Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks With Node-Exclusive Spectrum Sharing

Xinzhou Wu
Flarion Technologies
Email: X.Wu@flarion.com

R. Srikant
Coordinated Science Lab and
Dept. Of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Email: rsrikant@uiuc.edu

Abstract— **We consider wireless networks with a special type of spectral allocation, where the only constraint is that a node cannot transmit to more than one receiver at a time and cannot receive more than one transmission at a time. We introduce a scheduling algorithm called *regulated maximal matching* which is fully distributed and guarantees a throughput that is at least half of the throughput achievable by a centralized algorithm.**

I. Introduction

Scheduling and routing algorithms allocate resources to competing flows in a multi-hop wireless network. One goal in designing these algorithms is *throughput optimality*: the resource allocation algorithm should achieve the largest possible throughput region. Assuming that a central authority has access to all the queue length information, a throughput-optimal algorithm was developed in [17]. Specifically, it was shown that, for any set of arrival rates between various source-destination pairs that lie within the capacity region of the network, the Markov chain describing the queue lengths in the network is positive recurrent. Since this original paper, the problem of designing decentralized algorithms that achieve the capacity region has been open. The problem is difficult even if we relax the throughput-optimality criterion and only ask for algorithms that achieve a guaranteed fraction of the capacity region.

In this paper, we study multihop wireless networks where the available frequency spectrum is divided such that no two nodes in a two-hop neighborhood transmit at the same frequency. Effectively, this means that the only constraint on multihop communications is that a node cannot transmit or receive data from more than one neighbor at any time instant. Such a model has been studied recently in [15], [9]. One motivation for studying such models is the bluetooth technology specifications [12] which allow such a spectrum sharing. We call such networks node-exclusive spectrum-sharing networks. We further assume that the route for a given source-destination pair is fixed, so that the only resource allocation to be designed is the scheduling algorithm, i.e., at each time instant, a set of links have to chosen to transmit data, such that the links do not interfere with each other. Even for such a simple network, distributed algorithms that achieve a guaranteed fraction of the capacity are difficult to obtain. There is a subtle difference in the models of [15] and [9]: while [15] does not allow a node simultaneously transmit and receive at the same, the model in [9] does. We assume the latter model in this paper but our results hold for the other model as well.

Research supported by AFOSR URI grant F49620-01-1-0365

Scheduling in wireless networks is somewhat analogous to scheduling in high-speed switches. In fact, the model in [17] is general enough to be considered either a wireless network or a high-speed switch. While distributed scheduling is not much of an issue in high-speed switches, complexity is. In the context of high-speed switches, the algorithm in [17] obtains a weighted maximum matching on a bipartite graph at each time instant [11]. The high computational complexity of weighted maximum matching has motivated the search for low-complexity algorithms that achieve either the full capacity of the switch or some fraction of the switch capacity. Of most relevance to us in this paper is the notion of maximal matching which has been shown to achieve at least half the capacity region of the switch with batch scheduling in [18] and with continuous scheduling in [3].

In a recent work, using a Lyapunov function that is a modification of the one used in [3] for high-speed switches, it has been shown that maximal matching can also be used to achieve at least half the capacity region in node-exclusive spectrum sharing multi-hop wireless networks [9]. However, it assumed that the load on a link is directly imposed on a link, rather than packets travelling hop-by-hop through the network. It is well-known in queueing theory that the stability of such a model does not guarantee that stability of the corresponding model where packets traverse the network one link at a time [8], [10], [14]. One solution to this problem is provide appropriate buffer priorities as in [10]. This issue has been recognized in [9] where the authors give priorities to flows with a smaller number of hops which is the analog of the FBFS and LBFS policies in [10] for wireless networks. However, the algorithm in [9] assumes that the number of packets that arrive for a route is instantaneously known to all nodes on the route. Thus, the stability issue is not fully resolved. Nevertheless, the work in [9] provides a nice starting point to obtain decentralized algorithms for node-exclusive spectrum-sharing networks.

