Cross Layer Adaptive Control for Wireless Mesh Networks

Michael J. Neely , Rahul Urgaonkar

Abstract— **This paper investigates optimal routing and adaptive scheduling in a wireless mesh network composed of mesh clients and mesh routers. The mesh clients are power constrained mobile nodes with relatively little knowledge of the overall network topology. The mesh routers are stationary wireless nodes with higher transmission rates and more capabilities. We develop a notion of *instantaneous capacity regions*, and construct algorithms for multi-hop routing and transmission scheduling that achieve network stability and fairness with respect to these regions. The algorithms are shown to operate under arbitrary client mobility models (including non-ergodic models with non-repeatable events), and provide analytical delay guarantees that are independent of the timescales of the mobility process. Our control strategies apply techniques of backpressure, shortest path routing, and Lyapunov optimization.**

Index Terms— **Stochastic Optimization, Queueing Analysis, Mobility, Routing, Dynamic Control, Scheduling, Diversity**

I. Introduction

This paper develops optimal algorithms for transmission scheduling, routing, and flow control in an ad-hoc wireless mesh network. Mesh networks can be viewed as special cases of ad-hoc networks where the architecture is augmented with a collection of special high capacity nodes, called *mesh routers*, that facilitate communication (see [1] for a general survey on mesh networking). All other nodes are *mesh clients*, as they are expected to use the network of mesh routers to deliver their data to the proper destinations. We assume that mesh clients have peak and average power constraints, have general mobility patterns, and have little knowledge of the network topology beyond their immediate set of current neighbors. The mesh routers are stationary wireless nodes distributed throughout the network area. These nodes are typically more powerful than the mesh clients, and are responsible for keeping track of mobile client locations and for performing operations of routing and packet delivery.

We assume the system operates in slotted time. Every timeslot, data exogenously arrives to the clients in packetized form, and all packets are assumed to have the same bit length. Each packet must be delivered to its own desired destination client. In the most basic model, the source clients transmit their data to the nearest mesh router, and the mesh routers send the data over multi-hop paths to reach a mesh router node that is currently in range of the destination client. We assume that each mesh router has two radios that operate simultaneously over independent channels. The first radio is used for uplink (client-to-router) and downlink (router-to-client) communication, while the second is used for router-to-router communication. Thus, the system of router-to-router communication links can be viewed as an independent backbone network that supports multi-hop data transfer, but where the destination of each packet in this backbone is potentially changing over time. We develop a simple dynamic routing algorithm that combines backpressure and differential backlog concepts of [2] [3] together with shortest path routing. The algorithm works in conjunction with flow control and scheduling algorithms implemented at the client nodes, and maintains low network delay while ensuring network stability and fairness under general client mobility models. In particular, we develop a notion of *instantaneous capacity regions* and show that throughput and fairness guarantees can be achieved with respect to these regions while maintaining end-to-end average delay that is independent of the timescales of the client mobility process.

To model the uplink and downlink transmissions, we consider the situation where each mesh router covers a designated (and non-overlapping) "cell region" of the network area. Each mesh client is assumed to be within exactly one of these cell regions at any given timeslot, and can only transmit to (or receive from) the corresponding mesh router for that cell. Each transmitting node chooses its *transmission rate* (i.e., the number of packets to be transmitted), and each packet transmission is received correctly with some success probability. We consider both channel-aware and channel-blind modes of operation. For channel-aware scheduling, current channel states are measured and the success probability for each transmission rate is assumed to be known. These probabilities are unknown in channel-blind scheduling mode. The router nodes coordinate the scheduling decisions in their respective cells. To simplify implementation, we further assume that inter-cell interference is negligible. This assumption is valid, for example, when transmissions are designed so that signals in neighboring cells are orthogonal.

We also briefly discuss extensions of this model, such as when the cell structure is removed and packets transmitted by clients can be overheard by potentially more than one mesh router. In this case, it is useful to exploit the concept of *multi-receiver diversity*, where the probability of successful reception in at least one node within a set of potential receivers can be significantly larger than the success probability of a particular receiver that is specified in advance. This feature is especially useful for maintaining high throughput in channel-

Michael J. Neely and Rahul Urgaonkar are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089. (website: http://www-rcf.usc.edu/∼mjneely)

This work is supported in part by one or more of the following: the National Science Foundation grant OCE 0520324, the DARPA IT-MANET program grant W911NF-07-1-0028.

blind scheduling mode. Finally, we briefly consider hybrid mesh network models that support client-to-client communication in cases where some clients are not within range of *any* mesh router. In this case, the clients must relay their data through other clients in order to reach the mesh routers.

The analysis and design methodology of this paper is based on adaptive queueing and Lyapunov optimization techniques. Such techniques were perhaps first applied to wireless networks in the landmark paper [2] by Tassiulas and Ephremides, where Lyapunov drift is used to develop a joint optimal routing and scheduling algorithm. The Tassiulas-Ephremides algorithm introduced techniques of backpressure routing and maximum weight link scheduling, and was shown to yield network stability whenever input rates are within the network capacity region. Lyapunov drift has since become a powerful technique for the development of stable scheduling strategies for satellite and wireless systems [4] [5] [6] [7], computer networks and switches [8] [9] [10], and ad-hoc mobile networks [3]. Methods for joint stability and utility optimization via Lyapunov drift are developed for stochastic networks in [11] [12] [13] [14] for application to flow control and energy minimization. Recent alternative approaches are developed in [15] [16] [17] using stochastic gradient theory and fluid model transformations. Lyapunov drift is applied to wireless networks with multi-receiver diversity in [18], where an optimal diversity backpressure routing algorithm (DIVBAR) is developed and shown to improve performance beyond that of related diversity algorithms that do not use backpresure, such as those in [19] [20] [21]. Alternative approaches to scheduling, routing, and flow control in wireline networks are developed in [22] [23] [24] [25] using linear and convex optimization theory. Related work for static wireless networks is developed in [26] [27] [28] [29] [30] [31] [32].

This paper applies the backpressure, Lyapunov networking, and multi-receiver diversity techniques described in the work above to the special case problem of a mesh network with mesh clients and routers. While these techniques were developed for more general scenarios, this paper is likely the first to use them directly within the mesh network paradigm. Further, there are several new aspects of this work that are not considered elsewhere, including the introduction of a time varying capacity region and the development of analytical stability and delay guarantees for mesh clients with arbitrary (possibly non-ergodic) mobility patterns.

A. Summary of Results

In the next two sections, we present the basic mesh network model and define the ergodic capacity region, the instantaneous capacity region, and the notion of throughput-utility. The flow control, routing, and scheduling algorithms are developed in Section IV and analyzed in Section V. We show that the algorithm always provides bounded congestion, and enables a tradeoff between congestion and throughput-utility that is affected by a control parameter V. The time average throughput-utility can be made arbitrarily close to (or larger than) the time average of the optimal throughput-utilities over the instantaneous capacity regions, with delay that is independent of the user mobility process. This holds for any mobility model, and implies that throughput approaches 100% in the special case when input rates are within the instantaneous capacity region on every timeslot. Further, when mobility is ergodic with a well defined steady state, the same algorithm can approach the optimal throughput-utility defined over the ergodic capacity region (which is a larger utility than that defined over the time average of instantaneous capacities). However, the penalty for approaching optimality in this case is a delay that depends on the mobility timescales. Extensions to channel-blind scheduling, multi-receiver diversity, and hybrid algorithms are briefly considered in Section VI. Example simulations are provided in Section VII.

II. Network Model and Capacity Region

Consider a network with N nodes, labeled $\{1, \ldots, N\}$. Let $\mathcal{C}$ represent the subset of *client nodes* (or "mesh clients"), and $\mathcal{R}$ represent the subset of *router nodes* (or "mesh routers"). For simplicity, we assume all nodes are either clients or routers, and cannot be both (so that $\mathcal{C} \cup \mathcal{R} = \{1, \ldots, N\}$, and $\mathcal{C} \cap \mathcal{R}$ is the empty set).

We assume that each router node $r \in \mathcal{R}$ has a fixed location within the network. Further, we assume that the network area is divided into disjoint *cell regions*, each containing a unique router node. The cell region of a given router node represents the area that this router is responsible for covering. Time is slotted with integral units $t \in \{0, 1, 2, \ldots\}$, and each client node $c \in \mathcal{C}$ is assumed to be located within a distinct cell for the duration of each timeslot. Let $r(c, t)$ represent the router node associated with the cell of client c during slot t (where $r(c, t) \in \mathcal{R}$). At the beginning of each timeslot, the router node of each cell makes a *client-router scheduling decision* according to the physical scheduling constraints in the cell. An example constraint is one where at most one transmission can take place in the cell per timeslot. In this case, the router node must select a single client within its cell and decide to either transmit data to this client (a *downlink operation*) or receive data from this client (an *uplink operation*). The router might also decide to remain idle in cases when no data is available or when it is desired to postpone transmissions to future slots.

At the end of the slot, clients either remain in their same cell or move to a new cell. The mobility process is arbitrary. For example, some nodes may be stationary, some nodes may take a simple random walk among one or more cells, other nodes might move according to a random waypoint model that is influenced by the locations of other nodes, etc. For each timeslot t, define $\mathcal{T}(t)$ as the *current client topology*, which represents the current cell locations of the clients.

Every timeslot the router nodes must also make *router-router* transmission decisions. For simplicity, we assume these operations can be performed simultaneously and independently of the uplink/downlink operations. This assumption is valid, for example, when routers are equipped with two radios that use independent channels. Each router node $r \in \mathcal{R}$ is assumed to have a subset Θ_r consisting of neighboring router nodes $m \in \mathcal{R}$ to which r can transmit. Because the router nodes have fixed locations, the resulting mesh router topology can be described by a directed graph $\mathcal{G}$ with nodes in the set $\mathcal{R}$ and where a directed edge (r, m) exists if and only if $r \in \mathcal{R}$ and $m \in \Theta_r$. The resulting graph is assumed to be *connected*, so that it is possible to send data from any router node to any other (possibly using a multi-hop path).

A. Time Varying Channel States

For each node pair $(i, j) \in \{1, \ldots, N\} \times \{1, \ldots, N\}$, define the *channel state* $S_{ij}(t)$. The $S_{ij}(t)$ value characterizes the current quality of link (i, j), and might represent ambient noise levels, current fading states, attenuation coefficients, or a simple quality label such as {"0," "Bad," "Medium," "Good"}. A link quality of $S_{ij}(t) =$ "0" indicates that link (i, j) either does not exist or is so poor that data transmission on the current timeslot is impossible. Note that a link (i, j) is part of the mesh router network if $i \in \mathcal{R}$, $j \in \mathcal{R}$, is a client-router uplink if $i \in \mathcal{C}$, $j \in \mathcal{R}$, and is a downlink if $i \in \mathcal{R}, j \in \mathcal{C}$.

Define $\boldsymbol{S}(t) = (S_{ij}(t))$ as the network-wide link state matrix, composed of the current state of all links. We assume that there are a finite number of matrices $\boldsymbol{S}(t)$, and that these matrices are conditionally independent and identically distributed given the current client topology state $\mathcal{T}(t)$. Define the conditional link state probability distribution $\pi_{\boldsymbol{S}}(\mathcal{T})$ as follows:

$$\pi_{\boldsymbol{S}}(\mathcal{T}) = Pr[\boldsymbol{S}(t) = \boldsymbol{S} \mid \mathcal{T}(t) = \mathcal{T}] \quad (1)$$

Channel state matrices $\boldsymbol{S}(t)$ are thus chosen independently with distribution $\pi_{\boldsymbol{S}}(\mathcal{T})$ for every timeslot in which $\mathcal{T}(t) = \mathcal{T}$. These probabilities are not necessarily known to the network controllers. However, for the first half of this paper, we assume a *channel aware* scheduling model, where each mesh router node r is aware of the current channel states $S_{cr}(t)$ and $S_{rc}(t)$ of each of its possible uplinks and downlinks. The alternate case of *channel-blind* scheduling, where current link state information is unknown, is considered in Section VI.

For simplicity, we assume that only uplinks and downlinks have time varying channel states. Thus, $S_{ij}(t)$ is a constant for all time if (i, j) is a mesh router link. In particular, we define $S_{ij}(t) = 1$ for all time t if (i, j) is a valid mesh router link (so that $i \in \mathcal{R}$, $j \in \mathcal{R}$, and $j \in \Theta_i$), and define $S_{ij}(t) = 0$ for all time if router link (i, j) does not exist.

B. The Uplink and Downlink Channel Model

Each mesh client $c \in \mathcal{C}$ uses a fixed power P_c^{tran} for transmission purposes, and expends a fixed power P_c^{rec} whenever receiving. The power required for other functions, such as passing control information, is not considered here. Let $P_c(t)$ denote the power that client c expends during slot t (so that $P_c(t) \in \{0, P_c^{rec}, P_c^{tran}\}$ for each $c \in \mathcal{C}$). Each mesh client c has an average power constraint P_c^{av}, so that the following time average inequality is required:

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} P_c(\tau) \leq P_c^{av} \quad \text{for all } c \in \mathcal{C} \quad (2)$$

Each client has a set of *variable rate codes* that can be used to select the desired number of packets to transmit in a single slot. Specifically, for each client $c \in \mathcal{C}$ and each router $r \in \mathcal{R}$, define $\mu_{cr}(t)$ as the transmission rate allocated for link (c, r) during slot t (taking units of packets/slot). Because clients can only transmit to the single router whose cell they are currently in, we have $\mu_{cr}(t) = 0$ whenever $r \neq r(c, t)$. Further, the value $\mu_{cr}(t)$ must be an integer between 0 and μ_c^{max}, where μ_c^{max} represents the maximum rate of client c. Hence, the $\mu_{cr}(t)$ decision variables must satisfy:

$$\mu_{cr}(t) \in \{0, 1, \ldots, \mu_c^{max}\} \quad , \quad \mu_{cr}(t) = 0 \text{ if } r \neq r(c, t) \quad (3)$$

The simplest case is when $\mu_c^{max} = 1$ for all $c \in \mathcal{C}$, so that each client can either transmit a single packet, or remains idle. Note that client c expends power $P_c(t) = P_c^{tran}$ on any timeslot in which $\mu_{cr}(t) > 0$.

Likewise define $\mu_{rc}(t)$ as the number of packets router node $r \in \mathcal{R}$ transmits to client $c \in \mathcal{C}$. These decision variables must satisfy $\mu_{rc}(t) = 0$ if $r \neq r(c, t)$, and $\mu_{rc}(t) \in \{0, \ldots, \mu_r^{max}\}$, where μ_r^{max} is the maximum number of downlink packets the router node r can transmit. We assume that a client cannot transmit and receive on the same timeslot. Note that each client c expends power $P_c(t) = P_c^{rec}$ on any timeslot in which it receives data from its current router node. Transmissions in the cell of each router node $r \in \mathcal{R}$ are further restricted by the *scheduling constraint set* Γ_r, specifying the collection of all $\mu_{cr}(t)$ and $\mu_{rb}(t)$ rates that can be scheduled simultaneously (where $c, b \in \mathcal{C}$). This scheduling constraint Γ_r is defined to limit interference within each cell. We assume it is such that setting any component rates of a feasible rate allocation to zero yields another feasible rate allocation. A typical constraint set is one that allows only one uplink or downlink transmission per cell per timeslot. More general constraints can be defined if multi-user detection is possible and/or if there are multiple frequencies over which to schedule transmissions.

