A Novel Distributed Scheduling Algorithm for Wireless Mesh Networks*

Yun Hou and Kin K. Leung

Imperial College
London SW7 2BT, United Kingdom
{yun.hou, kin.leung}@imperial.ac.uk

***Abstract*— Wireless multi-hop, mesh networks are being considered as a candidate to backhaul data traffic from access networks to the wired Internet. These mesh networks are referred to as wireless backhaul networks. Existing medium access control (MAC) protocols and scheduling algorithms are devised for wireless access. So although they have been adopted for the wireless backhaul networks, they do not yield good performance. In this paper, we propose a novel distributed scheduling algorithm, composed of a framework and a new utility function definition, for wireless backhaul networks. We show by analysis and simulation that the algorithm converges to the desired throughput allocation in a long run, which can be specified by the routing protocol in use to guarantee quality of service. Moreover, in terms of interference, we show that our framework maintains strong temporal correlation of interference, which is required to ensure proper channel predictions for scheduling gain and for distributed power control. Finally, simulation results reveal that the new algorithm takes advantage of the multi-user diversity in achieving high overall network throughput, when compared with the tree-structure algorithm.**

I. INTRODUCTION

Backhaul networks transfer data between access points to gateway nodes, which in turn are connected to the wired Internet. Traditionally, backhaul networks are based on wired technologies such as ADSL, T1 and optical fibre. For low cost and ease of deployment, wireless networks are being considered to provide the backhaul capability. Due to proliferation of access points as well as reduction of cell size to meet growing traffic demands in cellular networks, there is a strong need to develop new wireless backhaul technologies. Notably, IEEE 802.16 specifications are devised to meet such need and indeed one of the expected applications of the 802.16 standards is to backhaul traffic to and from 802.11 access points.

We consider here wireless mesh networks for the backhaul application where data can be forwarded through multiple hops before reaching the desired gateway node. As shown in Figure 1 for example, backhaul traffic is collected from a number of sources (e.g., access points, BS's). Therefore, a key design challenge for the wireless backhaul networks is to provide very high network throughput while meeting the quality-of-service (QoS) requirements for various data flows from the entry to the exit nodes of the backhaul networks. Approaches to overcoming this challenge include use of intelligent antennas to improve data rate at the physical layer and advanced network protocols (e.g., scheduling and routing) to realize the throughput gain and ensure QoS at the network layer.

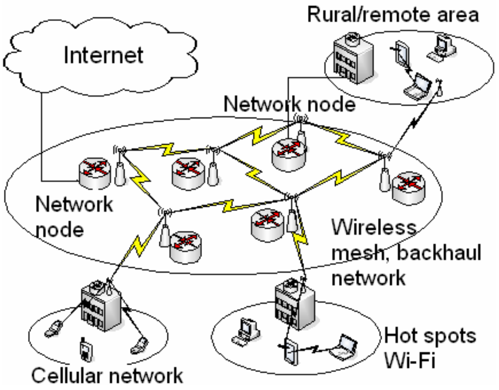

Figure 1 An example for wireless backhaul networks

In this paper, we focus on the scheduling issue in wireless backhaul networks. Scheduling for wireless mesh networks has drawn a lot of research attention recently. However, existing scheduling algorithms proposed for wireless mesh networks do not fit the backhaul features very well and we briefly examine the reasons as follows.

Centralised scheduling algorithms [1-3] are based on graph theory. These methods assume that there is a central controller having full knowledge about all links in the network. The method finds the optimal set of non-overlapping links, i.e., a perfect matching with the highest total throughput of the graph. However, it has been proved [4;5] that to find such an optimal link set in the graph is NP-complete. As a result, centralized methods are not feasible in large-scale backhaul networks.

The distributed coordination function (DCF) with the request-to-send (RTS) and clear-to-send (CTS) mechanism proposed in IEEE 802.11 ad-hoc mode [6] is commonly used in multi-hop wireless ad-hoc networks. Since it is basically selecting the active transmission in ad-hoc networks, it has been used as a candidate scheduling algorithm for wireless

---

*This research is financially supported by the EU IST FP6 MEMBRANE project (contract number 027310).

multi-hop ad-hoc networks. In addition, the election-based scheduling algorithm specified in the IEEE 802.16 standard [7] is another scheduling scheme for the wireless mesh networks. Both 802.11 and 802.16 scheduling algorithms are fully distributed and collision free. However, due to the completely random link selection, neither of the algorithms takes advantage of multi-user diversity in the wireless environments.

