

Heat Diffusion Algorithm for Resource Allocation and Routing in Multihop Wireless Networks

Reza Banirazi, Edmond Jonckheere, Bhaskar Krishnamachari

Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089

E-mail: {banirazi, jonckhee, bkrishna}@usc.edu

Abstract—We propose a new scheduling and routing approach, the Heat Diffusion (HD) protocol, using combinatorial analogue of the heat equation in mathematical physics. The algorithm holds for systems subject to time-varying network conditions with general packet arrivals and random topology states, including ad-hoc networks with mobility. Compared to the well-known backpressure policy, the HD protocol is generalized in form and optimized in performance, which considers link penalties and node capacities in the routing. It mitigates the packet looping behavior of backpressure and attempts to communicate less over links of higher costs and with the nodes of lower capacities. While HD policy shows benefits over backpressure, it is developed using the same underlying control laws. Therefore, it can easily leverage all the theoretical works that have been done in improving the original backpressure. For the same reason, it provides a relatively easy path-way to modify existing applications of backpressure to the optimized versions using HD protocol.

I. INTRODUCTION

Backpressure is a well-known algorithm for resource allocation and data routing in wireless networks, which achieves maximum throughput in the presence of time-varying network conditions and without precisely knowing arrival rates. It assigns a weight to each link, equal to the maximum differential backlog between transmitting and receiving nodes, and then chooses link rates to maximize the sum of the products of link rates and link weights. The algorithm, first called max-weight, was originally proposed in [1] for routing traffic over a multi-hop packet radio network with random packet arrivals and fixed set of link selection options. Then the idea was extended to ad-hoc mobile networks and was also combined with optimization techniques [2]–[6].

In this paper, inspired by the heat diffusion process on smooth manifolds, we introduce a new paradigm for the problem of resource allocation and packet routing in multihop wireless networks, referred to as Heat-Diffusion (HD) protocol. We develop the HD protocol along the same line as backpressure, as the differential queue length plays a main role in the routing decision, and as we use a multiple stage strategy to implement the algorithm. Our approach differs from backpressure, as each stage of the algorithm is formulated according to a metaphorical referral to the heat equation. The ubiquitousness of heat equation in mathematics and physics not only gives us a deeper insight into the resource allocation and routing problem, but also provides us with a high flexibility to derive an optimal solution for a very general class of routing networks. However, for the heat equation to be more

than a metaphor, we need to bring it from smooth geometry in which the heat lives to a purely combinatorial domain in which the data network runs. To this end, we utilize the theory of *combinatorial calculus*, which works with a *cell complex* as a countable discrete domain, and preserves such fundamental differential and integral operators as the Laplacian that convey most of the physics of the process.

Before proceeding, it is valuable to have a heuristic comparison between HD protocol and the original backpressure. Recall that the backpressure routing is anecdotally described by the flow of water through a network of pipes under pressure gradients. Certainly, the flow of packets in a data network is restricted by the capacity of each link, and hence the model is more relevant if pipes are replaced with buckets. Then we can imagine a barrel of packets at each node, where the packets may move from barrel i to barrel j if there is a communication link from node i to node j , and for transferring packets, each link uses a bucket whose capacity is equal to the link capacity. Having this analogy, the original backpressure is based on two assumptions: (i) all barrels are of the same storage capacity, and (ii) all buckets require the same amount of work to transfer the same number of packets. Technically, the first assumption implies that all nodes have equal *capacity*, which may reflect power, processing, and storage capability, while the second assumption means that all links are of equal *cost*, which may reflect power consumption, channel quality, and distance.

In contrast to the backpressure, the HD protocol considers a variety of node capacities and link penalties on the network. In this sense, the backpressure policy becomes a special case of the HD protocol, while the latter also provides a better performance under the influence of its ancestor, the heat diffusion process. In particular, the HD protocol inherits from the heat equation the property of jointly optimizing queue backlogs versus capacity of each node, and routing cost versus communication expense of each link. Furthermore, it is proved that the HD protocol achieves maximum throughput of the network in the sense that it guarantees stability of the queue buffers under any arrival rate such that there exists a stabilizing routing policy for it. We also notice that the notions of node capacity and link cost may reflect different constraints from the network. An example is a wireless sensor network in which communications happen over lossy channels and nodes are subject to different memory restrictions. Obviously, the former can be modeled by link costs and the latter by node capacities.

Closing this introduction, in Sec. II we describe the dynamics and definitions of a standard wireless packet network. Section III develops a pure combinatorial model of heat diffusion on a directed and undirected graph, which provides a solid theoretical background for the HD policy. We introduce the HD algorithm in Sec. IV, where its designing lines are compared with those from the backpressure policy. Performance analysis and stability properties of the new policy are discussed in Sec. V. The paper is concluded by Sec. VI.

II. SYSTEM MODELING AND DEFINITIONS

A *routing network* is described as a *connected* graph with N *stationary nodes* and L *directed edges*. The network operates in *timeslots* of the same length, denoted by n . A packet is an atomic entity that must transmit from its source to its destination, which are two distinct nodes in the routing network. All packets are of equal size, and each packet leaves the network layer as soon as reaching its destination. Each node installs separate internal queues for packets with different destinations. A packet destined to node d is called a d -*packet*. At each timeslot, a network layer *routing policy* decides either to transmit a packet to a neighboring node along one of the outgoing edges, or to keep it in the queue at its current node.

