EXTENDED ABSTRACT

Making Distributed Rate Control using Lyapunov Drifts a Reality in Wireless Networks

Avinash Sridharan, Scott Moeller and Bhaskar Krishnamachari
Ming Hsieh Dept. of Electrical Engineering
University of Southern California, Los Angeles, CA 90089, USA
{asridhar, smoeller, bkrishna}@usc.edu

I. INTRODUCTION

In recent years the literature on wireless networks has been enriched by several theoretical results that have developed new mathematical frameworks for design of optimal cross-layer protocols [1]. A particularly appealing stochastic network optimization approach that yields simple distributed algorithms for dynamic scenarios is based on the use of Lyapunov drifts [2].

The Lyapunov drift based techniques described in [2] provide for distributed rate control based purely on local observations of neighborhood queues. They guarantee the stability of the system and also provide mechanisms to achieve optimization with respect to given utility functions. While attractive on theoretical grounds, to our knowledge these techniques have yet to be implemented in real wireless networks [1]. One reason it has not been easy to translate the Lyapunov drift methodology from theory to practice is that it has been primarily developed under the assumption of a time-slotted system, implying a TDMA MAC. TDMA-based wireless networks are generally harder to implement due to the challenges associated with time-synchronization, particularly across multiple-hops.

[1] Although we have anecdotal evidence that an implementation of similar theoretically-derived queue back-pressure algorithms is being attempted in an ongoing DARPA-funded project, we are not aware of any prior published implementations of such techniques.

Our primary contribution in this work is to show how rate control algorithms based on the Lyapunov drift framework can be built over asynchronous CSMA-based MAC protocols that are widely available in practice. We model the available bandwidth in CSMA-based networks using the concept of receiver capacity, which corresponds to a linear rate region approximation, and use virtual queues to share available bandwidth resources through an explicit representation.

Our second contribution in this work is the experimental implementation of this distributed queue-based rate control algorithm on a real low-power wireless platform (specifically the Tmote Sky from Moteiv) over a CSMA MAC (the CC2420 communication stack implemented in TinyOS 2.x). We give details on our implementation and present experimental results, which highlight the validity of our analysis, bringing Lyapunov-based rate control algorithms for wireless networks a step closer to reality.

II. Lyapunov Optimization Formulation

Prior work in the area of stochastic network optimization [2], [4], [5] provides a detailed methodology for minimizing the Lyapunov Drift of not only physical queue backlogs in a queueing network, but also virtual queues introduced to constrain other system resources.

In order to translate this methodology to the domain of CSMA-based MAC protocols, we make use of the notion of receiver capacity, introduced in [6]. The basic idea is to associate a constant bandwidth capacity with each receiver in the network which must be shared by all transmitters within interference range of that receiver. In particular, for any node i, the rates allocated to a) all nodes sending data to that node i, b) all transmitting nodes within interference range of i, and c) the transmissions made by that node, must not exceed that node's receiver capacity. This model essentially corresponds to a linear approximation of the capacity region for each receiver. Intuitively, one would expect such a linear rate region approximation to be reasonable for CSMA precisely because it minimizes collisions through carrier sense (yielding similar sum-rates for different levels of contention between a set of users). We have previously validated the appropriateness of this approximation through experiments with real wireless devices [7].

In this section, we introduce receiver-capacity inspired virtual queues to a utility maximizing Lyapunov optimization. Though not necessary in a purely theoretical framework, these virtual queues are intended to provide system stability in a real-world implementation where channel capacities may not be directly tractable.

A. Table of Variables and Parameters

Symbol	Description
$\mu_{ij}(t)$	Attempted transmit rate from node i to node j in timeslot t
$\vec{S}(t)$	The vector of channel state information during time slot t
$U_i(t)$	The queue backlog for node i at time slot t
$Z_i(t)$	Virtual queue backlog for node i's collision domain at time slot t
$X_i(t)$	The attempted transmission volume by node i in time slot t
$R_i(t)$	The admitted exogenous arrivals to node i in time slot t
B_i	The receiver capacity of node i
D_i	The collision domain of node i, includes neighbors and node i
C_i	The set of one-hop children for which i is the forwarding agent

B. Lyapunov Optimization with Receiver Bandwidth Virtual Queues

The receiver capacity model described above yields one throughput constraint for each node within the network. Under a Lyapunov framework, we can equate this constraint to a set of virtual queues. Each node in the network would update this virtual queue using virtual arrivals equal to the rate of all overheard traffic and virtual services at the receiver capacity rate.