Another scheduling technique for multi-hop mesh networks has been proposed in [8]. The method is referred to as the tree-structure (TREE) scheduling technique that maps the backhaul network into a tree such that the gateway node to the Internet corresponds to the root of the tree, as illustrated in Figure 2. Note that nodes in the figure are the wireless routers of the wireless backhaul network, although they are assumed to be located according to the typical hexagonal layout for illustration purposes. Nodes at the same tree level have identical "duplexing status". Specifically, at each time slot of a time-slotted system, every node at the same tree level chooses one of its parent or children nodes to transmit data to. In this way, the TREE method is distributed and partially opportunistic. The main shortcoming is that the tree-mapping only considers part of the network links as scheduling candidates (denoted by solid lines in Figure 2), and misses some "horizontal links" (represented by dashed lines) between nodes closely located but mapped into different branches.

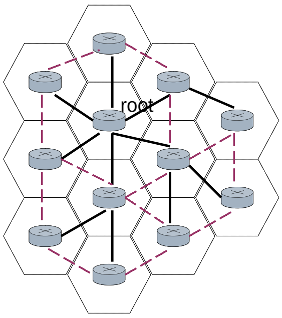

Figure 2 The network layout and its tree mapping.

To ensure QoS, the routing algorithm in use has to reserve and allocate radio resources for various competing data flows during selecting appropriate routes for them. Such resource allocation must be enforced by the scheduling algorithm. Otherwise, the QoS promised by the routing protocol to its applications cannot be guaranteed. It is worth noting that none of these existing scheduling algorithms explicitly consider the resource allocation to cooperate with the routing decision. In this paper we propose a novel distributed opportunistic scheduling algorithm for backhaul networks. We show that our new algorithm is distributed, opportunistic and enforcing throughput allocation in the long run. Furthermore, our proposed scheduling algorithm maintains strong temporal correlation for interference, without which channel quality and interference cannot be tracked and predicted with reasonable accuracy, as required by any opportunistic scheduling [9] and power control algorithms [10;11] for proper operations.

The remainder of the paper is organized as follows. In section II, the framework of the new scheduling algorithm is introduced. Section III derives the new utility definition and proves its convergence. Section IV presents our simulation results to portray the properties of the proposed algorithm, and section IV concludes the paper.

II. FRAMEWORK OF THE NEW ALGORITHM

The network nodes of the backhaul network studied are wireless routers (WR's) in hexagonal layout as shown in Figure 2. Note that although we adopt the hexagonal layout of nodes to illustrate our ideas here, our algorithm is applicable to general network topologies and layouts. From a point view of graph theory, the hexagonal layout is a regular graph of degree $r = 6$. In our assumption, each node can only transmit/receive to/from its one-hop neighbouring nodes, whereas it can be interfered by all the other concurrently active transmissions in the network. For the sake of discussion convenience, we call the "one-hop neighbour" as "neighbour" from now on. Besides, the time-division duplex (TDD) system where each link can only transmit or receive signal but not both at any given time is considered here.

In this paper we assume that each node schedules one of the links associated with it at a time. Then the objective of our scheduling algorithm is to identify not only the duplexing mode (transmitting or receiving) but also the specific direction (to which neighbour) of the next communication in an opportunistic manner. For example, if a node is receiving a great deal of interference, it may be more appropriate for the node to choose to transmit, provided that the intended receiver is expected to receive properly. On the contrary, if a node finds that one of its incoming links of the highest profit among its entire associated links, then the node may prefer to receive from that link. In our scheduling algorithm, every directional link is assigned with a utility representing the benefit of transmitting on this link in next time slot, and the opportunistic approach is to choose a combination of concurrent links with the highest aggregated instantaneous utility. The precise definition of utility function and its convergence will be discussed in details in next section. In this section, we focus on the framework with generic utility definition. Since it has been shown in [7] that a collision free method for utility exchange is feasible, we assume here that utility values of both incoming and outgoing links is available to the node, and the two ends of the link keep the same latest utility version to make scheduling decision.

The first stage of the framework is for each node to choose the link with the highest utility among all the incoming and outing links to activate for the next time slot. Then in the ideal case, links with the highest $2/N$ utilities will be chosen to activate in an N-node mesh network.

