Admission Control for Multihop Wireless Backhaul Networks with QoS Support

Seungjoon Lee, University of Maryland, College Park , MD
Girija Narlikar, Bell Laboratories, Murray Hill, NJ
Martin Pál, Bell Laboratories, Murray Hill, NJ
Gordon Wilfong, Bell Laboratories, Murray Hill, NJ
Lisa Zhang, Bell Laboratories, Murray Hill, NJ

Abstract— **Despite improvements in wireless access technologies such as 3G or 802.11x, ubiquitous data access has remained a challenge, mainly due to the lack of inexpensive, pervasive backhaul connections from access points to the Internet. With the recent WiMAX standard for high-speed, non-line-of-sight fixed wireless links, multihop wireless backhauls might now overcome this bottleneck. However an important remaining challenge is to provide rate and delay guarantees for customer connections similar to wired backhauls.**

We provide several schemes for performing admission control for connections with QoS requirements over a multihop wireless backhaul. This is the first work to address both rate *and* delay requirements for connections. Our admission control algorithms first construct appropriate tree-based topologies connecting wireless backhaul nodes to a wired gateway and then admit the best subset of connections while respecting their rate and delay requirements. Alternately, we admit all the connections with appropriate degradation of their QoS requirements.

I. Introduction

Cheap, ubiquitous data access has long been the holy grail for service providers. Recent advantages in wireless technologies such as 802.11x and 3G have enabled fast and convenient data access to users. However, the backhaul, which connects access points (base stations) to the core network continues to be a bottleneck in either price or performance. Fast wired connections such as fiber or T3/E3 are expensive to lease or install in many situations, while slower wired connections like T1/E1, DSL or cable provide inadequate bandwidth and/or poor QoS guarantees.

A recent longhaul, non-line-of-sight, high-speed wireless standard, 802.16-2004 (also called WiMAX), has enabled the possibility of a fast wireless backhaul. A *multihop wireless* backhaul has the unique advantage of providing extended coverage in difficult to reach areas or areas with little existing wired infrastructure. In addition, with careful planning, shorter, multiple hops help increase throughput due to lower pathloss and greater spatial reuse[15].

With time-division multiplexing (TDM) and centralized scheduling, physical layers like WiMAX now allow the possibility of a wireless backhaul with wireline-like QoS guarantees. Centralized scheduling for QoS in a single-hop wireless network has been well studied in the context of both TDMA [9], [22], [18], [1] and 802.11-like CSMA/CA [3] physical layers. Recent work has also explored QoS scheduling for multihop (mesh) networks [21], [19]. However, these solutions only guarantee *bandwidth* as part of QoS, and to the best of our knowledge, no work has addressed explicit delay requirements on multihop wireless networks. If a multihop wireless backhaul is to become reality, we believe that many interactive applications such as VoIP, interactive video, remote access and gaming need to be supported. Therefore, delay *and* bandwidth guarantees need to be an important component of QoS services provided by the backhaul.

In this paper, we show how a service provider can provide strict QoS (bandwidth and delay) guarantees to its individual backhaul customers based on their connection requirements. Given locations and demands of customers, the service provider must: (a) generate a topology for the multihop wireless backhaul that connects all the customers to an Internet gateway, and, (b) admit some or all of the customers' demands. Based on a scheduling and routing framework that provides upper bounds on end-to-end delays in a multihop wireless backhaul [15], we outline techniques for admission control of backhaul traffic. When customer demands, namely, connection rates and delay requirements, are known a priori, there are two approaches to admission control: either we can admit a subset of the requested connections while guaranteeing the specified rates and delays, or we can admit all the connections while scaling down their rates and/or increasing their delay tolerances in a fair fashion. We present solutions along both approaches. To the best of our knowledge, this is the first attempt to perform admission control for individual connections on a wireless backhaul such that both their rates and delays are deterministically satisfied.

Previous work. A vast body of work has explored bounded-delay scheduling algorithms in the wireline world: the most prominent of these are weighted fair queuing (WFQ) [4], also known as Packet-by-Packet Generalized Processor Sharing (PGPS) [16], [17], and Coordinated EDF (CEDF) [2], [13]. Given the connection rate, maximum burst size, packet size and number of hops, both algorithms provide guaranteed upper bounds on the end-to-end delay experienced by individual packets. Algorithms have been presented for connection admission control on a single GPS server for connections with delay requirements [20], [12]. A measurement-based admission control algorithm for multihop wired networks that increases network utilization by allowing occasional delay violations was presented in [8]; a survey of admission control algorithms can be found in [7].

In the wireless domain delay-aware QoS schedulers have been proposed in the context of single-hop cellular networks [9], [22], [18], [1]. Delay and rate aware scheduling in the context of 802.11-based multihop networks has been proposed using a probabilistic approach using priority back-off [10]. However, very little work has been done to deterministically guarantee delays in multihop wireless networks; previous work on QoS had focused mainly on guaranteeing rates [19], [21].