We endow each routing edge (i, j) with two independent parameters: (i) a *link capacity* $\mu_{ij}(n)$, also known as link transmission rate, and (ii) a *link cost* $\rho_{ij}(n)$. Furthermore, we endow each routing node i with a *node capacity* $C_i(n)$. We assume that these parameters are fixed within a timeslot n , while they may potentially change across the slots. Specifically, link parameters can change randomly based on the channel conditions and network topology states, and node capacities can be updated according to the node power and processing constraints.

Let $Q_i^{(d)}(n)$ represent the number of d -packets queued at the network layer of node i at the beginning of timeslot n , where $Q_i^{(i)}(n) = 0$ for all n . Notice that the queue contains both random exogenous packet arrivals from the transport layer at node i and endogenous packets forwarded through the network layer. Let *stochastic process* $A_i^{(d)}(n)$ represent the exogenous traffic generated at node i in timeslot n , where $A_i^{(i)}(n) = 0$ for all n . We assume *no congestion control* at the transport layer, so that all incoming traffic arrives directly to the network layer. Let $f_{ij}^{(d)}(n)$ denote the *actual* number of d -packets transmitted via link (i, j) at timeslot n , constrained as

$$\sum_d f_{ij}^{(d)}(n) \leq \mu_{ij}(n).$$

Assume that routing decision happens at the beginning of each timeslot and packets are transferred to the network layer at the end of each timeslot. Then the dynamics of the queue length process for d -packets at node i is described as

$$Q_i^{(d)}(n+1) = Q_i^{(d)}(n) - D_i^{(d)}(n) + E_i^{(d)}(n) \quad (1)$$

$$D_i^{(d)}(n) = \sum_{b \in \text{out}(i)} f_{ib}^{(d)}(n) \quad (2)$$

$$E_i^{(d)}(n) = A_i^{(d)}(n) + \sum_{a \in \text{in}(i)} f_{ai}^{(d)}(n) \quad (3)$$

where $D_i^{(d)}(n)$ and $E_i^{(d)}(n)$ represent the total number of d -packets departed from and arrived at node i , respectively, in the timeslot n , and $\text{out}(i)$ and $\text{in}(i)$ denote the sets of outgoing and incoming neighbors of node i , respectively.

Definition 1 (Network Stability): A queue $Q_i^{(d)}(n)$ is stable, if $\limsup_{n \rightarrow \infty} \mathbb{E}\{Q_i^{(d)}(n)\} < \infty$. A network is stable if all its queues are stable.

Contrary to wireline networks where links are independent resources, in the wireless case two links cannot be simultaneously activated if they have inter-link interference. A *schedule* is defined as a set of links in which no two links interfere with each other, and is called *maximal* if no more links can be added to that without violating the interference constraints. Let each maximal schedule \mathcal{S} be represented as a $\{0, 1\}^{N \times N}$ matrix, referred to as *scheduling matrix*, such that the entry associated with link (i, j) is set to 1 if the link is included in the maximal schedule, and to 0 otherwise. We define *scheduling set* \mathcal{S} as a collection of *all possible maximal schedules* in the network, which may be described by a finite (but arbitrarily large) set. Note that this generalizes one-hop interference (primary interference), two-hop interference (secondary interference), or generally k -hop interference models popularly used in literature [7]–[10].

Assume that the arrival processes are mutually independent with long-term average expected values as

$$\lambda_i^{(d)} = \lim_{\tau \rightarrow \infty} 1/\tau \sum_{n=0}^{\tau-1} \mathbb{E}\{A_i^{(d)}(n)\}$$

and then construct the overall traffic rate matrix $\lambda = [\lambda_i^{(d)}]$ for $i, d = 1, \dots, N$. We further assume that there is a deterministic upper bound on the total exogenous arrivals to each node, i.e., $\sum_{d=1}^N A_i^{(d)}(n) \leq A_i^{\max}$ for all n . At each timeslot n , a routing policy selects a scheduling matrix $\mathcal{S}(n) \in \mathcal{S}$, where $\mathcal{S}(i, j)(n) = 1$ indicates that the link (i, j) is activated and may transmit at most $\mu_{ij}(n)$ packets from node i to node j . A routing policy stabilizes a traffic rate matrix λ , if it stabilizes the network under λ , in the sense of Definition 1.

Definition 2 (Network Capacity): Given a routing policy, its *stability region* is the set of all traffic rate matrices that can be stabilized by it. Network *capacity region* Λ is the union of the stability regions achieved by all possible routing policies, including those of perfect precognition about random events.

Definition 3 (Throughput Maximizing Policy): A routing policy is throughput-maximal, if it stabilizes all *admissible* traffic rate matrices $\lambda \in \Lambda$.