In this paper, we will further investigate the impact of MM (maximal matching) scheduling in wireless networks. We will use the contention model in [9] and study the stability region of such networks. Unlike in [9], we have a fixed number of users in the wireless network, each generating arrivals according to some stochastic process which we will describe later. Letting the average arrival rate be λ_s for user s, the stability region Λ is defined to be the set of $[\lambda_s]$ such that the queue sizes in the system are stable. The main contributions of this paper is as follows:

- We first introduce a prioritized maximal matching algorithm that requires coordination throughout the wireless network. Under this algorithm, a maximal matching is found for one-hop flows, then and only then is a maximal matching obtained for flows in their second hop. Thus, for example, in a network where the number of hops on a route can vary from one to five, a carefully time-synchronized maximal matching has to be executed five successive times. Such an algorithm may be appropriate for networks with a small number of hops.
- Next, we show that the MM-based scheduling rule can be implemented as a fully distributed algorithm by inserting "regulators" between the various nodes. In contrast to prioritized maximal matching, no coordination among nodes is required; however, the mean arrival rates of each flow is required to implement this. We will comment on this more later.

The rest of the paper is organized as follows. In Section II, we describe the network model studied in this paper. MM scheduling for wireless networks will be defined in Section III and prioritized MM-scheduling will also be introduced in this section. In Section IV, we will present the regulated MM-scheduling algorithm. In Section V, we will make some concluding remarks. All proofs in the paper are deferred to the appendix.

II. Channel Model

We consider a time-slotted multi hop wireless network which can be modelled by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the set of all directed edges. A directed edge (i, j) is established between node $i, j \in \mathcal{V}$ if and only if node j is in the transmission coverage area of node i. A special case of this directed graph is that each edge is bi-directional. This is represented in our graph by two links, one from i to j, and the other from j to i.

We assume there are S users who have traffic going through the network with a fixed path. The path information is contained by the routing matrix $\mathbf{H} = [H^s_{ij}, (i, j) \in \mathcal{E}; 1 \leq s \leq S]$, where H^s_{ij} is an indicator function and is determined as follows

$$H^s_{ij} = \begin{cases} 1 & \textit{if } (i,j) \in \mathcal{E} \textit{ is on the path of user } s; \\ 0 & \textit{otherwise.} \end{cases} \quad (1)$$

Each user s's arrival process $A_s(t)$ is a random process with mean arrival rate λ_s. For simplicity, we assume that $A_s(t)$ is i.i.d. across time slots and the arrival processes of different users are assumed to be independent. Meanwhile, at each link $(i, j) \in \mathcal{E}$, a queue has to be maintained to buffer the the incoming packets which are to be transmitted over link (i, j). We assume all the links are have an infinite buffer.

In a wireless network, simultaneous transmissions over nearby links in the same time slot will interfere with each other and cause a *collision*. As a consequence, only a subset of links $\mathcal{E}$ can be used at the same time slot. We will focus on the following simple *interference constraint* described in the previous section: *in any given time slot, a transmitter cannot send packets intended for more than one neighbor in its transmission range. Similarly, a receiver cannot receive packets from more than one transmitter. However, we allow simultaneous transmission and reception at any node.* When link (i, j) is scheduled for transmission during a time slot, we assume that a maximum of one packet can be transmitted.

A scheduling policy is a rule to identify a set of links to be scheduled at each time slot which does not violate the interference constraint described earlier. We let $\pi_{ij}(t)$ denote the indicator function of the event that a transmission is scheduled over link $(i, j) \in \mathcal{E}$ at time t. Further, $\pi^s_{ij}(t)$ is the indicator function of the event that link (i, j) is scheduled for a transmission of a packet from user s.

We let q^s_{ij} denote the queue length of user s on link (i, j) and q_{ij} without the superscript s denote the total queue length at (i, j), i.e., $q_{ij} = \sum_s q^s_{ij}$ where the summation is only over those users whose route passes through link (i, j). We can now write our the update rule for the queueing network as follows:

$$\begin{aligned} q_{ij}(t+1) &= (q_{ij}(t) - \pi_{ij}(t))^+ + \sum_s A_s(t) H^s_{ij} I_{o(s)=i} \\ &\quad + \sum_s \sum_{m:(m,i)\in\mathcal{E}} \pi^s_{mi}(t) H^s_{mi} H^s_{ij}, \end{aligned} \quad (2)$$

where $o(s)$ denoted the origin or the first node in user s's route.