Organization. We present our model in detail in Section II. In Section III we study the objective of maximizing the profit of admitted connections when the QoS requirements of each admitted connection are satisfied. In Section IV we focus on an alternative objective of admitting all connections with relaxed

QoS.

II. Model

Input. Our input is a set of customer locations V along with a gateway g, which has a high-speed wired connection to the Internet. The customer locations must be connected to g via the multihop, wireless backhaul network. We henceforth also refer to customer locations or the gateway as *nodes*. The demands for each customer $v \in V$ are specified as a set of connections $\Omega(v)$. The two endpoints in the backhaul for the connections in $\Omega(v)$ are the customer location v and the gateway. Each connection in $\Omega(v)$ has packets traveling in the uplink and downlink directions; uplink packets must travel from node v to the gateway g, while downlink packets must travel from g to v. QoS requirements for connection i are expressed as a minimum rate ρ_i^u and a maximum delay D_i^u for packets in the uplink direction, and a minimum rate ρ_i^d and a maximum delay D_i^d in the downlink direction. In general, rate requirements may be highly asymmetric (e.g., streaming video or data downloads) or symmetric (e.g., VoIP); delay requirements may be highly heterogenous (e.g., VoIP vs data downloads). Each connection i also has a reward $w(i)$, denoting how valuable it is to admit this connection.

Individual links in the wireless backhaul are assumed to be directional, connecting a pair of nodes; the set of all links is E. We are given the capacity $C(e)$ of each link $e \in E$; this is the maximum transmission rate for the link in the absence of interference from other transmissions. Links connecting nodes that are very far apart will have very low or zero capacity. We assume negligible fast fading on backhaul links since both endpoints are static; however, shadow fading may affect link capacities. All packets in the backhaul network are assumed to be of a fixed size L; larger packets are fragmented at the source and reassembled at the destination.

Physical Layer. We assume that the wireless backhaul links use the WiMAX (802.16d) standard (or some similar longhaul NLOS technology), with links communicating in the time division duplex (TDD) mode. Further, the MAC layer (as in 802.16d) is assumed to schedule data to multiple receivers across timeslots using time division multiple access (TDMA) with a timeslot of duration τ. We also assume the use of subchannelization within a timeslot. Subchannelization allows the available spectrum to be subdivided into multiple orthogonal subchannels (subcarriers in an OFDM system [11] or spreading codes in a CDMA system), so that multiple separate data streams can be received in the same timeslot over separate subchannels at the same or nearby receivers with no interference[1]. We assume there are a total of M subchannels for wireless backhaul transmission. The *effective capacity* $c(e)$ for a link e that uses $m \leq M$ subchannels is given as $c(e) = C(e) \cdot m/M$. The wireless access technology (e.g., 802.11x) used by individuals at customer locations is assumed to be in a separate frequency band compared to the backhaul transmissions, and therefore the access and backhaul transmissions do not interfere with each other.

Interference. We assume a node may simultaneously receive on multiple subchannels coherently; and it may simultaneously transmit on multiple subchannels. However, we assume that a node transmitting on any of the subchannels cannot simultaneously receive data, even if the reception is on different subchannels [], because the transmission may overpower the received signal. **Girija, could you verify “overpower”?** This model differs from some previous works [] that assume subchannels to be completely orthogonal, allowing simultaneous transmission and reception at a node on different subchannels. Their approach requires subchannelization in the frequency domain ensuring that individual subchannels are spaced far apart, resulting in significant wastage of spectrum. We do not consider cross interference between links that do not share a common node in our model. This is reasonable for highly directional transmissions. See Section V for further discussion.

Backhaul topology and scheduling framework. A natural and easy way to maintain topology for a backhaul network is a tree rooted at the gateway; for example, the WiMAX standard assumes a tree topology for multihop networks when links are centrally scheduled. Henceforth we assume that our backhaul topology is a tree. To ensure end-to-end delay guarantees, we assume the Even-Odd framework as described in detail in previous work [15]. Briefly, our backhaul network labels each node as either an *even* or *odd* node, depending on its depth in the tree. Thus all backhaul tree links connect an even node and an odd node. Every even timeslot, all even nodes are in the *transmitting mode*, transmitting to one or more neighbors. At this time all odd nodes are in the *receiving mode*. Similarly, every odd timeslot, all odd nodes are in the transmitting mode and all even nodes are in the receiving mode. In the transmitting mode, all transmissions from a node must be on different subchannels; in the receiving mode all receptions at a node must be on different subchannels. Since all directional links are active half the time, we can view the wireless backhaul as a *half-idle* network.