Given a wireless network with time varying topology and inter-link interference, for a traffic rate matrix λ to be admissible, a necessary and sufficient condition is to have a *hyper-flow* that jointly satisfies flow conservation for all nodes, and rate constraints for all links, in a long-term average sense [11]. Let $\bar{f}_{ij}^{(d)}$ denote the long-term average expected number of d -packets transmitted through link (i, j) , which takes the value 0 for either $i = j$ or $i = d$. Let scheduling set be $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_{|\mathcal{S}|}\}$. Hyper-flow feasibility conditions in long-term average are described by

$$\lambda_i^{(d)} = \sum_{b \in \text{out}(i)} \bar{f}_{ib}^{(d)} - \sum_{a \in \text{in}(i)} \bar{f}_{ai}^{(d)} \quad (4)$$

$$\sum_{i=1}^N \lambda_i^{(d)} = \sum_{a \in \text{in}(d)} \overline{f_{ad}^{(d)}} \quad (5)$$

$$\sum_{d=1}^N \overline{f_{ij}^{(d)}} \leq \sum_{k=1}^{|\mathcal{S}|} \alpha_k \mathbf{S}_k(i, j) \overline{\mu_{ij}} \quad (6)$$

where $\overline{\mu_{ij}} = \lim_{\tau \rightarrow \infty} 1/\tau \sum_{n=0}^{\tau-1} \mu_{ij}(n)$, and $\sum_{k=1}^{|\mathcal{S}|} \alpha_k = 1$, $\alpha_k \geq 0$. For long-term average expected rates of d -packets, (4) and (5) refer to flow conservation at intermediate nodes $i \neq d$ and destination nodes d , respectively.

Remark 1: Note that the resulting long-term averages of link hyper-flows are not fixed, but depend on the resource allocation and routing policy that determines which scheduling matrix is picked up at each timeslot. In this sense, the convex factor α_k represents the long-term average fraction of timeslots at which scheduling matrix \mathbf{S}_k is chosen. Also note that the constraints (4)–(6) imply that the network capacity region Λ is convex, closed, and bounded. Furthermore, if $\lambda \in \Lambda$ then $\lambda' \in \Lambda$ for any traffic rate $\lambda' \preceq \lambda$.

III. COMBINATORIAL HEAT EQUATION

In *combinatorial calculus*, one works with a *cell complex* as a countably discrete domain. In contrast to *space discretization*, where the main goal is to find an accurate triangulation of a space as the computational grid, here we deal with a discrete domain, e.g., a graph, entirely as its own entity with no reference to an underlying continuous process. In other words, while the numerical partial differential equation puts emphasis on the fidelity of the discrete approximation to the desired analytical solution, combinatorial calculus establishes a separate, equivalent framework that operates on a pure discrete domain, of which a graph is a special case. Consequently, the concerns of numerical discretization about approaching a continuous solution in the limit are irrelevant in the context of combinatorial calculus. Due to the space limitation, here we use the related terminology and results from combinatorial calculus without precisely introducing them. For detailed explanations, we refer interested readers to [12]–[15].

To derive the exact combinatorial version of heat equation, we begin with the law of mass conservation, which requires that the amount of heat in any region must either leave through the boundary or have an external source at any time t ,

$$\frac{\partial}{\partial t} \iiint_{\mathcal{X}} Q dV = - \iint_{\partial \mathcal{X}} \langle F, n \rangle dA + \iiint_{\mathcal{X}} \Lambda dV$$

where $Q(x, y, z, t)$ is a scalar field representing the spatial distribution of heat on a smooth manifold \mathcal{X} , charted in (x, y, z) local coordinates; $F(x, y, z, t)$ is the flux of heat through the boundary $\partial \mathcal{X}$; scalar field $\Lambda(x, y, z, t)$ represent heat sources, where a positive value indicates a source and a negative value a sink; n and dA denote outward unit normal vector field and surface element on the boundary, respectively; and dV is the volume element on the manifold. A change in internal energy per unit volume, ∇Q , is proportional to the change in temperature, ∇U , through spatial heat capacity $C(x, y, z)$, i.e., $\nabla Q = C \nabla U$. Choosing zero energy at absolute zero temperature, this is rewritten as

$$Q(x, y, z, t) = C(x, y, z)U(x, y, z, t) \quad (7)$$

Since \mathcal{X} is arbitrary, considering (7), the mass conservation equation is equivalent to

$$C \frac{\partial U}{\partial t} = -\text{div} F + \Lambda \quad (8)$$

By Fick's law of diffusion, the amount of flux is proportional to the temperature gradient,

$$F = -\sigma \nabla U \quad (9)$$

where $\sigma(x, y, z)$ represents spatial *thermal diffusivity*, which is proportional to *thermal conductivity*. Equation (9) states that at each point, the amount of heat flux is proportional to the temperature gradient. Putting (8) and (9) together yields

$$C \frac{\partial U}{\partial t} = \sigma \nabla^2 U + \Lambda \quad (10)$$

where ∇^2 denotes *Laplace-Beltrami operator*.

