection maintains a random number between 1 and 200 time unit. In the simulation scenarios, a connected topology construction of the network is formed by channel assignment algorithm in [4] before our joint algorithm implemented.

We use connection request blocking ratio to make performance evaluation. The blocking ratio is the ratio between the number of blocked connections and the total number of connection request. We show the performance of our joint bandwidth guaranteed scheduling and routing algorithms (BGSR) comparing with the minimum hop-cont (shortest) path routing algorithm (MSP) under different system parameters of network size(n), available non-overlapping channels(C), the numbers of NICs(Q) and the channel capacity (c). According to IEEE 802.11 specifications, 802.11a has 12 non-overlapping channels and 11 Mbps channel capacity. IEEE 802.11a has 3 non-overlapping channels and 54 Mbps channel capacity. Our comparison is under different above five parameters which can influence the performance. Notice that we set k=1 and k=4 for the bandwidth guaranteed k-shortest path in our BGSR scheme respectively.

Fig.1 to Fig. 5 shows the simulation results of the blocking ratio with connection request of our BGSR scheme (k=1, k=4) and MSP scheme under various system parameters. Our BGSR scheme always outperforms MSP scheme, for we consider bandwidth requirement on route discovery, and eliminate impropriate path at first. BGSR scheme with k=4 outperforms the situation of k=1, because k-shortest path routing consider the violation of principle of optimum, and maintains k-shortest path towards the source in the routing discovery. Then we verify these k paths, and choose the shortest one for our communication. Evidently, 4-shortest path has more chances for selecting bandwidth required routing path comparing with 1-shortet path, and it reduces the probability for routing path to be blocked. Furthermore, we can see that the blocking ratio increases when the maximal bandwidth requirement increased, because more bandwidth needs to be allocated for each connection, it brings the probability for the path to be blocked. Comparing fig.1 with fig.2, and fig.3 with fig.4, we can see that the blocking ratio is increased when network size is relatively large. Because the bigger network size incur more interference and makes the available resources decreased. Comparing fig.1 with fig.3, and fig.2 with fig.4, we can see that the blocking ratio is reduced when network uses more channels and has more capacity. Because the more channels network has make the nodes have more probability to be assigned with different channels so that interference is cut down and it brings the admitting ratio increase of connection requirement. Comparing fig. 4 and fig.5, it shows that the increase of network NIC will cut down the blocking ratio. Each NIC can be tuned to a communication channel, so more NICs on a node can increase the probability for nodes to communicate on different channel, so more connection request is admitted.