Let $\phi_{ij}(\mu_{ij}(t), S_{ij}(t))$ represent the *success probability* associated with each packet transmitted over link (i, j) when the transmission rate is $\mu_{ij}(t)$ and the current channel state is $S_{ij}(t)$. For simplicity, we assume this probability depends only on the transmission rate and channel state over link (i, j), so that there are no interference effects associated with transmissions over other links.[1] This assumption is valid, for example, when the constraint sets Γ_r allow at most one transmission per cell per slot, and when transmissions in neighboring cells use orthogonal signals. Thus, when $\mu_{ij}(t)$ packets are transmitted over link (i, j), the expected number of successful receptions is equal to $\mu_{ij}(t)\phi_{ij}(\mu_{ij}(t), S_{ij}(t))$. For simplicity, we assume that ACK/NACK feedback about each transmitted packet is received by the transmitter at the end of the slot. Note that a simple special case is when all transmissions are successfully received with probability 1 whenever transmission rates are less than or equal to a threshold $\mu_{ij}^{th}(t)$ that depends on the link state $S_{ij}(t)$, in which case the ACK/NACK feedback is not required.

C. Flow Control and the Transport Layer

Data exogenously arrives to the clients in packetized form, and we let $A_c^{(d)}(t)$ represent the number of exogenous packets

[1] More general probability functions can be defined as in [18] to include possible channel interference.

that arrive to client c during slot t that have destination client d (destinations are assumed be be within the client set $\mathcal{C}$). All packets associated with a given destination d are defined as *commodity* d packets. To reduce congestion, it may be necessary to restrict the number of new packets admitted into the network. Define the *flow control decision variable* $\alpha_c^{(d)}(t)$ as the (integer) number of commodity d packets client c allows into its network layer during slot t. Any packet that is not admitted is either placed into a *transport layer storage reservoir* [12], or dropped if there is no room for storage. Transport layer reservoirs can have finite buffers (possibly of size zero). In the special case of a size zero transport buffer, all packets that are not immediately admitted to the network are necessarily dropped.

Let $L_c^{(d)}(t)$ represent the number of commodity d packets in the transport layer of client c at time t. The flow control decisions $\alpha_c^{(d)}(t)$ are chosen by each client $c \in \mathcal{C}$ according to the constraints:

$$0 \leq \alpha_c^{(d)}(t) \leq A_c^{(d)}(t) + L_c^{(d)}(t)$$
$$\sum_d \alpha_c^{(d)}(t) \leq \alpha_{max}$$

where α_{max} is a parameter that bounds the number of new packets that can enter the network layer of each client during one timeslot (specified in Section IV). We say that the flow controllers are "turned off" if we set $\alpha_{max} \triangleq \infty$ and choose $\alpha_c^{(d)}(t) = A_c^{(d)}(t)$ for all clients, destinations, and time.

D. Network Layer Scheduling and Queueing Dynamics

Data that is admitted into the network is queued at each client according to its destination. Define $U_c^{(d)}(t)$ as the number of commodity d packets currently stored in the network layer of client c. Typically, each client would be the source of only one or two active sessions, so that the number of distinct active queues in each client would be only one or two. For simplicity, all of these queues are assumed to have large enough buffers so that no data is ever dropped at the network layer (it is shown in Theorem 1 that this can be accomplished using a suitably large but finite buffer size).

Let $\mu_{cr}^{(d)}(t)$ represent the number of commodity d packets that client c attempts to transmit to its router node r during slot t. These transmission decision variables must be integers that satisfy the following constraints:

$$0 \leq \mu_{cr}^{(d)}(t) \leq U_c^{(d)}(t) \quad , \quad \sum_d \mu_{cr}^{(d)}(t) \leq \mu_{cr}(t) \tag{4}$$

where the transmission rate $\mu_{cr}(t)$ must satisfy the constraints (3). The first inequality above states that $\mu_{cr}^{(d)}(t)$ cannot be more than the number of commodity d packets available for transmission at client c, and the second inequality ensures that the total number of packets transmitted by the client does not exceed the transmission rate. Each of the $\mu_{cr}^{(d)}(t)$ packets will be independently and probabilistically received with probability $\phi_{cr}(\mu_{cr}(t), S_{cr}(t))$. Define $\tilde{\mu}_{cr}^{(d)}(t)$ to be the number of such commodity c packets that are successfully received. The one-step queueing dynamics for each of the client queues thus proceeds as follows:

$$U_c^{(d)}(t+1) = U_c^{(d)}(t) - \tilde{\mu}_{c,r(c,t)}^{(d)}(t) + \alpha_c^{(d)}(t) \tag{5}$$

where we recall that $r(c,t)$ is the router node whose cell contains client c during slot t.

Similarly, each router node also maintains separate queues for each commodity, and we define $U_r^{(d)}(t)$ as the number of commodity d packets currently in router $r \in \mathcal{R}$. Define $\mu_{rb}(t)$ as the transmission rate over link (r,b) during slot t (choice of these rates is discussed in Subsection II-E below). Define $\mu_{rb}^{(d)}(t)$ as the number of commodity d packets transmitted over link (r,b) during slot t. If $b \in \mathcal{R}$, then link (r,b) is a mesh router link. If $b \in \mathcal{C}$, then link (r,b) is a router-client downlink. The transmission decision variables must be non-negative integers that satisfy:

$$\sum_b \mu_{rb}^{(d)}(t) \leq U_r^{(d)}(t) \quad , \quad \sum_d \mu_{rb}^{(d)}(t) \leq \mu_{rb}(t) \tag{6}$$

Define $\tilde{\mu}_{rb}^{(d)}(t)$ as the number of commodity d packets successfully received over link (r,b) during slot t. The queue dynamics at each router node $r \in \mathcal{R}$ satisfy:

$$U_r^{(d)}(t+1) = U_r^{(d)}(t) - \sum_b \tilde{\mu}_{rb}^{(d)}(t) + \sum_a \tilde{\mu}_{ar}^{(d)}(t) \tag{7}$$

E. The Mesh Router Model

The sub-network consisting only of mesh router nodes $r \in \mathcal{R}$ and links (n,m) (where $n \in \mathcal{R}, m \in \mathcal{R}$) shall be referred to as the *mesh router network*. This network is responsible for delivering the client data over potentially multi-hop paths to reach a mesh router node that can transmit the data to the destination client. However, the locations of the clients may change over time, and hence the mesh routers must "chase a moving target." Therefore, one responsibility of the mesh router network is to maintain and disseminate location information for each of the mobile clients. Another responsibility is to make intelligent transmission decisions between router nodes.

The mesh router topology is fixed, and hence we assume a pre-computation has been performed to construct a *control tree*: a tree containing all router nodes over which control information and client location updates can be passed. The root node of the tree is chosen to minimize the maximum distance between itself and any other node of the tree. For each router node $r \in \mathcal{R}$ and each client node $c \in \mathcal{C}$, define $\hat{n}_r(c,t)$ as router node r's *current location estimate* for the client c (where $\hat{n}_r(c,t) \in \mathcal{R}$). That is, if $\hat{n}_r(c,t) = n$, then router r currently thinks client c is located in the cell of router node n. We assume that each router node knows exactly which clients are in its *own* cell, and hence $\hat{n}_r(c,t) = r$ whenever $r(c,t) = r$. That is, the estimated location of client c is exact for the router r whose cell it is currently located.

Location updates are broadcast to all other router nodes when necessary. Specifically, whenever a client moves to a cell of a new router and/or its whereabouts are queried, the updated information is sent to the root of the control tree, and the root passes it to all other router nodes. We assume such location updates are received within a maximum of D slots. The location estimates are initialized arbitrarily. The channels of the control tree are assumed to be independent of the data channels, and might operate over faster timescales than one timeslot. Hence, it is possible to have a D value that is less than the depth of the tree (indeed, in some networks it may be meaningful to assume $D = 0$, so that all routers have "immediate" knowledge of current client locations). The effects of the control tree are thus modeled by the location update parameter D, and we do not further discuss operation of the control tree in this paper.

To model packet delivery over the mesh router network, recall that $\mu_{rb}(t)$ represents the current transmission rate over link (r, b) in the mesh network (where r and b are both in the set $\mathcal{R}$). Rather than viewing these rates as decision variables, we view them as being determined by a pre-computed periodic or psuedo-random transmission schedule. As an example, if the router network is arranged as a two dimensional grid, then the transmissions might be scheduled according to the 4-periodic cycle that allows transmissions in the North, South, East, and West directions. We assume that $\mu_{rb}(t) = 0$ for all t whenever mesh router link (r, b) does not exist (i.e., when $b \notin \Theta_r$). For simplicity, we assume the mesh router links are fully reliable (so that all transmitted packets are received with probability 1). The resulting process of mesh router transmissions is assumed to enable multi-hop communication between any two router nodes.

While the computation of the transmission rate schedule for the mesh links is not the focus of this paper, it is an important consideration as it affects the throughput region the network can support. The schedule must generally enable high throughput, without knowing in advance how much traffic there is for each commodity type (and without knowing the client locations in advance). A simple and highly effective approach is to schedule so as to maximize the all-to-all communication rate λ supportable by the mesh routers. Specifically, we say that the router network supports an all-to-all rate λ if multi-commodity flows can be formed over the mesh router graph that deliver a total flow rate of λ for each of the $|\mathcal{R}|(|\mathcal{R}| - 1)$ router node pairs (where $|\mathcal{R}|$ is the number of router nodes). The multi-commodity flow must satisfy the link rate constraints, so that the sum flow rate over any link is less than or equal to the time average transmission rate of that link. Periodic or randomized schedules with this property can be computed for wireless networks with interference using techniques in [12] [28] [3]. The transmissions in the mesh router network are then used to support the actual traffic that arises from the clients. Using simple load balancing and traffic uniformization principles [33] [34] [35], is not difficult to show that the resulting throughput region of the mesh router network under this transmission scheduling policy is within a factor of 2 of the *optimal symmetric capacity region* for the mesh router sub-network (defined as the region of throughputs such that the sum throughput being sourced or sinked at any router node is less than or equal to λ_{max}, where λ_{max} is the maximum scalar with this property).

F. Discussion of Model Assumptions

The above network model assumes that client-to-client communication is impossible. Network capacity might be improved by allowing this feature (discussed further in Section VI), although this would require clients to cooperate and expend energy receiving and transmitting another client's data, possibly involving pricing [36] [37] [38]. The mesh router network is assumed to have node location updates based on a control tree. Strictly speaking, such location information is not necessary to achieve network capacity [2] [3] [11], but shall be useful in reducing network delay. Finally, we have assumed transmissions in the mesh router links are pre-computed. A dynamic link scheduling algorithm could enable higher network throughput, but might involve complex and time expensive computations to maximize a weighted sum of transmission rates over the network, where the transmission rates on each link may be coupled through interference [2] [3] [11]. The pre-computed link schedule is assumed for simplicity, as is the assumption of fully reliable mesh router links.

III. Network Capacity

Here we specify two types of capacity regions, the *ergodic capacity region* and the *instantaneous capacity region*. Both regions assume the network structure as specified in the previous section. Specifically, clients move according to a client topology process $\mathcal{T}(t)$, channel states are i.i.d. given the topology state $\mathcal{T}$, with conditional distributions $\pi_S(\mathcal{T})$ (defined in (1)), transmission rates are restricted as described in the previous section, and the transmission rate process for the mesh router links is pre-defined.

We further assume that exogenous arrivals at each client are bounded by a value A_{max}, so that:

$$\sum_d A_c^{(d)}(t) \leq A_{max} \quad \text{for all } c \in \mathcal{C} \tag{8}$$

Finally, in describing the capacity region, we assume the flow controllers are turned off (so that $\alpha_c^{(d)}(t) = A_c^{(d)}(t)$ for all t), and consider all possible routing and scheduling strategies with the structure described in the previous section. Let $\lambda_c^{(d)}$ be the input rate of commodity d data at source client c (where $c \in \mathcal{C}$, $d \in \mathcal{C}$). Let $\boldsymbol{\lambda} = (\lambda_c^{(d)})$ be the resulting matrix of input rates. We assume throughout that $\lambda_c^{(c)} = 0$ for all c (so that no client desires to send data to itself).

A. Queue Stability

Let $U(t)$ represent the number of packets in a discrete time queue with a general input and service rate process.

Definition 1: A queue is *strongly stable* if:

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left\{U(\tau)\right\} < \infty$$

A *network of queues* is defined to be strongly stable if all individual queues are strongly stable. Throughout this paper, we use "stability" to refer to strong stability.

B. Ergodic Capacity

Assume the client topology state process $\mathcal{T}(t)$ evolves according to an ergodic process with well defined steady state probabilities. Further assume the transmission rates for links in the mesh router network are ergodic with a well defined steady state distribution.

Definition 2: The *ergodic network capacity region* Λ is the closure of the set of all rate matrices $\boldsymbol{\lambda} = (\lambda_c^{(d)})$ ($c \in \mathcal{C}$, $d \in \mathcal{C}$) that can be stably supported by some network control algorithm for routing and transmission rate scheduling that conforms to the constraints of Section II.

The ergodic capacity region is a function of the steady state client topology distribution, channel probabilities, and time average mesh router transmission rates. A characterization of this region for general stochastic networks is provided in [3] [11]. We note that supporting throughput within this region may require delay to be on the order of the timescales associated with the node mobility process.

C. Instantaneous Capacity

Consider an arbitrary (perhaps non-ergodic) user mobility process $\mathcal{T}(t)$. Suppose that $\mathcal{T}(t) = \mathcal{T}$ at a particular timeslot t, and recall that channel probabilities have conditional distribution $\pi_{\boldsymbol{S}}(\mathcal{T})$.

Definition 3: Fix a timeslot t, and define $\mathcal{T}^* \triangleq \mathcal{T}(t)$, $\pi_{\boldsymbol{S}}^* \triangleq \pi_{\boldsymbol{S}}(\mathcal{T})$. The *instantaneous capacity region* $\Lambda^*(t)$ at a given time t is the ergodic capacity region associated with a network where the client locations are fixed at $\mathcal{T}^*$ for all time, with i.i.d. channel states with distribution $\pi_{\boldsymbol{S}}^*$ for all time.

Therefore, the instantaneous capacity region is defined as the throughput region that would be supportable if the current node locations never changed. In the next section, we show that if the instantaneous capacity region is sufficiently large for all timeslots t, then a stable scheduling algorithm can be developed with a guaranteed end-to-end average delay that is independent of the timescales of the user mobility process. The algorithm will *also* stabilize the system whenever the throughput matrix is inside the ergodic capacity region. However, delay in this case may depend on the mobility process if the throughput matrix exceeds some of the instantaneous capacity regions at various timeslots.

