A Counterexample in Congestion Control of Wireless Networks

Vivek Raghunathan [a,⋆] P. R. Kumar [b,⋆]

[a] *Department of Electrical and Computer Engineering, and Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, 1308 W. Main St, Urbana IL 61801, USA.*

[b] *Department of Electrical and Computer Engineering, and Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, 1308 W. Main St, Urbana IL 61801, USA.*

Abstract

This paper studies the interaction between TCP congestion control and wireless interference. One of the triumphs of wireline network research of the last decade has been the casting of the Internet congestion control problem within an optimization framework based on utility functions. Such an approach has provided a sound theoretical understanding of the underlying stability and fairness issues, as well as a post-facto justification of the scalability and stability of TCP-like additive-increase multiplicative-decrease (AIMD) algorithms. This paper provides counterexamples showing that the same result cannot be extended to wireless networks, at least not in a straightforward manner.

The fundamental difference is that wireless networks are of a broadcast nature. There is no strict notion of a "link", since transmissions from nearby nodes interfere with each other. We consider a fairly general model of interference in wireless networks, and present a counterexample of a wireless network in which the congestion control mechanism has an unstable equilibrium point at the desired fair solution. ns-2 simulations of this counterexample manifest an oscillatory throughput behavior that is orders of magnitude worse than the corresponding wired networks. Surprisingly, this oscillatory throughput behavior appears to be fairly typical of simulations in wireless networks, with almost all randomly chosen network simulation examples manifesting it. This loss of stability leads us to suggest that perhaps TCP should be modified for use in wireless networks, and that a cross-layer re-design of wireless TCP and MAC is needed to explicitly account for the effects of the wireless nature of interference.

Key words: Wireless multi-hop networks; wireless interference; TCP congestion control; fairness; stability; cross layer design; IEEE 802.11; resource allocation

1 INTRODUCTION

The communication protocols used on the Internet were originally designed in an experimental and heuristic manner. The exponential growth of the Internet has placed great stress on these protocols and raised the specter of a congestion collapse [1]. Consequently, there has been great interest in designing more scalable and stable Internet transport protocols using various methods, including mathematical traffic modeling techniques.

In a seminal paper, Kelly et al [2] proposed a fundamentally different approach that ties congestion control to resource allocation using convex optimization theory. Given utility functions that capture users' value for bandwidth, they suggested that congestion control be viewed as a distributed mechanism to solve the resource allocation optimization problem. In other words, the goal of congestion control is to allocate rates to users so as to maximize the sum of user utilities subject to link capacity constraints. It is possible to decompose this resource allocation problem in order to obtain a distributed solution and design globally asymptotically stable (primal and dual) algorithms that converge to the weighted proportional fair allocation that is the solution of the optimization problem. These algorithms are amenable to distributed implementation on the Internet: in the primal algorithm, users adapt their rates based on the sum of link prices along their paths. This approach provides a theoretical framework to model the congestion behavior of the Internet: in this framework, TCP congestion control can be viewed as a user rate-adaptation strategy and RED/AQM can be viewed as a link congestion pricing strategy. Different TCP variants correspond to implicitly optimizing for different utility functions [3].

In this paper, we attempt to understand the interaction between TCP congestion control and wireless interference. To do so, we examine the extension of the resource allocation approach to *wireless multi-hop networks*. The wireless channel is a broadcast medium and transmissions by "nearby" nodes cannot proceed simultaneously. These constraints introduce various forms of interference:

(1) *Self-interference*: This occurs when a flow's transmission on a hop interferes with its transmission on the previous and next hops.

⋆ This material is based upon work partially supported by DARPA/AFOSR under Contract No. F49620-02-1-0325, AFOSR under Contract No. F49620-02-1-0217, USARO under Contract Nos. DAAD19-00-1-0466 and DAAD19-01010-465, and NSF under Contract Nos. NSF ANI 02-21357, CCR-0325716, and NSF CNS 05-19535, and DARPA under Contact Nos. N00014-0-1-1-0576 and F33615-0-1-C-1905. Vivek Raghunathan is also supported by a Motorola Center for Communication Fellowship and a Vodafone Graduate Fellowship.

(2) *Inter-flow interference*: This occurs when flows with no common link or node still interfere with each other.

These interference constraints result in correlations between the instantaneous capacities of wireless "links", and introduce a spatial nature to congestion in wireless networks. In contrast, wired networks can be accurately modeled by a graph model with independent capacity constraints on the links. The absence of link interference and the independent nature of the capacity constraints in wired networks ensure that:

(1) A flow can obtain the needed congestion feedback information from just links along its own path (and no other links).
(2) Conversely, the links along a flow's path are the only links whose congestion is affected by the flow's traffic. No other links are affected by a flow's traffic.

It turns out that these two facts are crucial to the stability of wireline Internet congestion control mechanisms. Such a wired graph model with link capacity constraints is however not an accurate model of wireless networks, where the congestion in a spatial neighborhood interferes with the directed edges along a flow's path. More precisely, we observe that:

(1) As in wired networks, with unmodified TCP, a flow can only obtain congestion feedback from every directed link along its path, and no other.
(2) However, nearby links not directly on the path can interfere with this flow, since congestion in wireless networks is of a spatial nature. Thus, traffic at nearby links affects the congestion at directed links along a flow's path. Conversely, a flow's traffic can affect links which are "near" a flow, but not on the flow's route.

This fundamental difference between wireless and wired networks leads one to suspect that unless the TCP congestion control mechanism is explicitly re-designed to account for such spatial nature of wireless interference, the adverse interactions between flows caused by interference may lead TCP to measure and adapt to congestion incorrectly. This may even cause a desirable equilibrium point of the congestion control mechanism to become unstable.

In this paper, using a simple model of wireless interference, we formalize this insight and provide counterexamples of wireless scenarios where the interactions do indeed lead to loss of stability of the desired fair rate allocation:

(1) We construct wireless scenarios where the fair equilibrium flow allocation is unstable in the sense of Lyapunov. These counterexamples provide insight into why the congestion control mechanisms become unstable. They hold for any link congestion function that is "highly" convex, a class that includes typical measures of congestion like link delay and

packet drops.

(2) We verify the theoretically predicted behavior of the counterexample using differential equation simulations.

