FLOW CONTROL FOR END-TO-END DELAY AND POWER CONSTRAINED WIRELESS MULTIHOP NETWORKS

Jennifer C. Fang and Ramesh R. Rao
University of California, San Diego, La Jolla, CA, USA.

ABSTRACT

In this paper, we solve the resource allocation problem of minimizing the total average power consumption for end-to-end delay constrained traffic in a multihop wireless network comprised of links with time-varying (Markov-modeled) wireless channels. We are given a set of source-destination pairs. Each source may use multiple routes to transport traffic to the destination at specified data rates. The traffic transported on each route between each src-dst pair is subject to an end-to-end delay guarantee. We present a 2-tier hierarchical solution to solve the above problem. At the bottom tier, each link transmits packets to minimize its long-term average power subject to long-term average delay constraints [4]. Given this packet transmission policy at every link and the associated energy-delay and energy-rate trade-off relations, we perform a network-wide optimization of traffic flows at the top tier. This problem is framed as a non-differentiable convex optimization problem and solved using an incremental sub-gradient optimization technique. We implement our algorithm over sample network topologies and compare its performance with alternate algorithms, noting significant insight of the flow allocation policy as well as substantial gains in energy efficiency and throughput.

I. INTRODUCTION

The large-scale deployment of IEEE 802.11a/b/g technologies has provided cheap access to the Internet wirelessly. However, currently deployed IEEE 802.11b networks are not capable of supporting constant bit rate applications like Voice-over-IP services, internet audio and video etc., despite of using multiple wireless hops [5]. Battery powered wireless devices are energy constrained and many applications they run are delay-limited. In this paper, we study the trade-off between conserving energy and minimizing delay in a wireless multihop network comprised of links with time-varying channel conditions.

Thus far, transmission policies for minimizing average power consumption have been determined only for a single link [1][4][6]. In [6], Goldsmith et. al. showed that the optimal transmission policy for a single link is to transmit at high powers when the channel quality is “good” and remain silent when the channel quality is poor. Transmissions over poor channels are only required when the desired data rates are relatively high. A similar problem was solved in [1][4]. Specifically, these papers solve the problem of minimizing the average power subject to an average delay constraint for a single wireless link by using a dynamic programming (DP) approach combined with duality arguments. Their policy is a simple function of the link’s backlog and the channel fade state and can be expressed in closed-form.

However, the DP approach is not suited even for a two-link serial network due to the sharp increase in state space for the DP algorithm. The increase in state space is attributed to the dependence of a link’s packet transmission policy on its own channel quality and backlog as well as the channel quality and backlog of its neighboring links [4]. In the simple case of a K-link serial network with a backlog of 100 packets and 2 channel states, the state space is given by a $(2 \times 100)^K$, where each state of the system requires the computation of the cost-to-go function. In addition, this line of approach is impractical in real-life wireless networks, because each link requires global state information, i.e., the channel state and backlog of other links along a route at the start of a slot. Even if such colossal amount of state information was available at the beginning of a slot, it is likely to be stale over fast-fading channels.

We circumvent the above problems encountered in a monolithic DP approach by using a 2-tier hierarchical approach. At the bottom tier of network operations, each link implements the optimal single-link transmission policy as exposted in [4]. Given this packet transmission policy at the bottom tier and the associated energy-delay and energy-rate trade-off curves, we perform a network-wide optimization of traffic flows at the top tier. This problem is framed as a non-differentiable convex optimization problem and solved using an incremental subgradient technique. This technique is an iterative algorithm that computes the optimal flow allocation along multiple routes for each src-dst pair as well as the delay along links in each path so as to meet the end-to-end average delay requirement. Each step in the iteration decrements the average power consumption, by shifting flows over to “shortest-paths,” and adjusting average delays along re-
spective links.

Our hierarchical approach has two major advantages over the monolithic DP approach. Firstly, it scales well with size of the wireless network, since each link in the network does not require knowledge of the global state information (i.e., backlog, fade state of all links). Secondly, the complexity of the iterative flow and delay adjustment is relatively low and is amenable to distributed implementation.

In section II, we describe our system model and set of assumptions. In section III, we formulate our problem. We solve the proposed problem using duality in section IV. We simulate our algorithm over a triple diamond topology in section V and elicit some rather insightful observations.