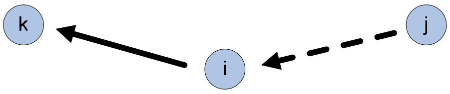

Figure 3 A conflict situation

However, the main difficulty in implementing this idea in a distributed way is the situation where a node makes a decision conflicting with one or more neighbours – the non-ideal cases. For example, in Figure 3, imagine that node *i* identifies that it is most profitable to transmit to the neighbouring node *k*, at the same time another neighbour node *j*, decides to send data to node *i*. Due to the TDD duplexing mode, node *i* cannot transmit and receive at the same time, and hence it is necessary to solve the conflict. In our opinion, it is not worth re-doing the scheduling on all the nodes in the network in order to find another solution, because fundamentally nodes have no knowledge about its neighbours' duplexing status before its decision making. Therefore, we retain the conflict-free decisions, and try to solve the conflict locally, i.e. between the two conflicting transmissions only. Simply, our suggestion is to give up the transmission with a collided destination, i.e. the dashed transmission in Figure 3. The reason is the following. As shown in Figure 3, the transmission (solid arrow) from node *i*, will be more profitable than the incoming transmission (dashed arrow) to node *i*, because it is the only reason why node *i* chose the solid arrow to be the next communication rather than the dashed one.

A formal description of our distributed scheduling framework is as follows. It is composed of four phases:

1. *Utility exchange*. Each node exchanges the utility function of each of its incoming and outgoing links with its neighbours.
2. *Initial decision*. Each node chooses the link with best utility to be the initial decision of next communication.
3. *Initial decision exchange*. Nodes with an initial decision of "transmit" – the best link of that node is an outgoing link – broadcast the initial "transmit" decision to all its one-hop neighbour nodes via a control channel. The control message indicates the IDs of the intended transmitting (origin) and receiving (destination) nodes associated with the initial decision.
4. *Final decision.*,
    - Each node with an initial decision of "transmit" checks if the desired receiving node is having the same "transmit" initial decision based on the control messages in Phase 3. If so, the node gives up the intended transmission. If not, the node starts transmission in that direction in next slot.
    - Each node with an initial decision of "receive" find out the best transmitter based on the control messages in Phase 3, and configures its physical layer to be ready to receive data from that direction in the next time slot.

Regardless of the specific utility function definition, so far the proposed framework has demonstrated following merits:

- It is fully distributed without deadlock. Nodes make scheduling decision simultaneously, and do not need to wait for other nodes' decisions to make its own decision.
- It takes advantage of multi-user diversity. Although in mesh networks it is very likely that the fluctuation of wireless links is weak, the multi-user diversity can be realized with other aspects such as differences in propagation loss (with random layout), independent uplink and downlink channel qualities and dynamic interference.
- It tends to generate a smooth interference, compared to random schedulers. Because the scheduling decisions are related to the instant utility, as long as the utility function is with strong time coherence, the link schedule shall be reasonably temporal correlated. From this point of view, it is sensible that tree-structure method will not perform well in terms of maintaining interference coherence, because the sets of active links in even and odd time slots are complementary to each other. Also, we do not expect smooth interference from IEEE 802.11 and 802.16 scheduling algorithms, because active links are decided based on random competition.

III. UTILITY DEFINITION AND CONVERGENCE

We have shown qualitatively that our framework provides some advantages with generic utility functions. However, to deliver a good backhaul service we need to ensure that the scheduling algorithm would be able to co-operate with routing algorithms. It is known that, in wireless networks, uncertainty in link capacity due to randomness of lower-layer protocols degrades performance of routing protocols. Furthermore, it is difficult to guarantee system performance if an opportunistic MAC layer is deployed, because opportunistic approaches usually introduce more fluctuating instantaneous performance at individual nodes. One of our major contributions of this paper is that we propose a utility function, or scheduling metric, which not only achieves opportunistic gain but also supports quality of service as committed by the routing algorithm in use.

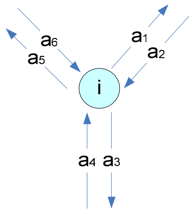

Figure 4 Target throughput allocation among links for one node when $r = 3$

The proposed co-operation between the scheduling and routing algorithms is in a "request-enforce" manner. Usually, the routing protocol determines routes for packet flows based on the total costs of paths. The cost of a link is usually computed using the long-time average capacity of that link. Thus, for routing protocol, to estimate the future link capacity is crucial in order to maintain an effective routing table for various source and destination node pairs. As a result, it is desirable for the routing layer to specify a target throughput allocation among links on each node and then request the scheduling algorithm to enforce such throughput allocation. Please note that rather than to achieve the precise target throughput for each link, the objective of our algorithm is to achieve the relative target throughput for each link scaled by a per-node (not per-link) proportionality constant. Thus achieving the relative target by the proposed scheduler effectively yields the actual throughput target.