We define the capacity region to be the set of vectors of arrival rates $[\lambda_s]$ for which the queueing system (2) is stable under some scheduling rule. Note that we have used the notation $[x_l]$ to denote a vector consisting of all elements of x_l, i.e., $[x_l] = (x_1, x_2, \cdots)$. The stability region of a general wireless network has been well characterized in [17] and for the network model we described above, the stability region is given by

$$\Lambda = \{[\lambda_s, s = 1, \cdots, S] : [\sum_{s=1}^{S} H^s_{ij} \lambda_s, (i, j) \in \mathcal{E}] \in \mathcal{C}o(\mathcal{R})\}, \quad (3)$$

where $\mathcal{C}o(\mathcal{R})$ is the set of all matchings that are consistent with our interference constraint.

The performance of a given scheduling rule $\pi = [\pi_{ij}, (i, j) \in \mathcal{E}]$ can be measured by the stability region C_π under this scheduling rule. A *throughput-optimal* scheduling rule [17] π_0 is a rule which stabilizes the queueing network under any set of user arrival rates that can be stabilized by another scheduling rule. For the network model where the routes of each user are fixed, the following scheduling rule can be shown to be *throughput-optimal*:

$$\pi(t) = \arg\max_{\pi \in \mathcal{R}} \sum_{(i,j)\in\mathcal{E}} \max_s (q^s_i - q^s_j) \pi_{ij}, \quad (4)$$

where q_i^s is the total number of packets of user s that are queued at node $i \in \mathcal{V}$. This scheduling rule is a *centralized* rule in the sense that at each time slot t, we have to globally optimize (4) over all possible rate allocation vectors $\pi \in \mathcal{R}$.

III. MM Scheduling for Wireless Networks

In a wireless multi hop network as we described in Section II, the most straightforward way to define an MM scheduling policy (based on the queue state $\{q_{ij}(t)\}$) is as follows: for any edge $(i,j) \in \mathcal{E}$ with a non-empty queue, i.e., $q_{ij} \geq 1$, (i,j) has to be in the schedule (i.e., $\pi_{ij} = 1$), if

$$\sum_{l:(i,l)\in\mathcal{E}} \pi_{il} + \sum_{u\neq i:(u,j)\in\mathcal{E}} \pi_{uj} = 0. \tag{5}$$

An equivalent way of describing this condition is that an MM schedule satisfies

$$\sum_{l:(i,l)\in\mathcal{E}} \pi_{il} + \sum_{u:(u,j)\in\mathcal{E}} \pi_{uj} \geq 1. \tag{6}$$

for any $(i,j) \in \mathcal{E}$ with $q_{ij} \geq 1$. Note that, even though we have used the term maximal matching to described this scheduling algorithm, it is not strictly a matching in the graph-theoretic sense where a matching is usually defined to be a set of edges, no two of which share a common node. Here, since we allow a node to simultaneously transmit and receive, the above is not a maximal matching; but we use this term anyway to avoid introducing additional terminology.

Unfortunately, in the multi-hop scenario that we consider in this paper, it is not clear if such simple MM scheduling schemes can be used to guarantee even a fraction of the capacity region. At this point, we have not been able to prove or disprove such a result for this scheduling rule even in our simple network model. In the following subsection, we define a variant of the above called prioritized MM scheduling which can be shown to yield at least half the capacity region.

A. Prioritized MM Scheduling

One way to improve the MM scheduling in a multi hop wireless network is to introduce a *prioritized* priority structure to it. Specifically, we give higher priority to the traffic which has travelled fewer hops at each node. We call such scheduling schemes *prioritized MM scheduling* schemes. The construction of this scheduling scheme is motivated by the proof techniques in [10], [9]. However, unlike in [9], we do not require instantaneous knowledge of arrivals on a route along all nodes in the route; instead, we construct a sequence of maximal matchings consistent with the priorities in a manner described below.

To differentiate between flows which have travelled different numbers of hops, we first introduce some extra notation. For any link $(i,j) \in \mathcal{E}$, we let $q_{ij}^{(k)}$ be the queue length at link (i,j) contributed by those users such that (i,j) is the kth hop on their path. Similarly, $q_{ij}^{(\leq k)}$ is the queue length at link (i,j) of the users such that (i,j) is within the k hops of their path, i.e.,

$$q_{ij}^{(\leq k)} = \sum_{l=1}^{k} q_{ij}^{(l)}.$$

We also let L be the number of hops on the longest route in the network. The prioritized MM scheduling algorithm is described below.