(3) A ns-2 simulation study indicates that TCP exhibits oscillatory throughput behavior over long time scales. Interestingly, randomly constructed wireless scenarios also exhibit this oscillatory throughput behavior fairly typically.

Our result is a negative one, and suggests that if stability of such solutions is important, then TCP will need to be specifically modified for use in wireless networks. It shows that a cross-layer TCP+MAC design for wireless multi-hop networks is required that explicitly takes into account the effects of interference.

The rest of this paper is organized as follows. In Section II, we briefly review the wireline congestion control model. In Section III, we introduce the wireless model and formulate the problem. In Section IV, we describe wireless counterexamples. We present simulations indicating the oscillatory throughput behavior of wireless TCP in Section V. Finally, we discuss related work in Section VI, and conclude in Section VII.

2 A BRIEF REVIEW OF THE WIRELINE MODEL

We begin by describing the original wireline model of [2]. Let S be the set of all flows in the network, assuming for simplicity that all users generate unipath flows. (We shall use the terms flow and user interchangeably.) A flow description consists of the set of links used by a flow, i.e., the route used by a flow from its source to its destination. Let A be the flow-route matrix, i.e., $A_{js} = 1$ if flow s passes through link j. We assume that this route matrix is fixed, at least at the time scales at which congestion control is operated. Let $x_s(t)$ be the rate allocated to user s at time t. As in [2], we let w_s be the "willingness-to-pay" for user s. Let $p_j(y)$ be the congestion price at link j as a function of the total load $y_j := \sum_s A_{js}x_s$ on the link. Assume that the $p_j(y)$'s are increasing, differentiable, and $p_j(0) = 0$.

Then, Kelly et al's primal algorithm for solving the network problem is:

$$\frac{d}{dt}x_s(t) = \kappa(w_s - x_s(t)\sum_j A_{js}p_j(y_j(t))). \tag{1}$$

It has the following motivation: users adjust their rates based on a TCP-like algorithm that adapts to congestion feedback received from the network. The interpretation of this algorithm is as follows:

(1) Users increase their rates using an *additive increase* term proportional to their willingness-to-pay w_s.
(2) Users decrease their rates based on *multiplicative decrease* proportional to the congestion feedback received. This is the $-\kappa x_s \sum_j A_{js} p_j(y_j)$ term.

Notice that an individual user's rate adaptation algorithm requires congestion information only from links along the path of the flow (since $A_{js} = 1$ only if link j is on flow s's path). It can thus be obtained by the source of a flow on an end-to-end basis using an implicit feedback mechanism like packet drops, or an explicit feedback mechanism like ECN [4]. Notice also that links need mark their price $p_j(y_j)$ based only on the total traffic y_j passing through them (since $y_j = \sum_s A_{js} x_s$). Thus, there is no need to maintain per-flow state at the intermediate routers, or for routers to exchange traffic information among themselves. Such a simple and distributed implementation framework is practical for the Internet.

It can be shown that this algorithm has a globally asymptotically stable equilibrium point. All trajectories $\{x_s(t) : s = 1, 2, ...\}$ of the system converge to the unique solution of the optimization problem:

$$\max_{\{x_s \geq 0\}} \sum_s w_s \log(x_s) - \sum_l \int_0^{y_l} p_l(r) dr. \tag{2}$$

The above optimization problem is a penalty function formulation of the system-wide resource allocation problem $\max_{\{x_s \geq 0\}} \sum_s U_s(x_s)$ subject to capacity constraints $Ax \leq C$ with each individual user s's utility function $U_s(x_s)$ being given by the log utility function. It turns out that solving this special version of the resource allocation problem with log utility functions provides weighted proportional fairness among users.

3 A MODEL FOR WIRELESS NETWORKS

We now modify this model to include wireless interference constraints. It should be noted that we present a simplification that does not capture complexities such as signal-to-interference plus noise ratios (SINR), fading, or even more fundamental information theoretic models; see Xue, Xie and Kumar [5] for such detailed models. The wireless network is represented by a directed graph $G = (E, V)$. Let C be the adjacency matrix of the graph, i.e., $C_{ij} = 1$ if there is a directed edge (i, j) between node i and node j, and 0 if not. A directed edge (i, j) indicates that node j can directly receive packets from node i. We assume bi-directional connectivity, i.e., $C_{ij} = C_{ji}$ and set $C_{ii} := 1$. (The directedness of the edges play a role in the interference graph, as we shall see

below.)

As in the wired model, we define the route flow matrix A in terms of the directed edges through which a flow passes. For example, the route flow matrix A_0 for Figure 1 is:

	e_1	e_2	e_3	e_4
flow 1	1	1	0	0
flow 2	0	1	1	0
flow 3	0	0	1	1

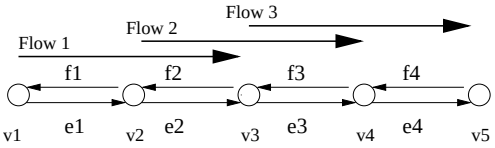

Fig. 1. Flow pattern with route-flow matrix A_0.

Kelly et al's model places independent capacity constraints on the links, as is common in network flow models. This is reasonable in wired networks since links do not interfere with each other, and transmissions on two links can proceed simultaneously. In wireless networks, transmissions on two directed links may or may not proceed simultaneously depending on the spatial locations of the transmitters and receivers. We model wireless interference effects as follows:

(1) The model of the physical constraint imposed by the wireless channel is that certain pairs of directed edges cannot transmit simultaneously. The pairs of directed edges that are disallowed from concurrent transmissions is decided by the particular wireless MAC protocol in operation. The traditional graph-theoretic way to capture such constraints is the notion of an interference graph, also called a conflict graph in the literature. The interference graph $I(G)$ is constructed from the original graph G as follows:
   (a) For every directed edge e in the graph G, place a vertex v_e in $I(G)$.
   (b) Two vertices v_{e_i} and v_{e_j} in $I(G)$ have a directed edge between them if a wireless transmission on directed edge e_j will destructively collide with an ongoing wireless transmission on directed edge e_i in G.