We define the per-node virtual queue $Z_i(t)$, per-node backlog queue $U_t(t)$, and Lyapunov Function as follows:

$$Z_i(t+1) \leq \max[Z_i(t) - B_i, 0] + \sum_{j \in D_i} X_j(t) \tag{1}$$

$$U_i(t+1) \leq \max[U_i(t) - X_i(t), 0] + \sum_{j \in C_i} X_j(t) + R_i(t) \tag{2}$$

$$L(\vec{U}(t), \vec{Z}(t)) = \sum U_i^2(t) + \sum Z_i^2(t)$$

As was done in [4], [5], we assume the existence of constants X_j^{max} and R_i^{max} such that $X_j^{max} \geq X_j(t)\ \forall t$ and $R_i^{max} \geq R_i(t)\ \forall t$. We can then define:

$$K_i \equiv B_i^2 + \left[\sum_{j \in D_i} X_j^{max}(t)\right]^2 \tag{3}$$

$$G_i \equiv [X_i^{max}]^2 + \left[\sum_{j \in C_n} X_j^{max} + R_i^{max}\right]^2 \tag{4}$$

Under these definitions and following the work of M. J. Neely et al in [4], [5], we arrive at the following general Lyapunov drift formula:

$$\begin{aligned}\Delta(\vec{U}(t),\vec{Z}(t)) \leq \quad & \textstyle\sum_i K_i + \sum_i G_i - 2\cdot\sum_i \left[Z_i(t)E\{B_i - \sum_{j\in D_i} X_j(t)\ |Z_i(t)\}\right] \\ & -2\cdot\sum_i \left[U_i(t)\cdot E\{X_n it) - \sum_{j\in C_i} X_j(t) - R_i(t)|U_i(t)\}\right]\end{aligned} \tag{5}$$

We subsequently define the reward function $Y(t)$ and subtract it from both sides of (5), yielding:

$$Y(t) = \sum_i g(R_i(t))$$

$$\begin{aligned}\Delta(\cdot) - V_{opt}\cdot E\{Y(t)|\vec{U}(t),\vec{Z}(t)\} \leq \quad & \textstyle\sum_i K_i + \sum_i G_i \\ & -2\cdot\sum_i \left[Z_i(t)E\{B_i - \sum_{j\in D_i} X_j(t)\ |Z_i(t)\}\right] \\ & -2\cdot\sum_i \left[U_i(t)\cdot E\{X_i(t) - \sum_{j\in C_n} X_j(t) - R_i(t)|U_i(t)\}\right] \\ & -V_{opt}\cdot E\{Y(t)|\vec{U}(t),\vec{Z}(t)\}\end{aligned} \tag{6}$$

In order to minimize the RHS bound of (6) in expectation, we ensure we minimize the RHS for every system state $\vec{U}(t)$,$\vec{Z}(t)$. We can neglect constant terms involving K_i and G_i. The remaining terms can be separated into coefficients multiplying $X_i(t)$ and $R_i(t)$. As was the case in prior Lyapunov Drift work, the resulting algorithm can be broken into two pieces: a transmission control decision and an admission control decision.

1) Control Decision : Consider node i with parent node k. The coefficient associated with transmission variable $X_i(t)$ is:

$$-\left[U_i(t) - \sum_{\forall j\ st\ i\in D_j} Z_j(t) - U_k(t)\right] \tag{7}$$

Therefore, if transmission rates $X_i(t)$ and $X_j(t)$ are independent $\forall i,j$, then in order to minimize the RHS of (6) we maximize $X_i(t)\ \forall i$ such that (7) is negative.

2) Admission Decision: Consider node i. The coefficient associated with admission variable $R_i(t)$ is:

$$-\left[\frac{V_{opt}}{2}\cdot g(R_i(t)) - U_i\cdot R_i(t)\right] \tag{8}$$

Therefore, in order to minimize the RHS of (6) we maximize $R_i(t)\ \forall i$ such that (8) is negative.

III. Systems Implementation

Our target platform was the Tmote sky class of devices. As mentioned earlier, our receiver capacity model has been validated empirically for these devices [7]. Tiny OS is the open source operating system that runs on the Tmote sky and hence our software architecture was designed specifically to work with the TinyOS infrastructure.