Algorithm 1: (Prioritized MM scheduling) At each time slot, we perform L rounds of maximal matchings. In the first round, the maximal matching is based on the queues $\{q_{ij}^{(1)}\}$. More specifically, only nodes with non-empty first hop traffic, i.e., source nodes of all traffic, are eligible to be a part of the maximal matching. For the second round of maximal matching, if a node in the first round has any outgoing links that are matched, then all of its outgoing links are removed from the graph. The second round of maximal matching is then implemented on the rest of the graph based on queues $\{q_{ij}^{(\leq 2)}\}$. Similarly, prior to the k^{th} round, if a node has any outgoing link that has been used in a matching in one of the prior rounds, then all of its outgoing links are removed from the graph. If a node has any incoming links that have been used in a matching in the prior rounds, then all of its incoming links are removed from the graph. Then, in k^{th} ($k \leq L$) round, the maximal matching is performed on the remaining graph based on $\{q_{ij}^{(\leq k)}\}$. In all, up to L rounds of matchings may have to be performed to implement prioritized MM scheduling.

Due to space limitations, we present the following theorem without proof. It uses ideas from [9]; however, the traffic model in [9] is quite different since they consider stability of file arrivals (user arrivals) and departures as opposed to packet arrivals and departures for a fixed number of users. Further, the idea of using multiple rounds of matchings in prioritized MM scheduling is new.

Theorem 1: For the multi hop wireless network model with the interference model described in Section I and using a prioritized MM scheduling policy as described in Algorithm 1, the system is stable for any set of rates $[\lambda_s]$ that lies strictly inside $\Lambda/2$. In other words, a prioritized scheduling policy can achieve a stability region of $\Lambda/2$. $\diamond$

IV. Regulated MM Scheduling

In this section we present a novel idea for completely distributing the implementation of the maximal matching algorithm. Although prioritized MM scheduling can achieve the half capacity region, there are significant difficulties in implementing this algorithm. In a large ad hoc network, the number of hops of a typical user scales approximately as $O(\log N)$, where N denotes the number of nodes in the network [5]. Hence, as the network size increases, the overhead of this scheduling algorithm also increases. In addition, network-wide coordination is required to ensure that nodes whose backlogged packets have all traversed at least one hop do not request transmissions on their links before a maximal matching is found among those nodes which have backlogged packets which are on their first hop. Similarly, nodes with backlogged packets which have already traversed two or more hops have to wait for nodes which have packets that have traversed fewer hops to complete their maximal schedules. Thus, global synchronization is required to implement this algorithm, which is not desirable in a fully distributed network. Even though the basic MM scheduling algorithm requires that time be synchronized for data transmission, the prioritized MM scheduling algorithm requires further synchronization (repeated many times) at the finer granularity of control packets (which are typically much smaller than data packets) which is considerably harder to approximate in practice.

Now, we present the main contribution of the paper which is a new algorithm, also based on the notion of maximal matching but one that does not suffer from the implementation issues of prioritized MM scheduling. The key idea in this algorithm is to introduce *regulators* at each node, one for each user using the node, such that the burstiness of the packets belonging to each user is regulated before entry into the node. A λ-regulator is a logical device with a maximum service rate λ, i.e., it generates packets for the node at its output at a maximum rate of λ. Specifically, at each time slot, a λ-regulator checks its buffer size and if it is non-empty, it transfers a packet to the user's queue at the node with probability λ. Otherwise, it transfers nothing.

It should be noted that there are other ways of constructing regulators and the regulator described above is just one example of them. We chose our particular type of regulator for technical convenience (specifically, this regulator ensure that the original queues at the wireless network nodes along with the regulator queues form a Markov chain) in the stability proof instead of performance considerations. Other implementations may provide better performance and a comparative performance of various implementations of regulators is a topic for further research. The idea of a regulator was originally suggested in the context of re-entrant lines in [6].

Denote the arrival rate vector consisting of all the users's arrival rates by $[\lambda_s]_{1 \leq s \leq S}$. We choose the regulators of a user s according to the following rule: for the first hop (node) along the path of user s, we use a λ_s-regulator at its input queue for user s; for k^{th} hop ($k \geq 2$), we use a $(\lambda_s + (k-1)\epsilon)$-regulator.

