# THROUGHPUT-DELAY TRADE-OFF IN WIRELESS MULTIHOP NETWORKS VIA GREEDY HYPERBOLIC EMBEDDING

E. STAI*, J. S. BARAS†, AND S. PAPAVASSILIOU ‡

**Abstract.** Real time applications over wireless multihop networks, demand routing/scheduling algorithms that achieve desirable delay-throughput trade-offs, with high throughput and low end-to-end delay. The backpressure algorithm, has received much attention by the research community in the past few years, as it satisfies the throughput optimal requirement. The backpressure algorithm performs routing and scheduling based on congestion gradients, by allowing transmission to the links that maximize the differential backlog between neighboring nodes. However, by deploying routing without using any information about the position or distance to the destination, it explores all possible source-destination paths leading to undesirable high delays. In this paper, we propose a method of restricting the number of paths used by the backpressure algorithm, with the aid of the greedy embedding of a network in the hyperbolic space. We propose two algorithms, the "Greedy" backpressure and the "Greediest" backpressure for both static and dynamic networks, which consider the network embedded in the hyperbolic space and combine the greedy routing in hyperbolic coordinates with the backpressure scheduling. We prove analytically their throughput optimality and study through simulations the induced improvement in the delay-throughput trade-off compared with the backpressure algorithm.

**Key words.** wireless networks, greedy embedding, throughput optimal scheduling, hyperbolic geometry

**AMS subject classifications.** 68M20, 60K25, 68M10

## 1. Introduction.

Real time applications over wireless multihop networks demand routing/scheduling algorithms that achieve desirable delay-throughput trade-offs, with high throughput and low end-to-end delay. The backpressure algorithm, introduced in its original form in [1], has received much attention by the research community in the past few years (i.e. [2, 3, 4]), as it satisfies the throughput optimal requirement. The backpressure algorithm performs routing and scheduling based on congestion gradients, by allowing transmission to the links that maximize the sum of differential queue backlogs in the network. However, by deploying routing without any information about the position or distance to the destination, the backpressure explores all possible source-destination paths leading to high delays even in the case of light traffic.

Several approaches have been developed for reducing the delay imposed by the pure backpressure scheduling/routing. The authors in [4] combine backpressure and shortest path routing by imposing hop-count constraints on each flow, assuming that each node knows a-priori its hop distance from all others. However, using this condition increases the computational complexity especially in dynamic network conditions. Following another approach, the authors in [5] use shadow queues for the backpressure scheduling/routing, improving in this way the delay of the backpressure algorithm while simultaneously reducing the number of real queues needed to be stored at each node to only one queue per neighboring node. In this paper, we propose an alternative method of restricting the number of paths used by the backpressure algorithm, with the aid of the greedy embedding of a network in hyperbolic space. A greedy embedding in hyperbolic space is a correspondence between nodes and hyperbolic coordinates such that the greedy routing algorithm, employed in the hyperbolic space, does not have local minima, i.e. every node can find at least one neighbor closer than itself to all possible destinations [6, 7]. In [6], a distributed implementation of a greedy embedding is

*Institute for Systems Research, University of Maryland, College Park, MD 20742, USA (estai@umd.edu).

†Department of Electrical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA (baras@umd.edu).

‡School of Electrical & Computer Engineering, National Technical University of Athens (NTUA), Athens, Zografou, 15780, Greece (papavass@mail.ntua.gr).

proposed, which can assign hyperbolic coordinates to new nodes without re-embedding the whole network. In this work, we consider a greedy embedding of the network following [6] and we impose routing constraints on the backpressure algorithm, by determining as possible next-hop neighbors for a specific destination only "greedy" neighbors, i.e. those that strictly reduce the hyperbolic distance to the destination.

We propose two algorithms, the "Greedy" backpressure and the "Greediest" backpressure for both static and dynamic networks. The first one performs routing by choosing as next hop node one of the greedy neighbors, while the Greediest backpressure chooses the greedy neighbor with the least hyperbolic distance to the destination. Both algorithms perform backpressure scheduling and as a result routing/scheduling is based on both congestion and distance gradients.

The rest of this paper is organized as follows. In Section 2 we summarize the properties of the hyperbolic space and the greedy embedding, while in sections 3, 4, 5, we describe and analyze the system model and the proposed algorithms. Section 6 provides numerical results, demonstrating the induced improvement in the throughput-delay trade-off.