IV. The Basic Backpressure Algorithm

Assume arrivals are i.i.d. over timeslots with rate matrix $\boldsymbol{\lambda} = (\lambda_c^{(d)})$ (where $\mathbb{E}\left\{A_c^{(d)}(t)\right\} = \lambda_c^{(d)}$). Let $\overline{\alpha}_c^{(d)}(t)$ be the time average expected rate of commodity d data admitted by client c during the first t timeslots:

$$\overline{\alpha}_c^{(d)}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\alpha_c^{(d)}(\tau)\right\}$$

As a measure of network fairness, for each commodity pair (c, d) (where $c, d \in \mathcal{C}$, $c \neq d$), define a utility function $g_c^{(d)}(\alpha)$ representing the satisfaction client c receives by sending commodity d data at a long term throughput α. Each utility function is assumed to be concave, continuous, non-decreasing, and to satisfy $g_c^{(r)}(0) = 0$. A typical example is the logarithmic utility function:

$$g_c^{(d)}(\alpha) = \log(1 + \alpha) \tag{9}$$

where $\log(\cdot)$ denotes the natural logarithm.

If the network has an ergodic capacity region Λ, define the ergodic optimal utility g^* as the solution to the following problem:

$$\begin{aligned} \text{Maximize:} \quad & g \triangleq \textstyle\sum_{c,d} g_c^{(d)}(\alpha_c^{(d)}) \\ \text{Subject to:} \quad & (\alpha_c^{(d)}) \in \Lambda \\ & \alpha_c^{(d)} \leq \lambda_c^{(d)} \end{aligned} \tag{10}$$

That is, g^* represents the optimal utility achievable by a stabilizing control algorithm with an ergodic capacity region Λ. Similarly, for each instantaneous capacity region $\Lambda^*(t)$, define the optimal utility $g^*(t)$ as the solution to the following maximization problem:

$$\begin{aligned} \text{Maximize:} \quad & g \triangleq \textstyle\sum_{c,d} g_c^{(d)}(\alpha_c^{(d)}) \\ \text{Subject to:} \quad & (\alpha_c^{(d)}) \in \Lambda^*(t) \\ & \alpha_c^{(d)} \leq \lambda_c^{(d)} \end{aligned} \tag{11}$$

A. The Cross-Layer Mesh Network Algorithm (CLC-Mesh)

The following cross-layer control algorithm is decoupled into separate algorithms for flow control at the clients, power-aware uplink/downlink transmission scheduling, and routing in the mesh router nodes. It uses a non-negative parameter V that can be chosen as desired to affect a utility-delay tradeoff. The resulting algorithm applies techniques from [11] [12] [13] to this mesh network setting, with the exception that we use a modified differential backlog weight to ensure worst case backlog guarantees.

Flow Control: For each client $c \in \mathcal{C}$, define Ω_c as the set of all commodities d such that $\lambda_c^{(d)} > 0$ (typically each client has at most one or two active sessions). For each $d \in \Omega_c$, define a *flow state queue* $Y_c^{(d)}(t)$ with an initial value $Y_c^{(d)}(0) = 0$, and with an update equation:

$$Y_c^{(d)}(t+1) = \max[Y_c^{(d)}(t) - \alpha_c^{(d)}(t), 0] + \gamma_c^{(d)}(t) \tag{12}$$

where $\alpha_c^{(d)}(t)$ is the flow control decision at slot t, and $\gamma_c^{(d)}(t)$ is a useful auxiliary variable (both chosen according to the computation below). These flow state queues are purely implemented in software at their respective clients.

Define $\alpha_{max} \triangleq A_{max}$, where A_{max} is the maximum number of packets that can arrive to a client in a single timeslot (satisfying (8)). For each timeslot t, each client $c \in \mathcal{C}$ chooses flow control variables $\alpha_c^{(d)}(t)$ as non-negative integers (represented as $\mathbb{N}^+$) that solve the following problem:

$$\begin{aligned} \text{Maximize:} \quad & \textstyle\sum_{d \in \Omega_c}[Y_c^{(d)}(t) - U_c^{(d)}(t)]\alpha_c^{(d)}(t) \\ \text{Subject to:} \quad & \textstyle\sum_{d \in \Omega_c} \alpha_c^{(d)}(t) \leq \alpha_{max} \\ & \alpha_c^{(d)}(t) \leq A_c^{(d)}(t) + L_c^{(d)}(t) \;\; \forall d \in \Omega_c \\ & \alpha_c^{(d)}(t) \in \mathbb{N}^+ \;\; \forall d \in \Omega_c \end{aligned} \tag{13}$$

Furthermore, the client chooses auxiliary variables $\gamma_c^{(d)}(t)$ for timeslot t as the solution to the following optimization:

$$\begin{aligned} \text{Maximize:} \quad & V g_c^{(d)}(\gamma_c^{(d)}) - \gamma_c^{(d)} Y_c^{(d)}(t) \\ \text{Subject to:} \quad & 0 \leq \gamma_c^{(d)} \leq \alpha_{max} \end{aligned} \tag{14}$$

The flow state queues $Y_c^{(d)}(t)$ are then updated via (12).

Flow control algorithms that use flow state queues in this manner were introduced in [12] and [11]. Note that this flow control algorithm can be implemented separately at each client, and uses only the $Y_c^{(d)}(t)$ values local to each client. The admission decisions (13) admit commodity d data into the transport layer at client c only if $U_c^{(d)}(t) \leq Y_c^{(d)}(t)$. The optimization in (13) is trivially solved by admitting as many packets as possible from the commodities d that maximize $Y_c^{(d)}(t) - U_c^{(d)}(t)$. In the case when there is only a single active session at each client, or when there are no transport layer storage reservoirs (so that $L_c^{(d)}(t) = 0$ for all t), the solution to (13) is given by: $\alpha_c^{(d)}(t) = \min[A_c^{(d)}(t) + L_c^{(d)}(t), \alpha_{max}]$ if $U_c^{(d)}(t) \leq Y_c^{(d)}(t)$, and $\alpha_c^{(d)}(t) = 0$ otherwise. The optimization of the auxiliary variables $\gamma_c^{(d)}(t)$ in (14) is a simple maximization of a concave function of a single real variable, and the solution can also be computed easily in real time. In the special case when utilities $g_c^{(d)}(\alpha)$ have the form in (9), we have the closed form solution:

$$\gamma_c^{(d)}(t) = \min[\max[V/Y_c^{(d)}(t) - 1, 0], \alpha_{max}]$$

Transmission Rate Variables: It is useful to modify the constraints of the transmission rate variables $(\mu_{ab}^{(d)}(t))$ as follows:

$$\mu_{ab}^{(d)}(t) \geq 0 \quad \text{for all } (a, b, d) \tag{15}$$

$$\sum_d \mu_{ab}^{(d)}(t) \leq \mu_{ab}(t) \quad \text{for all } (a, b) \tag{16}$$

These new constraints are less restrictive than the constraints (4) and (6) as they do not involve queue backlog. Our algorithm shall compute transmission rates according to the above constraints. In cases where a certain node a does not have enough data of a particular commodity d to fill the transmission rates $\mu_{ab}^{(d)}(t)$ over all outgoing links (a, b) for which $\mu_{ab}^{(d)}(t) > 0$, then *idle fill* packets are used. The actual packets are distributed over the links arbitrarily, with the only assumptions being that: (i) If $U_a^{(d)}(t) \geq \sum_b \mu_{ab}^{(d)}(t)$, then no idle fill packets are used. (ii) If $U_a^{(d)}(t) < \sum_b \mu_{ab}^{(d)}(t)$, then all $U_a^{(d)}(t)$ packets must be used on outgoing links (a, b) with positive rates $\mu_{ab}^{(d)}(t)$ (placing them on each link arbitrarily) before idle fill packets are used. All of our analysis and optimality results hold regardless of the decisions made in cases when there is a choice of this nature, provided the above two conditions are maintained. For implementation purposes, we use the particular strategy of filling all downlinks (to destinations) before filling router-router links.

Power-Aware Uplink/Downlink Scheduling: To ensure the average power constraints are met, we use the *virtual power queue* technique developed in [13]. Specifically, for each client $c \in \mathcal{C}$, define a *virtual power queue* $X_c(t)$ as a measure of the excess power expenditure over and above the average power constraint. The initial value is set to zero (so that $X_c(0) = 0$), and the virtual power queue is updated in software according to the update rule:

$$X_c(t + 1) = \max[X_c(t) - P_c^{av}, 0] + P_c(t) \tag{17}$$

where P_c^{av} is the average power constraint of client c, and $P_c(t)$ is the power expenditure of the client during slot t, where:

$$P_c(t) = \begin{cases} P_{tran} & \text{if } \mu_{c,r(c,t)}(t) > 0 \\ P_{rec} & \text{if } \mu_{c,r(c,t)}(t) = 0 \text{ and } \mu_{r(c,t),c}(t) > 0 \\ 0 & \text{otherwise} \end{cases}$$

The intuition is that stabilizing each virtual power queue ensures the time average input rate is less than or equal to the time average transmission rate, so that (2) is satisfied.

The transmission rates $\mu_{cr}(t)$ and $\mu_{rc}(t)$ within each cell are computed as follows. On each timeslot t, each client $c \in \mathcal{C}$ informs its current router node $r = r(c, t)$ of its backlog values $U_c^{(d)}(t)$ and its virtual power queue level $X_c(t)$. (If each source client has only one active session and hence only one non-zero queue $U_c^{(d)}(t)$, then this involves the exchange of only 2 scalar values $U_c^{(d)}(t)$ and $X_c(t)$). Define the *differential backlog* $W_{cr}^{(d)}(t)$ between a client c and its router node r as follows:

$$W_{cr}^{(d)}(t) \triangleq U_c^{(d)}(t) - U_r^{(d)}(t) \tag{18}$$

Define η as the largest right derivative of any utility function $g_c^{(d)}(\alpha)$ at the point $\alpha = 0$ (assumed finite). In the case of logarithmic utilities as in (9), we have $\eta = 1$. Define $\hat{W}_{cr}^{(d)}(t)$ as follows:

$$\hat{W}_{cr}^{(d)}(t) \triangleq \begin{cases} \max[W_{cr}^{(d)}(t), 0] & \text{if } U_r^{(d)}(t) < \eta V \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Define $\mathcal{C}_r(t)$ as the set of clients currently contained in the cell of router node r. Each router node r then observes the uplink and downlink channel states for its client nodes, and chooses transmission rates $\{\mu_{rc}(t)\}, \{\mu_{cr}(t)\}, \{\mu_{cr}^{(d)}(t)\}$, and $\{\mu_{rc}^{(d)}(t)\}$ that maximize:

$$\sum_{d \in \mathcal{C}_r(t)} U_r^{(d)}(t) \mu_{rd}^{(d)} \phi_{rd}(\mu_{rd}, S_{rd}(t)) +$$
$$\sum_{c \in \mathcal{C}_r(t)} \left[-X_c(t) P_c(t) + \sum_{d \in \mathcal{C}} \hat{W}_{cr}^{(d)}(t) \mu_{cr}^{(d)} \phi_{cr}(\mu_{cr}, S_{cr}(t)) \right] \tag{20}$$

subject to the constraint set Γ_r and the feasibility constraints (15) and (16). The virtual power queues are then updated according to (17).

In the special case where Γ_r is the constraint set that restricts only one uplink or downlink transmission per cell per slot, and when each client $c \in \mathcal{C}$ is the source of only one commodity d_c, then the above maximization is accomplished in each cell by comparing the value of (20) for each single uplink and downlink possibility, as follows:

Downlink optimization: For a given router r, consider the control option of choosing a downlink transmission over link (r, d) at a non-zero rate μ_{rd} (for a particular client $d \in \mathcal{C}_r(t)$). Note that we must transmit commodity d over this link (as a downlink transmission must send data to its destination), and hence $\mu_{rd}^{(d)} = \mu_{rd}$. Further, this downlink transmission precludes uplink transmissions or other downlink transmissions in this cell, and hence (20) reduces to:

$$U_r^{(d)}(t) \mu_{rd} \phi_{rd}(\mu_{rd}, S_{rd}(t)) - X_d(t) P_{rec} \tag{21}$$

where $P_d(t) = P_{rec}$ because client d receives a downlink transmission. Define μ_{rd}^* as the optimal feasible transmission rate for this downlink (i.e., μ^*_{rd} maximizes (21)). Define d^*_{down} as the client that maximizes $U_r^{(d)}(t)\mu^*_{rd}\phi_{rd}(\mu^*_{rd}, S_{rd}(t)) - X_d(t)P_{rec}$ over all $d \in \mathcal{C}_r(t)$. If the corresponding maximum value is non-negative, define Q^*_{down} as this maximum, else define $Q^*_{down} = 0$. Thus, Q^*_{down} represents the maximum value of the expression (20) over all downlink transmission options, including the option to remain idle. If $Q^*_{down} > 0$, then the optimal downlink is (r, d^*_{down}).

Uplink optimization: For a given router r, consider a transmission over uplink (c, r) at a non-zero rate μ_{cr} (for a given client $c \in \mathcal{C}_r(t)$). Thus, $\mu_{cr}^{(d_c)} = \mu_{cr}$ (because each client c is the source of only one commodity d_c). The resulting expression (20) becomes:

$$\hat{W}_{cr}^{(d_c)}(t)\mu_{cr}\phi_{cr}(\mu_{cr}, S_{cr}(t)) - X_c(t)P_{tran} \quad (22)$$

Define μ^*_{cr} as the feasible transmission rate that maximizes (22). Define c^*_{up} as the commodity that maximizes $\hat{W}_{cr}^{(d_c)}(t)\mu^*_{cr}\phi_{cr}(\mu^*_{cr}, S_{cr}(t)) - X_c(t)P_{tran}$ over all $c \in \mathcal{C}_r(t)$, and let Q^*_{up} represent the corresponding maximum value (if the maximum is non-negative), and else let $Q^*_{up} = 0$. Thus, Q^*_{up} represents the maximum value of the expression (20) if the controller at router node r decides to either transmit over an uplink channel or remain idle. If $Q^*_{up} > 0$, then the optimal uplink is (c^*_{up}, r).

If $Q^*_{down} = Q^*_{up} = 0$, no transmissions take place during slot t in the cell of node r. Otherwise, if $Q^*_{down} > 0$ and $Q^*_{down} \geq Q^*_{up}$, then choose a downlink transmission for the cell of router node r, using link (r, d^*_{down}) with transmission rate $\mu^*_{rd^*_{down}}$. Finally, if $Q^*_{up} > 0$ and $Q^*_{up} > Q^*_{down}$, then choose an uplink transmission, using link (c^*_{up}, r) and transmission rate $\mu^*_{c^*_{up}r}$.

Note that this scheduling algorithm is decoupled cell-by-cell, and only uses knowledge of the channel states, client backlogs, and virtual power queues local to each cell.

Mesh Network Routing: Before network operation, the mesh router sub-network computes a *shortest path routing table* for each pair of routing nodes. Specifically, for each node pair (n, b) such that $n \in \mathcal{R}, b \in \mathcal{R}$, we define H_{nb} as the distance required to travel from node n to node b over its shortest path. Distributed algorithms for computing such H_{nb} values are well known (see, for example, [39]), and a variety of link weight metrics might be used in the computation. For example, the link weight of each router link can be set to 1, in which case the value H_{nb} represents the shortest number of hops from router node n to router node b. The value of H_{nn} is defined to be 0 for all router nodes n. Define H_{max} as the largest value of $|H_{rn} - H_{bn}|$ over all router nodes r and n and all neighbors $b \in \Theta_r$. This is the largest change in the distance to a destination node n if a packet is sent from a node r to a neighbor node b.