Once the regulators are introduced, there is no need to perform repeated MM scheduling in many steps as in prioritized MM scheduling. We define the combination of the regulators and maximal matching to be a regulated MM scheduling algorithm. For completeness and to clarify what we mean by queue length, we describe the matching part of the regulated MM scheduling algorithm below:

Algorithm 2: (Regulated MM scheduling) At each time slot, only one round of maximal matching is performed based on the queue length (not including the buffer size of the regulators) at each node, i.e., all nodes with non-empty queues are eligible for matching while all nodes with empty queues and possibly non-empty regulator buffers are not eligible. $\diamond$

The main result of this paper is given below:

Theorem 2: For a multi hop wireless network, a regulated MM scheduling algorithm can achieve a stability region of $\Lambda/2$. $\diamond$

Remarks: We make the following comments on the regulated MM algorithm:

- We now provide an intuitive argument for why MM scheduling, with regulators inserted at the nodes, stabilizes the system for any rate within the $\Lambda/2$ region. The system with regulators behaves like two queueing systems: one consisting of the queues that were in the network before the regulators were introduced (which we will call the original queues) and the other consisting of the regulator buffers (which we will call the regulator queues). The original queues are stable since the inputs to these queues are the outputs of the regulators which are rate *regulated* to ensure that set of the arrival rates at the original queues fall within half the capacity region. The regulators effectively decouple the the original queues at each node from the original queues at the other nodes and thus, simulates a *well-behaved* queueing system where inputs are directly fed into each node along the path of all users. On the other hand, the regulator buffers are fed by stable queues and thus, the arrival rates at the regulators must be equal to the departure rates of the original queues feeding them. Thus, the stability of the regulator queues is now obvious since they are just a set of queues whose average arrival rate is strictly less than the service rate that they can provide. In the appendix, we establish a strong form of this stability result, i.e., we prove that the Markov chain consisting of both the original queueing system and the regulator buffers is positive recurrent. However, we believe simpler proofs exist if weaker notions of system stability are sufficient (such as the upper-boundedness of the long-run time average of the first moments of each of the queues) because of the intuitive argument that we have presented above.
- Regulated MM scheduling is attractive due to the ease of its implementation. The regulators can be thought of as an additional part (can be a logic buffer) of the queueing system at each node. It should be noted that to use the regulated MM scheme in this paper, one regulator per user is required at each node on the user's path. Thus, we are trading off additional computation to achieve a fully decentralized solution. After adding the regulators, all that is left to do is simple maximal matching based on the queue length information which is fully distributed and requires no coordination among the nodes. As compared to the prioritized MM scheduling policy, we have to do only one MM at each time slot and thus the overhead caused by scheduling is very small.
- The implementation of the regulators requires knowledge of the arrival rates from the various users. Recent work on joint scheduling and congestion control in wireless networks [4], [13], [16] suggests that it is possible to control arrival rates within the capacity region using feedback from the network. Here, we would additionally require that arrival rates be communicated to the nodes in the packet headers. While current versions of TCP

do not support this, protocols such as XCP which are currently being considered for the Internet of the future would allow such information to be part of the packet header [7].

V. Conclusions

In this paper, we have developed a new distributed scheduling algorithm for a simple multi hop wireless network model. Our main results show that maximal-matching based scheduling policies can achieve a capacity region of $\Lambda/2$, where Λ is the network stability region under a perfect (centralized) scheduler. Regulated MM scheduling combines the simplicity of maximal matching to wireless networks with the notion of per-flow traffic regulators to regulate the burstiness of traffic at each node in the network. Regulated MM scheduling is fully distributed and is easy to implement, and thus can be potentially used in future wireless networks.

Appendix I
Proof of Theorem 2

Let p_{ij}^s denote the length of the regulator queue on link (i,j) for user s. It is easy to see that the whole system $(\mathbf{q}(t),\mathbf{p}(t))$ is a Markov Chain. We define the following Lyapunov function for the system:

$$V(\mathbf{q},\mathbf{p}) = V_1(\mathbf{q}) + \xi V_2(\mathbf{p},\mathbf{q}), \tag{7}$$

where $V_1(\mathbf{q})$ is given by

$$V_1(\mathbf{q}) = \frac{1}{2}\sum_{i\in\mathcal{V}}\left(\sum_{l:(i,l)\in\mathcal{E}} q_{il}\right)^2 + \frac{1}{2}\sum_{j\in\mathcal{V}}\left(\sum_{u:(u,j)\in\mathcal{E}} q_{uj}\right)^2,$$

and V_2 is defined as follows

$$V_2(\mathbf{p},\mathbf{q}) = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}}\sum_s (p_{ij}^s + q_{ij}^s)^2.$$

ξ is a positive parameter.