**2. Greedy Network Embedding in Hyperbolic Space.** The whole infinite hyperbolic plane can be represented inside the finite unit disc $\mathcal{D} = \{z \in \mathcal{C} \| z| < 1\}$ of the Euclidean space; the Poincaré Disc model. The greedy embedding used in this work is based on the Poincaré Disc model. The hyperbolic distance function $d_H(z_i, z_j)$, for two points $z_i, z_j$, in the Poincaré Disc model is given by [6, 8]:

$$\cosh d_H(z_i, z_j) = \frac{2|z_i - z_j|^2}{(1-|z_i|^2)(1-|z_j|^2)} + 1 \tag{2.1}$$

The Euclidean circle $\partial\mathcal{D} = \{z \in \mathcal{C} \| z| = 1\}$ is the boundary at infinity for the Poincaré Disc model. In addition, in this model, the shortest hyperbolic path between two nodes is either a part of a diameter of $\mathcal{D}$, or a part of a Euclidean circle in $\mathcal{D}$ perpendicular to $\partial\mathcal{D}$.

The greedy embedding is constructed by choosing a spanning tree of the graph of the initial network and then embedding the spanning tree into the hyperbolic space according to the algorithm of [6]. If a spanning tree of the graph is greedily embedded in the hyperbolic space then the whole graph is also embedded [7]. From definition, the greedy embedding ensures the existence of at least one greedy path between each source-destination pair in the case of static networks. Every pair of nodes $i, j$ is connected through a unique path, let us denote it as $i,\ i_1,\ i_2, \ldots\ i_k,\ j$, lying on the spanning tree which is embedded in the hyperbolic space. Due to the particular embedding, $i_1$ is at least one greedy neighbor of $i$ for $j$ and $i_k$ is a greedy neighbor of $j$ for $i$.

**3. System Model and Capacity Region.** We consider slotted time $t$ and a wireless multihop network with $N(t)$ nodes at each time $t$. We consider the case where $N(t)$ is constant (static network) and the case of node churn, where existing nodes can leave and new nodes can join the network. The number of packets that arrive in node $i$ for destination $d$ at time $t$ is $A_i^d(t)$, with expected value $\lambda_i^d$ for every $t$. We suppose that each node $i$ stores a queue $q_i^d(t)$ for each destination $d$. We denote with $\mu_{ij}(t)$ the communication traffic between the neighboring nodes $i, j$ at time $t$, and with the $N \times N$ matrix, $[\mu_{i,j}(t)]$, the traffic over all the links, at time $t$. Also, we denote with $\mu_{ij}^d(t)$ the communication traffic on the link between nodes $i, j$ for destination $d$ at time $t$. The arrival and service rates are considered bounded. We denote with $dist_H(i,d)$ the hyperbolic distance between nodes $i, d$ (Section 2) and with $\mathcal{N}(i)$, the one-hop neighborhood of node $i$. Finally, we use the term $I_{S(t)}$ to refer to the set of service rate vectors of all possible independent sets of the graph at time $t$, i.e. maximal sets of links that do not interfere with each other. $I_{S(t)} = I_S$ is a constant set if the network is static and the channel conditions do not change. For brevity and simplicity, we do not always specify the range of the summations which is considered as the number of nodes $N(t)$.

We adapt the capacity region of [9], so as to include the routing constraints of the proposed algorithms. Let us denote as "greedy" paths, the paths consisting of nodes with strictly decreasing distances to the destination. Therefore, the capacity region should allow routing only through greedy paths. The capacity region $\Lambda_G$ is the set of all input rate matrices $(\lambda_i^d)$ with $\lambda_i^d \neq 0$ if $i \neq d$ and $(i,d)$ is a source destination pair, such that there exists a rate matrix $[\mu_{ij}]$ satisfying the following constraints:

- Efficiency constraints: $\mu_{ij}^d \geq 0$, $\mu_{ii}^d = 0$, $\mu_{dj}^d = 0$, $\sum_d \mu_{ij}^d \leq \mu_{ij}$, $\forall\ i,d,j$.
- Flow constraints: $\lambda_i^d + \sum_l \mu_{li}^d \leq \sum_l \mu_{il}^d, \forall\ i,d:\ i \neq d$.
- Routing constraints for the Greedy backpressure: $\mu_{ij}^d = 0$ if $i$ has at least one greedy neighbor for $d$ and $j$ is not one of the $i's$ greedy neighbors;
  and for the Greediest backpressure: $\mu_{ij}^d = 0$ if $i$ has at least one greedy neighbor for $d$ and $j$ is not $i's$ greedy neighbor with the shortest hyperbolic distance to the destination.