For each timeslot t, every router node r computes its *mixed differential backlog weight* $W_{rb}^{(d)}(t)$ for each neighboring router node $b \in \Theta_r$ and each commodity $d \in \mathcal{C}$ as follows:

$$W_{rb}^{(d)}(t) \triangleq U_r^{(d)}(t) - U_b^{(d)}(t) + \omega(H_{r,\hat{n}_r(d,t)} - H_{b,\hat{n}_r(d,t)}) \quad (23)$$

where $\omega \geq 0$ is a pre-specified constant. This is a variation of the Enhanced Dynamic Routing and Power Control (EDRPC) algorithm of [3]. Note that $(H_{r,\hat{n}_r(d,t)} - H_{b,\hat{n}_r(d,t)})$ is an *estimate* of the amount the distance would decrease between a commodity d packet and its destination if it is transmitted from node r to node b. This is an estimate because it assumes the true location of the client d is in the cell of router $\hat{n}_r(d, t)$. In the case when network backlog is large and $U_r^{(d)}(t) - U_b^{(d)}(t)$ is much larger than ωH_{max}, the weight $W_{rb}^{(d)}(t)$ is dominated by the differential backlog term. Conversely, in lightly loaded situations the differential backlog term may be small, so that the $W_{rb}^{(d)}(t)$ weight indicates the next hop for commodity d if it were to follow its estimated shortest path. Note that the $W_{rb}^{(d)}(t)$ weights can be computed at each node r using only the current queue backlogs $U_b^{(d)}(t)$ in neighboring nodes. Such backlogs can be determined by passing backlog updates to neighboring nodes at the end of each timeslot. Note also that in the case when the parameter ω is set to zero, the shortest path weights and the location estimates are not needed. This significantly reduces implementation complexity while possibly increasing delay. Define $\hat{W}_{rb}^{(d)}(t)$ as follows:

$$\hat{W}_{rb}^{(d)}(t) \triangleq \begin{cases} \max[W_{rb}^{(d)}(t), 0] & \text{if } U_b^{(d)}(t) < \eta V \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

At each timeslot t, each router node r computes the weights $\hat{W}_{rb}^{(d)}(t)$ as in (24) for each neighbor node b. It then observes the current transmission rate $\mu_{rb}(t)$ for each of its possible outgoing links (r, b). For each such link (r, b), the *optimal weight* $\hat{W}^*_{rb}(t)$ and *optimal commodity* $d^*_{rb}(t)$ are determined as follows:

$$\hat{W}^*_{rb}(t) \triangleq \max_{d \in \mathcal{C}} \hat{W}_{rb}^{(d)}(t)$$

$$d^*_{rb}(t) \triangleq \arg\max_{d \in \mathcal{C}} \hat{W}_{rb}^{(d)}(t)$$

If $\hat{W}^*_{rb}(t) > 0$, node r transmits $\mu_{rb}(t)$ packets of commodity $d^*_{rb}(t)$ over link (r, b) (not including the packets it first decided to send over a downlink channel at node r). If there are not enough packets of a particular commodity to fill the transmission rates $\mu_{rb}(t)$ over each outgoing link of node r, the links are filled with packets of the optimal commodities arbitrarily, and any residual transmission rate is filled either with *null packets* (i.e., dummy fill data) or with actual packets of different commodities d for which $\hat{W}_{rb}^{(d)}(t) > 0$.

The following important fact follows from the above scheduling algorithm:

Fact 1: If commodity d data is transmitted over a link (i, r) during slot t, where $i \in \mathcal{C} \cup \mathcal{R}$ and $r \in \mathcal{R}$, then the differential backlog between nodes i and r must satisfy:

$$\hat{W}_{ir}^{(d)}(t) > 0 \quad (25)$$

B. Algorithm Performance

Suppose arrivals are i.i.d. over timeslots with any arbitrary heterogeneous traffic matrix $\boldsymbol{\lambda} = (\lambda_c^{(d)})$. Suppose the above Cross-Layer Mesh Network Algorithm (CLC-Mesh) is implemented, using parameters $V \geq 0$, $\omega \geq 0$, and using arbitrary shortest path values H_{nb} with corresponding max differential value H_{max}. Recall from (1) that channel state matrices are conditionally i.i.d. over all timeslots with identical client topology states $\mathcal{T}(t)$. Assume $\mathcal{T}(t)$ is determined by *any arbitrary client mobility process*. Specifically, we assume there exists some probability law that describes the $\mathcal{T}(t)$ evolution, but this law is arbitrary and potentially non-ergodic. Let $\Lambda^*(t)$ be the resulting instantaneous capacity region for each timeslot. Assume all virtual and actual queues of the network are empty at time 0. Finally, for simplicity of exposition, we assume here that the mesh router link scheduling is done every timeslot according to a random or pseudo-random decision rule, so that the link rate matrix $(\mu_{rb}(t))$ (for $r \in \mathcal{R}$, $b \in \mathcal{R}$) is i.i.d. over timeslots.[2]

Define μ_{max}^{out} as the maximum number of packets that can be transmitted out of any node during a timeslot (considering both clients and routers, and summing over all outgoing links). Likewise, define μ_{max}^{in} as the maximum number of endogenous packets that can enter a node during a single slot.

Theorem 1: (Algorithm Performance) For arbitrary (possibly non-ergodic) client mobility processes, the mesh network algorithm stabilizes all queues in the network and meets all average power constraints. Specifically, we have:

(a) $U_n^{(d)}(t) \leq U_{max} \triangleq \eta V + \max[2\alpha_{max}, \mu_{max}^{in}]$ for all time t and all client or router queues (n, d).

(b) If $P_{rec} > 0$, then $X_c(t) \leq X_{max} \triangleq \frac{U_{max}\mu_{max}^{out}}{\min[P_{tran}, P_{rec}]} + \max[P_{tran}, P_{rec}]$ for all t and for all virtual power queues $X_c(t)$. In particular, the total power expended by a client $c \in \mathcal{C}$ over any interval of size T slots is no more than $P_c^{av}T + X_{max}$ (for any non-negative interval size T). If $P_{rec} = 0$, then $X_c(t) \leq U_{max}\mu_{max}^{out}/P_{tran} + P_{tran}$.

(c) The achieved throughput-utility satisfies:

$$\liminf_{t\to\infty} \sum_{c,d} g_c^{(d)}(\overline{\alpha}_c^{(d)}(t)) \geq \liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{g^*(\tau)\right\} - \frac{B}{V}$$

where $\overline{\alpha}_c^{(d)}(t)$ is defined:

$$\overline{\alpha}_c^{(d)}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\left\{\alpha_c^{(d)}(\tau)\right\}$$

and where B is a constant that depends on ωH_{max} and the number of network nodes, but is independent of V and independent of the user mobility process (the constant B is computed explicitly in the proof in Section V-A).

(d) If $dg_c^{(d)}(\alpha)/d\alpha > 0$ for $0 \leq \alpha \leq \alpha_{max}$ (for all client utilities (c, d)), and if $\boldsymbol{\lambda} \in \Lambda^*(t)$ for all timeslots t, then the component-wise $\liminf$ of the time average admitted rate matrix is within $O(1/V)$ of the input rate matrix $\boldsymbol{\lambda}$.

(e) Suppose the condition of part (d) holds, and define $\tilde{\Lambda}^*(t)$ as an extended instantaneous capacity region associated with the topology state $\mathcal{T}(t)$, but with $|\mathcal{R}||\mathcal{C}|$ new dimensions for all entries (r, d) with $r \in \mathcal{R}, d \in \mathcal{C}$, representing possible exogenous arrival processes of commodity d sourced at each router node. If there exists an $\epsilon > 0$ such that $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \tilde{\Lambda}^*(t)$ for all t (where $\boldsymbol{\epsilon} = (\epsilon_i^{(d)})$ is a matrix with entries $\epsilon_i^{(i)} = 0$, $\epsilon_i^{(d)} = \epsilon$, for all $i \neq d$ such that $i \in \mathcal{C} \cup \mathcal{R}$ and $d \in \mathcal{C}$), then the $\limsup$ average queue backlog can be bounded by a constant with a size of $O(1/\epsilon)$, i.e., a size that does not depend on V or on the timescales associated with the user mobility process.

(f) If the topology state process $\mathcal{T}(t)$ is ergodic with a well defined steady state, and if Λ represents the ergodic network capacity region, then the throughput-utility satisfies:

$$\liminf_{t\to\infty} \sum_{c,d} g_c^{(d)}(\overline{\alpha}_c^{(d)}(t)) \geq g^* - B_{mobile}/V$$

where g^* is the optimal solution of (10), and B_{mobile} is a constant that is independent of V but that possibly depends on the timescales associated with the client mobility process, as in [3].

Note that the queueing bounds are quite strong and specify bounded worst case queue backlog. In particular, all network queues are bounded by a value U_{max} that depends linearly on the V parameter. By Little's Theorem, this implies an end-to-end average delay bound that is also linear in V. Likewise, the virtual power queues are bounded by X_{max}, ensuring not only that the time average power constraint (2) is satisfied, but that the excess power expenditure beyond the average power constraint P_c^{av} over *any interval* is no more than X_{max} (where X_{max} is also linear in V). Part (c) of the theorem indicates that the time average throughput utility is within B/V of the target time average utility $\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} g^*(\tau)$. This difference B/V can be made arbitrarily small, with a corresponding (linear) tradeoff in X_{max} and U_{max}.

Parts (d) and (e) of the theorem indicate that, in the case when the traffic rates are strictly interior to the instantaneous capacity region for all slots t, then flow control is not necessary, so that choosing a very large value of V (which corresponds to accepting almost everything into the network) yields a packet drop rate of $O(1/V)$ and an average congestion and delay that depends only on the smallest distance ϵ between the traffic matrix and each instantaneous capacity region (and does not depend on V). This proves that, in such cases, almost 100% throughput can be achieved in situations of arbitrary (possibly non-ergodic) client mobility, with delay that is independent of the timescales associated with the mobility process. Part (f) specifies that if the mobility process is ergodic, then the algorithm can also come arbitrarily close to achieving the optimal utility g^* associated with the *ergodic capacity region* (regardless of the input rate matrix $\boldsymbol{\lambda}$), provided that the V parameter is sufficiently larger than a constant B_{mobile} that depends on the timescales of the client mobility process. Note that Theorem 1 holds for any buffer size at the transport layer storage reservoirs (possibly size 0).

Proof: (Theorem 1 part (a)) Consider any client node $c \in \mathcal{C}$ with queue backlog $U_c^{(d)}(t)$ and flow state queue $Y_c^{(d)}(t)$.

Claim 1: $Y_c^{(d)}(t) \leq \eta V + \alpha_{max}$ for all t.

To prove the claim, consider the optimization (14) that determines the $\gamma_c^{(d)}(t)$ variable. Because $g_c^{(d)}(0) = 0$ and η is the largest right derivative of any utility function, we have:

$$\begin{aligned} Vg_c^{(d)}(\gamma_c^{(d)}) - \gamma_c^{(d)}Y_c^{(d)}(t) &\leq V\eta\gamma_c^{(d)} - \gamma_c^{(d)}Y_c^{(d)}(t) \\ &= (V\eta - Y_c^{(d)}(t))\gamma_c^{(d)} \end{aligned}$$

From the above inequality, it follows that if $Y_c^{(d)}(t) > \eta V$, then the maximum of $Vg_c^{(d)}(\gamma_c^{(d)}) - \gamma_c^{(d)}Y_c^{(d)}(t)$ is equal to

[2] The algorithm achieves similar performance for periodic link scheduling, provided that the V parameter is increased by a factor equal to the scheduling period.

zero, and is achieved only when $\gamma_c^{(d)} = 0$. Thus, $\gamma_c^{(d)}(t) = 0$ whenever $Y_c^{(d)}(t) > \eta V$. It follows from (12) that $Y_c^{(d)}(t)$ cannot further increase when it exceeds ηV, so that $Y_c^{(d)}(t) \leq \eta V + \alpha_{max}$ for all t.

Claim 2: $U_c^{(d)}(t) \leq \eta V + 2\alpha_{max}$ for all $c \in \mathcal{C}$ and all t.

To prove Claim 2, note by (5) that at any timeslot t, $U_c^{(d)}(t)$ can only increase if its arrivals $\alpha_c^{(d)}(t)$ during slot t are non-zero. However by (13) we have that $\alpha_c^{(d)}(t) = 0$ whenever $U_c^{(d)}(t) > Y_c^{(d)}(t)$. By Claim 1, it follows that $\alpha_c^{(d)}(t) = 0$ whenever $U_c^{(d)}(t) > \eta V + \alpha_{max}$. Thus, $U_c^{(d)}(t)$ can only increase if it is less than or equal to $\eta V + \alpha_{max}$. Because queues are initially empty and the largest possible increase at any time is α_{max}, it follows that $U_c^{(d)}(t) \leq \eta V + 2\alpha_{max} \leq U_{max}$ for all t. This holds for all client queues $c \in \mathcal{C}$.

To prove the result for the router queues $r \in \mathcal{R}$, suppose that at a particular time t we have that $U_r^{(d)}(t) \leq U_{max}$ for all (r, d) (this certainly holds at time $t = 0$ when all queues are initially empty). We prove the same holds for time $t + 1$, which by induction proves the result for all time. Consider any queue $U_r^{(d)}(t)$ at time t. If this queue does not receive any new arrivals during slot t, then $U_r^{(d)}(t+1) \leq U_r^{(d)}(t) \leq U_{max}$ and we are done. If this queue receives new data from a client c during slot t, then the differential backlog metric $\hat{W}_{cr}^{(d)}(t)$ must be positive (by (25) of Fact 1), which (by the definition (19)) implies $U_r^{(d)}(t) < \eta V$. The most this queue can increase in any timeslot is given by μ_{max}^{in}, and so $U_r^{(d)}(t+1) \leq \eta V + \mu_{max}^{in} \leq U_{max}$.

Finally, if the queue $U_r^{(d)}(t)$ does not receive data from a client during slot t, but *does* receive data from other router nodes, then the differential backlog $\hat{W}_{ar}^{(d)}(t)$ must be positive for some other router node $a \in \mathcal{R}$. By (24), it must be that $U_r^{(d)}(t) < \eta V$. It thus again follows that $U_r^{(d)}(t+1) \leq \eta V + \mu_{max}^{in}$, proving the result. □

Proof: (Theorem 1 part (b)) Consider a virtual power queue $X_c(t)$ for any particular client $c \in \mathcal{C}$. Suppose that $X_c(t) \leq X_{max}$ at a particular time t (this clearly holds at $t = 0$). We show the same holds for time $t + 1$. If this queue does not increase during slot t, then clearly $X_c(t+1) \leq X_{max}$. Otherwise, this queue can only increase if $P_c(t) > 0$, that is, if client c is scheduled either to receive or transmit during slot t (see the virtual queue dynamic equation (17)). If it is scheduled to transmit (so that $P_c(t) = P_{tran}$), then by the uplink/downlink scheduling algorithm specified in (20), it must be the case that $\hat{W}_{c,r}^{(d)}(t)\mu_{cr}^{(d)}(t)\phi_{cr}(\mu_{cr}(t), S_{cr}(t)) - X_c(t)P_{tran} \geq 0$ for a particular commodity d (where $r = r(c,t)$). In particular, because probabilities are upper bounded by 1:

$$\hat{W}_{c,r}^{(d)}(t)\mu_{max}^{out} \geq X_c(t)P_{tran}$$

But $\hat{W}_{cr}^{(d)}(t) \leq U_c^{(d)}(t) \leq U_{max}$ for all t. Therefore, the virtual power queue cannot increase if $X_c(t) > U_{max}\mu_{max}^{out}/P_{tran}$. It follows that $X_c(t+1) \leq U_{max}\mu_{max}^{out}/P_{tran} + P_{tran} \leq X_{max}$.