The queue update equations for this system are

$$\begin{aligned} q_{ij}(t+1) &= q_{ij}(t) - \pi_{ij}(t) + \sum_s R_{\cdot i}^s(t) H_{ij}^s && (8)\\ p_{ij}^s(t+1) &= (p_{ij}^s(t) - R_{ij}^s)^+ + \pi_{ij}^s(t), && (9)\end{aligned}$$

where $R_{\cdot i}^s(t)$ is the output of regulator of user s before node i. Due to our definition of the regulator, R_{ij}^s can only be non-zero when p_{ij}^s is non-zero. Hence, we can remove the projection in (9) as well and we have

$$p_{ij}^s(t+1) = p_{ij}^s(t) - R_{ij}^s + \pi_{ij}^s(t).$$

We have to upper bound

$$E\left[V(\mathbf{q}(t+1),\mathbf{p}(t+1)) - V(\mathbf{q}(t),\mathbf{p}(t))|\mathbf{q}(t),\mathbf{p}(t)\right]$$

by a negative number for all states $(\mathbf{q},\mathbf{p})(t)$ except possibly in a bounded region, where the drift should simply be finite.

We consider the contribution of $V_1(\mathbf{q})$ and $V_2(\mathbf{p},\mathbf{q})$ separately for now. Thus, we first look at

$$\begin{aligned} &\Delta V_1(\mathbf{q})\\ =\ & E\left[V_1(\mathbf{q}(t+1)) - V_1(\mathbf{q}(t))|\mathbf{q}(t),\mathbf{p}(t)\right]\\ =\ & \frac{1}{2}E\Bigg[\sum_{(i,j)\in\mathcal{E}}(q_{ij}(t)+q_{ij}(t+1))\\ & \Gamma_{ij}([\sum_s R_{\cdot i}^s(t)H_{ij}^s - \pi_{ij}^s(t)])|\mathbf{q}(t),\mathbf{p}(t)\Bigg]\\ \le\ & \frac{1}{2}E\Bigg[\sum_{(i,j)\in\mathcal{E}}(q_{ij}(t)+q_{ij}(t+1))\\ & \Gamma_{ij}([\sum_s(\lambda_{\cdot i}^s H_{ij}^s - \pi_{ij}^s(t)])|\mathbf{q}(t),\mathbf{p}(t)\Bigg]\\ \le\ & \frac{1}{2}E\Bigg[\sum_{(i,j)\in\mathcal{E}}(q_{ij}(t)+q_{ij}(t+1))\\ & \Gamma_{ij}([\sum_s(\lambda_s + L\epsilon)H_{ij}^s - \pi_{ij}^s(t)])|\mathbf{q}(t),\mathbf{p}(t)\Bigg], \end{aligned}$$

where Γ_{ij} is defined as follows:

$$\Gamma_{ij}([r_{ij}]) = \sum_{l:(i,l)\in\mathcal{E}} r_{il} + \sum_{u:(u,j)\in\mathcal{E}} r_{uj}. \tag{10}$$

.

With some effort, one can show the following:

$$\Delta V_1(\mathbf{q}) \le -\eta\sum_{ij} q_{ij}(t) + C, \tag{11}$$

where C is a constant independent of $\mathbf{q}$ and $\mathbf{p}$.

For the contribution of V_2, we first notice that

$$\begin{aligned} & q_{ij}^s(t+1) + p_{ij}^s(t+1) && (12)\\ =\ & q_{ij}^s(t) + p_{ij}^s(t) + R_{\cdot i}^s(t) - R_{ij}^s(t) && (13)\end{aligned}$$