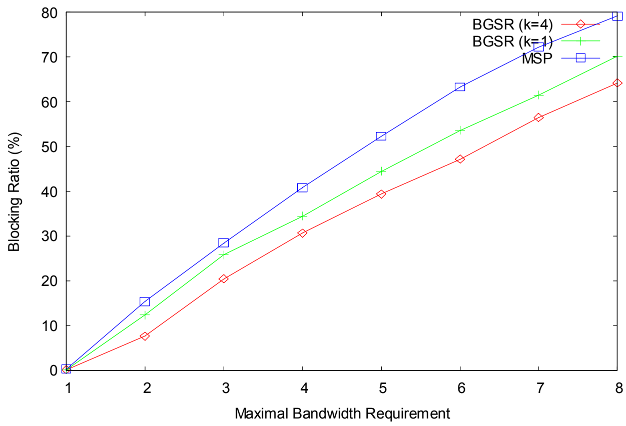

Fig.1. Blocking ratio of BGSR and MSP when n=25, C=3, Q=2, c=11

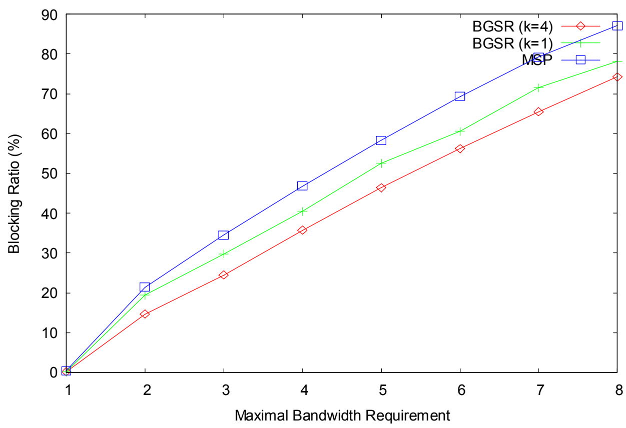

Fig.2. Blocking ratio of BGSR and MSP when n=40, C=3, Q=2, c=11

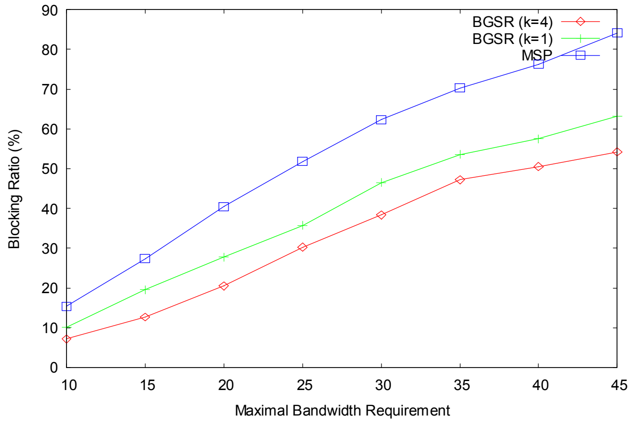

Fig.3. Blocking ratio of BGSR and MSP when n=25, C=12, Q=2, c=54

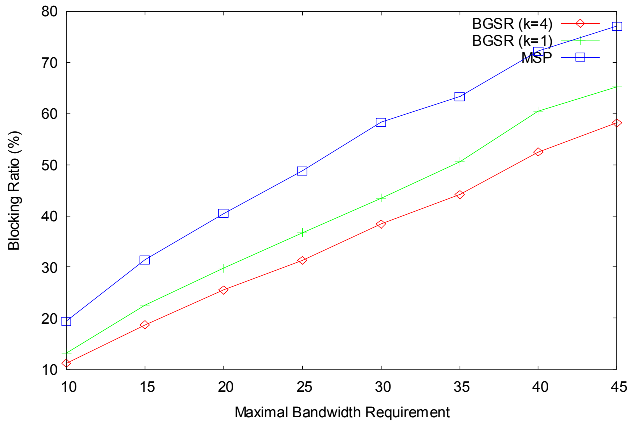

Fig.4. Blocking ratio of BGSR and MSP when n=40, C=12, Q=2, c=54

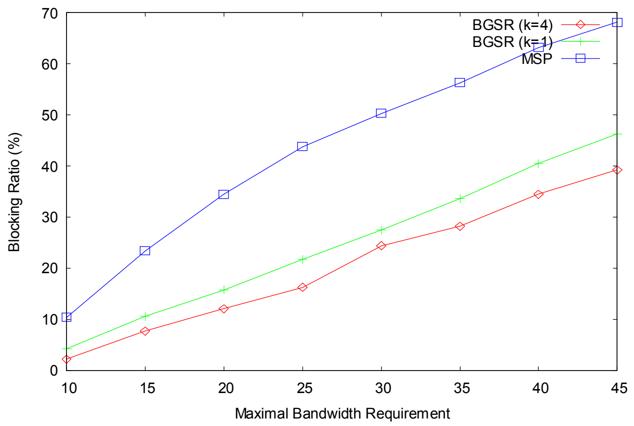

Fig.5. Blocking ratio of BGSR and MSP when n=40, C=12, Q=3, c=54

7. Conclusion

Wireless mesh networks are a promising technology for next generation wireless networking. Many applications scenarios are stimulating its rapid development. However, due to the influence of interference, the large number of users and the emergence of real-time multimedia applications, bandwidth guarantee becomes one of the important requirements of application in multi-hop wireless mesh networks due to interference between links. In this paper, we study on the joint scheme of interference-aware bandwidth guaranteed scheduling and routing in WMN. Our goal is to provide bandwidth guarantee shortest path for data communication. The problem is complicated for co-channel link interference share the medium. We define the con-channel interference to capture the influence of interference. Then we propose distributed algorithms to solve the problem. The scheduling algorithm uses TDMA and provides the feasibility condition of flows with bandwidth requirements. The routing algorithm is based on k-shortest path routing and verifying shortest paths considering interference constraint and flow admission control. The simulation evaluates the performance of our algorithms over various system parameters. The results show that our algorithm performs much better than minimum hop-count path routing and our algorithm satisfy bandwidth requirements with lower connection request blocking ratio.

For future work, we would like to work on multicast routing for bandwidth guaranteed scheduling and routing problems and study distributed algorithm on it.

References

[1] I.F. Akyildiz, X. Wang, W. Wang, "Wireless mesh networks: a survey," *Elsevier Journal of Computer Networks*, 2005.

[2] Mesh Networks Inc. Mesh networs technology overview. http://www.meshnetworks.com.

[3] P. Gupta, P.R. Kumar, "The capacity of wireless networks," *IEEE Transaction on Information Theory,* vol. 46, 2000, pp.388-404.