II. SYSTEM MODEL

Consider a network of $N$ stationary, randomly distributed nodes interconnected with each other via a set of $L$ wireless links. In our model, each of these $L$ links is modeled in the same manner as in [4]. Due to space limitations, we will not duplicate the description of modeling a single link.

In [4], Collins and Cruz derive a simple transmission policy for a single link as a function of its backlog and fade state. Their transmission policy (see eq. (13) in [4]) minimizes the average power required to transmit information generated at the link at a given average rate, when information transmitted is given an average delay guarantee. They plot the link's energy-delay response for a given arrival rate. As expected, the energy consumption is convex and decreasing in the average delay, and is proven in [1]. Similarly, we can also plot the energy-rate curves for a given delay. Again, as expected, the energy consumption is convex and increasing in the average rate. We approximate each response curve using piecewise linear functions to a desired degree of precision by setting the maximum difference between the actual response curve and the modeling curve to $\epsilon$. Let $H_l(D_l, w_l)$ denote the minimum long-term average power consumption of a single link $l$ for a given average delay $D_l$ and average arrival rate $w_l$ of packets. Thus, we have the bivariate function $H_l(D_l, w_l)$ in closed form for each link $l$ that is convex in $D_l$ and $w_l$. Each link precomputes these functions for many values of $D_l$ and $w_l$ and stores them.

For the multihop networks, we assume that transmissions along one link do not interfere with transmissions along other links that are simultaneously active. Such interference-free transmissions along links can be achieved by using FDMA or OFDM frequency (hopping) allocation among links or by using orthogonal codes in CDMA.

Each node in the network can relay information for other nodes, or it can be a source of information which must be transmitted to its corresponding destination. Denote a src-dst pair by a 2-tuple $\sigma$, where $\sigma = \{s, d\}$. Let $S$ denote the set of all src-dst pairs. Let $Y_\sigma$ be the average rate of packet generation at the source of a src-dst pair $\sigma$. We assume that the source nodes generate packets according to a known probability distribution (e.g., exponential on-off or Bernoulli on-off). Packets generated at a source are allowed to be transmitted on one or more routes to their destination, and each route may span multiple hops employing other nodes in the network to act as relays. We assume that the set of allowed routes for each src-dst pair have been precomputed.

The data transmitted on each link incident on an intermediate transmitter is shaped by an $E_l$-shaper. An $E_l$-shaper is a network element with an infinite buffer that serves packets in its buffer at a constant rate of $E_l$ packets/sec [8]. Fig. 1 depicts a sample network with $E_l$-shapers placed at each intermediate transmitter. The $E_l$-shaper then outputs to link $l$'s buffer, which subsequently transmits over the wireless channel.

A. THE NECESSITY OF TRAFFIC MODELING AND $E_l$-SHAPERS

For a multihop network, the departure process of one link is the arrival process of another. Recall that the optimal transmission policy of each link in a slot depends on its backlog and the current fading state of its outgoing link [4]. Hence, while the traffic is generated at the source at a constant rate, the traffic entering the node upstream is bursty due to the Markovian nature of the channel. Burstiness of traffic at intermediate nodes has traditionally been a problem encountered in almost every G/G/1 queuing system, barring some simple ones (e.g., M/M/1, M/D/1). In order to study the end-to-end performance of hard-to-characterize stochastic arrivals at intermediate nodes, the customary approach is to model the departure process of a link by an analytically tractable process that captures or bounds the statistics of the actual departure process to a reasonable accuracy.

In our framework, the traffic departing from a link is modeled by a Geometric ON-OFF process with the same average arrival rate as the actual departure process. The average delay experienced by the ON-OFF process at the node upstream is
higher than the average delay experienced by the actual departure process and is hence an upper bound to the actual arrival process.