A link e of capacity $C(e)$ requires $F(e)/C(e)$ subchannels to transport a total flow of $F(e)$. Since each node spends half its time in the receiving mode and half its time in the transmitting mode, and the subchannels used on each of its incident links must be disjoint in any given mode, the even-odd framework specifies the following admissibility conditions [15] on rates. First, we define $E_{in}(v)$ and $E_{out}(v)$ for every node v as follows:

$$\begin{array}{rcl} E_{in}(v) & = & \{(w,v) \mid (w,v) \in E\} \\ E_{out}(v) & = & \{(v,w) \mid (v,w) \in E\} \end{array} \qquad \forall v \in V.$$

If $F(e)$ is the total bit rate of all selected connections on link e, then the selected connections are *rate-admissible* in the Even-Odd scheduling framework if and only if

$$\left\{ \begin{array}{lcl} \sum_{e \in E_{in}(v)} \frac{F(e)}{C(e)} & \leq & 1/2 \\ \sum_{e \in E_{out}(v)} \frac{F(e)}{C(e)} & \leq & 1/2. \end{array} \right. \qquad \forall v \in V. \qquad (1)$$

Delay bounds. The Even-Odd framework allows us to compare the operation of a half-idle wireless network with a fully busy wireline network with the corresponding topology and effective edge capacities. In particular, any wireline scheduling policy can now be applied *locally* at each node of the wireless network. The resulting end-to-end delays experienced in the wireless network were proved to be approximately twice the end-to-end delays in the corresponding wireline network [15] for many scheduling policies, e.g. WFQ and CEDF. In the rest of this paper, we assume that the WFQ scheduling policy is utilized in our backhaul network. For a connection i in a single direction with a burst size of σ_i, a rate ρ_i, and a path

[1] 802.16d currently supports subchannelization in units of 1, 2, 4, 8, or 16 OFDM subcarriers in the uplink.

P_i with K_i hops where the kth has link effective capacity c_k, the wireline delay Δ_i^{wired} was shown to be bounded by [17]:

$$\Delta_i^{wired} \leq \tfrac{\sigma_i}{\rho_i} + (K_i - 1) \cdot \tfrac{L}{\rho_i} + \textstyle\sum_{1 \leq k \leq K_i} \tfrac{L}{c_k}$$

Since connection i is routed along path P_i, the effective capacities c_k of all the edges along the path must be at least ρ_i. Therefore we can rewrite the above equation as

$$\Delta_i^{wired} \leq \tfrac{\sigma_i}{\rho_i} + 2 \cdot K_i \cdot \tfrac{L}{\rho_i} \tag{2}$$

The Even-Odd framework [15] allows us to bound the corresponding delay Δ_i on our backhaul wireless network as:

$$\begin{aligned} \Delta_i &\leq 2 \cdot \Delta_i^{wired} + (K_i + 5) \cdot \tau \\ &\leq 2 \cdot \frac{\sigma_i}{\rho_i} + 4 \cdot K_i \cdot \frac{L}{\rho_i} + (K_i + 5) \cdot \tau \end{aligned} \tag{3}$$

We denote the right hand side of (3) as a function of burst, rate and hop count, $D(\sigma_i, \rho_i, K_i)$.

III. Topology Construction and Admission Control

In this section, our goal is to construct a good tree topology and select the "best" set of connections for which we can satisfy the rate and delay (QoS) requirements. When evaluating a set of admitted connections, we use the following definitions: $T(v)$ denotes the total reward of all connections in $\Omega(v)$, and $A(v)$ denotes the total reward of *admitted* connections among $\Omega(v)$. We consider two objectives: $\sum_{v \in V} A(v)$ and $\min_{v \in V} \frac{A(v)}{T(v)}$.

Theorem 1: Given a set of nodes and demands, it is NP-hard to construct a backhaul tree that optimizes either objective.

In Section III-A we present an ILP formulation for the combined problem of tree construction and connection admission. Later we decouple the problem into two subproblems. In Section III-B we present fast heuristics for constructing tree topologies. In Section III-C we show that once a tree topology is specified, the best set of admissible connections can be selected in polynomial time.