   In this model, the adjacency matrix $T(I(G))$ of the interference graph, which we shall refer to as the interference matrix, is an $|E| \times |E|$ matrix that captures the interference constraints. Different wireless MAC protocols may lead to different interference models, and consequently different

interference matrices T:

(a) *IEEE 802.11 physical carrier sense*: IEEE 802.11 physical carrier sense prevents a node from transmitting if the received energy from any other transmission in its spatial neighborhood is greater than a certain threshold, called the carrier-sensing threshold. The typical value of this threshold results in a IEEE 802.11 carrier sensing range of two hops. In other words, $T_{e_i e_j} = 1$, if the respective transmitters at the head of directed edges e_i and e_j are withing two hops of each other. For example, for the graph in Figure 1, the interference matrix T with the physical carrier sense model is:

	e_1	e_2	e_3	e_4	f_1	f_2	f_3	f_4
e_1	1	1	1	0	1	1	0	0
e_2	1	1	1	1	1	1	1	0
e_3	1	1	1	1	1	1	1	1
e_4	0	1	1	1	1	1	1	1
f_1	1	1	1	1	1	1	1	0
f_2	1	1	1	1	1	1	1	1
f_3	0	1	1	1	1	1	1	1
f_4	0	0	1	1	0	1	1	1

(b) *IEEE 802.11 virtual carrier sense*: IEEE 802.11 virtual carrier sensing uses a four-way handshake to acquire the channel and solve the hidden and exposed terminal problems: the sender uses an RTS (Request-to-Send) frame to silence all nodes in its neighborhood. If the intended receiver receives the RTS, it responds with a CTS (Clear-to-Send) silencing all nodes in the receiver's neighborhood. On receiving the CTS frame, the sender proceeds with the payload DATA frame, and the receiver replies with an ACK to complete the handshake. In this model, $T_{e_i e_j} = 1$ if the transmitter/receiver of one is within reception range of the transmitter/receiver of the other. For example, for the graph in Figure 1, the interference matrix T with

IEEE 802.11 virtual carrier sense is:

	e_1	e_2	e_3	e_4	f_1	f_2	f_3	f_4
e_1	1	1	1	0	1	1	1	0
e_2	1	1	1	1	1	1	1	1
e_3	1	1	1	1	1	1	1	1
e_4	0	1	1	1	0	1	1	1
f_1	1	1	1	0	1	1	1	0
f_2	1	1	1	1	1	1	1	1
f_3	1	1	1	1	1	1	1	1
f_4	0	1	1	1	0	1	1	1

(c) *Node exclusive interference model*: The node exclusive interference model is a simple idealized model that is often used in the wireless scheduling literature. In this model, the only interference constraint placed on transmissions is that a node cannot transmit or receive simultaneously. Thus, $T_{e_i e_j} = 1$ if the directed edges e_i and e_j have the same transmitter, or the same receiver, or the transmitter of one is the receiver of the other. For example, for the graph in Figure 1, the interference matrix T with the node exclusive interference model is:

	e_1	e_2	e_3	e_4	f_1	f_2	f_3	f_4
e_1	1	1	0	0	1	1	0	0
e_2	1	1	1	0	1	1	1	0
e_3	0	1	1	1	0	1	1	1
e_4	0	0	1	1	0	0	1	1
f_1	1	1	0	0	1	1	0	0
f_2	1	1	1	0	1	1	1	0
f_3	0	1	1	1	0	1	1	1
f_4	0	0	1	1	0	0	1	1

We observe in passing that all these interference models result in a symmetric interference matrix T (although this is not necessary).

(2) It is intuitive that a flow that passes through the interference neighbor-

hood of a directed edge causes congestion at that directed edge, and should therefore increase the congestion price reported by it. It is also clear that a flow that does this on multiple occasions should cause even greater congestion at the directed edge and be weighted accordingly. To capture this effect, we let B be the flow-interference matrix, with $B_{js} = n$ if flow s causes interference of "magnitude" n at directed edge j, by passing n times through the interference neighborhood of j. In other words, $B = TA$. We note in passing that, in a wired network, transmissions on links do not interfere with each other and thus (with some abuse of notation), $T = I$ and $B = A$.

Analogous to the wireline case, this leads to the following model for the wireless congestion control algorithm:

$$\frac{d}{dt}x_s(t) = \kappa(w_s - x_s(t)\sum_j A_{js}p_j(y_j(t))). \tag{3}$$

where $y_j := \sum_s B_{js}x_s$.

We note that the big difference lies in the definition of y_j that involves the flow-interference matrix $B = TA$ rather than A.

The interpretation of the algorithm (3) is as follows:

(1) As in TCP, sources adapt their rates using an additive increase and multiplicative decrease mechanism.
(2) We assume that the implementation of congestion feedback is unchanged. Thus, sources receive congestion feedback only from directed edges along their path from source to destination, i.e., only from directed edges with $A_{js} = 1$, either implicitly by detecting packet drops or explicitly by using ECN marks.
(3) We assume that directed edges mark their prices based on the total interference they experience. Thus, the price $p_j(y_j)$ marked by a directed edge is based on $y_j = \sum_s B_{js}x_s$. One could argue, though not too strenuously, that this is a realistic assumption in wireless networks because the congestion pricing/packet dropping mechanisms typically used implicitly enforce this assumption:
    (a) If packet drops are used for feedback, then increased interference in the neighborhood of a directed edge increases the contention for the wireless channel and buffer occupancy at the head of the edge, and, consequently, the number of packet drops due to buffer overflow.
    (b) If congestion marking using an active queue management scheme like RED/REM is used, then the presence of increased interference (y_j) in the neighborhood of a directed edge increases the average RED/REM queue size at the head of the edge. Since the RED/REM price p_j is

an increasing function of the average queue size, it also increases.

4 COUNTEREXAMPLES OF INSTABILITY

We begin by assuming for simplicity that IEEE 802.11 physical carrier sense is used to model interference.

4.1 IEEE 802.11 physical carrier sense interference model

Suppose that all directed edges in the network use the same marking/pricing strategy and drop the subscript j on the p_j. This leads to the following flow rate adjustment dynamics:

$$\frac{d}{dt}x_s(t) = \kappa(w_s - x_s(t)\sum_j A_{js}p(y_j(t))), \tag{4}$$

where $y_j := \sum_s B_{js}x_s$ and $B := TA$.