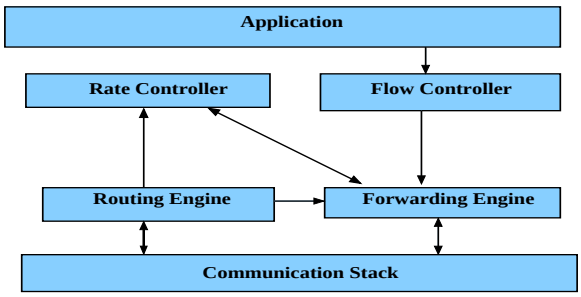

Fig. 1. Software architecture for the distributed rate control using lyapunov drifts on TinyOS 2.0.2

Figure 1 presents the software architecture of our implementation. Because the objective was to perform distributed rate control over a static tree, we use the routing engine provided by the collection tree protocol in TinyOS-2.0.2 [8]. The routing engine helps build a tree rooted at a specific node, acting as the sink. The forwarding engine maintains a FIFO queue which receives packets from the node's children, communicating through the communication stack, or from the application via the flow controller. The communication stack consists of the default TinyOS CSMA stack for the CC2420 radios which is the 802.15.4 radio present on the Tmote sky platforms.

The rate controller and flow controller blocks implement the control and admission decisions respectively, dictated by equations 7 and 8. Though our analysis presented in section II assumes that all events occur on slotted time boundaries, this is not the case for a real CSMA based MAC. Here we implement a timer that fires every T seconds, then carries out the control decision to be active for the duration of the next timer period. Although each node is performing control decisions in slots of T second duration, there is no synchronization between nodes. Additionally, true Lyapunov Optimization would require transmission rates to be allocated such that the sum of all terms described by equation 7 would be minimized. Under a CSMA approach, node transmission rates $X_i(t)$ are not independent from one another. Therefore, minimization would require exact knowledge of the system capacity region and coordination among the nodes. This would break the decentralized implementation for our Tmote sky deployment. Our implementation therefore allocates a maximum value (B_i^{max}) to each $X_t(t)$ for which equation 7 is negative. We show in section IV that the newly introduced virtual queues bring the system to a stable operating throughput, within the capacity region, even while per-timeslot transmission admission rates to the CSMA MAC may exceed the capacity region.

In addition to the control decision, the rate controller block updates the virtual queue using equation 1. To do this it needs to know its receiver capacity, its current transmission rate and the transmission rate of

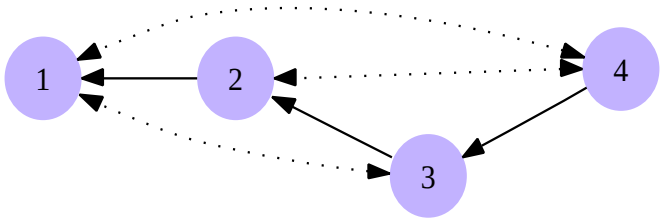

Fig. 2. The experimental 4 node fully connected topology.

all its neighbors who are senders in its broadcast domain. It establishes its receiver capacity by setting it to the saturation throughput corresponding to the number of neighbors within its broadcast domain [7]. The rate controller block estimates the node's current transmission rate by maintaining an exponential weighted moving average of the number of packets transmitted by the forwarding engine, slot by slot.

Note that in order to carry out the control decision of equation 7, knowledge is needed of the current transfer rates ($X_i(t)$) and virtual queue sizes ($Z_i(t)$) of all neighbors. Therefore, once every timeslot, each node stores their transmission rate and virtual queue size to the next transmitted MAC header and broadcast the data packet instead of unicasting it to the parent. This allows all neighbor nodes to overhear this information and retrieve it from the headers.

The flow controller block makes an admission decision on every packet that it receives from the application layer. For simplicity of implementation we assume linear utility functions, the admission decision (equation 8) then reduces to the following: if $\frac{V_{opt}}{2} * Utility \leq U_i(t)$ then admit the application-layer send.