We can bound V_2 as follows:

$$\begin{aligned} &\Delta V_2(\mathbf{p},\mathbf{q})\\ =\ & E\left[V_2(\mathbf{p}(t+1),\mathbf{q}(t+1)) - V_2(\mathbf{p}(t),\mathbf{q}(t))\right.\\ & \left.|\mathbf{q}(t),\mathbf{p}(t)\right]\\ =\ & E\Bigg[\sum_{(i,j)\in\mathcal{E}}\sum_s(p_{ij}^s(t+1)+q_{ij}^s(t+1))^2\\ & -\sum_{(i,j)\in\mathcal{E}}\sum_s(p_{ij}^s(t)+q_{ij}^s(t))^2|\mathbf{q}(t),\mathbf{p}(t)\Bigg]\\ =\ & E\Bigg[\sum_{(i,j)\in\mathcal{E}}\sum_s\{p_{ij}^s(t+1)+q_{ij}^s(t+1)\\ & +p_{ij}^s(t)+q_{ij}^s(t)\}\{p_{ij}^s(t+1)+q_{ij}^s(t+1)\\ & -p_{ij}^s(t)-q_{ij}^s(t)\}|\mathbf{q}(t),\mathbf{p}(t)\Bigg]\\ =\ & E\Bigg[\sum_{(i,j)\in\mathcal{E}}\sum_s\left(p_{ij}^s(t)+q_{ij}^s(t)\right)\left(p_{ij}^s(t+1)\right. \end{aligned}$$

$$+q^s_{ij}(t+1)-p^s_{ij}(t)-q^s_{ij}(t)\big)\,|\mathbf{q}(t),\mathbf{p}(t)\Big]$$

$$+\frac{1}{2}E\left[\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t+1)+q^s_{ij}(t+1)\right.$$

$$\left.-p^s_{ij}(t)-q^s_{ij}(t)\big)^2\,|\mathbf{q}(t),\mathbf{p}(t)\right]$$

The second term above can be easily bound by a constant C_3 independent of $\mathbf{P}(t)$ and $\mathbf{Q}(t)$ as follows:

$$\begin{aligned}
&E\left[\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t+1)+q^s_{ij}(t+1)\right.\\
&\left.-p^s_{ij}(t)-q^s_{ij}(t)\big)^2\,|\mathbf{q}(t),\mathbf{p}(t)\right]\\
\leq\;& 2E\left[\sum_{(i,j)\in\mathcal{E}}\sum_s\big(R^s_{ij}(t)\big)^2+\big(R^s_{\cdot i}(t)\big)^2\,|\mathbf{q}(t),\mathbf{p}(t)\right]\\
\leq\;& 2\sum_{(i,j)\in\mathcal{E}}\sum_s\big((\lambda_s+L\epsilon)\big)^2+\big((\lambda_s+L\epsilon)\big)^2=C_3
\end{aligned}$$

The first term can be bounded by

$$\begin{aligned}
&E\left[\textstyle\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t)+q^s_{ij}(t)\big)\right.\\
&\left.\big(p^s_{ij}(t+1)+q^s_{ij}(t+1)-p^s_{ij}(t)-q^s_{ij}(t)\big)\,|\mathbf{q}(t),\mathbf{p}(t)\right]\\
&=\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t)+q^s_{ij}(t)\big)\,E\left[R^s_{\cdot i}(t)-R^s_{ij}(t)|\mathbf{q},\mathbf{p}\right]\\
&=\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t)+q^s_{ij}(t)\big)\left[\lambda^s_{\cdot i}I_{p^s_{\cdot i}(t)\geq 1}-\lambda^s_{ij}I_{p^s_{ij}(t)\geq 1}\right]\\
&\leq\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t)+q^s_{ij}(t)\big)\left[\lambda^s_{\cdot i}-\lambda^s_{ij}I_{p^s_{ij}(t)\geq 1}\right]
\end{aligned}$$

From our design of the regulators, we know that

$$\lambda^s_{ij}=\lambda^s_{\cdot i}+\epsilon,$$

for any (i,j) on user s's path. From this, we have

$$\begin{aligned}
&\Delta V_2(\mathbf{p},\mathbf{q})\\
&\leq\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t)+q^s_{ij}(t)\big)\left(\lambda^s_{\cdot i}-\lambda^s_{ij}I_{p^s_{ij}(t)\geq 1}\right)\\
&+C_2.
\end{aligned}\tag{14}$$