Because the scheduling framework is fully distributed, in the discussion of the new utility derivation, we focus on one individual node. Here we treat the incoming and outgoing links equally as competitors. Then for a certain node *i*, every time slot, it has $2r$ candidate links for choice, given that r is the number of neighbours to the node. As shown in Figure 4, periodically, the routing algorithm estimates the routing demand in a certain future (up to thousands of packet durations) and passes the scheduling a target throughput allocation $\mathbf{a}_\mathbf{i} = \left(a_i(1), a_i(2), \cdots, a_i(2r)\right)$ for that duration. Then the challenge is to find out the appropriate utility definition, with which the scheduler's allocation of the long-run throughput $\mathbf{R}_\mathbf{i} = \left(R_i(1), R_i(2), \cdots, R_i(2r)\right)$ is proportional to the target allocation $\mathbf{a}_\mathbf{i}$, i.e. $\mathbf{R}_\mathbf{i}^* = c_i \cdot \mathbf{a}_\mathbf{i}$, c_i is any positive constant as the proportionality constant for that node. In order to identify the utility definition, first we formulate an optimization problem, at node *i*, by defining an objective function $f(\mathbf{R}_\mathbf{i})$, which is a function of $\mathbf{R}_\mathbf{i}$. Then we prove that the optimal solution $\mathbf{R}_\mathbf{i}^*$ associated with this objective function is proportional to the target throughput allocation. After that we define a utility function and prove that if a scheduler chooses a link according to this utility, the allocated throughputs by this scheduler, in the long run, converge to the optimal solution $\mathbf{R}_\mathbf{i}^*$ to the defined optimization $f(\mathbf{R}_\mathbf{i})$.

Lemma 1: If an optimization problem is to maximize the objective function

$$\max_{\mathbf{R}_\mathbf{i}} f\left(\mathbf{R}_\mathbf{i}\right) = \sum_{k=1}^{2r} a_i(k) \cdot \log R_i(k), \quad s.t. \quad \sum_{k=1}^{2r} R_i(k) \le C \tag{1}$$

then the optimal solution $\mathbf{R}_\mathbf{i}^* = \left(R_i^*(1), R_i^*(2), \cdots, R_i^*(2r)\right)$ is strictly proportional to $\mathbf{a}_\mathbf{i} = \left(a_i(1), a_i(2), \cdots, a_i(2r)\right)$.

Proof: it is a classic constrained optimization problem, which could be solved by Lagrange multipliers. Then the new Lagrange problem with multiplier λ_i for node *i* is:

$$L = \sum_{k=1}^{2r} a_i(k) \log R_i(k) - \lambda_i \cdot \left(\sum_{i=k}^{2r} R_i(k) - C \right) \tag{2}$$

The first order (necessary) optimality condition of (2) is:

$$\begin{cases} \nabla L = 0 \\ \lambda_i \cdot \left(\sum_{k=1}^{2r} R^*{}_i(k) - C \right) = 0 \end{cases} \tag{3}$$

Because the constraint is binding, $\lambda_i \ne 0$ and $\sum_{k=1}^{2r} R_i^*(k) - C = 0$, the first part of (3):

$$\nabla L = 0 \Leftrightarrow \frac{a_i(k)}{R_i^*(k)} = \lambda_i \qquad k = 1, 2, \cdots, 2r \tag{4}$$

Equivalently to:

$$R_i^*(k) = \lambda_i^{-1} a_i(k) \qquad k = 1, 2, \cdots, 2r \tag{5}$$

Note that i is the node index and λ_i is the same for all all links $k = 1, 2, \cdots, 2r$ associated with node *i*. That means the final throughput allocation, after a long run, is proportional to the target throughput allocation, $\mathbf{R}_\mathbf{i}^* = \lambda_i^{-1} \cdot \mathbf{a}_\mathbf{i}$, with the per-node proportionality constant equal to λ_i^{-1} ■

Lemma 2: If a objective function is defined as (1), then a single-link scheduler with scheduling utility/metric for link *k* defined as

$$M_i(k) = a_i(k) \frac{\rho_i(k)}{R_i(k)} \tag{6}$$

achieves this objective on a iterative basis., given that $\rho_i(k)$ is the instantaneous supportable data rate for the *k*-th link of node *i*, and $R_i(k)$ is the long time average of $\rho_i(k)$.