As aforementioned, we used the greedy embedding of [6] which works for both static and dynamic networks. At this point, we define the notion of strong stability of the queues, which will be used in the proofs that follow. According to the Definition 3.1 of [9], a queue, $q_i^d$ is strongly stable if $\limsup_{t->\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E(q_i^d(\tau)) < \infty$. If all the queues of the network are strongly stable, then the whole network is strongly stable.

**4. Static Networks.** In this section, we develop our algorithms in the case of static wireless networks, i.e. the number of nodes and their positions are fixed. The proposed algorithms differ from the classic backpressure algorithm in the definition of the weight $P_{ij}(t)$ for each link $ij$, due to their greedy routing constraints. Algorithm 1 describes how the pure backpressure algorithm is modified to follow only greedy paths.

It is important to mention that due to the greedy embedding that ensures the existence of a greedy path for every source-destination pair, the routing constraints of Algorithm 1 are well defined and there does not exist local minima that can cause the packets to get stack at a specific node. Therefore, with probability one the packets will be routed to the destination under Algorithm 1. The following theorem shows the throughput optimality of Algorithm 1 for static networks.

THEOREM 4.1. *If we assume that the arrival rates $\lambda_i^d$ lie inside the capacity region $\Lambda_G$, then the queues of the network are strongly stable, under the Greedy (Greediest) backpressure algorithm for static networks.*

*Proof.* We define two indicator functions dependent on the type of the backpressure algorithm (Greedy or Greediest).
For the Greedy backpressure:
$I_1 = \{dist_H(i,d) > dist_H(j,d) \bigwedge (j \in \mathcal{N}(i))\}$, $I_2 = \{dist_H(i,d) < dist_H(j,d) \bigwedge (i \in \mathcal{N}(j))\}$,
while for the Greediest backpressure:
$I_1 = \{dist_H(i,d) > dist_H(j,d)\ \bigwedge\ dist_H(j,d) = \min_{l \in \mathcal{N}(i)} dist_H(l,d) \bigwedge (j \in \mathcal{N}(i))\}$
$I_2 = \{dist_H(i,d) < dist_H(j,d)\ \bigwedge\ dist_H(i,d) = \min_{l \in \mathcal{N}(j)} dist_H(l,d) \bigwedge (i \in \mathcal{N}(j))\}$
where we observe that $I_2$ equals $I_1$ if we replace $i,j$ with $j,i$ correspondingly.
The queue dynamics in the case of Algorithm 1 are

$$q_i^d(t+1) = \max\{q_i^d(t) - \sum_{j|I_1} \mu_{ij}^d(t), 0\} + \sum_{j|I_2} \mu_{ji}^d(t) + A_i^d(t). \tag{4.1}$$

We denote by $Q(t) = (q_i^d(t))$, the vector of queues of the network. We define the Lyapunov

**Algorithm 1:** Greedy (Greediest)-backpressure algorithm for Static Wireless Networks, performed at time slot $t$

**for** *each directed link* $(i,j)$ **do**
  **for** *each destination* $d$ **do**
    %*Greedy backpressure*%
    **if** $dist_H(i,d) > dist_H(j,d)$ **then**
      $P^d_{ij}(t) = q^d_i(t) - q^d_j(t)$;
    (%*OR Greediest backpressure*%
    **if** $dist_H(i,d) > dist_H(j,d)$ *and* $dist_H(j,d) = \min_{l\in\mathcal{N}(i)} dist_H(l,d)$ **then**
      $P^d_{ij}(t) = q^d_i(t) - q^d_j(t)$;)
    **else**
      $P^d_{ij}(t) = -\infty$;
  %*Define the weight* $P_{ij}(t)$ *as follows* :
  $P_{ij}(t) = \max(\max_d P^d_{ij}(t), 0)$;
  $d^*(i,j) = arg\max_d P^d_{ij}(t)$;
*Choose the communication traffic matrix through the maximization* :
$[\mu_{ij}(t)] = arg_{\mu'\in\mathbf{I_S}} \max \sum_{(i,j)} \mu'_{ij} P_{ij}(t)$
**for** *each directed link* $(i,j)$ **do**
  **if** $\mu_{ij}(t) > 0$ **then**
    *the link* $(ij)$ *serves* $d^*(i,j)$ *with* $\mu^{d^*}_{ij}(t) = \mu_{ij}(t)$;
  *For* $d \neq d^*$ *we set* $\mu^d_{ij}(t) = 0$