We proceed to analyze the stability of this algorithm in the absence of delays. Any equilibrium point of this set of differential equations must be a solution of the set of non-linear equations:

$$w_s = x_s \sum_j A_{js}p(\sum_{s'} B_{js'}x_{s'}) \quad \text{for } s = 1,\ ...,\ S. \tag{5}$$

First, we observe that in general, this system of equations has multiple solutions and thus, global asymptotic convergence to an equilibrium point is simply not possible for wireless congestion control.

Example 1

Let the form of $p(y)$ be like the delay in an M/M/1 queue with a server of rate 2. In other words, $p(y) = \frac{1}{1-\frac{y}{2}} - 1$ (we make $p(0) = 0$ to avoid multiplicative decrease at zero network load). Consider an eight node ring G such that every node interferes with two nodes to its right and two nodes to its left, as shown in Figure 2. The flow pattern A consists of eight five-hop flows, also shown in Figure 2. Let $w_s = \frac{1.999*p(1.999)}{5} = 799.2002$ for all s.

Then, it can be verified that all the following are solutions to the set of equations (5):

1. $x = [0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799]$.

2. $x = [0.0874\ 0.0491\ 0.0889\ 0.0924\ 0.0874\ 0.0491\ 0.0889\ 0.0924]$.

3. $x = [0.0924\ 0.0874\ 0.0491\ 0.0889\ 0.0924\ 0.0874\ 0.0491\ 0.0889]$.

4. $x = [0.0889\ 0.0924\ 0.0874\ 0.0491\ 0.0889\ 0.0924\ 0.0874\ 0.0491]$.

5. $x = [0.0491\ 0.0889\ 0.0924\ 0.0874\ 0.0491\ 0.0889\ 0.0924\ 0.0874]$.

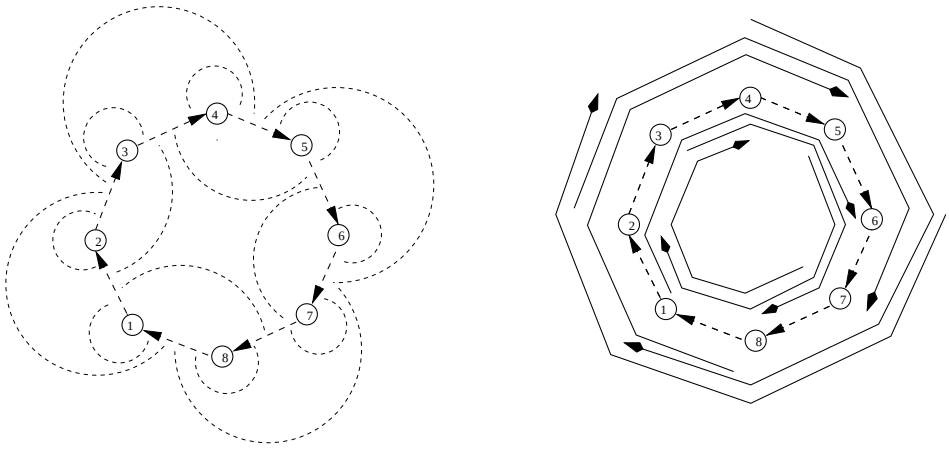

Fig. 2. Wireless network and flow pattern for example 1. In the figure on the left, the dashed lines indicate interference from the two-hop neighborhood using the IEEE 802.11 physical carrier sense model. In the figure on the right, the solid lines indicate the flows.

Since there are five equilibrium solutions, global convergence to a particular solution cannot take place. □

In the example above and the ones described later, our choice of graph G, flow pattern A and willingness-to-pay values W ensure that all users (source-destination pairs) are equivalent to each other. In all such symmetric examples, we can distinguish between two types of equilibria:

(1) The "fair" (or "desirable") equilibrium point is the one where each (equivalent) user s receives identical rate x_s. In the example above, this is $x = [0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799\ 0.0799]$.
(2) The other equilibrium points where different users s receive different rates x_s are "unfair".

In such a symmetric situation, it is of course desired that the system converge to the "fair" equilibrium. While we have shown above that global asymptotic convergence to the "fair" equilibrium point does not take place, it may be possible that the "fair" equilibrium point of (4) is locally asymptotically stable. If this is indeed the case, then it seems plausible that mechanisms like

TCP slow-start may bring the system close to this equilibrium, and the local stability of the "fair" equilibrium point will then guarantee convergence to the fair rate allocation. To examine this, we carry out a local stability analysis of the equilibrium points of (4).

Let $\hat{x}_s,\ 1 \leq s \leq S$ be a fixed point of the system of equations (4). Let us represent a flow as a perturbation of a fixed point by introducing $\{g_s(t) : 1 \leq s \leq S\}$, implicitly defined by $x_s(t) = \hat{x}_s + \sqrt{\hat{x}_s} g_s(t)$.

Defining $G(t) := (g_1(t), ..., g_S(t))$ and denoting the first derivative of the price function $p(y)$ by $p'(y)$, it can be shown that the linearized model of the system is:

$$\dot{G} = LG, \tag{6}$$

where

$$-L := WX^{-1} + X^{\frac{1}{2}} A^T Q B X^{\frac{1}{2}},$$

$$W := diag(w_1,\ ...,\ w_S)\ \ ,\ \ X := diag(\hat{x}_1,\ ...,\ \hat{x}_S),\ \text{and}$$

$$Q := diag(p_1'(\hat{y}_1),\ ...,\ p_N'(\hat{y}_N),$$

with

$$\hat{y}_j := \sum_s B_{js} \hat{x}_s.$$

From Lyapunov's direct method, we know that the system of differential equations is not locally stable in the sense of Lyapunov if the linearized system of differential equations is not locally stable, i.e., the matrix $L = -(WX^{-1} + X^{\frac{1}{2}} A^T Q T A X^{\frac{1}{2}})$ has a positive eigenvalue.

We first prove that there exists a price function $p(y)$ for which the system of differential equations does indeed manifest local instabilities.