An ON-OFF arrival process with a big difference in the peak and average rates entering the node upstream can result in excessive power consumption due to the sporadic nature of arrivals. A relatively smoother traffic process on the other hand results in a steady evolution of the backlog and thereby lowers power consumption\(^2\). This motivates the use of a traffic shaper at the ingress point of each link. The advantage of using a traffic shaper are manifold. First of all, a traffic shaper would be ideal for multihop networks where nodes would like to filter traffic coming from other nodes (e.g. firewalling). Another major advantage of using a shaper is that it acts as a shock absorber by smoothening out highly bursty traffic at intermediate nodes. This not only reduces the power consumption at intermediate nodes but also allows for analytic tractability of the problem since the traffic process entering each link’s buffer is well-characterizable and its power-delay trade-off curves can be employed in the network optimization. The only downside of using a traffic shaper is that it imposes additional delay in the end-to-end flow of traffic. However, we have found through simulations that this additional delay incurred is less than the delay incurred when letting bursty traffic directly into the link’s buffer.

In our framework, the traffic departing an \(E_l\)-shaper and entering the link’s buffer is modeled precisely by a Geometric On-Off random process with peak rate \(E_l\). The first \((I)\) and second \((\bar{I}^2)\) moment of the active time of the \(E_l\)-shaper can be characterized precisely and is given by equation (5) in [7]:

\[
\overline{I}_l = \frac{K_l \omega_l}{\pi_l} \frac{(\pi_l - \omega_l)}{(E_l - \omega_l)} \quad \text{and} \quad \overline{I}^2_l = \frac{K_l E_l}{\pi_l} \frac{(\pi_l - \omega_l)^3}{(E_l - \omega_l)^3} \frac{K_l w^2}{\pi^2}.
\]

Utilizing the DP equations 9–10 in [4] and the traffic characteristics described above, each link computes its optimal transmission policy by calculating the average cost-to-go function evaluated at its buffer. Note that as a special case, for zero delay across the shaper, we can set \(E_l = \pi_l\).

III. PROBLEM DEFINITION

As stated in the previous section and illustrated in Fig. 1, we assume that all traffic entering a node and destined on link \(l\) is shaped by an \(E_l\)-shaper. Let \(\omega_l^p\) be the average rate of traffic flow of src-dst pair \(\sigma\) over link \(l\). Let \(\Omega(l)\) denote the set of all src-dst pairs whose traffic is carried on link \(l\). Clearly, \(\sum_{\sigma \in \Omega(l)} \omega_l^p = \omega_l\) for each link \(l \in L\).

Since many src-dst pairs may share a common link to transmit their data, we must distinguish between the delays experienced by each src-dst pair’s stream. Let \(D_l\) be the average queuing delay experienced by packets in link \(l\)’s buffer, and let \(D_l^\sigma\) be the average queuing delay experienced by packets belonging to src-dst pair \(\sigma\) at link \(l\)’s buffer. Similarly, define \(\Delta_l\) as the average delay at the traffic shaper ahead of link \(l\); and define \(\Delta_l^\sigma\) as the average delay experienced by packets belonging solely to src-dst pair \(\sigma\) at the shaper ahead of link \(l\).

Denote \(\pi_l\) as the peak rate of the traffic on link \(l\), i.e., \(\pi_l = P_{\text{max}}(l)\). Let \(\phi(\sigma)\) be the set of allowable routes for src-dst pair \(\sigma\), and \(E(l)\) the set of all links incident on \(l\). Denote \(\gamma_{\text{set}}(l)\) as the prespecified end-to-end average delay guarantee for each src-dst pair \(\sigma\).

Then the average delay \(\Delta_l\) experienced at the \(E_l\)-shaper by a Geometric ON-OFF arrival process with average rate \(\omega_l\) and peak rate \(\pi_l\) is given by \(\Delta_l\), where

\[
\Delta_l = \frac{2K_l \omega_l}{(E_l - \omega_l)^2} \frac{(\pi_l - E_l)(\pi_l - \omega_l)}{(\pi_l)^2},
\]

and \(K_l = \frac{1}{\lambda_{XY}(l)} + \frac{1}{\lambda_{CB}(l)}\) is a constant for link \(l\) (\(\lambda_{XY}(l)\) is the probability of the channel transitioning from state \(X\) to \(Y\) of link \(l\)). In our formulation, we assume that the delay across the shaper \(\Delta_l\) is set to a predetermined value (\(\Delta_l(0)\)).