function $L(Q(t)) = \sum_{i,d} q^d_i(t)^2$ and take the expected value of the difference $L(Q(t+1)) - L(Q(t))$ so as to compute the Lyapunov Drift:

$$E(L(Q(t+1)) - L(Q(t))|Q(t)) = E\left[\sum_{i,d} q^d_i(t+1)^2 - \sum_{i,d} q^d_i(t)^2 \,\middle|\, Q(t)\right],$$

$$\overset{(4.1),[9]}{\leq} \sum_{i,d}\left\{q^d_i(t)^2 + E\left[\left(\sum_{j|I_1}\mu^d_{ij}(t)\right)^2 + \left(\sum_{j|I_2}\mu^d_{ji}(t) + A^d_i(t)\right)^2 \,\middle|\, Q(t)\right]\right\}$$

$$-\sum_{i,d}\left\{2q^d_i(t)E\left[\left(\sum_{j|I_1}\mu^d_{ij}(t) - \sum_{j|I_2}\mu^d_{ji}(t) - A^d_i(t)\right)\middle| Q(t)\right]\right\} - \sum_{i,d} q^d_i(t)^2,$$

$$\leq B + 2\sum_{i,d} q^d_i(t)\lambda^d_i - 2\sum_{i,d} q^d_i(t) E\left[\left(\sum_{j|I_1}\mu^d_{ij}(t) - \sum_{j|I_2}\mu^d_{ji}(t)\right)\middle| Q(t)\right], \tag{4.2}$$

where $\mu^d_{ij}(t)$ are the service rates computed by Algorithm 1 and $B > 0$ is an upper bound of $\sum_{i,d} E\left[\left(\sum_{j|I_1}\mu^d_{ij}(t)\right)^2 + \left(\sum_{j|I_2}\mu^d_{ji}(t) + A^d_i(t)\right)^2 \middle| Q(t)\right]$.
If the $\lambda^d_i$ lie inside the capacity region, then from corollary 3.9 in [9], there exist rates $\hat{\mu}^d_{ij}(t)$ determined according to the network topology and independently of the queue backlog satisfying $\lambda^d_i + \epsilon = E\left[\sum_{j|I_1}\hat{\mu}^d_{ij}(t) - \sum_{j|I_2}\hat{\mu}^d_{ji}(t)\right]\ \forall\ i,d,\ \epsilon > 0$.

The Greedy (Greediest) backpressure maximizes $\sum_{i,d} q^d_i(t) E\left[\left(\sum_{j|I_1}\mu^d_{ij}(t) - \sum_{j|I_2}\mu^d_{ji}(t)\right)\middle| Q(t)\right]$, so

$$\sum_{i,d} q_i^d(t) E\left[\sum_{j|I_1} \mu_{ij}^d(t) - \sum_{j|I_2} \mu_{ji}^d(t) \middle| Q(t)\right] > \sum_{i,d} q_i^d(t) E\left[\sum_{j|I_1} \hat{\mu}_{ij}^d(t) - \sum_{j|I_2} \hat{\mu}_{ji}^d(t)\right]. \tag{4.3}$$

Therefore the Lyapunov drift (Eq. (4.2)) becomes

$$E[L(Q(t+1)) - L(Q(t))|Q(t)] \leq B - 2\sum_d \sum_i q_i^d(t)\epsilon, \tag{4.4}$$

and from lemma 4.1 of [9], the network is strongly stable. □

The delay performance of a scheduling/routing algorithm for each flow, is closely related to the sum of queues of the nodes on the source-destination path. As stated in [5] the total backlog of a path increases with the increase of its hop-length. As the spanning tree used for the hyperbolic embedding of the network determines the greedy paths used by the flows, it is therefore expected that the choice of the spanning tree will affect the number and hop-length of the greedy paths and therefore the delays experienced by the flows. As a result, different types of spanning trees like minimum weight spanning tree or shortest path spanning tree should be studied for their effect on the delay performance.