We now construct the combinatorial version of (7)–(10) on an *undirected* graph as a 1-complex, where manifold \mathcal{X} is replaced by a 0-chain vector η_0 ; variables Q , U , and Λ are replaced by 0-cochain vectors \mathbf{Q} , \mathbf{U} , and $\boldsymbol{\lambda}$, respectively, that act as node functions on the graph; and flux F is replaced by a 1-cochain vector \mathbf{f} that acts as an edge function on the graph. Furthermore, the effects of nonuniform heat flux due to σ and inhomogeneous heat capacity due to C are reflected as an edge weight matrix $\boldsymbol{\sigma}$ and a node weight matrix \mathbf{C} on the graph, respectively. Thus the *exact* combinatorial version of the heat equations on an undirected, weighted graph is obtained as

$$\frac{d(\eta_0^\top \mathbf{Q})}{dt} = -(\mathbf{B} \eta_0)^\top \mathbf{f} + \eta_0^\top \boldsymbol{\lambda} \quad (11)$$

$$\mathbf{Q} = \text{diag}(\mathbf{C}) \mathbf{U} \quad (12)$$

$$\mathbf{f} = \text{diag}(\boldsymbol{\sigma}) \mathbf{B}^\top \mathbf{U} \quad (13)$$

$$\text{diag}(\mathbf{C}) \frac{d\mathbf{U}}{dt} = -\mathbf{B} \text{diag}(\boldsymbol{\sigma}) \mathbf{B}^\top \mathbf{U} + \boldsymbol{\lambda} \quad (14)$$

where $\text{diag}(\mathbf{v})$ denotes the diagonal matrix expansion of vector \mathbf{v} . Given a graph, matrix \mathbf{B} represents *node-edge incidence matrix*, in which $B(i, \ell)$ is 1 if node i is the sending node of oriented edge ℓ , is -1 if i is the receiving node, and is 0 otherwise. Here, the term “orientation” refers to an arbitrary choice of edge orientation in a cell complex, which should not be confused with the edge direction in a directed network.

Remark 2: In the field of combinatorial calculus, the matrix $\boldsymbol{\Delta}_w^0 = [\text{diag}(\mathbf{C})]^{-1} \mathbf{B} \text{diag}(\boldsymbol{\sigma}) \mathbf{B}^\top$ is known as weighted 0-Laplace-deRham operator, which is a generalization of the standard graph Laplacian $\mathbf{B} \mathbf{B}^\top$ for an undirected graph.

We now consider how the diffusion changes when the underlying graph is *directed*. With no loss of generality, let the edge orientation match the edge direction. Assuming a directed graph, the conservation of mass in (11) is unchanged, but Fick's law (13) is modified to permit the flow in only one direction, that is,

$$\mathbf{f} = \max\{0, \text{diag}(\boldsymbol{\sigma}) \mathbf{B}^\top \mathbf{U}\}. \quad (15)$$

Therefore, diffusion on a directed graph is governed by

$$\text{diag}(\mathbf{C}) \frac{d\mathbf{U}}{dt} = -\mathbf{B} \text{diag}(\boldsymbol{\sigma}) \mathbf{B}^\top \text{diag}(\mathbf{1}_{(\mathbf{B}^\top \mathbf{U} > 0)}) \mathbf{U} + \boldsymbol{\lambda}$$

where $\mathbf{1}_{(v>0)}$ is an indicator vector function, in which the i th entry is 1 if $v_i > 0$, and 0 otherwise. One may tempt to consider matrix $[\text{diag}(\mathbf{C})]^{-1} \mathbf{B} \text{diag}(\boldsymbol{\sigma}) \mathbf{B}^\top \text{diag}(\mathbf{1}_{\mathbf{B}^\top \mathbf{U} > 0})$ as the Laplacian operator for a directed weighted graph, though it is an operand-dependent operator.

IV. HEAT DIFFUSION ALGORITHM

This section introduces the main result of this paper, i.e., a throughput-maximal scheduling and forwarding mechanism for a generalized routing network with the vectors of link costs and node capacities as presented in Sec. II, using the combinatorial heat equation derived in Sec. III. As the spirit of our design resembles the backpressure algorithm, we first provide the details of backpressure policy as follows.

Definition 4 (Backpressure Policy): At a timeslot n , for each link (i, j) , define $Q_{ij}^{(d)}(n) = Q_i^{(d)}(n) - Q_j^{(d)}(n)$ and do the following three stages (ties broken arbitrarily): **Weighting:** find the link optimal d -packet to be served as $d_{ij}^*(n) = \arg \max_d Q_{ij}^{(d)}(n)$ and the link optimal weight as $w_{ij}^*(n) = \mu_{ij}(n) Q_{ij}^{(d^*)}(n)$. **Scheduling:** choose a scheduling matrix $\mathbf{S}(n) \in \arg \max_{\mathbf{S} \in \mathcal{S}} \sum_{(i,j)} \mathbf{S}(i, j) w_{ij}^*(n)$. **Forwarding:** serve the queue holding packets destined to node $d_{ij}^*(n)$ over the activated link (i, j) at maximum rate $\mu_{ij}(n)$. If there are not enough d^* -packets to send, transmit null packets.

The same way as backpressure policy, we design the HD algorithm in three stages, albeit to formulate each stage we inspire ourselves from the combinatorial heat equations of Sec. III. In doing so, we associate with the flow of each d -packet a corresponding flow of heat on the network, for which node d is the single sink. In this analogy, the number of d -packets at each node plays the role of heat quantity in the combinatorial heat equations, where d -temperature at node i is defined by $U_i^{(d)} = Q_i^{(d)}/C_i$. At each timeslot, the algorithm computes the weight attributed to each link in accordance with a metaphorical view on the heat equation, which directly influences the stages of weighting and scheduling. Furthermore, in the forwarding stage, the number of transferring packets along each link is subject to homogenizing temperature over the network as far as possible, by the same token that the heat diffuses through a manifold.