A. Combined topology construction and connection selection

We formulate the topology construction and connection admission as a combined ILP as described below. We focus on maximizing the total reward of admitted connections, $\sum_v A(v)$. However, we can easily extend it for the other objective, $\min_v \frac{A(v)}{T(v)}$. We omit many details of the ILP due to space limitations.

$$\max \sum_i w_i \cdot s_i \qquad \text{subject to :}$$

$$\text{Flow conservation on } flow_i^z(e) \qquad \forall i, z \tag{4}$$

$$flow_i^z(e) \leq X(\{e, \bar{e}\}) \qquad \forall i, z, e \tag{5}$$

$$\sum_e X(\{e, \bar{e}\}) = |V| - 1 \tag{6}$$

$$r_i^z \geq \rho_i^z \qquad \forall i, z \tag{7}$$

Admissibility condition (1) with

$$F(e) = \sum_{i,z} r_i^z * flow_i^z(e) * s_i \qquad \forall v \tag{8}$$

$$K_i^z = \sum_e flow_i^z(e) \qquad \forall i, z \tag{9}$$

$$D(\sigma_i^z, r_i^z, K_i^z) \leq D_i^z + (1 - s_i) * H \qquad \forall i, z \tag{10}$$

In the above ILP, the superscript $z \in \{u, d\}$ indicates uplink or downlink transmission; the subscript i indicates connection i. Therefore, i combined with z specifies connection i in the transmission direction z. All links e are directed.

For each connection i, recall w_i is the profit of connection i and we define a binary variable s_i that indicates whether or not i is selected. We also define a binary variable $flow_i^z(e)$ to indicate whether or not e carries traffic for connection i in direction z. Flow conservation (4) can be expressed in linear constraints in terms of $flow_i^z(e)$. We also define a binary variable X on every pair $\{e, \bar{e}\}$ where $\bar{e}$ is link e in the opposite direction. (We view $\{e, \bar{e}\}$ as the *undirected* version of e and $\bar{e}$.) Constraint (5) forces $X(\{e, \bar{e}\}) = 1$ if either e or $\bar{e}$ carries any traffic. Constraint (6) enforces a tree topology, which is used for both uplink and downlink transmission.

We let variable $r_i^z \geq \rho_i^z$ represent the provisioned rate of each connection i in direction z if i is admitted. For admitted connections, we use constraints (8) and (10) to enforce the rate admissibility condition and the delay constraint. Note that constraint (8) contains the product of three linear variables, and is therefore nonlinear. However, since $flow_i^z(e) * s_i$ is binary and $r_i^u + r_i^d$ is bounded, there are well-known methods to linearize $F(e)$. We omit details here.

For delay constraint (10), we choose a sufficiently large parameter H so that this constraint is trivially satisfied for any connection not selected. Also, as written this constraint is not linear since $D(\sigma_i^z, r_i^z, K_i^z)$ has r_i^z in the denominator but we can remove it by multiplying D_i^z by r_i^z. Constraint (10) involves K_i^z, the hop count of the connection path. Since we have ensured the resulting topology is a tree, each connection i has one possible path that consists of links e where $flow_i^z(e)$ is set to 1.

In the ILP we have enforced that the uplink and downlink transmission share a common tree, and each connection uses a common path for uplink and downlink transmission. We can, however, have a separate tree for each direction z by defining $X^z(e)$ to indicate if e carries traffic in transmission direction z. The links with $X^z(e)$ set to 1 define a tree for direction z. Other constraints remain unchanged.

B. Heuristics for Tree Construction

Solving the above ILP for even moderate problem sizes is difficult. Therefore we next consider fast heuristics for tree construction.

- **MST**: Builds a minimum spanning tree using $1/C(e)$ as edge costs.
- **Star**: Connect every node to the gateway by a direct link.
- **SP-order**: This heuristic and the next are based on the following. For a set of connections in a tree, let $F(e)$ be the resulting total bit rate routed over e. Then the *node load, $L(v)$,* is defined as follows: $L(v) = \max(\sum_{e \in E_{out}(v)} F(e)/C(e), \sum_{e \in E_{in}(v)} F(e)/C(e))$. Suppose each node has a single unit-rate uplink connection. We iteratively build up a tree by attaching a node not yet in the tree via a link to a node in the tree. The link chosen is one that minimizes the resulting maximum node load. In this heuristic, the vertices are added in the order given by the shortest path distance to the gateway where the edge lengths are given by $1/C(e)$. These distances can easily be computed using Dijkstra's algorithm.
- **Depth-order**: We iteratively build a tree as in SP-order except we order the nodes as follows. For each connection

i, we set $r_i^x = \rho_i^x$, $x = u, d$ and calculate how long the path for i (in terms of hop count) can be without violating the uplink or downlink delay requirement based on Inequality 3. Then, since each node can have multiple connections, for each node we take the average of these hop count values for the connections at that node and order the nodes in ascending order of this average value.

In Section III-D, we present an experimental evaluation of these heuristics, using algorithm AdmitBest to compute a measure of quality for each topology.

C. Selecting connections given tree topology

In this section we assume that the backhaul tree topology has been determined, e.g. using one of the heuristics given above. We now turn to the problem of selecting a set of connections that can be simultaneously supported while respecting all capacity and delay constraints. Recall we have two objectives, $\max \sum_{v \in V} A(v)$ and $\max \min_{v \in V} \frac{A(v)}{T(v)}$.