IV. Evaluation

For this work, to verify the performance of our implementation, we present the results for the simple 4 node fully connected topology shown in figure 2. The dotted lines represent the interference links and the solid lines represent a parent child relationship. We hope to solve the following optimization problem for the 4 node fully connected topology:

$$\begin{aligned} \max : w_2 \times r_2 + w_3 \times r_3 + w_4 \times r_4 \\ s.t\ r_2 + 2r_3 + 3r_4 \leq B \ \in \ \Lambda \end{aligned} \tag{9}$$

The constraints for the 4 node topology can be derived using the receiver capacity model presented in [6]. The receiver capacity B in the above equation is identical for nodes 1, 2, 3 and 4 since all

Utilities	$\mathbf{V_{opt}}$	Node 2 (Pkts/sec)	Node 3 (Pkts/sec)	Node 4(Pkts/sec)
$w_2 = 2, w_3 = 3, w_4 = 1$	40	32.63	3.42	0.074
$w_2 = 1, w_3 = 1, w_4 = 5$	20	0.1263	0.1	12.044
$w_2 = 1, w_3 = 5, w_4 = 5$	20	0.98	16.51	3.308

TABLE I

GOODPUTS ACHIEVED BY NODES 2, 3 AND 4 FOR DIFFERENT COMBINATION OF WEIGHTS ASSOCIATED WITH FLOWS ORIGINATING FROM EACH OF THE NODES.

nodes are fully connected and overhear the same number of senders (3 in this case). The value of B can be obtained from the saturation throughput curve presented in [7] corresponding to the value of 3 senders ($\sim 90 pkts/sec$).For our experiments we vary the weights w_2, w_3 and w_4 and observe the rate allocation achieved by our Lyapunov drift implementation. Since figure 2 is a linear topology with a linear utility function, different combination of the weights w_2, w_3 and w_4 should lead to different nodes being activated.

Table I presents the goodput achieved by nodes 2, 3 and 4 for different combinations of these weights. In row 1 of table I when $w_2 = 2$ and $w_3 = 3$, the optimal solution allocates maximal flow to node 2 and zero flow to nodes 3 and 4. This solution corresponds closely with our labortory results. While enlarging V_{opt} was seen to bring labortory simulations closer to optimal, the Tmote sky device only had room for a queue of sixty 33-Byte packets. This restricts V_{opt} values to be less than or equal to $\frac{60*2}{U_{max}}$ where U_{max} is the maximum utility assigned to any node in the topology. Similar congruence can be seen between the optimal solution and laboratory results for the second and third scenarios shown in table I.

V. ONGOING AND FUTURE WORK

The attempt to translate theoretical ideas to an experimental setting opens up many exciting directions. We describe below some of our ongoing and near-term efforts that we plan to provide more details on in the full version of this paper.

In this extended abstract, we have presented experimental results from a small four-node network with a fully connected topology. We are currently working to implement the algorithm on a larger topology (20-40 nodes) on the USC Tutornet wireless network testbed [3]. If this abstract is accepted, we plan to provide experimental results from such large topologies in the full paper.

On the theoretical front, we plan to provide analytical bounds for the average queue size and the optimal solution that can achieved for a given V_{opt}. Also, one of the key gaps between the analysis we

have presented and our current implementation is that the analysis is done with the assumption that the buffer sizes are infinite, in reality resource constraints on the wireless devices we work with limit queue sizes to 50 at most. We believe that the effects of finite buffer sizes on the optimal solutions that can be achieved by our implementation would be more severe in larger topologies with richer interference sets, and thus finite-buffer effects need to be investigated more thoroughly.

Finally, we plan to compare the performance with and without the use of virtual queues, both analytically and experimentally. This would help us better understand the benefits obtained by having an explicit representation of the available bandwidth using the CSMA receiver capacity model.

VI. Acknowledgement

We would like to acknowledge Prof. Michael J. Neely, who took the time and effort to introduce the authors to the technique of using Lyapunov drifts for the purposes of stochastic network optimization.

References

[1] M. Chiang, S.H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE, vol. 95, no. 1, pp. 255-312*, January 2007.

[2] L. Georgiadis, M.J. Neely, and L. Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking, Vol. 1, no. 1, pp. 1-144*, 2006.

[3] Embedded Networks Laboratory. *http://testbed.usc.edu*, 2007.

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

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

[6] A. Sridharan and B. Krishnamachari. Maximizing network utilization with max-min fairness in wireless sensor networks. *5th Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2007.

[7] A. Sridharan and B. Krishnamachari. Maximizing network utilization with max-min fairness in wireless sensor networks. *special issue of ACM/Kluwer Wireless Networks, Available as ANRG Working Paper 07-10-02 from http://ceng.usc.edu/ bkrishna*, 2008.

[8] TinyOS working group. *http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html*, 2007.