To assign a weight to each link, let us look at the trait of heat diffusion on a single link during one timeslot as an *adiabatic process*. For ease of notation, we assume that there is only one destination node of *infinite capacity* in the network. Consider link (i, j) at timeslot n with differential queue backlog $Q_{ij}(n) = Q_i(n) - Q_j(n)$. For a while, assume that there is no limit on capacity with links. This is required, because combinatorial heat equation is not concerned with link capacities, where we will relax this assumption very soon. Let t denote the continuous-time within timeslot n of length T . From (15), by $\sigma_{ij} = 1/\rho_{ij}$, one obtains

$$\tilde{f}_{ij}(t) = \max \left\{ 0, \frac{1}{\rho_{ij}} \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} \tilde{U}_i(t) \\ \tilde{U}_j(t) \end{pmatrix} \right\}, t \in [0, T]$$

with initial conditions $\tilde{U}_i(0) = U_i(n)$ and $\tilde{U}_j(0) = U_j(n)$.

Here a tilde on the top signifies that we are working in continuous-time.

Neglecting time constant of the process compared with the length of timeslot, T , we get $\lim_{t \rightarrow T} \tilde{f}_{ij}(t) = 0$. This is well-justified in the sense that by a link capacity equal to μ_{ij} , one can transmit μ_{ij} packets in a duration of time equal to the link propagation delay, which is practically negligible. By the law of mass conservation, $\tilde{Q}_i(0) + \tilde{Q}_j(0) = \tilde{Q}_i(T) + \tilde{Q}_j(T)$, where $\tilde{Q}_i(T) = \lim_{t \rightarrow T} \tilde{Q}_i(t)$. Furthermore, since the process is thermally adiabatic and so insulated, if $\tilde{U}_i(0) > \tilde{U}_j(0)$, then $\tilde{U}_i(T) = \tilde{U}_j(T)$. Otherwise, no diffusion happens and neither \tilde{U}_i nor \tilde{U}_j is changing during $t \in [0, T]$. This equivalently means that at the end of timeslot n , we have $C_j \tilde{Q}_i(T) = C_i \tilde{Q}_j(T)$, iff $U_i(0) \geq U_j(0)$. By considering the laws of mass conservation and adiabatic condition together,

$$\tilde{Q}_i(T) = \frac{C_i(\tilde{Q}_i(0) + \tilde{Q}_j(0))}{C_i + C_j}, \tilde{Q}_j(T) = \frac{C_j(\tilde{Q}_i(0) + \tilde{Q}_j(0))}{C_i + C_j}$$

for $\tilde{U}_i(0) \geq \tilde{U}_j(0)$. Subtracting these two equations, and replacing the initial conditions $\tilde{Q}_i(0)$ and $\tilde{Q}_j(0)$ by $Q_i(n)$ and $Q_j(n)$, respectively, one calculates the maximum number of packets that can be transmitted from node i to node j during timeslot n as

$$f_{ij}^{\max}(n) = \max \left\{ 0, \left\lceil \frac{C_j Q_i(n) - C_i Q_j(n)}{C_i + C_j} \right\rceil \right\} \quad (16)$$

where $\lceil x \rceil$ returns the ceiling value of x .

To fix our initial assumption that link (i, j) is of infinite capacity, we need to restrict $f_{ij}^{\max}(n)$ to the link capacity $\mu_{ij}(n)$. Furthermore, the number of transmitted packets can be at most as many number of packets as $Q_i(n)$ holds. Putting all these together, the actual number of packets transmitted from node i to node j during timeslot n , under an adiabatic heat diffusion condition on link (i, j) , is given by

$$f_{ij}(n) = \min \left\{ f_{ij}^{\max}(n), Q_i(n), \mu_{ij}(n) \right\}. \quad (17)$$

In the following, we design three stages of heat-diffusion algorithm using expression (17).

Weighting Stage: At a timeslot n , for each link (i, j) , the algorithm first finds the optimal d -packet to transmit as

$$U_{ij}^{(d)}(n) = \max \left\{ 0, \frac{Q_i^{(d)}(n)}{C_i(n)} - \frac{Q_j^{(d)}(n)}{C_j(n)} \right\} \quad (18)$$

$$d_{ij}^*(n) = \arg \max_d U_{ij}^{(d)}(n)$$

where ties are broken arbitrarily. To attribute a weight to each link, the algorithm performs the following:

$$C_{ij}(n) = \frac{C_i(n)C_j(n)}{C_i(n) + C_j(n)}$$

$$f_{ij}^{(d^*)}(n) = \min \left\{ \left\lceil C_{ij}(n) U_{ij}^{(d^*)}(n) \right\rceil, Q_i^{(d^*)}(n), \mu_{ij}(n) \right\} \quad (19)$$

$$w_{ij}^*(n) = \left(f_{ij}^{(d^*)}(n) \right) \left(U_{ij}^{(d^*)}(n) \right) \quad (20)$$

where $C_{ij}(n) = C_i(n)$, or equivalently $C_j(n) = \infty$, whenever $j = d_{ij}^*(n)$. This simply implies that *each node is of infinite*

capacity for the packets destined to it. It turns out that equating queues is not applied when the receiving node is the destination, where the maximum possible packets, i.e., $\min\{Q_i^{(d^*)}(n), \mu_{ij}(n)\}$, are transmitted.