Theorem 2: For either objective, we can find an optimal subset of connections via dynamic programming.

Since the route for each connection is now fixed, the delay experienced by each connection i, according to Equation (3), depends only on the inverse of its rate ρ_i^z. Hence, in order to satisfy the required delay bound D_i^z, we adjust the actual rate ρ_i^z of each connection to a potentially higher *declared rate*

$$r_i^z \quad = \quad \max\left\{\rho_i^z, \frac{2 \cdot \sigma_i^z + 4 \cdot K_i \cdot L}{D_i^z - (K_i + 5) \cdot \tau}\right\}. \tag{11}$$

From now on, it suffices to find a subset of connections that are rate-admissible with respect to their declared rates r_i^z, as delay guarantees will be satisfied automatically.

We shall find an optimal set of admissible connections by dynamic programming. To this end, we discretize the rate allocations into equal-sized units (e.g., of 500kbps). For example, multiple VoIP connections are bundled together by the customer into a single connection request, similar to wireline allocations in terms of bundled voice channels over a T1 or a fractional T1. Input rate demands r_i^u, r_i^d are assumed to be multiples of this increment. We use $n(u)$ to denote the number of children of node u in the tree; g denotes the gateway or tree root. The AdmitBest algorithm builds a set of tables in a bottom-up fashion. There is a table for each parent-child pair: $\mathrm{R}(v, i)$ is the table corresponding to the i-th child of v ($i \leq n(v)$). In addition, we have a table $\mathrm{R}(v, 0)$ corresponding to an artificial "zeroth" child.

The table entry $\mathrm{R}(v, i)[F^u, F^d, m^x, m^r]$ denotes the maximum reward at node v when we use a total of m^x subchannels for transmission and m^r subchannels for reception at v on links between node v and its first i children, the aggregate upstream bandwidth of first i subtrees (one subtree for each child) is F^u, and the aggregate downstream bandwidth of i subtrees is F^d.

Then, we want to maximize

$$\max_{F^u, F^d, m^x, m^r} \mathrm{R}(g, n(g))[F^u, F^d, m^x, m^r].$$

Base Case. Recall that $\Omega(v)$ is the set of all connections associated with node v. As the base case, we consider $\mathrm{R}(v, 0)[F^u, D^d, m^x, m^r]$ to handle a subset of connections in $\Omega(v)$. Since access connections do not share spectrum with backhaul transmissions, we can admit any connection without using subchannels at or below node v; we will only require subchannels to transport their packets above node v (between v and its parent). Therefore, we initialize the base case as follows:

$$R(v, 0)[F^u, F^d, 0, 0] = \max_{S \subset A} \left\{ \sum_{j \in S} w_j \;\middle|\; \sum_{j \in S} r_j^u \leq F^u \;\wedge\; \sum_{j \in S} r_j^d \leq F^d \right\}$$

We also fill in other entries: $\mathrm{R}(v, 0)[F^u, F^d, m^x, m^r] = \mathrm{R}(v, 0)[F^u, F^d, 0, 0]$, for $0 < m^x, m^r \leq M$.

Inductive Step. Suppose that for v we have computed the entries $\mathrm{R}(v, i-1)$ and $\mathrm{R}(u_i, n(u_i))$, where u_i denotes the i-th child of v. We use F_j^u, F_j^d, m_j^x, and m_j^r to denote each of those entries, where we use $j = 1$ for $\mathrm{R}(v, i-1)$ and $j = 2$ for $\mathrm{R}(u_i, n(u_i))$. Then, we compute $\mathrm{R}(v, i)$ as follows:

$$\begin{aligned} \mathrm{R}(v, i)[F^u, F^d, m^x, m^r] = & \\ \max_{F_j^u, F_j^d, m_j^x, m_j^r; j=1,2} & \{\mathrm{R}(v, i-1)[F_1^u, F_1^d, m_1^x, m_1^r] \\ & + \mathrm{R}(u_i, n(u_i))[F_2^u, F_2^d, m_2^x, m_2^r]\} \end{aligned} \tag{12}$$

where the maximum is over all $F_j^u, F_j^d, m_j^x, m_j^r$ satisfying

$$F_1^u + F_2^u \leq F^u \tag{13}$$

$$m^{u_i \to v} = \left\lceil \frac{2F_2^u}{C(u_i, v)} \right\rceil \cdot M \tag{14}$$

$$m_2^x + m^{u_i \to v} \leq M \tag{15}$$

$$m^r \geq m_1^r + m^{u_i \to v} \tag{16}$$

In Constraint (14), $m^{u_i \to v}$ denotes the number of subchannels required for u_i to transmit its upstream traffic to its parent v. Note that we multiply the traffic amount by 2 because in the Even-Odd framework, an admitted flow i transmits at the rate of $2r_i$ every other time slot, where r_i is given from Eq. 11. Constraint (15) limits the number of subchannels used by u_i's transmission by M. Since v also needs to use $m^{u_i \to v}$ subchannels to receive the upstream traffic, we have Constraint (16). Similar to the upstream requirement, we can write the constraints for downstream traffic.