Procedure for construction of a scenario with unstable equilibrium points

The key observation is that the matrix $-L$ only depends on $p(y)$ and $p'(y)$ at the equilibrium point. Consider a graph G with N nodes and non-singular interference matrix T, a flow pattern A, and a diagonal matrix Q such that QT has a negative eigenvalue. (It is possible to do this in a number of ways, e.g., choose $Q = I$, and T a 0-1 matrix with a negative eigenvalue. The matrix

Q has no physical significance and is merely chosen to ensure that QT has a negative eigenvalue.) Let route-flow matrix $A = I$ (i.e., N one-hop flows) and $X = I$, so that $X^{\frac{1}{2}}A^TQTAX^{\frac{1}{2}}$ has a negative eigenvalue.

Now consider any candidate price function that satisfies the constraint imposed by the matrix price derivatives, i.e., at N different points y_i, $p'(y_i) = Q(i)$. The values of $p(y_0)$ and $p'(y_0)$ at any point y_0 can be chosen independently of each other and thus, we can choose the value of the price $p(y_i)$ at these N points arbitrarily. Since we are interested in a negative eigenvalue for $-L = WX^{-1} + X^{\frac{1}{2}}A^TP'TAX^{\frac{1}{2}}$, and the second term has a negative eigenvalue, $-L$ will also have a negative eigenvalue, provided the first term is small enough. WX^{-1} is a diagonal matrix with positive entries that depend only on A and $p(y)$. Hence, we can choose $p(y_i)$ arbitrarily small so that $-L$ has a negative eigenvalue. □

We now return to the example in Figure 2 and show that the "fair" equilibrium point is indeed unstable in the sense of Lyapunov. This counterexample also gives intuition into why path-based feedback fails in wireless environments. We obtain the required instability by using cyclic graph structures and cyclic flow patterns.

Example 1 (contd.)

The node positions in the wireless network are such that the corresponding graph is a cycle graph C_N of size $N = 8$ with every node able to receive from a node to its right and a node to its left. With IEEE 802.11 physical carrier sensing, the two-hop interference matrix of the graph can be represented by the circulant matrix T_N with first row [1 1 1 0 0 0 1 1]. (We note that a circulant matrix is a square matrix whose rows can be obtained from the first row by cyclic shifts). The eigenvalues of a circulant matrix are the Discrete Fourier Transform (DFT) of the first row [6] and thus, the eigenvalues of the interference matrix T_N of such a cycle graph of size N are:

$$\lambda_k = 1 + 2cos(2\pi\frac{k}{N}) + 2cos(4\pi\frac{k}{N}), \quad k = 0, 1, ..., N-1. \tag{7}$$

The minimum eigenvalue of T_8 is $\lambda_{min} = -1$. For $T = T_8$, T^{-1} is a circulant matrix with first row [−0.6 0.4 0.4 −0.6 0.4 −0.6 0.4 0.4]. Also, T^2 and T^3 are circulant matrices with first row [5 4 3 2 2 2 3 4] and [19 18 16 13 12 13 16 18] respectively, and T^3 has a negative eigenvalue at -1.

Fix the operating point of the network as $\hat{y} = ke$, where e is the row vector of all ones. Note that the flow pattern consists of eight flows, each passing through five successive directed edges, i.e., $A = T$. Choose the willingness-to-pay values as $w_s = \frac{kp(k)}{5}$ for $s = 1,\ 2, ...,\ 8$.

We can compute the rate vector at equilibrium $\hat{x} = B^{-1}\hat{y} = A^{-1}T^{-1}\hat{y} = \frac{k}{25}e$. Also, at the fair equilibrium, we have:

$$\frac{w_s}{\hat{x}_s} = 5p(k) \text{ for s} = 1, \ \ldots \ , \ 8$$

Thus, $-L = 5p(k)I + \frac{k}{25}p'(k)T^3$. T^3 has a negative eigenvalue at -1. Finally, recall that for an M/M/1 type delay function, $p(y) = \frac{1}{1-y} - 1$, if the operating point of the system is chosen at $1 > y_0 > \frac{125}{126}$, then $125p(y_0) < \frac{125}{1-y_0} < y_0p'(y_0)$. Thus, the eigenvalues of L are the eigenvalues of T^3 shifted by an amount less than 1 and thus, $-L$ has a negative eigenvalue. Hence, the "fair" equilibrium point is unstable. $\square$

The local instability of the "fair" equilibrium point also applies to other realistic node pricing functions like that used by AQM packet dropping mechanisms.

Example 2

Suppose $p(y)$ grows like the penalty function used in [2] to solve the resource allocation problem, i.e.,

$$p(y) = \frac{(y - C + \epsilon)_+}{\epsilon^2}, \tag{8}$$

where $\epsilon > 0$ is a small number.

The same graph G, a flow pattern A and a willingness-to-pay matrix W used in Example 1 will yield a system of differential equations that has a locally unstable equilibrium for this case too. The proof of this fact is identical to the earlier proof in Example 1; observe that the only property of the price function $p(y)$ used in Example 1 was the fact that there exists a y_0 such that $125p(y_0) < y_0p'(y_0)$. This ensures that the first term in the expression for the linearized matrix $-L$ is too small to keep the equilibrium point stable.

To complete the proof, given any ϵ, we choose the operating point of the network as $k = \alpha(C - \epsilon)$, where $1 < \alpha < \frac{125}{124}$, by appropriately choosing the willingness-to-pay values. At this operating point, it can be shown that there exists a y_0 such that $125p(y_0) < p'(y_0)y_0$. $\square$

4.2 Extensions to other interference models

The same technique can be used to construct counterexamples for the other interference models described in Section 3. For example, suppose IEEE 802.11 virtual carrier sensing is used to model interference. Consider the same example as in Figure 2 with eight nodes in a cyclic ring, and eight five-hop flows, and $w_s = 799.2002$ for all s. Under IEEE 802.11 virtual carrier sensing, the interference graph for this example is the same as that of the physical carrier sensing model. Thus, T and A are represented by the same circulant matrix as in IEEE 802.11 physical carrier sensing, with first row [1 1 1 0 0 0 1 1], and counterexamples can be constructed in a manner identical to Examples 1 and 2.