Proof: Because the objective function (1) is convex, the sufficient and necessary optimality condition [12] of optimality for this convex optimization problem is

$$\nabla f \odot \Delta \mathbf{R}_\mathbf{i} \le 0 \quad \text{at } \mathbf{R}_\mathbf{i} = \mathbf{R}_\mathbf{i}^* \tag{7}$$

Where $\mathbf{R}_\mathbf{i}^*$ is the optimal long-run throughput vector, and $\mathbf{R}_\mathbf{i}$ is any of the non-optimal average throughput vector, and $\Delta\mathbf{R}_\mathbf{i} = \mathbf{R}_\mathbf{i} - \mathbf{R}_\mathbf{i}^*$.

Expand (7):

$$\sum_{k=1}^{2r} a_i(k) \frac{R_i(k) - R_i^*(k)}{R_i^*(k)} \le 0 \tag{8}$$

$$\Leftrightarrow \mathrm{E}\left\{ \sum_{k=1}^{2r} a_i(k) \frac{\rho_i(k) - \rho_i^*(k)}{R_i^*(k)} \right\} \le 0 \tag{9}$$

Then the instantaneous rate vector in every iteration round $\boldsymbol{\rho}_\mathbf{i} = \left(\rho_i(1), \rho_i(2), \cdots, \rho_i(2r),\right)$ maximizing the following (10) will satisfy (9):

$$\max_{\boldsymbol{\rho}_i} \sum_{k=1}^{2r} a_i(k) \frac{\rho_i(k)}{R_i(k)} \tag{10}$$

For a single-link scheduler, (10) becomes

$$\max_k a_i(k) \frac{\rho_i(k)}{R_i(k)}. \tag{11}$$

That is, the scheduler choosing the link with the largest metric (11) to serve in each time slot will achieve the objective function (1). ■

Based on both lemma 1 and 2, we conclude that a single-link scheduling algorithm with scheduling metric/utility defined as (11) is maximizing the objective function defined as (1) which leads to a actual throughput allocation equal to the target allocation. Simply speaking, the scheduler with utility function (11) allocates the throughput strictly proportional to the rooting demands in the long run.

Combining the framework and the utility definition, we obtain a novel integrated, distributed and opportunistic scheduling algorithm for wireless mesh networks.

IV. NUMERICAL RESULTS

In this section, we investigate the performance of our proposed scheduling algorithm by simulation, in terms of the throughput allocation convergence, the interference temporal correlation and the average network throughput. In our simulation, the hexagonal layout network consists of 37 wireless routers. Independent Rayleigh-fading channels between different sender/receiver pairs are implemented. Low Doppler frequency of 5~20Hz, corresponding to 1.08~4.32kmph velocity for 5GHz carrier frequency, is assumed due to the stationary topology.

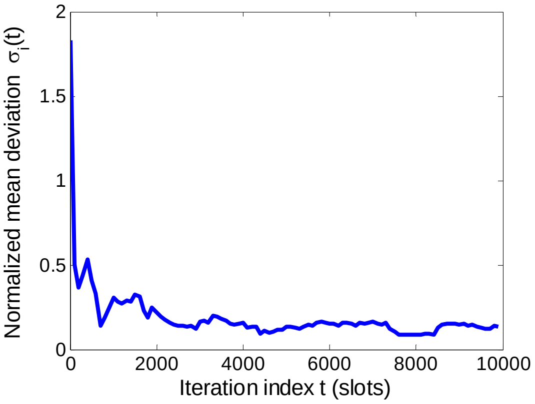

Figure 5 Normalized mean deviation of the ratio vector

To assess the convergence of our algorithm in terms of throughput allocation, we randomly generate objective throughputs in our simulation, and observe the actual time-averaged throughput of each link slot by slot. The actual throughput gained by each links is recorded and compared relative to the predefined throughputs. The comparison is examined by the ratio below at the central-position node of the network:

$$\beta(k,t) = \frac{R(k,t)}{a(k)} \quad k = 1,2,\cdots,2r \tag{12}$$

In (12) the numerator is the actual long-run throughput of the candidate links up to time slot *t*, while the denominator is the target throughput allocation. Note that what matters here is the similarity between the elements of the vector $\boldsymbol{\beta}$: the more similar to each other, the better the convergence. Imagine the ideal case where all the links have the same ratio, which means:

$$\beta^*(t) = \frac{R(k,t)}{a(k)} \quad k = 1,2,\cdots,2r\,,$$

then we have $\mathbf{R}(t) = \beta^*(t)\cdot\mathbf{a}$, which is our original objective. Along this line, we assess the similarity of ratios by taking the normalized mean deviation of $\boldsymbol{\beta}$ every slot:

$$\sigma(t) = \frac{1}{2r}\sum_{k=1}^{2r}\left|\frac{\beta(k,t)-\mu_R(t)}{\mu_R(t)}\right|.$$

Where t is the time index, and $\mu_R(t)$ is the average of $\{\beta_i(k,t), k = 1,2,\cdots.2r\}$. $\beta_i(k,t)$ is the ratio on the *k*-th link in *t*-th time slot. Figure 5 unfolds that our algorithm converges to the target throughput allocation among links for the central node, by showing that the normalized mean deviation of the ratio falls down rapidly from the original high value (1.82) to near zero (0.15).

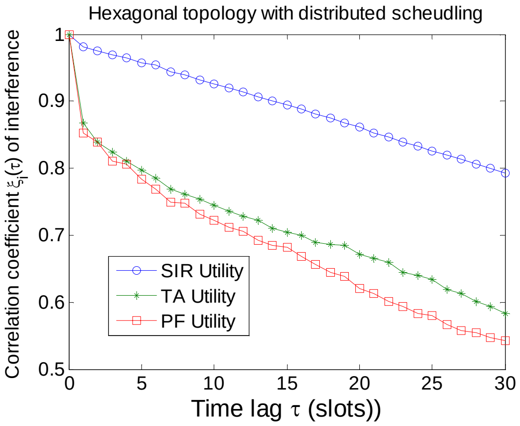

Figure 6 Temporal correlation coefficient of interference for distributed scheduling

Regarding to the temporal correlation of interference, we measured the interference at the central-position node every time-slot and plot out its autocorrelation coefficient after a simulation running 10,000 slots:

$$\xi_I(\tau) = \mathrm{E}\left\{\frac{\left|\left\{\left(I(t)-\mu_I\right)\left(I(t-\tau)-\mu_I\right)\right\}\right|}{\sigma_I^2}\right\}. \tag{13}$$

Theoretically, the coefficient $\xi_I(\tau)$ is in the range of $[0,1]$, where $\xi_I(\tau)=0$ implies temporally independent process, and $\xi_I(\tau)=1$ indicates very strong temporal correlation. Figure 6 shows that the proposed scheduling framework is able to maintain a reasonable interference temporal correlation for all utilities tested here. The signal-to-noise-ratio (SIR) utility brings higher temporal correlation than the other two methods. Moreover, our proposed throughput-allocation (TA) utility generate slightly more correlated interference than the commonly used proportional-fair (PF) utility. It is resulted from the fact that PF utility is a special case to our TA utility with identical throughput allocation to all the links. The more the allocation between links diverse, the more the scheduling decisions coherent in time, and hence the more temporally correlated interference received. Another insight is that if the target throughput is proportional to the observed actual link capacity, which is likely to happen when adaptive routing algorithms exist, then the interference temporal correlation provided by our TA utility will be as good as that provided by the SIR utility. This is because of the fact that the target allocation $\mathbf{a}_i$ and the actual throughput $\mathbf{R}_i$ in (11) will be cancelled, and the scheduling metric becomes purely opportunistic SIR.

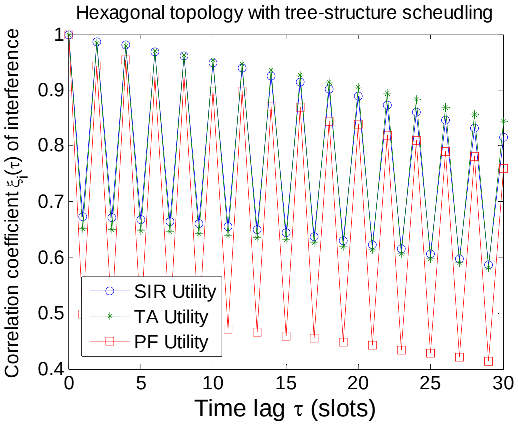

Figure 7 Temporal correlation coefficient of interference for TREE scheduling

With the same setting, Figure 7 indicates that the tree-structure generates oscillating interference no matter which utility is taken. It is known that both distributed power control and scheduling algorithms rely on temporally coherent interference to predict the future channel quality. The oscillating interference characteristic degrades the scheduling and power control algorithms because it causes erroneous channel predictions and hence wrong power control and scheduling decisions.