Our approach of assigning a delay budget across the traffic shaper is similar to the approach in [8]. Let \(Y_\sigma\) denote the average packet generation rate at the source of src-dst pair \(\sigma\), which is known, and \(x_\sigma^p\) as the average amount of traffic along route \(r\) for src-dst pair \(\sigma\). The stability relation ensures that \(\sum_{r \in \phi(\sigma)} x_\sigma^p = Y_\sigma\). Also, \(w_l = \sum_{\sigma \in S} \sum_{l \in r \in \phi(\sigma)} x_\sigma^p\).

Our objective is to minimize the total average power consumption in order to support a given end-to-end traffic flow between a prespecified set of src-dst pairs subject to end-to-end average delay constraints per src-dst pair. Mathematically, we seek to

\[
\min \left\{ \sum_{l=1}^{L} H_l(D_l, \omega_l) \right\}, \quad \text{subject to}
\]

\[
\sum_{l \in \phi} D_l^\sigma + \Delta_l^\sigma \leq \gamma_{\text{set}}(l) \forall r \in \phi(\sigma), \sigma \in S,\quad (3)
\]

\[
\sum_{m \in E(l)} \sum_{\sigma \in \Omega(m)} w_m^p D_l^\sigma = w_l D_l \quad \forall l \in L, \quad (4)
\]

\[
\sum_{\sigma \in \Omega(l)} w_l^\sigma \Delta_l^\sigma = \omega_l \Delta_l(0) \quad \forall l \in L, \quad (5)
\]

\[
0 \leq w_l \leq w_l^{\text{max}}.
\]
By solving problem (2)–(6), we find the optimal flow allocation \( x^*_\sigma \) along route \( r \) for src-dst pair \( \sigma \) and the optimal delay allocation \( D^*_r \). Constraint (3) ensures that the end-to-end average delay for each traffic flow along every route \( r \) is bounded by \( D^*_{\text{tot}} \), which is presupposed for each src-dst pair \( \sigma \). The average delay of each src-dst pair’s data stream is related to the average delay of a typical packet incurred at the shaper by Little’s formula (5), and is likewise related to the average delay incurred at the link’s buffer (4). Constraint (6) ensures the feasibility of the achievable rates along links.

IV. CONVEX DUALITY APPROACH

We solve the convex optimization problem (2)–(6) using a convex duality approach. Define a dual variable \( \alpha^\sigma \) per constraint, the dual function \( Q(\alpha) \) can be simply written as

\[
Q(\alpha) = \min_{D^*_r, w_l} \left\{ \sum_{l=1}^{L} H_l(D_l, w_l) + \sum_{\sigma \in S} \sum_{r \in \psi(\sigma)} \alpha^\sigma \left( \sum_{i \in \mathcal{E}} D^*_i + \Delta^*_i - D^*_{\text{tot}} \right) \right\},
\]

where \( D_l = \frac{1}{w_l} \sum_{m \in \mathcal{E}(l)} \sum_{\sigma \in \Omega(m)} w^\sigma_m D^*_m \).

Notice here that \( \Delta^*_i \) is a slack variable in the optimization problem. For simplicity of exposition, we shall ignore this term (and constraint), although it is relatively straightforward to include it in the optimization problem. The dual problem is stated as

\[
\max Q(\alpha) \quad \text{subject to} \quad \alpha \geq 0.
\]

Evaluation of the dual function for a given \( \alpha \) can be done in an unconstrained manner. Since it involves a nondifferentiable cost function \( H_l(D_l, w_l) \), we use an iterative sub-gradient based approach to evaluate the dual function. Each iteration is broken up in 3 phases, which take place in sequence. In phase A of an iteration, the flow allocation along links in the network is kept constant. Each link determines its delay \( D^*_l \) that would minimize the cost function for fixed values of \( \alpha \) and \( \bar{w} \). The value \( D^*_l(k) \) in iteration \( k \) can be obtained by letting each link \( l \) compute

\[
\min_{D^*_l} \{ H_l(D_l, w_l) + \sum_{\sigma \in S} \sum_{r \in \psi(\sigma)} \alpha^\sigma \Delta^*_l \},
\]

where \( D_l = \frac{1}{w_l} \sum_{m \in \mathcal{E}(l)} \sum_{\sigma \in \Omega(m)} w^\sigma_m D^*_m \).