For the node exclusive interference model, we consider the topology and flow pattern shown in Figure 3. The interference matrix is a circulant matrix with first row [1 1 0 0 0 0 0 1]. The flow pattern consists of eight three hop flows, with the flow matrix A the same as that of the interference matrix. Suppose the price function is the delay of a M/M/1 server with rate 2, i.e., $p(y) = \frac{1}{1-\frac{y}{2}} - 1$ and the willingness-to-pay values are $w_s = \frac{1.99 * p(1.99)}{3} = 132.003$ for all s. Then, following the lines of Example 1, one can establish that:

(1) There are multiple solutions to the system of equations 5, with a "fair equilibrium" and multiple "unfair equilibria".
(2) The linearized matrix L about the "fair equilibrium" has a negative eigenvalue. Thus, the "fair equilibrium" is locally unstable in the sense of Lyapunov.

The corresponding counterexample for the price function in Example 2 is similar.

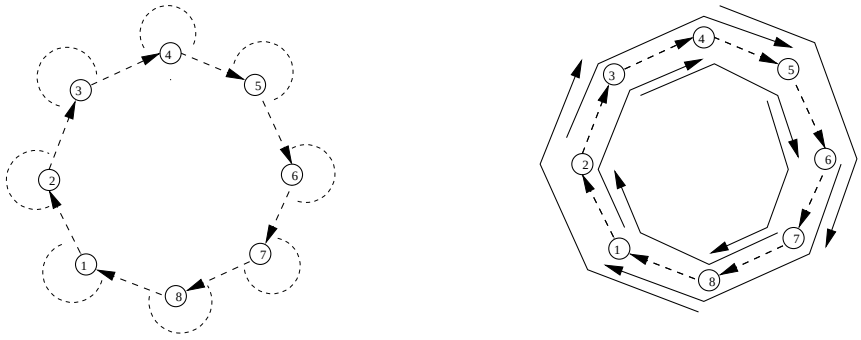

Fig. 3. Wireless network and flow pattern for counterexample for node exclusive interference model. In the figure on the left, the dashed lines indicate the interference between directed edges. In the figure on the right, the solid lines indicate the flows.

4.3 Discussion

A natural question to consider at this point is whether it is possible to choose price functions $p(y)$ that yield the desired stability. Unfortunately, the space of counterexamples is rather rich. The same graph G, flow pattern A and willingness-to-pay matrix W yield a system of differential equations that has a locally unstable equilibrium for any price function $p(y)$ that satisfies one of the following properties (the proofs are straight-forward):

(1) There exists a y_0 such that $125p(y_0) < y_0p'(y_0)$.
(2) There exists a point y_0 such that $p(y_0) = cy_0^n$, and $n > 125$.

In fact, we can show that pricing functions of the form $p(y) = cy^n$ have counterexamples of instabilities for all $n \geq 6$ in a manner similar to Examples 1 and 2. We shall establish this for the node exclusive interference model (proofs for other models are similar and are omitted). One need only choose the interference matrix T so that the negative real eigenvalue has magnitude as large as possible. To do so, we observe that cyclic interference graphs with big interference neighborhoods have the required property. Further, the first term WX^{-1} counteracts the effect of the negative eigenvalue and should be kept as small as possible. This term depends only on the flow pattern A and the actual price function $p(y)$ at the operating point. We use these facts to construct a stronger counterexample where the "fair equilibrium" is unstable for all $p(y) = cy^n$ with $n \geq 6$, as well as the ones considered earlier.

Example 3

Let $N = 12M + 2$ and consider a N node cyclic wireless graph with node positions such that there are $6M - 3$ neighbors each to the right and left respectively. There are $E = N(6M - 3)$ edges in this graph and thus, the interference matrix has dimension $N(6M - 3) \times N(6M - 3)$. In our counterexample construction, we will focus only on the $N \times N$ subsection of this matrix corresponding to the directed edges between nearest neighbors. For example, in Figure 4, the directed edges between the nearest neighbors, and the sub-section of the interference matrix corresponding to these directed edges is shown for $M = 2$. When the node exclusive interference model is used, this component of the interference matrix is a circulant matrix T_0 with first row [1 1 ... 1 0 0 0 0 0 0 0 1 1 .... 1], where the first $6M - 2$ entries are ones, the next seven are zeros, and the last $6M - 3$ entries are ones. (If IEEE 802.11 physical or virtual carrier sensing had been used instead as an interference model, the interference matrix would have had many more non-zero entries.) Since we wish to keep the first term WX^{-1} as small as possible, we choose a flow pattern consisting of N flows, each passing through three successive

nearest neighbor directed edges and one shifted from the previous flow. Thus, the flow matrix A is a $N(6M-3) \times N$ matrix, where the component of A corresponding to the nearest neighbor directed edges is a $N \times N$ circulant matrix A_0 with first row [1 1 0 0 0 ... 1] (one of these flows is shown in Figure 4). The matrix T and A are as follows:

$$T = \left[\begin{array}{c|c} T_0 & T_{01} \\ \hline T_{10} & T_{11} \end{array}\right] \begin{array}{l} \text{nearest neighbor edges} \\ \text{other edges,} \end{array}$$

$$A = \left[\begin{array}{c} A_0 \\ \hline 0 \end{array}\right] \begin{array}{l} \text{nearest neighbor edges} \\ \text{other edges} \end{array}$$

.

We choose the operating point of the network as $x = \frac{k}{3(12M-5)} \cdot e$, where e is the row vector of all ones with appropriate dimension. With this choice of x, $y = TAx = [\ y_0\ y_1\ ]^t$, where y_0 is the component corresponding to the nearest neighbor edges and y_1 is the other component. It can be seen that with these choices of A_0 and T_0, $y_0 = k \cdot e$. Further, we have:

$$P = \left[\begin{array}{c|c} P_0 & 0 \\ \hline 0 & P_1 \end{array}\right] \begin{array}{l} \text{nearest neighbor edges} \\ \text{other edges} \end{array},$$

and

$$Q = \left[\begin{array}{c|c} Q_0 & 0 \\ \hline 0 & Q_1 \end{array}\right] \begin{array}{l} \text{nearest neighbor edges} \\ \text{other edges} \end{array},$$

where $P_0 = k^n I$ and $Q_0 = nk^{n-1} I$ correspond to the nearest neighbor directed edges and P_1 and Q_1 correspond to the other edges. Then, some algebraic manipulation establishes that:

$$-L = WX^{-1} + X^{1/2} A^T Q T A X^{1/2}$$

$$= 3k^n I + \frac{nk^n}{3(12M-5)} A_0^T T_0 A_0.$$

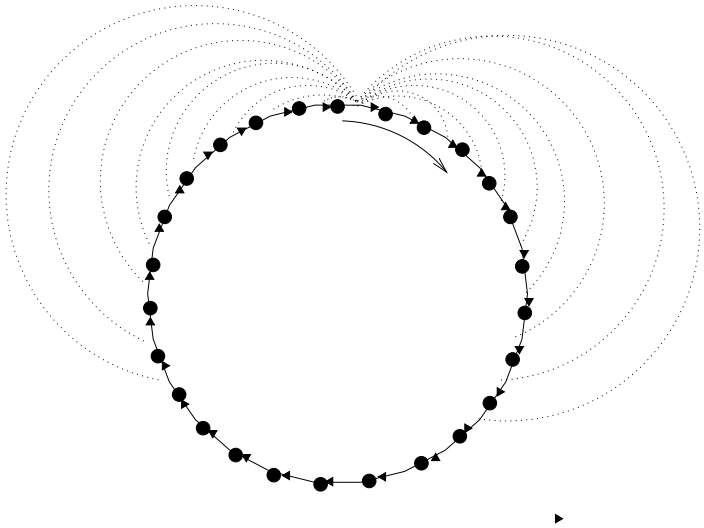

Fig. 4. Wireless network and flow pattern for Example 3 indicating the interference between nearest neighbor directed edges.

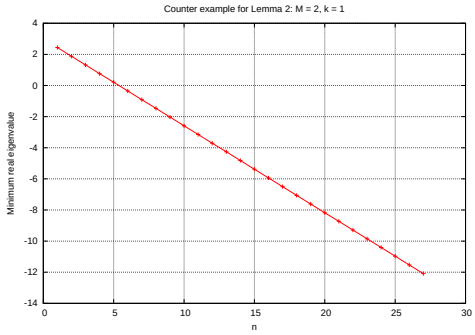

Fig. 5. Minimum real eigenvalue using Example 3 for price function $p(y) = y^n$.

Since the product of circulant matrices is a circulant matrix and there is a closed form solution for the eigenvalues of a circulant matrix, we can easily compute the eigenvalues of $-L$ for various values of M. The minimum real value of $-L$ for $M = 2$ is shown as a function of n is Figure 5. Observe that for all $n \in [6, 27]$, this value is negative and this interference matrix provides the required counterexample. (In fact, it provides a counterexample for $n \geq 5.38$.) As before, this shows that the "fair" equilibrium point is locally unstable. □

These counterexamples lead us to conjecture that a universal counterexample exists for all convex price functions $p(y)$. However, we have as yet been unable to prove this conjecture.

5 SIMULATION RESULTS

The examples presented in the previous section suggest that wireline Internet congestion control mechanisms may suffer from instabilities if used without modification in wireless networks. In order to understand how these instabilities manifest in practice, we carry out:

(1) Matlab simulations of the differential equations.
(2) ns-2 simulations of TCP over IEEE 802.11.

5.1 Matlab simulations of the underlying differential equations

In order to understand the non-linear dynamics of the system of differential equations (4), we simulated the example in Figure 2 using Matlab's ode45 differential equation solver.

With the graph $C = C_8$, the flow pattern $A = A^T = T_8$, link price function corresponding to an M/M/1 delay function, i.e., $p(y) = \frac{1}{1-\frac{y}{2}} - 1$, and willingness-to-pay matrix $\frac{1.999.p(1.999)}{5} I$, the set of equations (5) have multiple solutions, as noted before in Example 1. In Figure 6, we show plots of differential equation simulations of the system with the desired operating point chosen as $y = 1.999e$. We observe that trajectories of the system converge to one of the locally asymptotically stable "unfair" equilibria, and always diverge away from the "fair" equilibrium point.

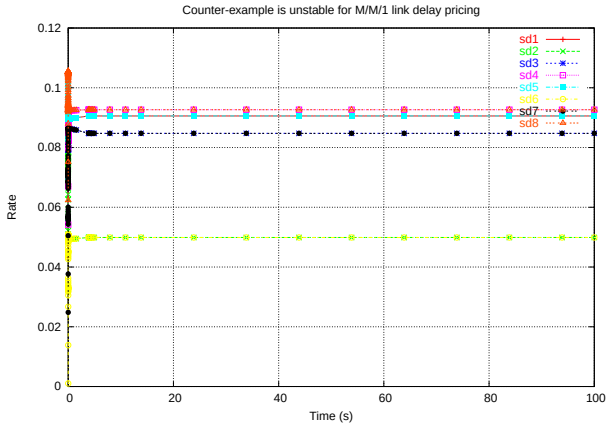

Fig. 6. Trajectories of differential equations for $p(y)$ =M/M/1 delay.

Similar behavior is observed for other price functions like the penalty function $p(y) = \frac{(y-C+\epsilon)_+}{\epsilon^2}$ used to solve the resource allocation problem in [2]. Plots of the differential equation simulation are shown in Figure 7 with this price function. In this simulation, $C = 1$, $\epsilon = 0.1$ and the desired operating point is

$y = k.e$, where $(C - \epsilon) < k = 0.905 < \frac{125}{124}(C - \epsilon)$.

It can be verified that for this choice of parameters, the differential equations have multiple fixed points:

1. $x = [0.0362\ 0.0362\ 0.0362\ 0.0362\ 0.0362\ 0.0362\ 0.0362\ 0.0362]$.

2. $x = [0.0433\ 0.0289\ 0.0289\ 0.0433\ 0.0433\ 0.0289\ 0.0289\ 0.0433]$.

3. $x = [0.0433\ 0.0433\ 0.0289\ 0.0289\ 0.0433\ 0.0433\ 0.0289\ 0.0289]$.

4. $x = [0.0289\ 0.0433\ 0.0433\ 0.0289\ 0.0289\ 0.0433\ 0.0433\ 0.0289]$.

5. $x = [0.0289\ 0.0289\ 0.0433\ 0.0433\ 0.0289\ 0.0289\ 0.0433\ 0.0433]$.