Finally, if this client is scheduled to *receive* (so that $P_c(t) = P_{rec}$), and if $P_{rec} > 0$, then it must be that $U_{r(c,t)}^{(c)}(t)\mu_{r(c,t),c}^{(c)}(t)\phi_{r(c,t),c}(\mu_{r(c,t),c}(t), S_{r(c,t),c}(t)) - X_c(t)P_{rec} \geq 0$. In particular:

$$\begin{aligned} X_c(t) &\leq U_{r(c,t)}^{(c)}(t)\mu_{max}^{out}/P_{rec} \\ &\leq U_{max}\mu_{max}^{out}/P_{rec} \end{aligned}$$

where the last inequality holds because queue backlogs are bounded by U_{max} for all t (from part (a)). Therefore, we again have $X_c(t+1) \leq X_c(t) + P_{rec} \leq X_{max}$, proving the result. □

Parts (c)-(f) of the theorem are proven in the next section using Lyapunov optimization techniques.

V. Stochastic Lyapunov Optimization

Let $\boldsymbol{Q}(t) = (Q_1(t), \ldots, Q_K(t))$ be a vector process of queue lengths for a discrete time stochastic queueing network with K queues (possibly including some virtual queues like the flow state and power queues defined in the previous subsections). Let $L(\boldsymbol{Q})$ be any non-negative scalar valued function of the queue lengths, called a *Lyapunov function*. Throughout this paper, we shall use the particular Lyapunov function $L(\boldsymbol{Q}) = \frac{1}{2}\sum_{i=1}^K U_i^2$, so that a large value of $L(\boldsymbol{Q}(t))$ represents a large network congestion during slot t. However, the following theorem uses any $L(\boldsymbol{Q})$ that is non-negative. Define the *conditional Lyapunov drift* $\Delta(\boldsymbol{Q}(t))$ as follows:[3]

$$\Delta(\boldsymbol{Q}(t)) \triangleq \mathbb{E}\left\{L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t)) \mid \boldsymbol{Q}(t)\right\} \quad (26)$$

Suppose that the network accumulates "rewards" every timeslot (where rewards might correspond to utility measures of control actions). Assume rewards are real valued, and let the stochastic process $f(t)$ represent the reward earned during slot t. Let $f^*(t)$ represent some target reward process. The following result from [12][11] specifies a drift condition that ensures the time average of the reward process $f(t)$ is close to meeting or exceeding the time average of $f^*(t)$:

Theorem 2: (Lyapunov Optimization [12][11]) Suppose there exist constants $V > 0$, $B > 0$, $\epsilon > 0$, and a non-negative function $L(\boldsymbol{Q})$ such that for every timeslot t and every possible value of $\boldsymbol{Q}(t)$, the conditional Lyapunov drift satisfies:

$$\Delta(\boldsymbol{Q}(t)) - V\mathbb{E}\left\{f(t) \mid \boldsymbol{Q}(t)\right\} \leq B - \epsilon\mathbb{E}\left\{h(t) \mid \boldsymbol{Q}(t)\right\} - V\mathbb{E}\left\{f^*(t) \mid \boldsymbol{Q}(t)\right\}$$

where $h(t)$ is some non-negative stochastic process that may depend on the queue state, then we have:

$$\begin{aligned} \liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{f(\tau)\right\} &\geq \liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{f^*(\tau)\right\} - \frac{B}{V} \\ \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{h(\tau)\right\} &\leq \frac{B + V\overline{(f-f^*)}}{\epsilon} \end{aligned}$$

where $\overline{(f-f^*)}$ is defined:

$$\overline{(f-f^*)} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{f(\tau) - f^*(\tau)\right\} \quad \square$$

[3]Strictly speaking, we should represent this conditional Lyapunov drift with the notation $\Delta(\boldsymbol{Q}(t), t)$, as the stochastics might also depend on the timeslot t. However, we use the simpler notation $\Delta(\boldsymbol{Q}(t))$ as a formal representation of the right hand side of (26).

From the theorem, it is clear that if the parameter V can be chosen arbitrarily large while still ensuring the drift condition of Theorem 2 is satisfied, then the time average of $f(t)$ can be pushed arbitrarily close to (or above) the time average of the target reward process $f^*(t)$, with a corresponding tradeoff in the time average of the process $h(t)$ that is linear in V. Using quadratic Lyapunov functions tends to yield $h(t)$ processes of the form $h(t) = \sum_{i=1}^{K} Q_i(t)$, in which case the V parameter directly affects an upper bound on time average congestion (and hence, by Little's Theorem, average delay). This can be seen more explicitly in cases when reward functions are bounded by constants f_{max}, f_{min}, so that $f(t) \leq f_{max}$ and $f^*(t) \geq f_{min}$ for all t. In this case the upper bound on the time average of $h(t)$ is given by $(B + V(f_{max} - f_{min}))/\epsilon$.

In the next subsections, we use the Lyapunov Optimization Theorem (Theorem 2) to prove the remaining portions of Theorem 1. We do so by comparing the Lyapunov drift of the CLC-Mesh algorithm described in Section IV-A to the drift of a stationary randomized control algorithm that makes decisions independent of queue backlog.

A. *Computing the Drift for CLC-Mesh*

Let $\boldsymbol{Q}(t)$ represent the collection of virtual and actual queue states $[\boldsymbol{U}(t), \boldsymbol{X}(t), \boldsymbol{Y}(t)]$ for the CLC-Mesh algorithm. Define the Lyapunov function:

$$L(\boldsymbol{Q}) = \frac{1}{2}\left[\sum_{n,d}(U_n^{(d)})^2 + \sum_{c}(X_c)^2 + \sum_{c,d}(Y_c^{(d)})^2\right]$$

Define $\Delta(\boldsymbol{Q}(t))$ as the conditional Lyapunov drift, and define $\Delta_U(\boldsymbol{Q}(t))$, $\Delta_X(\boldsymbol{Q}(t))$, $\Delta_Y(\boldsymbol{Q}(t))$ respectively as the drift components associated with the actual queues, virtual power queues, and flow state queues, respectively, so that:

$$\Delta(\boldsymbol{Q}(t)) = \Delta_U(\boldsymbol{Q}(t)) + \Delta_X(\boldsymbol{Q}(t)) + \Delta_Y(\boldsymbol{Q}(t))$$

To compute the drifts, the following inequality shall be useful. For any non-negative values u, μ, a, we have:

$$\frac{1}{2}\left(\max[u-\mu, 0] + a\right)^2 \leq \frac{u^2 + \mu^2 + a^2}{2} - u(\mu - a)$$

Using the above inequality with the dynamics (17) for the virtual power queues $X_c(t)$ and the dynamics (12) for the flow state queues $Y_c^{(d)}(t)$, it is not difficult to show:

$$\Delta_X(\boldsymbol{Q}(t)) \leq B_X - \sum_{c} X_c(t)\mathbb{E}\left\{P_c^{av} - P_c(t) \mid \boldsymbol{Q}(t)\right\} \quad (27)$$

$$\Delta_Y(\boldsymbol{Q}(t)) \leq B_Y - \sum_{c,d} Y_c^{(d)}(t) \times \mathbb{E}\left\{\alpha_c^{(d)}(t) - \gamma_c^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \quad (28)$$

where

$$B_X \triangleq \frac{1}{2}\left[|\mathcal{C}|\max[P_{tran}^2, P_{rec}^2] + \sum_{c\in\mathcal{C}}(P_c^{av})^2\right]$$

$$B_Y \triangleq \frac{\alpha_{max}^2\left[|\mathcal{C}| + |\mathcal{A}|\right]}{2}$$

where $|\mathcal{C}|$ is the number of clients, and $|\mathcal{A}|$ is the number of active sessions $A_c^{(d)}(t)$ in the network. Similarly, the drift $\Delta_U(\boldsymbol{Q}(t))$ can be derived as the sum of drift terms $\Delta_{U_c}(\boldsymbol{Q}(t))$ and $\Delta_{U_r}(\boldsymbol{Q}(t))$ corresponding to client and router nodes (using queueing dynamics (5) and (7)):

$$\Delta_{U_c}(\boldsymbol{Q}(t)) \leq B_1 - \sum_{c\in\mathcal{C},d} U_c^{(d)}(t) \times \mathbb{E}\left\{\tilde{\mu}_{c,r(c,t)}^{(d)}(t) - \alpha_c^{(d)}(t) \mid \boldsymbol{Q}(t)\right\}$$

$$\Delta_{U_r}(\boldsymbol{Q}(t)) \leq B_2 - \sum_{r\in\mathcal{R},d} U_r^{(d)}(t) \times \mathbb{E}\left\{\sum_b \tilde{\mu}_{rb}^{(d)}(t) - \sum_a \tilde{\mu}_{ar}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\}$$

where

$$B_1 \triangleq |\mathcal{C}|\left[(\mu_{max}^{out})^2 + \alpha_{max}^2\right]/2$$

$$B_2 \triangleq |\mathcal{R}|\left[(\mu_{max}^{out})^2 + (\mu_{max}^{in})^2\right]/2$$

Recall that $\mu_{ab}^{(d)}(t)$ is a transmission rate that may include both actual and idle fill packets of commodity d transmitted over link (a, b), while $\tilde{\mu}_{ab}^{(d)}(t)$ represents the number of *actual* packets sucessfully received from this transmission. It is useful to define $\hat{\mu}_{ab}^{(d)}(t)$ to be the number of *actual or idle fill packets* successfully received (where an idle fill packet is viewed as being "successful" with the same probability as the actual packets for that transmission). It follows that:

$$\tilde{\mu}_{ab}^{(d)}(t) \leq \hat{\mu}_{ab}^{(d)}(t) \quad (29)$$

Further, because idle packets will not be sent if there are enough actual packets available:

$$\tilde{\mu}_{ab}^{(d)}(t) = \hat{\mu}_{ab}^{(d)}(t) \ \text{ if } U_a^{(d)}(t) \geq \mu_{max}^{out} \quad (30)$$

Using (29) and (30) in the $\Delta_{U_c}(\boldsymbol{Q}(t))$ and $\Delta_{U_r}(\boldsymbol{Q}(t))$ expressions yields:

$$\Delta_{U_c}(\boldsymbol{Q}(t)) \leq B_1 + |\mathcal{C}|\,(\mu_{max}^{out})^2 - \sum_{c\in\mathcal{C},d} U_c^{(d)}(t) \times \mathbb{E}\left\{\hat{\mu}_{c,r(c,t)}^{(d)}(t) - \alpha_c^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \quad (31)$$

$$\Delta_{U_r}(\boldsymbol{Q}(t)) \leq B_2 + |\mathcal{R}|\,(\mu_{max}^{out})^2 - \sum_{r\in\mathcal{R},d} U_r^{(d)}(t) \times \mathbb{E}\left\{\sum_b \hat{\mu}_{rb}^{(d)}(t) - \sum_a \hat{\mu}_{ar}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \quad (32)$$

Therefore, summing the right hand sides of (27), (28), (31), and (32), we have the following Lyapunov drift bound:

$$\begin{aligned}\Delta(\boldsymbol{Q}(t)) - V\mathbb{E}\left\{\textstyle\sum_{c,d} g_c^{(d)}(\gamma_c^{(d)}(t)) \mid \boldsymbol{Q}(t)\right\} &\leq \tilde{B} \\ &- \textstyle\sum_c X_c(t)\mathbb{E}\left\{P_c^{av} - P_c(t) \mid \boldsymbol{Q}(t)\right\} \\ &- \textstyle\sum_{c,d} Y_c^{(d)}(t)\mathbb{E}\left\{\alpha_c^{(d)}(t) - \gamma_c^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \\ &- \textstyle\sum_{c\in\mathcal{C},d} U_c^{(d)}(t)\mathbb{E}\left\{\hat{\mu}_{c,r(c,t)}^{(d)}(t) - \alpha_c^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \\ &- \textstyle\sum_{r\in\mathcal{R},d} U_r^{(d)}(t)\mathbb{E}\left\{\sum_b \hat{\mu}_{rb}^{(d)}(t) - \sum_a \hat{\mu}_{ar}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \\ &- V\mathbb{E}\left\{\textstyle\sum_{c,d} g_c^{(d)}(\gamma_c^{(d)}(t)) \mid \boldsymbol{Q}(t)\right\} \quad (33)\end{aligned}$$

where we have subtracted the utility metric from both sides, and where:

$$\tilde{B} \triangleq B_X + B_Y + B_1 + B_2 + |\mathcal{C}|(\mu_{max}^{out})^2 + |\mathcal{R}|(\mu_{max}^{out})^2$$

It turns out that the cross layer control algorithm is designed to *come within an additive constant C of minimizing the right hand side of (33) over all alternative control actions*, where C is a constant that does not depend on V. To see this, note that the $\alpha_c^{(d)}(t)$ and $\gamma_c^{(d)}(t)$ variables affect the right hand side of (33) only through terms of the form:

$$\sum_{c,d}[U_c^{(d)}(t) - Y_c^{(d)}(t)]\mathbb{E}\left\{\alpha_c^{(d)}(t) \mid \boldsymbol{Q}(t)\right\}$$

and

$$\sum_{c,d}\mathbb{E}\left\{Y_c^{(d)}(t)\gamma_c^{(d)}(t) - Vg_c^{(d)}(\gamma_c^{(d)}(t)) \mid \boldsymbol{Q}(t)\right\}$$

and hence the flow control algorithm in (13) and (14) chooses decision variables that minimize these terms over all alternative flow control actions that satisfy $0 \leq \gamma_c^{(d)}(t) \leq \alpha_{max}$ as well as the general flow control constraints of problem (13). That the remaining control variables come within an additive constant C of minimizing the right hand side is proven in Appendix A, where the value C is also computed.