Implementation. For each parent-child pair, we have $(F_{max} M)^2$ entries, where F_{max} is the number of possible bandwidth values due to discretization. In a naıve implementation of the above algorithm, we would need to enumerate all possible values of the eight free variables $F_j^u, F_j^d, m_j^x, m_j^r$ to compute the value of each table entry. We can, however, reduce the number of free variables from eight to two (F_2^u and F_2^d). This is because all the intermediate tables are non-decreasing in all coordinates (increasing any resource coordinate can only increase the objective function value), and hence we only need to consider configurations in which all the inequality constraints in (13)–(16) are satisfied with equality. Thus, for each entry, we need to do $(F_{max})^2$ work. As a result, the total running time is $O(n(F_{max})^4 M^2)$, where n is the total number of nodes.

D. Experiment Results

We have performed simulation experiments to evaluate the performance of the approaches described in Sections III-A and III-C. In our experiments, we use a single gateway located in the middle of 5km-by-5km square, where other nodes are placed uniformly at random. We assume the fixed wireless pathloss model [5] and a fixed transmit power at each node

	BW (in Kbps)		Delay (in ms)		Reward	Fraction
	Up	Down	Up	Down		
Interactive Audio	500	500	15	15	10	25%
Interactive Video	500	500	15	15	10	25%
Streaming Audio	0	500	–	40	4	20%
Streaming Video	0	500	–	40	4	20%
Data	0	500	–	100	3	10%

TABLE I

BANDWIDTH (BW) AND DELAY REQUIREMENTS, REWARD, AND FRACTION OF EACH TRAFFIC TYPE

Number of nodes	Feasible ILP Solution	SP-order	Depth-order	MST	Star
8	26.7	22.0	19.0	20.0	22.0
9	26.7	24.7	18.0	21.7	26.3
10	38.0	26.7	19.7	26.3	29.3

TABLE II

COMPARISON BETWEEN ILP SOLUTIONS AND HEURISTICS.

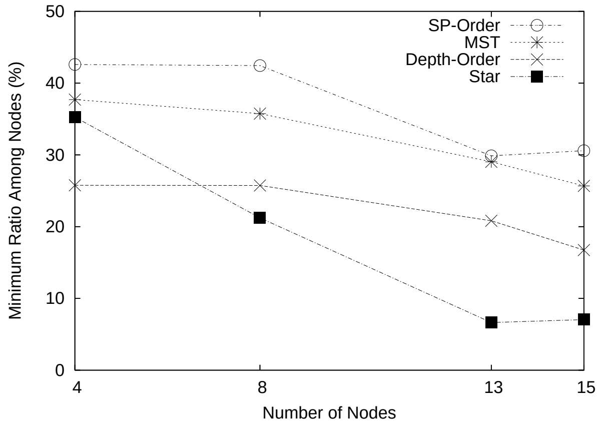

Fig. 1. Minimum $A(v)/T(v)$ ratio among nodes when we vary the number of nodes. The number of connections is fixed at 84.

(e.g., 36 dBm). Then, we can compute the resulting signal-to-noise ratio (SNR) at the receiver and translate this SNR into a bit rate for links between each pair of nodes.[2] We assume all packet sizes are 300 bytes. In our experiments, we consider a number of connection types and use the values specified in Table I. We randomly generate a number of scenarios with different node placement and connections assigned to each node.

In our first experiments, we compare our tree construction heuristics with the optimal ILP solutions when we try to maximize the total reward. Since finding an exact optimal solution for a large instance is time-consuming, we use small instances where there are up to 10 nodes each with 1 connection. We also assume that the maximum link bandwidth is 7.5 Mbps so that we can admit only a subset of connections. When finding an optimal solution, CPLEX tries to find a tighter upper bound in addition to improving a feasible integer solution. In our experiments, we stop a CPLEX run when the current feasible integer solution is within 25% of the upper bound.[3] We report the average of three experiments.

In Table II, we tabulate the performance of each scheme when we vary the number of nodes in the network. In this table, the performance of the tree construction heuristics are comparable to the optimal solutions. Interestingly, Star performs well in these experiments in which many connections have tight delay requirements. It is because we use the objective that does not consider the fairness among nodes. In this case, we can achieve reasonable performance by admitting all connections associated with only a few nodes close to the gateway while admitting no connections from other nodes. Although we do not present the results here, we also experimented with other scenarios (e.g., using different fractions for traffic types), and SP-order and MST heuristics usually performed better than Star, especially when delay requirements of user connections were less tight.