Remark 3: In (19), the left most term inside the braces is nonnegative, due to (18). Also note that this term is equivalent to (16) for d^* -packet. This is addressed through replacing $U_{ij}^{(d^*)}(n)$ by (18) for $d = d^*$, and the fact that $C_j(n)Q_i^{(d^*)}(n) - C_i(n)Q_j^{(d^*)}(n) = 0$, iff $U_{ij}^{(d^*)}(n) = 0$.

Scheduling Stage: After assigning the optimal weight (20) to each link, the scheduling matrix is chosen by solving a centralized optimization problem (ties broken arbitrarily) as

$$\mathbf{S}(n) \in \arg \max_{\mathbf{S} \in \mathcal{S}} \sum_{(i,j)} \mathbf{S}(i,j) w_{ij}^*(n) / \rho_{ij}(n). \quad (21)$$

Remark 4: Take note of some important distinctions between backpressure and HD algorithms: (i) Using $U_{ij}^{(d^*)}(n)$ in (20), in lieu of $Q_{ij}^{(d^*)}(n)$ of backpressure policy, the HD algorithm tends to transmit packets to the nodes of highest capacity. (ii) Including link cost $\rho_{ij}(n)$ in (21), the HD algorithm gives priority to the links of lowest penalty, when possible to keep the network stable. (iii) Weighting based on the actual number of transmittable packets $f_{ij}^{(d^*)}(n)$ in (20), in lieu of $\mu_{ij}(n)$ of backpressure policy, the HD algorithm attempts to optimize the performance of routing policy in light and moderate traffic rates. As an example, consider two links a and b with differential queue backlogs of 1 and 3, and link capacities of 10 and 3, respectively. So the consequence of activating link a is transmission of 1 packet, versus 3 packets for link b . Observe that the backpressure policy selects link a , while the HD algorithm chooses link b .

Forwarding Stage: Subsequent to the scheduling stage, each activated link transmits $f_{ij}^{(d^*)}(n)$ number of packets in accordance with (19).

Remark 5: Despite the backpressure policy that forwards the maximum possible number of d^* -packets across activated links, here packet forwarding is governed by diffusion mechanism. This mitigates the *packet looping* behavior of the backpressure policy. As an example, consider a network with only 2 packets at node i , and with no new arrivals. Let bidirectional link (i,j) be of the highest capacity, and greater than 2, among all links connected to nodes i and j . Obviously, routing under the backpressure policy loops these two packets between nodes i and j forever. On the contrary, one can see that the heat-diffusion algorithm successfully transmits both packets to the destination without looping.

V. STABILITY AND PERFORMANCE

In this section, we will show that the HD algorithm supports the entire capacity region Λ , in the sense that it stabilizes the network whenever the rate matrix lies within Λ .

Proposition 1 (Throughput-Maximizing Policy): Consider a routing network with N wireless nodes. The HD routing algorithm stabilizes the network for any traffic rate matrix λ strictly interior to the network capacity region Λ .

Proof: For simplicity, we consider only one destination node in the network. We further assume C_i constant for all nodes and $\rho_{ij} = 1$ for all links. A complete proof for the general case of multi-destination network with time-varying C_i and ρ_{ij} is given in [19]. Consider a Lyapunov energy function $V(n) = 1/2 \sum_i Q_i(n)^2 / C_i$ and define Lyapunov drift $\Delta V(n) = V(n+1) - V(n)$. In the remaining part, we drop n from all variables for the ease of notation. Using (1), some mathematical manipulation yields

$$\Delta V = \sum_i \left[\frac{1}{2C_i} (E_i^2 + D_i^2) + \frac{Q_i}{C_i} (E_i - D_i) - \frac{E_i D_i}{C_i} \right]$$

From (2) and (3) we obtain

$$\Delta V \leq BN + \sum_i \frac{Q_i}{C_i} A_i - \sum_i \frac{Q_i}{C_i} \sum_{a,b} (f_{ib} - f_{ai}) \quad (22)$$

$$B = \frac{1}{2N} \sum_i \frac{1}{C_i} \left[(A_i^{\max} + \sum_a \mu_{ai}^{\max})^2 + (\sum_b \mu_{ib}^{\max})^2 \right]$$

where constant μ_{ij}^{\max} stands for a deterministic upper bound on $\mu_{ij}(n)$. Since λ is assumed strictly interior to the capacity region Λ , there exists a vector ϵ with positive entries such that $\lambda + \epsilon \in \Lambda$. Therefore, there exists a randomized routing policy which stabilizes λ based only on the current topology state and so independent of the queue occupancies [16]. Also from feasibility condition (4), for any stabilizable traffic rate, there exists a hyper-flow in long-term average such that

$$\mathbb{E} \left\{ \sum_{a,b} (f'_{ib} - f'_{ai}) \right\} = \lambda_i + \epsilon_i \quad (23)$$

for some f'_{ib} and f'_{ai} . It is proved in [19] that the HD policy minimizes the Lyapunov drift (22) compared to any other policy, including backpressure. Also observe that

$$\sum_i \frac{Q_i}{C_i} \sum_{a,b} (f_{ib} - f_{ai}) = \sum_{(i,j)} f_{ij} \left(\frac{Q_i}{C_i} - \frac{Q_j}{C_j} \right).$$