It follows that the drift expression on the left hand side of (33) is less than or equal to the resulting expression when the right hand side of (33) is modified by replacing $\tilde{B}$ with $\tilde{B}+C$, and by replacing variables

$$\{P_c(t)\}, \{\alpha_c^{(d)}(t)\}, \{\hat{\mu}_{ab}^{(d)}(t)\}, \{\gamma_c^{(d)}(t)\}$$

with variables corresponding to any other feasible control actions:

$$\{P_c^*(t)\}, \{\alpha_c^{*(d)}(t)\}, \{\hat{\mu}_{ab}^{*(d)}(t)\}, \{\gamma_c^{*(d)}(t)\}$$

B. Replacing the Decision Variables

In [3], the capacity region for a general network is characterized. Specifically, it is shown that for any rate matrix within the capacity region, there exists a *stationary randomized* scheduling policy that chooses transmission rates independently of queue backlog, together with multi-commodity flows that support the traffic rates, such that the sum flow over any link is less than or equal to the expected link transmission rate with respect to the stationary randomized algorithm. This result can be extended to the scenario of this paper (which additionally considers average power constraints at each client and probabilistic reception) to yield the following lemma (proof omitted for brevity, see [3] [13] [18] for related proofs):

Lemma 1: (Existence of a Stationary Randomized Policy) Consider any given slot t with client topology state $\mathcal{T}(t)$ and instantaneous capacity region $\Lambda^*(t)$. For any input rate matrix $(x_c^{(d)}) \in \Lambda^*(t)$, there must exist a stationary and randomized control algorithm that chooses decision variables $P_c^*(t)$ and $\hat{\mu}_{ab}^{*(d)}(t)$ independently of queue backlog, and that yields (for all $d \in \mathcal{C}$, all t, and all $\boldsymbol{Q}(t)$):

$$\mathbb{E}\left\{P_c^*(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} \leq P_c^{av} \quad \forall c \in \mathcal{C} \quad (34)$$

$$\mathbb{E}\left\{\hat{\mu}_{c,r(c,t)}^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} = x_c^{(d)} \quad \forall c \in \mathcal{C} \quad (35)$$

$$\mathbb{E}\left\{\sum_b \hat{\mu}_{rb}^{*(d)}(t) - \sum_a \hat{\mu}_{ar}^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} = 0 \quad \forall r \in \mathcal{R} \quad (36)$$

Note that, because these stationary randomized decisions are independent of queue backlog, the conditional expectation with respect to $\boldsymbol{Q}(t)$ could be replaced by an unconditional expectation. The inequalities and equalities in the theorem can be intuitively understood as follows: Inequality (34) implies that the stationary randomized power allocations $P_c^*(t)$ yield expected power expenditures that satisfy the power constraints for all clients c. Next, we can define $f_{ab}^{*(d)} \triangleq \mathbb{E}\left\{\hat{\mu}_{ab}^{*(d)}(t)\right\}$, so that these expected transmission rates can be intuitively interpreted as *multi-commodity flows* $f_{ab}^{*(d)}$. With this interpretation, the equality (35) states that the flow of commodity d out of any client is equal to the input rate $x_c^{(d)}$. Likewise, the inequality (36) can be interpreted as a conservation equality stating that the sum flow of commodity d data into any router node r is equal to the sum outflow of that commodity. It is important here to recall that the instantaneous capacity region $\Lambda^*(t)$ is defined in terms of a virtual network with a constant topology state for all time (given by $\mathcal{T}(t)$). Hence, although the above expectations are conditioned on $\mathcal{T}(t)$, by the law of large numbers they would represent the true time average powers and transmission rates if this stationary randomized policy were implemented every slot on such a virtual network.

Now consider a particular timeslot t, with client topology state $\mathcal{T}(t)$ and instantaneous capacity region $\Lambda^*(t)$. Let $(x_c^{*(d)}(t))$ represent the optimal variables that solve (11), so that $g^*(t) = \sum_{c,d} g_c^{(d)}(x_c^{*(d)}(t))$. It follows that $(x_c^{*(d)}(t)) \in \Lambda^*(t)$ and $x_c^{*(d)}(t) \leq \lambda_c^{(d)}$ for all (c,d). Consider the simple (and feasible) strategy that chooses the stationary randomized decisions $P_c^*(t)$ and $\hat{\mu}_{ab}^{*(d)}(t)$ to yield (34)-(36) with respect to the rate matrix $(x_c^{*(d)}(t))$, and additionally chooses flow control decision variables:

$$\gamma_c^{*(d)}(t) = x_c^{*(d)}(t)$$

$$\alpha_c^{*(d)}(t) = \begin{cases} A_c^{(d)}(t) & \text{with probability } x_c^{*(d)}(t)/\lambda_c^{(d)} \\ 0 & \text{otherwise} \end{cases}$$

It follows that:

$$\begin{aligned}
\gamma_c^{*(d)}(t) &= x_c^{*(d)}(t) \\
\mathbb{E}\left\{\alpha_c^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &= x_c^{*(d)}(t) \\
\mathbb{E}\left\{P_c^*(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &\leq P_c^{av} \\
\mathbb{E}\left\{\hat{\mu}_{c,r(c,t)}^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &= x_c^{*(d)}(t) \\
\mathbb{E}\left\{\sum_b \hat{\mu}_{rb}^{*(d)}(t) - \sum_a \hat{\mu}_{ar}^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &= 0
\end{aligned}$$

Taking expectations of both sides of the above equalities and inequalities with respect to the conditional distribution of $\mathcal{T}(t)$ given $\boldsymbol{Q}(t)$ and using the law of iterated expectations removes the conditioning on $\mathcal{T}(t)$. Plugging the resulting expectations of this alternative strategy into the right hand side of the drift expression (33) thus creates many terms that can be cancelled, and yields the following drift bound for the CLC-Mesh algorithm:

$$\Delta(\boldsymbol{Q}(t)) - V\mathbb{E}\left\{\sum_{c,d} g_c^{(d)}(\gamma_c^{(d)}(t)) \mid \boldsymbol{Q}(t)\right\} \leq \tilde{B} + C - V\mathbb{E}\left\{g^*(t) \mid \boldsymbol{Q}(t)\right\} \quad (37)$$

where we have used the fact that $g^*(t) \triangleq \sum_{c,d} g_c^{(d)}(x_c^{*(d)}(t))$. The above expression is in the exact form for application of the Lyapunov Optimization Theorem (Theorem 2), and hence we have:

$$\liminf_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c,d}\mathbb{E}\left\{g_c^{(d)}(\gamma_c^{(d)}(\tau))\right\} \geq \liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{g^*(\tau)\right\} - \frac{\tilde{B}+C}{V}$$

Using concavity of the utility functions $g_c^{(d)}(\cdot)$ together with Jensen's inequality yields:

$$\liminf_{t\to\infty}\sum_{c,d} g_c^{(d)}(\overline{\gamma}_c^{(d)}(t)) \geq \liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{g^*(\tau)\right\} - \frac{\tilde{B}+C}{V}$$

where

$$\overline{\gamma}_c^{(d)}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\gamma_c^{(d)}(\tau)\right\}$$

However, note that the $\gamma_c^{(d)}(t)$ variables are inputs to the flow state queues $Y_c^{(d)}(t)$, and the $\alpha_c^{(d)}(t)$ variables represent the time varying service rate of these queues. Because $Y_c^{(d)}(t) \leq Y_{max} \triangleq \eta V + \alpha_{max}$ for all t (see Claim 1 in proof of Theorem 1 part (a)), we have for all t:

$$\sum_{\tau=0}^{t-1}\gamma_c^{(d)}(t) \leq \sum_{\tau=0}^{t-1}\alpha_c^{(d)}(t) + Y_{max}$$

Indeed, the above inequality follows because $\alpha_c^{(d)}(t)$ can be viewed as the server rate process of the virtual queue $Y_c^{(d)}(t)$ defined in (12), and thus the total sum of all service rates plus the worst case backlog Y_{max} must be larger than or equal to the total sum of all arrivals. Taking expectations, dividing by t, and using the fact that $\alpha_c^{(d)}(t) \geq 0$ yields:

$$\overline{\alpha}_c^{(d)}(t) \geq \max[\overline{\gamma}_c^{(d)}(t) - Y_{max}/t, 0]$$

Because utilities are non-decreasing, with largest right derivative η, we have for all $g_c^{(d)}(\cdot)$:

$$\begin{aligned} g_c^{(d)}(\overline{\alpha}_c^{(d)}(t)) &\geq g_c^{(d)}(\max[\overline{\gamma}_c^{(d)}(t) - Y_{max}/t, 0]) \\ &\geq g_c^{(d)}(\overline{\gamma}_c^{(d)}(t)) - \eta Y_{max}/t \end{aligned}$$

Therefore

$$\begin{aligned} \liminf_{t\to\infty}\sum_{c,d} g_c^{(d)}(\overline{\alpha}_c^{(d)}(t)) &\geq \liminf_{t\to\infty}\sum_{c,d} g_c^{(d)}(\overline{\gamma}_c^{(d)}(t)) \\ &\geq \liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{g^*(\tau)\right\} \\ &\quad -(\tilde{B}+C)/V \end{aligned}$$

proving the result of part (c) of Theorem 1.

C. Proof of Part (d) of Theorem 1

Define the $\liminf$ admission rate into queue (c,d) as follows:

$$\overline{\alpha}_c^{(d)} \triangleq \liminf_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\alpha_c^{(d)}(\tau)\right\}$$

Here we show that if the conditions of part (d) of Theorem 1 hold, then $\overline{\alpha}_c^{(d)} \geq \lambda_c^{(d)} - O(1/V)$ for all (c,d). First note that if $\boldsymbol{\lambda} \in \Lambda^*(t)$ for all t, then from (11) it is clear that:

$$g^*(t) = g^* \triangleq \sum_{c,d} g_c^{(d)}(\lambda_c^{(d)}) \quad \text{for all } t$$

Part (c) of Theorem 1 thus implies the $\liminf$ of time average utility is within B/V of g^*:

$$g^* - B/V \leq \liminf_{t\to\infty}\sum_{c,d} g_c^{(d)}(\overline{\alpha}_c^{(d)}(t)) \quad (38)$$

However, because for all t and all (c,d) we have

$$\sum_{\tau=0}^{t-1}\alpha_c^{(d)}(\tau) \leq \sum_{\tau=0}^{t-1}A_c^{(d)}(\tau)$$

it follows that the $\limsup$ admitted rate is less than or equal to $\lambda_c^{(d)}$:

$$\overline{\alpha}_c^{(d)} \leq \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\alpha_c^{(d)}(\tau)\right\} \leq \lambda_c^{(d)} \quad (39)$$

Using (39) together with the fact that utilities are continuous and non-decreasing, it follows that for any particular utility function (a,b), we have:

$$\liminf_{t\to\infty}\sum_{c,d} g_c^{(d)}(\overline{\alpha}_c^{(d)}(t)) \leq g_a^{(b)}(\overline{\alpha}_a^{(b)}) + \sum_{(c,d)\neq(a,b)} g_c^{(d)}(\lambda_c^{(d)})$$

Using this inequality in (38) yields:

$$\begin{aligned} g^* - B/V &\leq g_a^{(b)}(\overline{\alpha}_a^{(b)}) + \sum_{(c,d)\neq(a,b)} g_c^{(d)}(\lambda_c^{(d)}) \\ &= g^* - \left[g_a^{(b)}(\lambda_a^{(b)}) - g_a^{(b)}(\overline{\alpha}_a^{(b)})\right] \quad (40) \end{aligned}$$

where (40) holds by definition of g^*. Now define ν as the smallest derivative of any utility function over the interval $0 \leq \alpha \leq \alpha_{max}$. The assumptions in part (d) of Theorem 1 imply that $\nu > 0$. Using the fact that $\lambda_a^{(b)} \geq \overline{\alpha}_a^{(b)}$ in the right hand side of (40) yields:

$$g^* - B/V \leq g^* - \nu[\lambda_a^{(b)} - \overline{\alpha}_a^{(b)}]$$

Therefore, for any (a,b) we have:

$$\overline{\alpha}_a^{(b)} \geq \lambda_a^{(b)} - B/(V\nu)$$

proving part (d) in Theorem 1.

D. Proof of parts (e) and (f) of Theorem 1

Part (e) is proven in a manner similar to the proof technique in Section V-B. Specifically, suppose $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \tilde{\Lambda}^*(t)$ for all t. Then we can replace the control decisions with alternative feasible control decisions, and it can be shown that there exist corresponding variables that yield:

$$\begin{aligned} \gamma_c^{*(d)}(t) &= \lambda_c^{(d)} \\ \mathbb{E}\left\{\alpha_c^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &= \lambda_c^{(d)} \\ \mathbb{E}\left\{P_c^*(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &\leq P_c^{av} \\ \mathbb{E}\left\{\hat{\mu}_{c,r(c,t)}^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &= \lambda_c^{(d)} + \epsilon \\ \mathbb{E}\left\{\sum_b \hat{\mu}_{rb}^{*(d)}(t) - \sum_a \hat{\mu}_{ar}^{*(d)}(t) \mid \boldsymbol{Q}(t), \mathcal{T}(t)\right\} &= \epsilon \end{aligned}$$

The only difference between the above expressions for the alternative control variables and those of Section V-B is that the transmission rate out of each client is increased by an amount ϵ, and the difference between transmission rates out of and into each router node r is now equal to ϵ (rather than zero). This is feasible because $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \tilde{\Lambda}^*(t)$, and hence the network can support an extra rate ϵ added to the original sources at each client, as well as new exogenous sources of rate ϵ added at each router node (for each commodity $d \in \mathcal{C}$).

Following the reasoning of Section V-B exactly, we can use these values to replace the right hand side of the drift expression (33) to yield (compare with (37)):

$$\begin{aligned} \Delta(\boldsymbol{Q}(t)) - V\mathbb{E}\left\{\sum_{c,d} g_c^{(d)}(\gamma_c^{(d)}(t)) \mid \boldsymbol{Q}(t)\right\} &\leq \tilde{B} + C \\ - \sum_{a \in \mathcal{C} \cup \mathcal{R}, d} U_a^{(d)}(t)\epsilon &- V\sum_{c,d} g_c^{(d)}(\lambda_c^{(d)}) \end{aligned}$$

The above drift expression is in the exact form for application of the Lyapunov Optimization Theorem (Theorem 2), and hence we have:

$$\begin{aligned} \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{a,d}\mathbb{E}\left\{U_a^{(d)}(\tau)\right\} \leq \frac{\tilde{B}+C}{\epsilon} \\ + \frac{V}{\epsilon}\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{c,d}\mathbb{E}\left\{g_c^{(d)}(\gamma_c^{(d)}(\tau)) - g_c^{(d)}(\lambda_c^{(d)})\right\} \end{aligned} \quad (41)$$

However, by concavity of each utility function $g_c^{(d)}(\cdot)$ we have:

$$\begin{aligned} \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{g_c^{(d)}(\gamma_c^{(d)}(\tau)) - g_c^{(d)}(\lambda_c^{(d)})\right\} \\ \leq \limsup_{t\to\infty} g_c^{(d)}(\overline{\gamma}_c^{(d)}(t)) - g_c^{(d)}(\lambda_c^{(d)}) \\ = g_c^{(d)}(\limsup_{t\to\infty}\overline{\gamma}_c^{(d)}(t)) - g_c^{(d)}(\lambda_c^{(d)}) \end{aligned} \quad (42)$$

where $\overline{\gamma}_c^{(d)}(t) \triangleq \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\left\{\gamma_c^{(d)}(\tau)\right\}$. The final equality holds because utilities are non-decreasing and continuous, and the variables $\overline{\gamma}_c^{(d)}(t)$ are in the interval $[0, \alpha_{max}]$ for all t. Because the $Y_c^{(d)}(t)$ queues are bounded by a finite Y_{max}, it follows that the $\limsup$ expected time average of the $\gamma_c^{(d)}(t)$ process is no more than the $\limsup$ expected time average of the $\alpha_c^{(d)}(t)$ process, which can be no more than $\lambda_c^{(d)}$. Hence, the right hand side of (42) is less than or equal to 0, which implies the final term of the right hand side of (41) is less than or equal to zero (because the $\limsup$ of a sum is less than or equal to the sum of $\limsup$s). This implies:

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{a,d}\mathbb{E}\left\{U_a^{(d)}(\tau)\right\} \leq \frac{\tilde{B}+C}{\epsilon} \quad (43)$$

proving the result of part (e) of Theorem 1.