[4] J. Tang, G. L. Xue and W.Y. Zhang, "Interference-aware topology control and Qos routuing in multi-channel wireless mesh networks," *In Proc. MOBICOM*, 2005.

[5] T.Goff, N. B. Abu-Ghazaleh, D. S. Phatak, and R. Kahvecioglu, "Preemptive maintenance routing in ad hoc networks," *In Proc. of MOBICOM,* 2001.

[6] A. Raniwala, K. Gopalan, T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM Mobile Computing and Communications Review (MC2R)*, vol. 8, 2004, pp. 50-65.

[7] M. Alicherry, R. Bhatia, L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," *In Proc. MOBICOM*, 2005.
[8] V.S. A Kumar, M.V. Marathe, S. Parthasarathy, A. Srinivasan, "Algorithmic aspects of capacity in wireless networks," *In Proc. SIGMETRICS*, 2005.
[9] D.B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *In Proc. Mobile Computing,* vol. 353, 1996.
[10] J. Jain, J. Padhye, V. Padmanabhan and L.Qiu, "Impact on interference on multihop wireless network performance," *In Proc. of ACM MOBICOM*, 2003.
[11] V.X. A. Raniwala and T. –C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh networks," *In Proc. of IEEE INFOCOM*, 2005.
[12] H. Zhai, J. Wang, Y.Fang, " Distributed packet scheduling for multihop flows in ad hoc networks," *In Proc. of IEEE WCNC* , 2004.
[13] M. Alicherry, R. Bhatia, L.Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," *In Proc. of MOBICOM*, 2005.
[14] Q. Xue, A. Ganz, "Ad hoc Qos on-demand routing (AQOR) in mobile ad hoc networks, " Journal of Parallel Distributed Computing, vol. 63, 2003, pp. 154-165.
[15] Z.F. Jia, B. Gupta, J. Walrand and P. Varaiya, "Bandwidth guaranteed routing for ad-hoc networks with interference consideration," In Proc. of IEEE Symposium on Computers and Communications (ISCC), 2005.
[16] Y. Yang and R. Kravets, "Contention-aware admission control for ad hoc networks," Technical report, UIUC, 2003.
[17] M.R. Garey, D.S. Johnson, "Computers and Intractability –A Guide to the Theory of NP-Completeness", *Freeman*, California, USA, 1979.

Ziming Zeng is now an assistant professor at school of Information Management in Wuhan University, China. He received his PH.D. degree in Computer Science from Wuhan University in 2006. His current research interests include E-commerce, Wireless Networks and Information Management.

Yuanyuan Zeng is now an assistant professor at school of Information and Engineering in Shanghai University of Finance & Economics, China. She received her PH.D. degree in Computer Science from Wuhan University in 2007. Her current research interests include Wireless networks and Intelligent Computing.