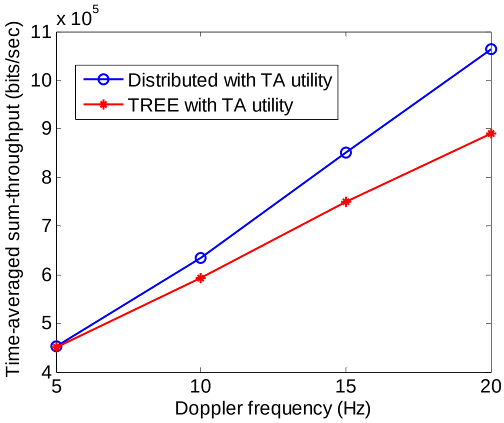

Figure 8 Network throughput for hexagonal topology

The network aggregated instantaneous throughput can be seen as a measurement of the opportunistic gain achievement. In Figure 8 an up to 19% enhancement in network throughput is provided by our proposed distributed algorithm compared to the TREE method. We see that the throughputs for both distributed and TREE scheduling are growing up as the channel fluctuates faster. It is because that as discussed preciously, the TREE method is also an opportunistic approach, it does takes advantage of channel fluctuation to some extent. However, due to the static tree mapping and missing of horizontal links, the network throughput obtained by TREE method is uniformly below that produced by our proposal. As a result, our distributed scheduling algorithm is superior to TREE method in terms of both interference temporal correlation and opportunistic gain achievement, besides the proposed TA utility is able to help routing algorithms guarantee QoS.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel scheduling algorithm for wireless mesh backhaul networks. The new scheduling algorithm is fully distributed, capable of selecting high utility link combinations to achieve opportunistic gain, and generating reasonably temporal-correlated interference to ensure distributed power control and scheduling performance. Both analysis and simulation results revealed that our algorithm is able to enforce the redefined throughput allocation to help routing algorithms guarantee QoS. In this paper, we only considered the single-link scheduler, MIMO antenna equipped networks and hence multi-link scheduling algorithms will be the next stage of our research.

REFERENCES

[1] L. Chen, S. H. Low, J. C. Doyle, and M. Chiang, "Cross-layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," *Proc. of IEEE INFOCOM*, Barcelona, Spain, 2006.

[2] S. Ramanathan, "Scheduling algorithms for multihop radio networks," *IEEE / ACM Transactions on Networking,* vol. 1, no. 2, pp. 166-177, 1993.

[3] K. Jain, "Impact of interference on multi-hop wireless network performance," *Wireless networks,* vol. 11, no. 4, pp. 471-487, 2005.

[4] L. Lovasz, *Matching theory,* North-Holland, 1986.

[5] M. R. Garey, *Computers and intractability : a guide to the theory of NP-completeness*, W. H. Freeman, 1979.

[6] "IEEE Std 802.11-1997 Information Technology-telecommunications And Information exchange Between Systems-Local And Metropolitan Area Networks-specific Requirements-part 11: Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications," IEEE Std 802. 11-1997, p. i-445, 1997.

[7] "IEEE Std. 802.16-2001 IEEE Standard for Local and Metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Std 802. 16-2001, pp. 0-322, 2002.

[8] G. Narlikar, G. Wilfong, and L. Zhang, "Designing Multihop Wireless Backhaul Networks with Delay Guarantees," *Proceeding of IEEE INFOCOM,* Barcelona, Spain, 2006.

[9] Y. Hou and K. K. Leung, "Performance Impacts of Erroneous Channel Predictions on Packet Scheduling in Wireless Networks," *Proceeding of Global Mobile Congress (GMC),* Beijing, 2006.

[10] K. Leung, "Power control by interference prediction for broadband wireless packet networks," *IEEE Transactions on Wireless Communications,* vol. 1, no. 2, pp. 256-265, 2002.

[11] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Transactions on Vehicular Technology,* vol. 42, no. 4, pp. 641-646, 1993.

[12] S. Boyd and L. Vandenberghe, *Convex optimization,* Cambridge University Press, 2004.