In our second set of results, we use larger networks and consider the second objective: maximizing $\min_v \frac{A(v)}{T(v)}$. The maximum bit rate in this experiment is 75 Mbps, and we use the averages from ten runs. In Figure 1, we plot the minimum $A(v)/T(v)$ ratios by different tree heuristics while varying the number of nodes. We observe that SP-order can achieve more than 30% of connection rewards for each node on average while MST and Depth-order can achieve 15–25% in the 15-node scenarios. In contrast, Star sometimes satisfies only 5–6% of rewards for some nodes. We can observe that as the number of nodes increases in these experiments, the performance of Star degrades significantly. It is because we cannot admit many connections associated with nodes that have poor direct link to the gateway. This result clearly shows the advantage of using multihop paths in wireless backhaul networks.

In the next section, we consider an alternative approach where we admit all requested connections with minimal performance degradation.

IV. RELAXING QOS REQUIREMENTS

In this section we assume each link reserves a fraction of its bandwidth for QoS sensitive connections (e.g. VoIP) whose delay and bandwidth requirements have to be met if admitted. We use heuristics and algorithms outlined in Section III for admission control of these connections. The remaining connections (e.g for file transfer and web browsing) are less sensitive to QoS. We admit all of them, possibly at a scaled down rate and a stretched end-to-end delay guarantee. In the following, we aim to provision as much bandwidth as possible for each connection while stretching its delay requirement as little as possible. We assume the tree topology is constructed already.

Let γ be the stretch factor of the delay requirement. Given γ a natural objective is to maximize λ, the fraction of the rate provisioned for each connection. This objective can be achieved optimally via linear programming. As in Section III-A we use superscript $z \in \{u, d\}$ to indicate the direction of a connection.

$$\max \lambda \quad \text{subject to:}$$

$$r_i^z \geq \lambda \rho_i^z \qquad \forall i, z \qquad (17)$$

$$D(\sigma_i^z, r_i^z, K_i^z) \leq \gamma D_i^z \qquad \forall i, z \qquad (18)$$

Admissibility condition (1) with

$$F(e) = \sum_{z,i:e \in P_i} r_i^z \qquad \forall v, z \qquad (19)$$

[2] In our experiments, capacities for links (u, v) and (v, u) are equal.

[3] We completed a few runs to find exact solutions. We observe that in those cases, the feasible solutions within the 25% error bound were actually optimal integer solutions after CPLEX further improved the upper bounds. Although we use the 25% error bound, we believe the integer solutions are closer to the actual optimal solutions.

Note that for a fixed λ, minimizing γ with constraints (17)-(19) is not a linear program. However, for γ from a desired range, we can carry out binary search on γ, optimize for λ for each fixed γ until we find an acceptable λ. Typically, a larger delay stretch of γ allows a larger common fraction, λ, of bandwidth from each connection to be admitted. To see this, for a connection i with a stringent delay requirement, it requires sufficiently large admitted connection rate r_i. Therefore, when γ gets smaller we admit a smaller common fraction of connection rate from each connection. This leaves excess link bandwidth that can be distributed among connections that require extra bandwidth to meet the delay requirement. On the one extreme, when γ gets sufficient small the necessary admitted rate to satisfy the delay requirement (18) can no longer satisfy the admissibility condition (19). On the other extreme, when γ gets sufficiently large, the admissibility condition becomes the sole bottleneck and the delay requirement is no longer constraining. Therefore, there is a certain tradeoff between λ and γ.

Similarly we can also optimize the following objectives, $\max \sum_{i,z} r_i^z/\rho_i^z$, $\max \sum_{i,z} r_i^z$ and $\max \min_{i,z} r_i^z$. They all correspond to linear programs and therefore can be solved efficiently. In addition, we can maximize $\sum_{i,z} \log r_i^z$. Since the objective is a concave function and the constraints are linear, the objective can be optimized efficiently [6]. Note that maximizing $\sum_{i,z} \log r_i^z$ achieves proportional fairness, see e.g. [14].

Theorem 3: Given γ, we can provision a connection rate of r_i^z and delay guarantee of γD_i^z for each connection i while maximizing any of the following objectives, $\min_{i,z} r_i^z/\rho_i^z$, $\sum_{i,z} r_i^z/\rho_i^z$, $\sum_{i,z} r_i^z$, $\min_{i,z} r_i^z$ and $\sum_{i,z} \log r_i^z$.

V. Discussion and Future Work

We have addressed admission control schemes when the nodes (customer locations and a gateway) and connections are given a priori. We have first considered the scenario where we strictly guarantee all the requirements of admitted connections. We also consider how to relax the rate and delay requirements while we admit all user connections.