The statement of part (f) is almost identical to that of part (c), with the exception that the system is assumed to have ergodic mobility patterns with a well defined steady state. The proof uses the Lyapunov Optimization Theorem (Theorem 2) in exactly the same manner as in the proof of part (c), with the exception that instead of using the one-step conditional drift, we use the K-slot Lyapunov drift, defined in [3], which gives rise to a constant B_{mobile} that depends on the time required for the mobility process to reach "near steady state" (formalized in [3]). Indeed, combining the proof techniques of [3] with that of part (c) of Theorem 1 establishes the result of part (f).

VI. Channel Blind Scheduling and Multi-Receiver Diversity

Here we consider two approaches to channel-blind scheduling. The first uses the same CLC-Mesh algorithm of Section IV-A, while the second augments the algorithm by partially removing the cell structure and incorporating multi-receiver diversity.

A. Using an Estimated Error Probability

Suppose that each client $c \in \mathcal{C}$ transmits either 0 or 1 packet per timeslot, so that $\mu_{cr}(t) \in \{0, 1\}$ for all uplink channels (c, r) and all t. Each router node $r \in \mathcal{R}$ uses a downlink transmission rate that is either 0 or k packets/slot (for some positive integer k), so that $\mu_{rc}(t) \in \{0, k\}$ for all downlink channels (r, c) and all t. Rather than measuring exact channel states, each router node r maintains a success probability estimate ϕ_r^{up} and ϕ_r^{down}, representing success probabilities in the cell of router r for uplink and downlink transmissions. These probabilities can easily be estimated based on long-term ACK/NACK history. The CLC-Mesh algorithm can then be implemented exactly as before using these probabilities as the actual channel success probabilities. If this system is modeled as if uplink and downlink successes in each cell $r \in \mathcal{R}$ are i.i.d. over timeslots with probability ϕ_r^{up} and ϕ_r^{down}, then this is a special case of the basic model specified in Section II. Hence, the same analytical results of Theorem 1 hold, where capacity regions are re-defined in terms of these new system assumptions. Of course, the ergodic capacity and instantaneous capacity regions under this channel-blind model are generally smaller than those of a model where channel state information gives more detail about current channel states.

B. Multi-Receiver Diversity

Suppose that the control algorithm in the mesh router subnetwork is the same as before, but that uplink and downlink transmissions are chosen randomly and without any channel state measurements. Specifically, every slot each router node $r \in \mathcal{R}$ independently enters *downlink mode* with probability q_r^{down}, i.i.d. over each slot and independent of the other downlinks. If node r enters downlink mode at the beginning of a timeslot, it sends a control signal to all clients within its cell that restricts them from transmitting. Otherwise, each client c in the cell of router node r enters *transmission mode* with probability q_c, i.i.d. over all timeslots in which the router of their current cell is not in downlink mode, and independent across clients. We assume that routers transmit either 0 or k packets in downlink mode (so that $\mu_{rc}(t) \in \{0, k\}$ for all downlink channels (r, c)), and that clients transmit either 0 or 1 packet (so that $\mu_{cr}(t) \in \{0, 1\}$ for all uplink channels (c, r)). For simplicity, here we neglect the average power constraints at the clients, and we assume that each client c has only one active session $A_c^{(d_c)}(t)$.

If a router node r transmits k packets over a downlink transmission to a destination client in its cell, each of these packets is received with a success probability given by the current channel state (note that the transmitter does not need to *measure* this channel state ahead of time). If a client c transmits a packet, that packet is received at the router node of its cell with a probability that may depend on the number of other clients transmitting in that cell. However, we assume it is also possible for one or more router nodes of *an adjacent cell* to successfully overhear the transmission (with some probability that depends on the distance between client and router node, the number of other clients transmitting, and the probability that this other router node is in uplink mode). We assume that all transmissions are ACKed or NACKed at the end of the timeslot (the absence of an ACK is considered equivalent to a NACK). Allowing multiple router nodes to overhear the same transmission creates a *multi-receiver diversity gain*, and can significantly increase the probability that a packet is successfully delivered to at least one of the mesh routers, particularly when clients are close to several router nodes simultaneously.

It remains only to specify the action taken when a client node transmits a packet and receives ACK information (possibly from one or more router nodes). Let $c \in \mathcal{C}$ represent a client that has just transmitted a single packet, and recall that this client transmits packets only from commodity d_c. If it receives no ACK, it assumes their were no successful receivers and hence keeps the packet in its queue for a future attempt. If it receives ACKs from one or more mesh router nodes r, it selects the successful router node r with the largest positive differential backlog metric $\hat{W}_{cr}^{(d_c)}(t)$, defined in (19), breaking ties arbitrarily. We assume each router node r that successfully heard the client transmission includes its queue backlog $U_r^{(d_c)}(t)$ in its ACK message, so that the $\hat{W}_{cr}^{(d_c)}(t)$ metric is computed at the clients quite easily. As a final part of the timeslot, the client then informs the router node r with the largest positive $\hat{W}_{cr}^{(d_c)}(t)$ value to take charge of the packet (using a reliable control channel). In this case, the client then removes the packet from its queue while the winning router node adds the packet to its queue. The other successful router nodes simply delete the packet (not including it in their backlog). If no successful router nodes r have positive differential backlog metrics $\hat{W}_{cr}^{(d_c)}(t)$, the client simply retains responsibility of the packet and retransmits at some future time. Note that this algorithm ensures every packet is in at most one queue at the end of every timeslot.

It can be shown that the structure of the algorithm maintains the $U_a^{(b)}(t) \leq U_{max}$ bound for all t and all client and router queues (a, b). This algorithm is a variant of the diversity backpressure routing algorithm DIVBAR developed in [18]. There, a Lyapunov drift argument is constructed and it is shown that this differential backlog index is an *optimal* way to pick the next hop node for each client transmission. It thus can be shown that, assuming control channel information and ACK/NACK feedback is perfect, the algorithm specified above achieves utility similar to the performance specified in parts (c)-(f) of Theorem 1), but where optimality is measured with respect to this new network structure. Specifically, instantaneous capacity, ergodic capacity, and utility optimality is measured over all possible scheduling and next-hop routing algorithms that use probabilities q_r^{down} and q_c with the probabilistic transmission and reception rules and multi-user diversity features described above. More precisely, we assume optimality is measured with respect to the assumption that nodes transmit whenever possible (using dummy packets if there are no actual packets), so that the time varying reception probabilities can be viewed as a time varying channel state that is independent of queue backlog (performance can of course be improved if dummy packets are not transmitted, but optimality is measured with respect to a larger capacity region in this case). Finally, we emphasize that optimality is with respect to the queueing structure as defined in this paper, and does not include cooperative communication or network coding strategies.

C. Hybrid Routing Policies

The multi-receiver diversity algorithm still maintains a partial cell structure for downlink transmissions. However, in cases when some clients may not be within range of *any* mesh router node, it may be necessary to have other clients relay data to and from this client. General policies of this form are considered in [2] [3] [12], and in [18] for general networks with multi-receiver diversity. Such approaches can also be incorporated into the mesh network paradigm of this paper. However, one disadvantage is that clients need to maintain separate queues for commodities associated with each other client. A simple *hierarchical* approach can be employed as a low complexity alternative, where we assume that clients desire to send their data to any router node, and that the router nodes are responsible for direct downlink delivery to the destinations. In this case, all data generated by clients can be treated as a *single commodity* that must be routed to any of the mesh routers, and this data is viewed as the exogenous arrivals to the mesh router sub-network. This keeps the client actions quite simple, but does not necessarily achieve network capacity (due to the fact that commodity information is ignored at the clients).

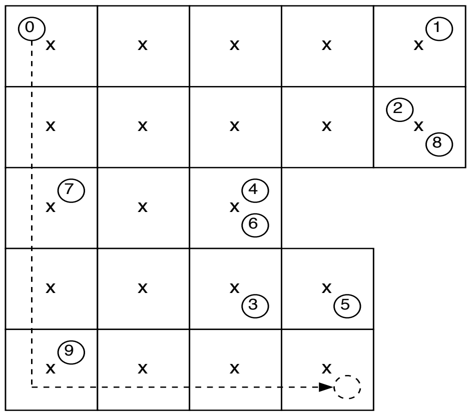

Fig. 1. An example mesh network with 21 router nodes and 10 clients.

VII. Simulations

For brevity, we provide a simple simulation of the basic CLC-Mesh algorithm for a cell partitioned network with 21 mesh router nodes, 21 cells, and 10 active clients (see Fig. 1). At most one uplink or downlink transmission can take place per cell per timeslot, and all transmissions send exactly one packet. We assume that $P_c^{tran} = 1$, $P_c^{rec} = 1$, and $P_c^{av} = 0.5$ for each client node $c \in \{0, \ldots, 9\}$. All uplink transmissions are successful with probability 0.8. Downlink transmissions are successful with probability 1.

The mesh network in this example is arranged according to a grid, and we assume mesh link scheduling takes place according to a 4-periodic cycle: On the first slot of the cycle, all mesh router nodes can transmit to the router node to their North (remaining idle if there is no other cell to the North). On the next three slots, transmission takes place in the West, South, and East directions (remaining idle when appropriate).[4] We use H_{ab} distances given by the shortest path hop count between mesh router node a and mesh router node b (so that the distance between the top right and the bottom right cells is equal to 7). We choose $\omega = 2$, so that mesh routers only transmit in a direction that deviates from the estimated shortest path if there is a backlog differential of more than 2 packets. To simplify the simulation, we assume client locations are exactly updated at every node every 8 timeslots.

We assume that each client is the source of a single independent session of rate λ, and that client source-destination pairs are specified as follows:

$$0 \leftrightarrow 1, 2 \leftrightarrow 3, \ldots, 8 \leftrightarrow 9$$

so that client 0 desires to send data at rate λ to client 1 and client 1 desires to send independent data at rate λ to client 0, client 2 desires to send data at rate λ to client 3 and client 3 desires to send data at rate λ to client 2, etc. Uplink/downlink transmissions for CLC-Mesh are thus scheduled by comparing (21) and (22). All exogenous inputs are independent Bernoulli processes, so that a single packet arrives i.i.d. with probability λ every timeslot. We assume there are no transport layer storage buffers, so that all packets that are not immediately admitted to the network layer at their source clients are necessarily dropped. Flow control is performed with respect to the logarithmic utilities in (9).

[4]Note that this 4-periodic cycle assumes that mesh router nodes can transmit and receive simultaneously. For example, an interior node can transmit a packet in the West direction while receiving a packet from the router to its East. This can easily be extended to an 8-periodic cycle in cases when the schedule requires transmission and reception to take place on separate slots.

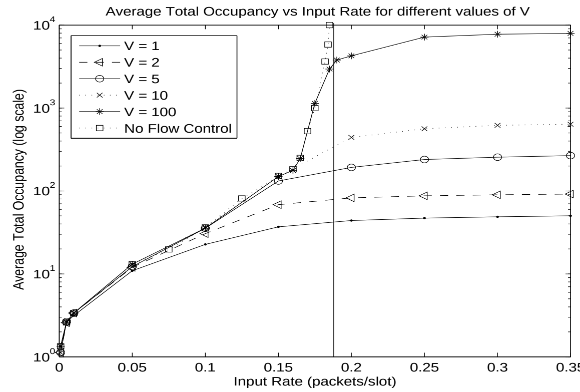

Fig. 2. Congestion versus input rate λ.

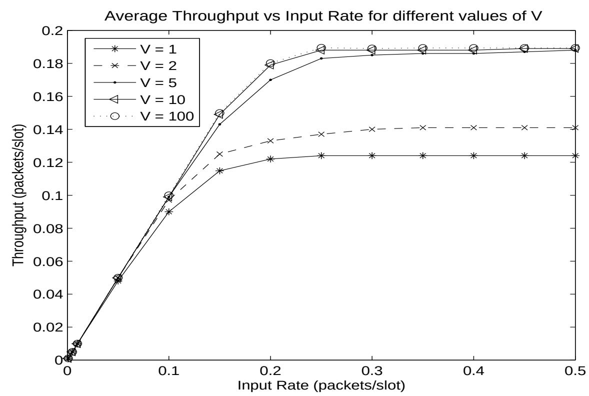

Fig. 3. Achieved throughput versus raw input rate λ for various V values.

All clients have initial locations as shown in Fig. 1. For our first experiment, we assume clients $c \in \{0, \ldots, 8\}$ are stationary, and that client 9 moves according to a Markov random walk, where every timeslot it stays in its same location with probability 0.5, and else it moves randomly one step in either the North, West, South, or East directions. If it decides to move in an infeasible direction, it stays in its same cell. In Fig. 2 we plot the average total occupancy (summing all packets in all queues of the network) versus the input rate λ. Each data point represents a simulation over 100,000 timeslots, and the different curves show values of the flow control parameter $V \in \{1, 2, 5, 10, 100\}$, and the case $V = \infty$ (no flow control) is also shown. The vertical asymptote appears roughly at $\lambda = 0.19$ for the simulations without flow control, which is the capacity of this network. Fig. 3 illustrates the resulting throughput versus the raw data input rate λ for various V parameters. The achieved throughput is almost identical to the input rate λ for small values of λ, and the throughput saturates at a value that depends on V, being very close to the 0.19 capacity level when V is large.

We next conduct a non-ergodic experiment: We fix $\lambda = 0.15$

(within the capacity region), $V = 10$, and simulate the system again for 100,000 timeslots. For the first $50,000$ slots, all nodes have the same mobility model as before. On timeslot $50,000$, node 1 moves from its initial location in the top left corner to a new location in the bottom right corner (arriving there after 7 slots), and remains in that location thereafter, as illustrated in Fig. 1. This single non-ergodic event takes place while the algorithm continues to run. Figs. 4 and 5 illustrate a moving average (over 1000 slots) of the total number of commodity 0 and commodity 1 packets in the network over the duration of the experiment. The network throughput does not diminish (yielding a throughput of 0.1495 packets/slot for each session, so that less than one percent of all data is dropped), and the system quickly reaches a "new" equilibrium with respect to the new topology state. It was also observed that the maximum number of packets in any queue at any instant of time was no more than $V + \max[2\alpha_{max}, \mu_{max}^{in}] = 12$ (as proven in Theorem 1).

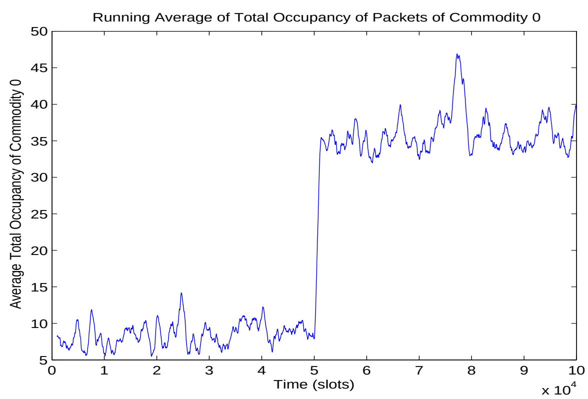

Fig. 4. Moving average number of commodity 0 packets versus time.