Determination of \( D^*_l(k) \) is relatively straightforward and is done using the incremental sub-gradient method detailed in [3]. This process involves computing the sub-gradient of each component function sequentially and taking steps along each component subgradient with intermediate adjustment of variables after processing each component function. It is easy to see that for a given value of \( H_l(D^*_l(\bar{w}), w_l) \), \( D^*_l \) is a convex and increasing function of \( w_l \). Hence, \( H_l(D^*_l(\bar{w}), w_l) \) is convex and increasing in \( w_l \).

In phase B of each iteration, the flow allocation along links is adjusted in a way that results in a decrease in the cost function. Notice that this sub-problem is similar to the problem in [2], except for the piecewise linearity of the cost function w.r.t. the link rates. For a given \( \alpha \geq 0 \), Bertsekas et al. show that shifting traffic from “longer” routes to “shorter” routes decreases the convex cost function. In the optimal policy, all traffic for each src-dst pair is exclusively transported over shortest paths.

The differences between our problem and [2] arise from the piecewise linearity of our problem, the cost function. Moreover, in our problem, the cost function is comprised of a sum of bivariate functions where both variables are convex functions of \( x^*_\sigma \). Owing to the piecewise linearity of our cost function, the approach in [2] is not directly applicable, as the Hessian matrix is not well defined. However, for this non-differentiable problem we use a similar approach called the incremental sub-gradient technique [3], to decrease the cost function in each iteration. This flow adjustment step can also be interpreted as shifting traffic from “longer” paths to “shortest” paths, where the weight of each link in the graph is given by \( \eta(l) \),

\[
\eta(l) = \frac{\partial H_l(D^*_l(\bar{w}), w_l)}{\partial D^*_l} \frac{\partial D^*_l}{\partial w_l} + \frac{\partial H_l(D^*_l(\bar{w}), w_l)}{\partial w_l},
\]

the partials being computed using the sub-gradient approach. Further, denoting the cost of path \( r \in \psi(\sigma) \) as \( x_r \), where \( x_r = \sum \eta(l) \), the flows along each path are adjusted as

\[
x^*_r(k + 1) = x^*_r(k) + \sum_{r \in \psi(\sigma)} \frac{\eta(l)}{\eta(r)} (x_r - x^*_r) \quad \forall r \neq r^*,
\]

where \( r^* \) denote shortest paths.

Having evaluated the dual function for a given value of \( \alpha \geq 0 \), we proceed to phase C, the incremental maximization of \( Q(\alpha) \), by simple adjustments to \( \alpha \). In this step the dual vector

\[
H_l(D^*_l(\bar{w}), w_l) \quad \text{is convex and increasing in} \quad w_l \quad \text{and therefore convex in} \quad x^*_\sigma.
\]

\footnote{"H_l(D^*_l(\bar{w}), w_l) \quad \text{is convex and increasing in} \quad w_l \quad \text{and therefore convex in} \quad x^*_\sigma."}
where $\Delta_{\alpha}$ is the step size. Paths that violate their respective end-to-end delay constraints increase their $\alpha^e_k$ by an amount proportional to their delay violation. Since each step of the iteration decrements the cost function and eventually converges to the globally optimal solution of our convex optimization problem. Proof of convergence of our algorithm to the optimal solution is omitted due to page limitations.

V. NUMERICAL EXPERIMENTS

Let us consider a randomly generated network depicted in Fig. 2(a). Three source nodes transmit at identical average data rates to a common “Access Point,” via a predetermined set of routes. For each link, $P_{\text{max}} = 200\text{mW}$. The probability transition matrix of each link is $\lambda_{GC} = 0.9$, $\lambda_{GB} = 0.1$, $\lambda_{BC} = 0.2$, and $\lambda_{BG} = 0.8$. The path gain of a link is modeled as $1/d^4$, where $d$ is the distance between the transmitter and the receiver of the link. The fade level of each link is the path gain of that link normalized by the ambient noise. The fade level of the “good” channel for each link is ten times better than the “bad” channel for all links, i.e., $s^G = 10s^B$. The average shaper delay budget is 4 slots.