The work can be extended to decide dynamically if a newly arrived connection can be admitted without violating its QoS requirements and without degrading the QoS of the already admitted connections.

The solutions presented for tree construction can also be extended to deal with the problem of cross interference. That is, the ILP and the various heuristics can be augmented so that any tree they produce will not have interfering links active at the same time in the Even-Odd framework.

While the work discussed in this paper was restricted to a tree topology, we are considering various generalizations. For example, we could have a number of overlapping trees each rooted at different gateways or even more general topologies. Also it would be interesting to study the problem of adding "relay nodes" that do not originate traffic but can relieve congestion. However many of these generalizations result in even more complex problems. For example, routing connections becomes a challenge in the presence of multiple potential paths and delay requirements, whether or not splitting paths is allowed.

References

[1] S. Abedi. Improved stability of qos provisioning for 3g systems and beyond: optimum and automatic strategy selection for packet schedulers. In *IEEE Int. Conf. Communications*, 2004.

[2] M. Andrews and L. Zhang. Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule. In *Proc. of IEEE INFOCOM*, pages 380 – 388, 1999.

[3] V. Bharghavan, S. Lu, and T. Nandagopal. Fair queueing in wireless networks: Issues and approaches. *IEEE Pers. Comm.*, 1999.

[4] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *ACM SIGCOMM*, 19(4):2–12, 1989.

[5] V. Erceg and K.V.S. Hari et al. Channel models for fixed wireless applications. *IEEE 802.16 Broadband Wireless Access Working Group*, IEEE 802.16.3c-01/29r4, 2001. http://ieee802.org/16.

[6] M. Grotschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1993.

[7] S. Jamin. *A measurement based admission control algorithm for integrated services packet network*. PhD thesis, USC, 1996.

[8] Sugih Jamin, Peter B. Danzig, Scott J. Shenker, and Lixia Zhang. A measurement-based admission control algorithm for integrated service packet networks. *IEEE/ACM Trans. Netw.*, 5(1):56–70, 1997.

[9] Z. Jiang, L. Chang, and N. K. Shankaranarayanan. Channel quality dependent scheduling for flexible wireless resource control. In *Proc. GLOBECOM*, 2001.

[10] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly. Distributed multi-hop scheduling and medium access with delay and throughput constraints. In *MobiCom '01: Proc. 7th Int. Conf. on Mobile computing and networking*, 2001.

[11] I. Koffman and V.Roman. Broadband wireless access solutions based on OFDM access in IEEE 802.16. *IEEE Comm. Mag.*, 40(4):96 – 103, 2002.

[12] K. Kumaran, G. E. Margrave, D. Mitra, and K. R. Stanley. Novel techniques for the design and control of generalized processor sharing schedulers for multiple QoS classes. In *Proc. IEEE Infocom*, 2000.

[13] C. Li and E. Knightly. Coordinated network scheduling: A framework for end-to-end services. In *Proc. of IEEE ICNP*, 2000.

[14] R. Maxumdar, L. Mason, and C. Douligeris. Fairness in network optimal flow control: Optimaility of product form. *IEEE Trans. on Comm.*, 39:775 – 782, 1991.

[15] G. Narlikar, G. Wilfong, and L. Zhang. Designing multihop wireless backhaul networks with delay guarantees. *Bell Labs Technical Memorandum*, August 2005. http://www.somepage.

[16] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Trans. on Networking*, 1(3):344 – 357, 1993.

[17] A. K. Parekh and R. G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: The multiple-node case. *IEEE/ACM Trans. on Networking*, 2(2):137 – 150, 1994.

[18] R. Srinivasan and J.S. Baras. Understanding the trade-off between multiuser diversity gain and delay - an analytical approach. In *IEEE Vehicular Technology Conference*, volume 5, 2004.

[19] S. Sriram, T. B. Reddy, B. S. Manoj, and C. S. R. Murthy. On the end-to-end call acceptance and the possibility of deterministic qos guarantees in ad hoc wireless networks. In *MobiHoc: Proc. 6th ACM Int. Symp. Mobile ad hoc networking and computing*, pages 169–180, 2005.

[20] R. Szabo, P. Barta, F. Nemeth, J. Biro, and C-G Perntz. Call admission control in generalized processor sharing (GPS) schedulers using non-rate proportional weighting of sessions. In *Proc. of IEEE Infocom*, 2000.

[21] J. Tang, G. Xue, and W. Zhang. Interference-aware topology control and qos routing in multi-channel wireless mesh networks. In *MobiHoc: Proc. 6th ACM Int. Symp. Mobile ad hoc networking and computing*, pages 68–77, 2005.

[22] D. Wu and R.Negi. Downlink scheduling in a cellular network for quality-of-service assurance. In *IEEE Trans. on Vehicular Technology*, volume 53, 2004.