A Realizable Cross-Layer Architecture for Wireless Mesh Networks *

Jinsung Lee, Junhee Lee, and Song Chong
School of EECS, KAIST, Daejeon, Korea
{ljs, junhee}@netsys.kaist.ac.kr, song@ee.kaist.ac.kr

Introduction

A wireless mesh network (WMN) is an emerging technology to provide open access to the Internet with high bandwidth and low cost while covering a wide area without wiring. Compared to ad-hoc networks, traditionally focusing on military and specialized civilian applications, WMNs can be thought as the generalized one for fulfilling the actual user requirements. Accordingly, WMNs have become one of the most prospective candidates for the Future Internet technology.

In the landmark paper [3], Tassiulas and Ephremides introduced *throughput optimal* technique of backpressure routing and maximum weight link scheduling which was shown to yield network stability whenever input rates are within the network capacity region. Specifically, the dynamic backpressure policy activates the non-interfering links by assigning high priority to those with large *differential backlogs*. The well-known further work is to combine flow control with routing and scheduling problem [2]. However, the backpressure policy has some critical problems that make it difficult to implement in practice. The first is low scalability with the increasing number of flows. Due to per-flow queueing, it requires considerable complexity in a large-scale network. Secondly, the policy requires unconventional rate control at each source, which means that users in the mesh network need to change their transport protocol into totally new one instead of TCP. Finally, an optimal scheduling requires centralized computation and even might be NP-hard in the worst case.

In this paper, we propose a realizable mesh router architecture using dynamic backpressure policy against the above problems. Specifically, for multi-hop link scheduling we consider the Greedy Maximal Scheduling (GMS) which is regarded as a well-known suboptimal scheduling [1]. But it has not been implemented in practice despite lots of mathematical works so far. As an alternative, our paper focuses on the issues related to implementing GMS. For flow control, we separate the mesh router architecture into access and backhaul parts so that the router only maintains the small number of queues even in a large-scale network and the users do not modify their transport protocol. Moreover, we can provide better fairness between uplink and downlink traffic at each router as a result of separation.

*This work was supported by the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MOST) (No. R01-2006-000-10753-0).

The Proposed Mesh Router Architecture

Network Model

Each user who can be either source or destination is assumed to have one 802.11 radio and each mesh router is assumed to have two radios that operate simultaneously over independent channels. The first radio is used for uplink (user-to-router) and downlink (router-to-user) communications, while the second one is used for backhaul (router-to-router) communication. Thus, the router has the *access part* for uplink and downlink communications and the *backhaul part* for multi-hop communication with other routers. All routers are fixed and have paths determined in priori among them. Although the latter assumption would result in the reduced capacity region in the network, it could improve QoS such as delay[1].

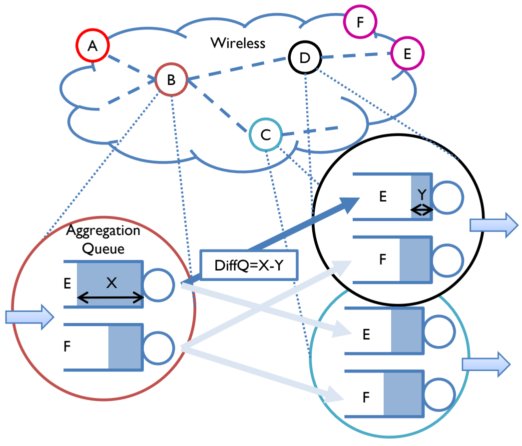
Mesh backhaul Architecture

Each mesh router collects the backlog information from its neighbor routers. As in Fig. 1(a), the link scheduling is performed according to a differential backlog strategy to maximize a sum of products of transmission rate and differential backlog over all the links within a network, which is the most complex part and we consider a suboptimal method based on GMS. Still it requires a linear complexity with the network size, which originally arises from exchanging of control messages such as backlog information among routers.