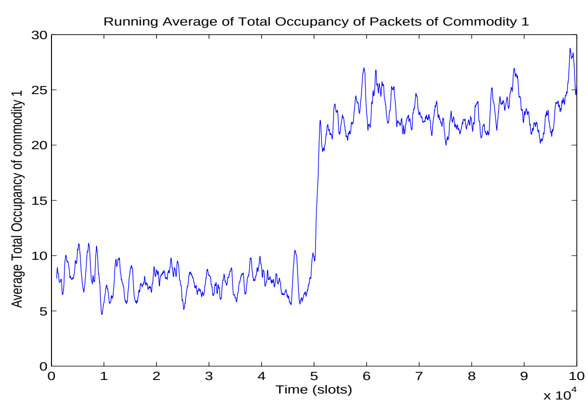

Fig. 5. Moving average number of commodity 1 packets versus time.

APPENDIX A – MINIMIZING DRIFT TO WITHIN C

In Section V-A, it was shown that the flow control actions of the cross layer control policy yield decision variables $\gamma_c^{(d)}(t)$ and $\alpha_c^{(d)}(t)$ that minimize their contribution to the right hand side of the drift expression (33). Similarly, here we show that the remaining control decision variables $P_c(t)$ and $\hat{\mu}_{ab}^{(d)}(t)$ affecting the right hand side of (33) come within an additive constant C of minimizing their contribution. These remaining terms in the right hand side of (33) can be re-written (switching the sums, and defining $\hat{\mu}_{ab}^{(d)}(t) = 0$ whenever no link (a, b) exists, and defining $U_c^{(c)}(t) = 0$ for all c):

$$\sum_c X_c(t)\mathbb{E}\left\{P_c(t) \mid \boldsymbol{Q}(t)\right\} - \sum_{a,b,d}\left[U_a^{(d)}(t) - U_b^{(d)}(t)\right]\mathbb{E}\left\{\hat{\mu}_{ab}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\}$$

The above expression can be further decomposed into a sum over downlinks, uplinks, and mesh router links:

$$\sum_c X_c(t)\mathbb{E}\left\{P_c(t) \mid \boldsymbol{Q}(t)\right\} \quad (44)$$

$$- \sum_{r\in\mathcal{R}, d\in\mathcal{C}_r(t)} U_r^{(d)}(t)\mathbb{E}\left\{\hat{\mu}_{rd}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \quad (45)$$

$$- \sum_{r\in\mathcal{R}, c\in\mathcal{C}_r(t), d}\left[U_c^{(d)}(t) - U_r^{(d)}(t)\right]\mathbb{E}\left\{\hat{\mu}_{cr}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \quad (46)$$

$$- \sum_{r\in\mathcal{R}, b\in\mathcal{R}, d}\left[U_r^{(d)}(t) - U_b^{(d)}(t)\right]\mathbb{E}\left\{\hat{\mu}_{rb}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \quad (47)$$

For simplicity of notation, define $\boldsymbol{\psi}(t) \triangleq [\boldsymbol{P}(t); (\hat{\mu}_{ab}^{(d)})]$ as the set of control decision variables used in the above expression (44)-(47), and let $f(\boldsymbol{\psi}(t))$ represent this expression. Define Ψ as the set of all feasible control choices. The goal is to show that the uplink/downlink transmission and mesh network routing algorithms of CLC-Mesh come within an additive constant of minimizing $f(\boldsymbol{\psi}(t))$ over all $\boldsymbol{\psi}(t) \in \Psi$. First define the following component functions:

$$f_1(\boldsymbol{\psi}(t)) \triangleq \sum_c X_c(t)\mathbb{E}\left\{P_c(t) \mid \boldsymbol{Q}(t)\right\} - \sum_{r\in\mathcal{R}, d\in\mathcal{C}_r(t)} U_r^{(d)}(t)\mathbb{E}\left\{\hat{\mu}_{rd}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} - \sum_{r\in\mathcal{R}, c\in\mathcal{C}_r(t), d} W_{cr}^{(d)}(t)\mathbb{E}\left\{\hat{\mu}_{cr}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\}$$

$$f_2(\boldsymbol{\psi}(t)) \triangleq - \sum_{r\in\mathcal{R}, b\in\mathcal{R}, d} W_{rb}^{(d)}(t)\mathbb{E}\left\{\hat{\mu}_{rb}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} + \sum_{r\in\mathcal{R}, b\in\mathcal{R}, d}\omega(H_{r,\hat{n}_r(d,t)} - H_{b,\hat{n}_r(d,t)})\mathbb{E}\left\{\hat{\mu}_{rb}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\}$$

It follows by the definition of $W_{cr}^{(d)}(t)$ in (18) and $W_{rb}^{(d)}(t)$ in (23) that $f(\boldsymbol{\psi}(t)) = f_1(\boldsymbol{\psi}(t)) + f_2(\boldsymbol{\psi}(t))$. Because $f_1(\boldsymbol{\psi}(t))$ and $f_2(\boldsymbol{\psi}(t))$ use different control variables, it follows that:

$$\inf_{\boldsymbol{\psi}(t)\in\Psi} f(\boldsymbol{\psi}(t)) = \inf_{\boldsymbol{\psi}(t)\in\Psi} f_1(\boldsymbol{\psi}(t)) + \inf_{\boldsymbol{\psi}(t)\in\Psi} f_2(\boldsymbol{\psi}(t))$$

Now define the modified function $\tilde{f}_1(\boldsymbol{\psi}(t))$:

$$\tilde{f}_1(\boldsymbol{\psi}(t)) \triangleq \sum_c X_c(t)\mathbb{E}\left\{P_c(t) \mid \boldsymbol{Q}(t)\right\} - \sum_{r\in\mathcal{R}, d\in\mathcal{C}_r(t)} U_r^{(d)}(t)\mathbb{E}\left\{\hat{\mu}_{rd}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} - \sum_{r\in\mathcal{R}, c\in\mathcal{C}_r(t), d}\max[W_{cr}^{(d)}(t), 0]\mathbb{E}\left\{\hat{\mu}_{cr}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} \quad (48)$$

The only difference between this modified function and $f_1(t)$ is that the variable $W_{cr}^{(d)}(t)$ is replaced with $\max[W_{cr}^{(d)}(t), 0]$. It follows that:

$$\inf_{\boldsymbol{\psi}(t)\in\Psi} f_1(\boldsymbol{\psi}(t)) = \inf_{\boldsymbol{\psi}(t)\in\Psi} \tilde{f}_1(\boldsymbol{\psi}(t)) \quad (49)$$

This holds because if $W_{cr}^{(d)}(t) < 0$, then the infimizing solution would set $\hat{\mu}_{cr}^{(d)}(t) = 0$ (recall that the feasible control space always allows a zero transmission rate in any component). Define $\delta_{max} \triangleq \max[2\alpha_{max}, \mu_{max}^{in}]$. Note that for any uplink channel (c, r), we have:

$$\hat{W}_{cr}^{(d)}(t) \leq \max[W_{cr}^{(d)}(t), 0] \leq \hat{W}_{cr}^{(d)}(t) + \delta_{max} \quad (50)$$

where $W_{cr}^{(d)}(t)$ and $\hat{W}_{cr}^{(d)}(t)$ are defined in (18) and (19). The last inequality holds because $U_r^{(d)}(t) \leq \eta V + \delta_{max}$ for all time (part (a) of Theorem 1), so that $\max[W_{cr}^{(d)}(t), 0] \leq \hat{W}_{cr}^{(d)}(t) + \delta_{max}$. Now note that:

$$\mathbb{E}\left\{\hat{\mu}_{rd}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\} = \mathbb{E}\left\{\phi_{rd}(\mu_{rd}(t), S_{rd}(t))\mu_{rd}^{(d)}(t) \mid \boldsymbol{Q}(t)\right\}$$

Hence, the uplink/downlink transmission algorithm of CLC-Mesh (in (20)) chooses control variables to minimize a function that is similar to $\tilde{f}_1(\boldsymbol{\psi}(t))$, with the exception that the variable $\max[W_{cr}^{(d)}(t), 0]$ is replaced by $\hat{W}_{cr}^{(d)}(t)$. Using (50) and the definition of μ_{max}^{out} it follows that: the uplink/downlink algorithm chooses transmission rates and client power levels that come within an additive constant $\delta_{max}\mu_{max}^{out}|\mathcal{C}|$ of minimizing $\tilde{f}_1(\boldsymbol{\psi}(t))$, and hence (by (49)) of minimizing $f_1(\boldsymbol{\psi}(t))$.

Similarly, it can be shown that (using (23) and (24)):

$$\hat{W}_{rb}^{(d)}(t) \leq \max[W_{rb}^{(d)}(t), 0] \leq \hat{W}_{rb}^{(d)}(t) + \delta_{max} + \omega H_{max}$$

and that the mesh routing decisions of CLC-Mesh come within an additive constant $|\mathcal{R}|\mu_{max}^{out}(2\omega H_{max} + \delta_{max})$ of minimizing $f_2(\boldsymbol{\psi}(t))$. Therefore, the resulting cross-layer control algorithm comes within an additive constant C of the solution that minimizes the right hand side of the drift bound (33), where:

$$C = |\mathcal{R}|\mu_{max}^{out}(2\omega H_{max} + \delta_{max}) + \delta_{max}\mu_{max}^{out}|\mathcal{C}|$$

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: A survey. *Computer Networks Journal (Elsevier)*, vol. 47, no. 4, pp. 445-487, Mar. 2005.

[2] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transacations on Automatic Control*, vol. 37, no. 12, Dec. 1992.

[3] M. J. Neely, E. Modiano, and C. E Rohrs. Dynamic power allocation and routing for time varying wireless networks. *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 89-103, January 2005.

[4] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *IEEE Trans. on Inform. Theory*, vol. 39, pp. 466-478, March 1993.

[5] N. Kahale and P. E. Wright. Dynamic global packet routing in wireless networks. *Proc. IEEE INFOCOM*, 1997.

[6] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, and P. Whiting. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, vol. 39, no.2, pp.150-154, 2001.

[7] M. J. Neely, E. Modiano, and C. E. Rohrs. Power allocation and routing in multi-beam satellites with time varying channels. *IEEE Transactions on Networking*, vol. 11, no. 1, pp. 138-152, Feb. 2003.

[8] P.R. Kumar and S.P. Meyn. Stability of queueing networks and scheduling policies. *IEEE Trans. on Automatic Control*, vol.40,.n.2, pp.251-260, Feb. 1995.

[9] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. *Proc. IEEE INFOCOM*, 1996.

[10] E. Leonardi, M. Mellia, F. Neri, and M. Ajmone Marsan. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. *Proc. IEEE INFOCOM*, 2001.

[11] L. Georgiadis, M. J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.

[12] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *Proc. IEEE INFOCOM*, March 2005.

[13] M. J. Neely. Energy optimal control for time varying wireless networks. *IEEE Transactions on Information Theory*, vol. 52, no. 7, July 2006.

[14] M. J. Neely. *Dynamic Power Allocation and Routing for Satellite and Wireless Networks with Time Varying Channels*. PhD thesis, Massachusetts Institute of Technology, LIDS, 2003.

[15] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *Proc. IEEE INFOCOM*, March 2005.

[16] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, vol. 50, pp. 401-457, 2005.

[17] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Opportunistic power scheduling for dynamic multiserver wireless systems. *IEEE Transactions on Wireless Communications*, vol. 5, no.6, pp. 1506-1515, June 2006.

[18] M. J. Neely. Optimal backpressure routing for wireless networks with multi-receiver diversity. *Proc. of Conf. on Information Sciences and Systems (CISS), Invited paper on Optimization of Communication Networks*, March 2006.

[19] M. Zorzi and R. Rao. Geographic random forwarding (geraf) for ad hoc and sensor networks: Multihop performance. *IEEE Trans. on Mobile Computing*, vol. 2, no. 4, Oct.-Dec. 2003.

[20] S. Biswas and R. Morris. Exor: Opportunistic multi-hop routing for wireless networks. *Proc. of Sigcomm*, 2005.

[21] F. Baccelli, B. Blaszczyszyn, and P. Muhlethaler. An aloha protocol for multihop mobile wireless networks. *IEEE Transactions on Information Theory*, 2006.

[22] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, vol. 8, pp. 33-37, 1997.

[23] F.P. Kelly, A.Maulloo, and D. Tan. Rate control for communication networks: Shadow prices, proportional fairness, and stability. *Journ. of the Operational Res. Society*, 49, p.237-252, 1998.

[24] S. H. Low and D. E. Lapsley. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, vol. 7(6): 861-75, Dec. 1999.

[25] S. H. Low. A duality model of tcp and queue management algorithms. *IEEE Trans. on Networking*, vol. 11(4), August 2003.

[26] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation for wireless networks. *Proc. of the 39th Annual Allerton Conf. on Comm., Control, Comput.*, Oct. 2001.

[27] J. W. Lee, R. R. Mazumdar, and N. B. Shroff. Downlink power allocation for multi-class cdma wireless networks. *Proc. IEEE INFOCOM*, 2002.

[28] R. Cruz and A. Santhanam. Optimal routing, link scheduling, and power control in multi-hop wireless networks. *Proc. IEEE INFOCOM*, April 2003.

[29] M. Chiang. Balancing transport and physical layer in wireless multihop networks: Jointly optimal congestion control and power control. *IEEE J. Sel. Area Comm.*, vol. 1, no. 23, pp. 104-116, Jan. 2005.

[30] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, vol. 41, no. 4, pp. 451-474, March 2003.

[31] B. Krishnamachari and F. Ordonez. Analysis of energy-efficient, fair routing in wireless sensor networks through non-linear optimization. *IEEE Vehicular Technology Conf.*, Oct. 2003.

[32] X. Lin and N. B. Shroff. Joint rate control and scheduling in multihop wireless networks. *Proc. of 43rd IEEE Conf. on Decision and Control, Paradise Island, Bahamas*, Dec. 2004.

[33] L. G. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, Vol. 11, No. 2, pp. 350-361, 1982.

[34] C-S Chang, D-S Lee, and Y-S Jou. Load balanced birkhoff-von neumann switches. *IEEE Workshop on High Performance Switching and Routing (HPSR)*, 2001.

[35] C.E. Koksal. *Providing Quality of Service over Electronic and Optical Switches*. PhD thesis, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems (LIDS), Sept. 2002.

[36] L. Buttyan and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 8, no. 5, pp. 579-592, Oct. 2003.

[37] J. Crowcroft, R. Gibbens, F. Kelly, and S. Ostring. Modeling incentives for collaboration in mobile ad-hoc networks. *presented at the 1st Int. Symp. Modeling and Optimization in Mobile, Ad-Hoc, and Wireless Networks (WiOpt '03), Sophia-Antipolis, France*, March 2003.

[38] M. J. Neely. Optimal pricing in a free market wireless network. *Proc. IEEE INFOCOM*, May 2007.

[39] D. P. Bertsekas and R. Gallager. *Data Networks*. New Jersey: Prentice-Hall, Inc., 1992.