**5. Dynamic Networks.** In dynamic networks the topology of the network (and the capacity region) changes due to the addition and deletion of nodes. We assume that the network remains always connected and that nodes come and leave in a much slower rate than the rate of the scheduling/routing process. In this case, new nodes admit a greedy embedding in hyperbolic space according to the embedding of [6]. However, when a node leaves, the greedy property can be locally destroyed and the greedy routing must be adapted in order to avoid possible local minima, without re-embedding the whole network. Thus, we adapt Algorithm 1, so as to perform the classic backpressure when the greedy property is locally lost: if a node has a greedy neighbor for a particular destination it performs either Greedy or Greediest backpressure while if the greedy property is lost for the specific destination, it performs the classic backpressure algorithm. As a result, the lines 4, 5, 9, 10 of Algorithm 1 for the Greedy backpressure are replaced with the following ones:

```
if i has at least one greedy neighbor for a destination d then
    if dist_H(i,d) > dist_H(j,d) then
        P_ij^d(t) = q_i^d(t) - q_j^d(t);
    else
        P_ij^d(t) = -∞;
if i has not greedy neighbor for a destination d then
    P_ij^d(t) = q_i^d(t) - q_j^d(t);
```

For the Greediest backpressure, lines $7-10$ change similarly, with the difference that the second in row if-condition becomes: $dist_H(i,d) > dist_H(j,d)$ *and* $dist_H(j,d) = \min_{l \in \mathcal{N}(i)} dist_H(l,d)$

Proposition 5.1. *If we assume a finite time interval $T$ then the queues of the network remain bounded for this finite time interval (Finite Time Stability), under the Greedy (Greediest) backpressure for both static and dynamic networks.*

*Proof.* The arrival and service rates are considered bounded, i.e. at each time $t$, we have that $\sum_d \sum_{j|I_1} \mu_{ij}^d(t) \leq \mu_{i,max}^{out}$, $\sum_d \sum_{j|I_2} \mu_{ji}^d(t) \leq \mu_{i,max}^{in}$ and $\sum_d A_i^d(t) \leq A_{i,max}$. We will prove the Finite Time Stability (FTS) by using the definition of finite time stability used in [10]. From our state equation, we have that

$$q_i^d(t+1) = \max\{q_i^d(t) - \sum_{j|I_1} \mu_{ij}^d(t), 0\} + \sum_{j|I_2} \mu_{ji}^d(t) + A_i^d(t) \leq q_i^d(t) + \sum_{j|I_1} \mu_{ij}^d(t) + \sum_{j|I_2} \mu_{ji}^d(t) + A_i^d(t),$$

$$\leq q_i^d(t) + \mu_{i,max}^{out} + \mu_{i,max}^{in} + A_{i,max} = q_i^d(t) + B_{\max}^i, \tag{5.1}$$

where $B_{\max}^i = \mu_{i,max}^{out} + \mu_{i,max}^{in} + A_{i,max}$. It follows that $q_i^d(t) \leq q_i^d(0) + B_{\max}^i \cdot t$. Therefore if the initial queues are bounded ($q_i^d(0) < \infty$), the queues at time $T$ are also bounded and this proves the FTS of our system. □

The following theorem shows under certain assumptions the throughput optimality of the Greedy (Greediest) backpressure for dynamic networks. In order to prove Theorem 5.2, we assume that at each time instant only one node can be added or deleted, also that a new node does not change any pre-existing connections in the network and finally that there is a controller that adapts the arrival rates to lie inside the capacity region, after a node churn process. It is important to mention that we use per destination queues, and therefore it is less likely to deal with the finite flow problem [11] when a node leaves, as other nodes will continue to communicate with the destinations of the deleted node.