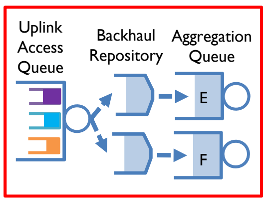
The CSMA mechanism makes use of broadcasting nature of wireless medium for carrier sensing and provides a clue that enables implicitly coordination among interfering links. Thus, every router maintains a local maximum

[1] In the original policy, routing loop may happen since it only considers relative difference of backlogs as a routing metric.

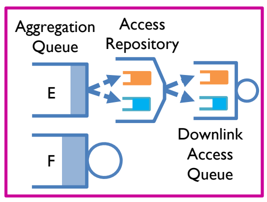
weight, i.e. differential backlog with interfering routers without any message exchange and it chooses its contention window of which size is inversely proportional to the weight. The higher weight a link has, the more chances it could get. This approach approximates GMS in a probabilistic manner while keeping low control overhead.



(a) Backhaul architecture for B, C and D



(b) Uplink architecture for A with 3 source users



(c) Downlink architecture for E with 2 destination users

Figure 1. An example mesh network with two aggregation flows: A, E, and F are edge routers while B, C, and D backhaul routers.

Mesh Access Architecture

Unlike the previous backhaul architecture, the access architecture can be implemented in a more intelligent way. A mesh user associates to the nearby mesh router[2] of which the access part creates per-flow queue for each user. The uplink architecture is shown in Fig. 1(b). Mesh users covered by one edge router contend for accessing channel using 802.11 DCF. The incoming packets enter the corresponding input queue defined as uplink access queue at an edge router. Since the router maintains the path to every destination a priori, it can easily find out the edge router containing the destination. When the packet moves from uplink access queue to *backhaul repository*, an aggregation takes place according to its destination router and we define the *aggregation flow* as one from source router to destination router, which belongs to the backhaul part. From the repository the flow control determines the amount of packets released to *aggregation queue* (output queue) using its local price, i.e. the backlog of aggregation queue. No packet drop is assumed due to a enough size of the repository. For example, if there are n destination routers in a network, each one can have at most $n-1$ aggregation flow queues. For downlink transmission, the packet is delivered to the aggregation queue (input queue) of the proper destination router over a multi-hop path and then it enters the *access repository* according to its destination. Contrary to the uplink case, the aggregation flow is divided on a per-user basis and then the flow control is carried out using their local backlogs. At the downlink queue, the packet is scheduled in a way that maximizes a sum of products of link rate and backlog over all flows and transmitted with high priority through the 802.11 DCF rule using IFS smaller than DIFS. Therefore, the downlink traffic is more likely to access channel over the uplink one.

In our approach, the access part in the edge router can handle both uplink and downlink transmissions. Thus if giving some intelligence such as priority assignment among different users, we can solve the 802.11 unfairness and even provide better QoS.

Conclusions

We presented the cross-layer mesh architecture for future Internet technology: the backhaul with dynamic backpressure policy and the access with QoS differentiation.

Currently, we are implementing the features mentioned here in our testbed[3], which requires some heuristics slightly deviated from theoretical optimum. There are still important concerns such as interaction with TCP. But we expect it can give a valuable insight to those who are interested in backpressure policy.

References

[1] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *IEEE Infocom*, Miami, FL, USA, Mar. 2005.

[2] M. J. Neely and R. Urgaonkar. Cross layer adaptive control for wireless mesh networks. *Ad Hoc Networks (Elsevier)*, 5(6):719–743, Aug. 2007.

[3] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. on Automatic Control*, 37(12):1936–1948, Dec. 1992.

[2] We term this mesh router an *edge router*, and otherwise, a *backhaul router*. Furthermore, the edge router is termed source router (or destination router) if it has source clients (or destination clients) in its coverage.

[3] WiSEMesh at KAIST, http://netsys.kaist.ac.kr/wisemesh/doku.php