By combining (11) and (14), we have

$$\begin{aligned}
&E\left[V(\mathbf{q}(t+1),\mathbf{p}(t+1))-V(\mathbf{q}(t),\mathbf{p}(t))|\mathbf{q}(t),\mathbf{p}(t)\right]\\
\leq\;& -\eta\sum_{(i,j)\in\mathcal{E}}q_{ij}(t)+C_3+\\
&\xi\sum_{(i,j)\in\mathcal{E}}\sum_s\big(p^s_{ij}(t)+q^s_{ij}(t)\big)\left(\lambda^s_{\cdot i}-\lambda^s_{ij}I_{p^s_{ij}(t)\geq 1}\right)\\
=\;& -\sum_{(i,j)\in\mathcal{E}}\sum_s\left[\eta-\xi\left(\lambda^s_{\cdot i}-\lambda^s_{ij}I_{p^s_{ij}(t)\geq 1}\right)\right]q^s_{ij}(t)+\\
&\xi\sum_{(i,j)\in\mathcal{E}}\sum_s p^s_{ij}(t)\left(\lambda^s_{\cdot i}-\lambda^s_{ij}I_{p^s_{ij}(t)\geq 1}\right)+C_3\\
\leq\;& -\sum_{(i,j)\in\mathcal{E}}\sum_s\left[\eta-\xi\right]q^s_{ij}(t)+\\
&\xi\sum_{(i,j)\in\mathcal{E}}\sum_s p^s_{ij}(t)\left(\lambda^s_{\cdot i}-\lambda^s_{ij}I_{p^s_{ij}(t)\geq 1}\right)+C_3\\
=\;& -\sum_{(i,j)\in\mathcal{E}}\sum_s\left[\eta-\xi\right]q^s_{ij}(t)-\epsilon\xi\sum_{(i,j)\in\mathcal{E}}\sum_s p^s_{ij}(t)+C_3.
\end{aligned}$$

We can easily choose ξ (independent of $\mathbf{P}(t)$ and $\mathbf{Q}(t)$, of course) here such that

$$\eta+\xi\geq c_0>0$$

and thus

$$\begin{aligned}
&E\left[V(\mathbf{q}(t+1),\mathbf{p}(t+1))-V(\mathbf{q}(t),\mathbf{p}(t))|\mathbf{q}(t),\mathbf{p}(t)\right]\\
\leq\;& -\sum_{(i,j)\in\mathcal{E}}\sum_s c_0q^s_{ij}(t)-\epsilon\xi\sum_{(i,j)\in\mathcal{E}}\sum_s p^s_{ij}(t)+C_3.
\end{aligned}$$

This concludes the proof of this theorem. $\diamond$

REFERENCES

[1] S. Asmussen. *Applied Probability and Queues*. Springer-Verlag, New York, 2003.

[2] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 1991.

[3] J.G. Dai and B. Prabhakar. The throughput of data switches with and without speedup. In *Proceedings of IEEE INFOCOM*, pages 556–564, 2000.

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

[5] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.

[6] C. Humes. A regulator stabilization technique: Kumar-Seidman revisited. *IEEE Transactions on Automatic Control*, pages 191–196, 1994.

[7] D. Katabi, M. Handley, and C. Rohrs. Internet congestion control for future high bandwidth-delay product environments. In *Proceedings of ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[8] P. R. Kumar and T. I. Seidman. Dynamic instabilities and stabilization methods in distributed real-time scheduling of manufacturing systems. *IEEE Transactions on Automatic Control*, pages 289–298, March 1990.

[9] X. Lin and N. Shroff. The impact of imperfect scheduling on cross-layer control in multihop wireless networks. In *Proceedings of IEEE INFOCOM*, 2005.

[10] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Transactions on Automatic Control*, pages 1406–1416, December 1991.

[11] N. McKeown, V. Anantharam, and J. Walrand. Achieving 100% throughput in an input-queued switch. In *Proceedings of IEEE INFOCOM*, pages 296–302, San Francisco, March 1996.

[12] B. Miller and C. Bisdikian. *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communication*. Prentice Hall, 2000.

[13] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. In *Proceedings of IEEE INFOCOM*, 2005.

[14] A.N. Rybko and A.L. Stolyar. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problems of Information Transmission*, 28, 1992. Translated from Problemy Peredachi Informatsii, vol. 28, no. 3, pp. 3-26, 1992.

[15] S. Sarkar and L. Tassiulas. End-to-end bandwidth guarantees through fair local spectrum share in wireless ad hoc networks. In *Proceedings of IEEE Conference on Decision and Control*, December 2005.

[16] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. Submitted.

[17] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, pages 1936–1948, December 1992.

[18] T. Weller and B. Hajek. Scheduling non-uniform traffic in a packet-switching system with small propagation delay. *IEEE/ACM Transactions on Networking*, pages 813–823, December 1997.