THEOREM 5.2. *Let $T_i$ be a time interval during which the number of nodes $N(T_i)$ stays constant. In addition, assume that the arrival rates of interval $T_i$, denoted by $\lambda_i^d(T_i)$, lie within the capacity region $\Lambda_G(T_i)$, also dependent on $T_i$. Then the queues $q_i^d(t)$ are strongly stable.*

*Proof.* We first prove Theorem 5.2 in the case of the Greedy backpressure algorithm. Let us consider consecutive time intervals $T_i$, $i = 1...\infty$, with constant number of nodes, $N(T_i)$, at each interval $T_i$. We suppose that the nodes get informed about a node addition or deletion just after the last slot of an interval and before the first slot of the next. We first prove that the Lyapunov drift becomes negative at each $T_i$, when the sum of queues exceeds a certain bound $B_{T_i}$, dependent on the interval $T_i$. We define four indicator functions:

$I_1 = \{dist_H(i,d) > dist_H(j,d) \bigwedge (j \in \mathcal{N}(i))\}$, $I_2 = \{dist_H(i,d) < dist_H(j,d) \bigwedge (i \in \mathcal{N}(j))\}$

$I_3 = \{(Node\ i\ has\ no\ greedy\ neighbor\ for\ destination\ d) \bigwedge (j \in \mathcal{N}(i))\}$

$I_4 = \{(Node\ j\ has\ no\ greedy\ neighbor\ for\ destination\ d) \bigwedge (i \in \mathcal{N}(j))\}$

where we observe that $I_2$ equals $I_1$ and $I_4$ equals $I_3$, if we replace $i, j$ with $j, i$ correspondingly. Also, $I_1 \cap I_3 = \emptyset$, $I_2 \cap I_4 = \emptyset$. The queue dynamics are

$$q_i^d(t+1) = \max\{q_i^d(t) - \sum_{j|\{I_1 \| I_3\}} \mu_{ij}^d(t), 0\} + \sum_{j|\{I_2 \| I_4\}} \mu_{ji}^d(t) + A_i^d(t), \tag{5.2}$$

where $E(A_i^d(t)) = \lambda_i^d(T_i)$ is constant for the period $T_i$ and assumed to lie inside the capacity region $\Lambda_G(T_i)$ for this time period. By following the same steps as in Theorem 4.1, the Lyapunov drift takes the form

$$E[L(q(t+1)) - L(q(t)) | q(t)] \leq B_{T_i} - \sum_{d=1,i=1}^{N(T_i)} q_i^d(t)\epsilon. \tag{5.3}$$

We have proved that for each interval $T_i$ with constant number of nodes $N(T_i)$, if the arrival rates are adapted through a controller to lie inside the capacity region of this interval $\Lambda_G(T_i)$, the sum of queues is bounded. Now we check the Lyapunov drift at the transitions between two intervals with different number of nodes and since we have assumed that at each transition only one node can be added or deleted, two consecutive intervals differ only by one node. If a node is added at time $t$ then, the other nodes get informed about its arrival and may start to send data to the new node. If a node is deleted, the network gets informed about this event and erase the queues destined to the deleted node. To simplify the theoretical proofs, we assume that the IDs of the nodes can be rearranged so that they take consecutive integer values when a node leaves the network.

In the case of a node addition, let us suppose that at time $t$ we have $N$ nodes, while at time $t+1$ we have $N+1$ nodes. Also, $q_i^{N+1}(t) = 0$ and $q_{N+1}^d(t) = 0, \forall d, i$, as at time $t$ the node with ID $N+1$ enters the network. The change in the Lyapunov function, between slots $t, t+1$, is computed as follows

$$L(t+1) - L(t) = \sum_{i=1}^{N+1}\sum_{d=1}^{N+1}(q_i^d(t+1))^2 - \sum_{i=1}^{N}\sum_{d=1}^{N}(q_i^d(t))^2 = \sum_{i=1}^{N+1}\sum_{d=1}^{N+1}(q_i^d(t+1))^2 - \sum_{i=1}^{N+1}\sum_{d=1}^{N+1}(q_i^d(t))^2. \tag{5.4}$$

By using the constant $B_{T_{i+1}}$ (which is computed as $B_{T_i}$ but for the $N+1$ nodes of $T_{i+1}$), instead of $B_{T_i}$, we can mimic the computation of the drift done in Eq. (5.3) to prove that the Lyapunov drift is negative if the sum of queue backlogs is high enough and if the $\lambda_i^d, \forall i, d$ lie inside the capacity region of the interval $T_{i+1}$.