Considering these two facts, taking conditional expectation with respect to $\mathbf{Q}(n)$ from (22), and using (23) yield

$$\mathbb{E} \{ \Delta V | \mathbf{Q} \} \leq BN - \sum_i \frac{Q_i}{C_i} \epsilon_i.$$

Defining $\|\mathbf{Q}\| = \sum_i Q_i$ and $\delta = \min_i(\epsilon_i) / \max_i(C_i)$ gives

$$\mathbb{E} \{ \Delta V | \mathbf{Q} \} \leq BN - \delta \|\mathbf{Q}\|. \quad (24)$$

Hence, for $\|\mathbf{Q}\| > BN/\delta$, we get $\mathbb{E} \{ \Delta V | \mathbf{Q} \} < 0$, and by Proposition 2 of [17], the queuing system is stable. ■

Proposition 2 (Guaranteed Average Congestion): For the wireless network of Proposition 1, the HD routing algorithm guarantees bounded average queue occupancies as

$$\lim_{\tau \rightarrow \infty} \frac{1}{\tau} \sum_{n=0}^{\tau-1} \sum_{i,d} \mathbb{E} \{ Q_i^{(d)}(n) \} \leq \frac{NB}{\delta}$$

for some finite constants $B > 0$ and $\delta > 0$.

Proof: Assume (24) holds for all timeslots. Taking expectation from (24), by the law of iterated expectations,

$$\mathbb{E} \{ V(n+1) \} - \mathbb{E} \{ V(n) \} \leq BN - \delta \sum_i \mathbb{E} \{ Q_i(n) \}.$$

Summing up the above inequality over timeslots $n=0, \dots, \tau-1$, and using telescopic series on the left hand side, we get

$$\mathbb{E} \{ V(\tau) \} - \mathbb{E} \{ V(0) \} \leq BN\tau - \delta \sum_{n=0}^{\tau-1} \sum_i \mathbb{E} \{ Q_i(n) \}$$

where $V(0)$ is assumed to be finite, and $V(\tau)$ will be also finite due to Proposition 1 that the HD policy stabilizes λ . Dividing by τ and taking limit as $\tau \rightarrow \infty$ yield the result. ■

Now, we introduce a heuristic fluid model of the heat-diffusion algorithm. We assume that time is continuous and the evolution of each queue is governed by the following differential equation:

$$\dot{\tilde{Q}}_i^{(d)} = - \sum_{b \in \text{out}(i)} \tilde{f}_{ib}^{(d)} + \sum_{a \in \text{in}(i)} \tilde{f}_{ai}^{(d)} + \lambda_i^{(d)} \quad (25)$$

for all $i, d = 1, \dots, N$, where dot on the top denotes the continuous-time derivative. The HD algorithm assigns link rates at every instant of time as described in Sec. IV. Then, the following asymptotic stability result holds.

Proposition 3 (Long Term Average Convergence): Starting from any initial condition $\tilde{Q}(0)$, the states of the continuous-time system (25) asymptotically converges to the states of a traditional heat diffusion on a directed graph with edge weights of $\sigma_{ij} = \bar{C}_{ij}/\bar{\rho}_{ij}$, where \bar{C}_{ij} and $\bar{\rho}_{ij}$ denote the long-term averages of C_{ij} and ρ_{ij} , respectively.

Proof: We give a sketch of the proof only for one destination node, with C_{ij} constant and $\rho_{ij} = 1$. A general version of this proposition is proved in [19]. We indicate variables in the traditional heat diffusion system with superscript *. Consider a node energy function as $V(i) = 1/2 C_i (\tilde{U}_i - \tilde{U}_i^*)^2$. Taking derivative, and substituting from (25) yield

$$\begin{aligned} \dot{V}(i) &= \lambda_i (\tilde{U}_i - \tilde{U}_i^*) - \tilde{U}_i \sum_b \tilde{f}_{ib} + \tilde{U}_i^* \sum_b \tilde{f}_{ib} \\ &\quad + \tilde{U}_i \sum_a \tilde{f}_{ai} - \tilde{U}_i^* \sum_a \tilde{f}_{ai}. \end{aligned}$$

Since λ_i is considered as the same input to both routing network and traditional heat graph, by the Fick's law (15) developed for the directed graph, we provide

$$\lambda_i = \sum_b \sigma_{ib} \tilde{U}_{ib}^* - \sum_a \sigma_{ai} \tilde{U}_{ai}^*$$

where from (12), $\tilde{U}_{ij}^* = \max \{0, \tilde{Q}_i^*/C_i - \tilde{Q}_j^*/C_j\}$. Using these equations in $\dot{V}(i)$ and summing up over all nodes,