Fixing a traffic demand, the iterative algorithm is run to obtain the average network power consumption for various values of end-to-end delay, which is set to be identical for all src-dst pairs. The results are shown in Fig. 3(a) for traffic demands of 5, 6, 7 and 8 Mbps.

The optimal flow allocations on each route indicate that under stringent end-to-end delay guarantees, sources prefer routes with smaller number of hops. This behavior of the routing policy is attributed to the convex and decreasing nature of the energy-delay curve for a single link. At low end-to-end delays, link delays on longer routes are more stringent due to number of hops and the shaper delay at each hop. In our case, although routes R7, R8 and R9 all serve source 3, in order to meet the low link delays, nodes on R7 and R9 must transmit at very high powers. R8 on the other hand uses fewer hops and carries a significant amount of traffic from source 3. Link L4 is then forced to become a bottleneck link because it must carry data for source 3 along with sources 1 and 2. However, when the end-to-end delay guarantee is more relaxed, the shaper delay becomes relatively insignificant. More traffic from source 3 is shifted to R7 and R9, consequently reducing the strain on bottleneck link L4. As a result, the network power consumption is decreased substantially. Our results suggest that providing delay guarantees over longer routes tends to be more difficult, because of the increased burstiness of the traffic at each of the intermediate hops.

A. VARYING CHANNEL MEMORY

The channel memory, defined as the average time that the channel spends in each state, is equal to $1/\lambda_{GB}$. Keeping all parameters constant as in the previous experiment and decreasing the memories of links L8 and L12, Fig. 3(b) shows that the average network power consumption is decreased.

Under stringent end-to-end delay guarantees, average network power consumption is similar to the previous experiment with identical memory links. However, when end-to-end delay is more relaxed, the network power consumption decreases significantly. This is especially evident when the traffic demand of each source is high (8 Mbps). This is attributed to the fact that lower-memory links (i.e., faster fading) consume less average power given identical delay allowance than the higher memory links. By decreasing the channel memories of L8 and L12, the longer routes R7 and R9 become more attractive and carry sizeable traffic, lowering the load on R8 and thus relieving the bottleneck link L4. The average power consumption is reduced significantly. This experiment suggests that longer routes are energy optimal for delay constrained traffic, provide that their links have lower channel memories.

B. COMPARISON WITH BASE POLICY

Finally, using the parameters of experiment 2, we compare the average network power consumption expended by the optimal policy with a base policy. The base policy uses the shortest route for each src-dst pair, i.e., source 1 uses R2, source 2 uses R5, and source 3 uses R8. All sources transmit at identical data rate. End-to-end delay is set to 18 slots for all src-dst pairs. The shaper delay is set to 4 slots. For each policy, we vary the traffic demand of each source. The network power consumption is plotted in Fig. 2(b).

For identical network throughput, our iterative algorithm provides savings in network power consumption. This is especially significant when the traffic demand is high. Since the total end-to-end delay is relatively high compared to the shaper delay, our algorithm takes advantage of the multiple routes and shifts traffic away from the bottleneck links and thereby reducing power consumption significantly.

In addition, the maximum supportable traffic demand using the base policy is limited to only 4 Mbps, while our algorithm can provide network throughput of up to four-fold. This suggests that using multiple routes per source not only increases the amount of traffic demand supportable in a network, it also greatly reduces the energy consumption to transport a given traffic demand compared to a single, shortest route policy.
(a) Network topology

(a) Same channel memory

(b) Single route vs. multiple routes

(b) Varying channel memory

Fig. 2. Triple diamond network and results

Fig. 3. Triple diamond network and results

VI. CONCLUSION

We solve the resource allocation problem of minimizing the total average power consumption for end-to-end delay constrained traffic in a multihop wireless network comprised of links with time-varying channels. Our 2-tier hierarchical approach scales well with the size of the wireless network and is amenable to distributed implementation due to low complexity of the iterative flow and delay adjustment. Our implementation over sample networks suggest that providing delay guarantees over longer routes tends to be more difficult, because of the burstiness of the traffic at intermediate hops. However, longer routes become more energy optimal for delay constrained traffic provide that their links have lower channel memories.

REFERENCES