In the case of node deletion, from interval $T_i$ to the $T_{i+1}$, at time $t \in T_i$ we have $N$ nodes, and at time $t' \in T_{i+1}$ we have $N-1$ nodes. By taking out the terms corresponding to the deleted node and rearranging indices, the change in the Lyapunov function is computed as

$$L(t+1) - L(t) \leq \sum_{i=1}^{N-1}\sum_{d=1}^{N-1}(q_i^d(t+1))^2 - \sum_{i=1}^{N-1}\sum_{d=1}^{N-1}(q_i^d(t))^2. \tag{5.5}$$

Similarly with node addition, we mimic the computation of the drift done in Eq. (5.3) by using the constant $B_{T_{i+1}}$ which is computed as $B_{T_i}$ but for the $N-1$ nodes of $T_{i+1}$. Therefore by choosing the highest constant $B_{T_i}$, denoted as $B_{\max} = \max_{\forall T_i}\{B_{T_i}\}$, at each time $t$ that the sum of queue backlogs is higher than $\frac{B_{\max}}{\epsilon}$, the Lyapunov drift will be negative.

$$E\left[E[L(q(t+1)) - L(q(t))|Q(t)]\right] \leq E\left[B_{\max} - \sum_{d,i} q_i^d(t)\epsilon\right],$$

$$E(L(q(T+1))) - E(L(q(0))) \leq T B_{\max} - E(\sum_{t=0}^{t=T}\sum_{d,i} q_i^d(t)\epsilon) \Rightarrow -E(L(q(0))) \leq T B_{\max}$$

$$-E\left(\sum_l \left[\sum_{s=\sum_{x=0}^{x=l-1} T_x+1}^{s=\sum_{x=0}^{x=l} T_x-1}\sum_{i=1,d=1}^{N(T_l)} q_i^d(s)\epsilon + \mathbf{1}_N \sum_{i=1,d=1}^{N(T_{l+1})} q_i^d(\sum_{x=0}^{x=l} T_x)\epsilon + \mathbf{1}_{l=K}\sum_{i=1,d=1}^{N(T_l)} q_i^d(\sum_{x=0}^{x=l} T_x)\epsilon\right]\right), \tag{5.6}$$

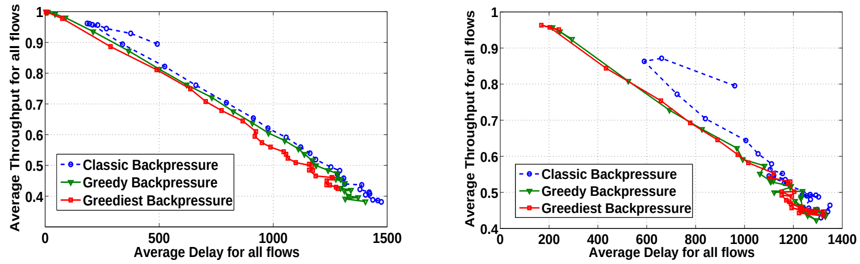
where $\mathbf{1}_N = (\mathbf{1}_{\{N(T_l)-N(T_{l+1})=1, l\neq K\}} + \mathbf{1}_{\{N(T_l)-N(T_{l+1})=-1, l\neq K\}})$, where $\mathbf{1}$ is the indicator function and $T_0 = 0$. Also $l$ spans as many integers $K$, so that we have $\sum_{l=1}^{l=K} T_l = T$ and $N(T_{K+1}) = N(T_K)$. By dividing the previous upper bound by $T$ and taking $T$ ($or$ $l$) $\to \infty$, we conclude the strong stability of the queues. □

Remark 1. *The analysis can be repeated for the Greediest backpressure algorithm by redefining the first two indicator functions as:*

$I_1 = \{dist_H(i,d) > dist_H(j,d) \wedge dist_H(j,d) = \min_{l\in\mathcal{N}(i)} dist_H(l,d) \bigwedge (j \in \mathcal{N}(i))\}$

$I_2 = \{dist_H(i,d) < dist_H(j,d) \wedge dist_H(i,d) = \min_{l\in\mathcal{N}(j)} dist_H(l,d) \bigwedge (i \in \mathcal{N}(j))\}$