$$\begin{aligned} \dot{V} &= \sum_i \dot{V}(i) \\ &= \sum_{(i,j)} \left[\tilde{f}_{ij}(\tilde{U}_i^* - \tilde{U}_j^*) - \tilde{f}_{ij}(\tilde{U}_i - \tilde{U}_j) \right. \\ &\quad \left. + \sigma_{ij} \tilde{U}_{ij}^* (\tilde{U}_i - \tilde{U}_j) - \sigma_{ij} \tilde{U}_{ij}^* (\tilde{U}_i^* - \tilde{U}_j^*) \right] \\ &= \sum_{(i,j)} (\sigma_{ij} \tilde{U}_{ij}^* - \tilde{f}_{ij}) \underbrace{((\tilde{U}_i - \tilde{U}_i^*) - (\tilde{U}_j - \tilde{U}_j^*))}_M. \end{aligned}$$

exploration on (18)–(21), one can see that if $M > 0$, then $\tilde{f}_{ij} > \sigma_{ij} \tilde{U}_{ij}^*$ and so $\dot{V} < 0$. Also if $M < 0$, it can be shown that necessarily $\tilde{f}_{ij} < \sigma_{ij} \tilde{U}_{ij}^*$ and so $\dot{V} < 0$. These imply that $\lim_{t \rightarrow \infty} V \rightarrow 0$, and equivalently $\lim_{t \rightarrow \infty} \tilde{U}_i \rightarrow \tilde{U}_i^*$, which concludes the proof. ■

VI. CONCLUSION

A new wireless routing strategy was developed that emulates the heat diffusion and shows some benefits over the backpressure policy. The new policy has been proved to stabilize the

network for any rate matrix in the interior of the capacity region. A challenging question is what network, in the sense of topology, has the largest capacity region. Preliminary results [18] indicate that Euclidean networks have larger capacity region than hyperbolic networks. In particular, Proposition 3 hinges the dynamics of HD routing policy with a traditional heat diffusion system on graphs, which can open a door to utilize the tools of continuous graph theory in the study of packet based, time slotted routing networks. The effectiveness of the HD algorithm has been examined through simulations, though due to the space limitation, the results were not included in this paper. For more detailed explanations and simulation results, refer to the technical report [19].

REFERENCES

- [1] L. Tassioulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multi-hop radio networks," *IEEE Tran. Automatic Control*, vol. 37, no. 12, pp. 1936–1949, 1992.
- [2] M. Neely, E. Modiano, and C. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE Conf. Computer Communications (INFOCOM)*, 2003.
- [3] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," *IEEE Conf. Decision Control (CDC)*, 2004.
- [4] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-Length-Based scheduling and congestion control," *IEEE Conf. Computer Communications (INFOCOM)*, 2005.
- [5] L. Georgiadis, M. Neely, and R. Tassioulas, "Resource allocation and cross-layer control in wireless networks," *Foundations Trends Networking*, vol. 1, no. 1, pp. 1–144, 2006.
- [6] J. Lee, R. Mazumdar, and N. Shroff, "Opportunistic power scheduling for dynamic multiserver wireless systems," *IEEE Trans. Wireless Communications*, vol. 5, no. 6, pp. 1506–1515, 2006.
- [7] H. Balakrishnan, C. Barrett, V. Kumar, M. Marathe, and S. Thite, "The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks," *IEEE J. Selected Areas Communications*, vol. 22, no. 6, pp. 1069–1079, 2004.
- [8] X. Lin and N. Shroff, "The impact of imperfect scheduling on cross-layer rate control in multihop wireless networks," *IEEE Conf. Computer Communications (INFOCOM)*, 2005.
- [9] L. Bui, A. Eryilmaz, R. Srikant, and X. Wu, "Joint asynchronous congestion control and distributed scheduling for multi-hop wireless networks," *IEEE Conf. Computer Communications (INFOCOM)*, 2006.
- [10] X. Wu, R. Srikant, and J. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *IEEE Trans. Mobile Computing*, vol. 6, no. 6, pp. 595–605, 2007.
- [11] L. Tassioulas, "Adaptive back-pressure congestion control based on local information," *IEEE Tran. Automatic Control*, vol. 40, pp. 236–250, 1995.
- [12] R. Abraham, J. Marsden, and T. Ratiu, *Manifolds, Tensor Analysis, and Applications*. Springer, 1988.
- [13] D. Lovelock, and H. Rund, *Tensors, Differential Forms, and Variational Principles*. Dover Publications, 1989.
- [14] T. Frankel, *The Geometry of Physics: An Introduction*. Cambridge University Press, 1997.
- [15] S. Carroll, *Spacetime and Geometry: An Introduction to General Relativity*. Addison Wesley, 2004.
- [16] L. Tassioulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Tran. Information Theory*, vol. 43, no. 3, pp. 1067–1073, 1997.
- [17] E. Leonardi, M. Melia, F. Neri, and M. Marson, "Bounds on average delays and queue size averages and variances in input-queued cell-based switches," *IEEE Conf. Computer Communications (INFOCOM)*, 2001.
- [18] E. Jonckheere, R. Banirazi, and R. Chatterjea, "Heat calculus approach to queue stability versus Ricci curvature of sensor networks," USC, Tech. Report, 2012. Available: <http://eudoxus2.usc.edu>.
- [19] R. Banirazi, E. Jonckheere, and B. Krishnamachari, "Heat diffusion policy: Pareto optimal dynamic routing for time-varying wireless networks," USC, Tech. Rep., 2012. Available: <http://eudoxus2.usc.edu>.