**6. Simulation Results.** In this section, we present some MATLAB simulation results that clarify the Pareto dominance of our algorithms over the classic backpressure algorithm concerning the throughput-delay trade-off. We consider a 4x4 grid topology, where each node at each time slot generates traffic for a random destination, with probability equal for all node pairs ranging from $\lambda = 0.025$ to $\lambda = 0.775$ with step increase 0.025. For each $\lambda$ we run the algorithms for 5000 slots. Each link can transmit one packet during a time slot. We consider the one-hop interference model. Throughput is expressed as the percentage of




(a) Greedy (Greediest) for static networks. (b) Greedy (Greediest) for dynamic networks.

FIG. 6.1. *Throughput-Delay Trade-off for the wireless network topology.*

packets that reach their destination divided by those sent from the source for each flow and both throughput and delay are expressed as averages for all flows (source-destination pairs).

In Fig. 6.1(a), the static network embedded in the hyperbolic space runs the Greedy, Greediest and classic backpressure algorithms. In Fig. 6.1(b), 4 nodes have been deleted from the grid while the network is still connected, thus the greedy property is locally lost and the network runs the version of the Greedy (Greediest) backpressure for dynamic networks. The bullets on the curves represent different values of $\lambda$. We observe that for all $\lambda$ and especially for light traffic, the Greedy and Greediest backpressure algorithms achieve a better throughput delay trade-off than the classic backpressure algorithm, as for the same values of throughput they lead to lower delay. Finally, in the dynamic network case, the curves of the proposed algorithms resemble the form of the curve of the classic backpressure, but being on the left and upper side of the latter, they correspond to higher throughput and lower delay.

**Acknowledgments.** The research of Eleni Stai and John Baras was partially supported by the US Air Force Office of Scientific Research under MURI grant FA9550-09-1-0538, and by the National Science Foundation (NSF) under grant CNS-1018346. Eleni Stai was also partially supported by the Foundation for Education and European Culture (IPEP).

REFERENCES

[1] L. Tassiulas, A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum throughput in Multihop Radio Networks", *IEEE Transactions on Automatic Control*, Vol. 37, No. 12, pp. 1936 - 1948, December 1992.

[2] A. Eryilmaz, R. Srikant, "Joint Congestion Control, Routing, and MAC for Stability and Fairness in Wireless Networks", *IEEE J. on Selected Areas in Com. (JSAC)*, Vol. 24, No. 8, pp. 1514 - 1524, July 2006.

[3] A. Eryilmaz, A. Ozdaglar, D. Shah, E. Modiano, "Distributed Cross-Layer Algorithms for the Optimal Control of Multi-hop Wireless Networks", *IEEE/ACM Tran. on Net.*, Vol. 18, No. 2, pp. 638 - 651, 2010.

[4] L. Ying, S. Shakkottai, A. Reddy, S. Liu, "On Combining Shortest-Path and Back-Pressure Routing Over Multi-hop Wireless Networks", *IEEE/ACM Tran. on Net.*, Vol. 19, No. 3, pp. 841 - 854, June 2011.

[5] L. Bui, R. Srikant, A.L. Stolyar, "Novel Architectures and Algorithms for Delay Reduction in Back-pressure Scheduling and Routing", *IEEE INFOCOM 2009*, pp. 2936 - 2940, April 2009.

[6] A. Cvetkovski, M. Crovella, "Hyperbolic Embedding and Routing for Dynamic Graphs", *IEEE INFOCOM 2009*, pp. 1647 - 1655, April 2009.

[7] R. Kleinberg, "Geographic Routing Using Hyperbolic Space", *IEEE INFOCOM*, pp. 1902 - 1909, May 2007.

[8] J. W. Anderson, "Hyperbolic Geometry", *2nd ed. Springer*, 2007.

[9] L. Georgiadis, M. J. Neely, L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks", *now publications*, 2006.

[10] F. Amato, M. Carbone, M. Ariola, C. Cosentino, "Finite-Time Stability of Discrete-Time Systems", *Proc. of American Control Theorem*, June - July 2004.

[11] P. van de Ven, S. Borst, S. Shneer, "Instability of Max Weight Scheduling Algorithms", *IEEE INFOCOM 2009*, pp. 1701 - 1709, April 2009.