Again, in this case, trajectories of the system converge to one of the "unfair" locally asymptotically stable equilibrium points and diverge away from the (desired) "fair" equilibrium point.

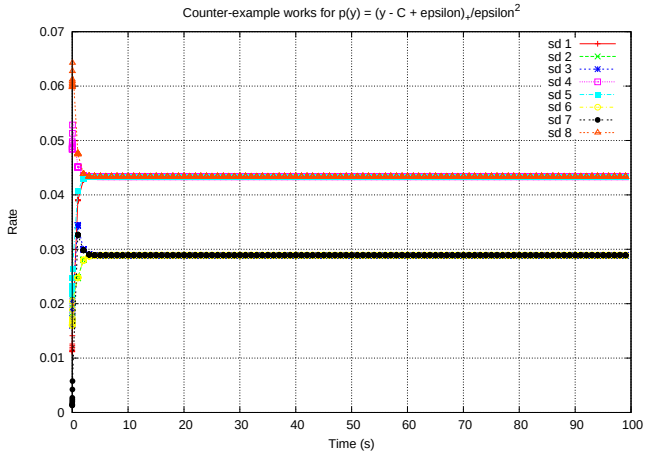

Fig. 7. Trajectories of differential equations for $p(y) = \frac{(y-C+\epsilon)_+}{\epsilon^2}$.

5.2 Oscillatory behavior of the wireless TCP counterexample in practice

The lack of global asymptotic convergence can possibly lead to fragility with respect to real-world issues like network feedback delays or the discontinuous nature of window based congestion control. We need to investigate the behavior of congestion control in a more realistic environment that models the complex window congestion control dynamics of TCP. TCP has two additional mechanisms that may affect stability. First, the window-based congestion control mechanism of TCP is a damping mechanism that may provide some implicit protection against instabilities. Also, it is possible that TCP slow-start may bring trajectories of the system close to one of the "unfair"

equilibrium points, after which local asymptotic stability may be sufficient to make it converge to an unfair allocation.

We study the underlying dynamics of the counterexamples using ns-2 with Monarch wireless extensions to investigate the impact of real-world protocol issues on wireless stability. The simulation parameters for wired and wireless simulations are shown in Table 1. In our simulations, the wired link bandwidth is chosen so that the wired throughput is (roughly) of the same order as the wireless throughput. For each of the simulations, we have run the experiment 10-30 times with different seed values to verify that the observed behavior is not a spurious effect. In our results below, we present observations for a typical run of each experiment. We must mention three additional important details:

(1) TCP has been reported to oscillate over long timescales even in a single bottleneck wireline network [7]. It seems that the behavior is strongly related to the buffer size at the bottleneck link. A useful heuristic is to use buffers that are large enough to absorb oscillations in offered load [8]. In our simulations, we did not observe oscillatory behavior in most of our wired topologies even with small buffers. However, in order to completely eliminate potential oscillatory effects due to small network buffers, we set the network buffer sizes to be larger than the bandwidth-RTT product.
(2) The routing required by the example in Figure 2 does not correspond to shortest-path routes and needs to be configured by hand. For wireless networks, we wrote a manual routing agent that can be configured using oTcl. For wired networks, we used the in-built manual routing agent.
(3) While the differential equations (4) use a logarithmic TCP Vegas utility function, we have observed identical oscillatory behavior for all variants of TCP. Unless otherwise mentioned, we have used TCP Reno for all our simulations.

To understand the underlying TCP dynamics of these examples, we use plots of individual user TCP throughputs over time, averaged over different timescales by smoothing. At the connection RTT timescale, we expect to observe the distinctive sawtooth curve of TCP congestion avoidance. As we increase the timescale of averaging to large multiples of RTT, the sawtooth nature of TCP congestion avoidance should average out to a constant value if TCP is non-oscillatory. This behavior is characteristic of wired networks; for example, in Figure 8, we show the TCP Reno throughput averaged over three timescales for a wireline network scenario with the same topology and connection pattern as the counterexample in Figure 2.

In contrast, in wireless networks, we observe remarkably different behavior. For example, in Figure 9, we show the TCP Reno throughput for the topology and connection pattern in Figure 2, averaged over 1 and 25 seconds respectively. The TCP throughput oscillates over long timescales for all the flows (see Figure

Table 1
Simulation parameters

Parameter	Value
Simulation Time	500 s
Packet Size	210 bytes
Simulation Runs	10 - 30
Wireless	
Wireless MAC	IEEE 802.11
Interface Queue (IFQ)	DropTail
Wired	
Link bandwidth	66Kb
Link delay	10 ms
Link output queue	DropTail

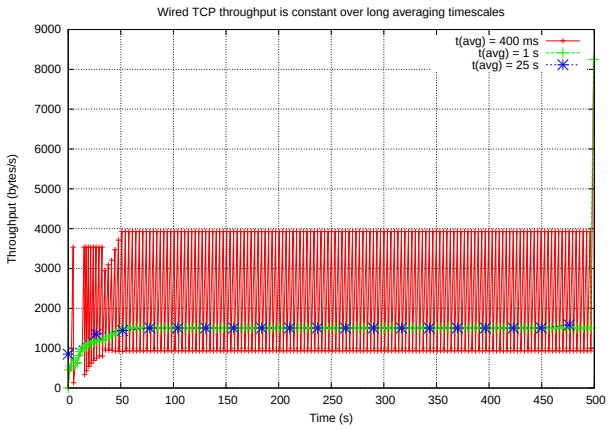

Fig. 8. Wired TCP throughput is stable. TCP throughput at the 400 ms timescale produces the sawtooth oscillations. TCP throughput at the 1 and 25 second time scale shows little or no fluctuations.

10).

This oscillatory behavior is prevalent irrespective of the TCP variant used, as seen in Figure 11 for TCP Vegas. (While TCP Vegas is known to behave badly even in wired scenarios, we did not observe such behavior in our simulations, and wired TCP Vegas throughput was stable when averaged over long timescales.)

We use the normalized variance nv of the throughput time series of a flow, averaged over the 25s time scale, as a measure of its oscillatory behavior. For a data series $\{x_1, x_2..., x_n\}$, the normalized throughput variance nv is defined

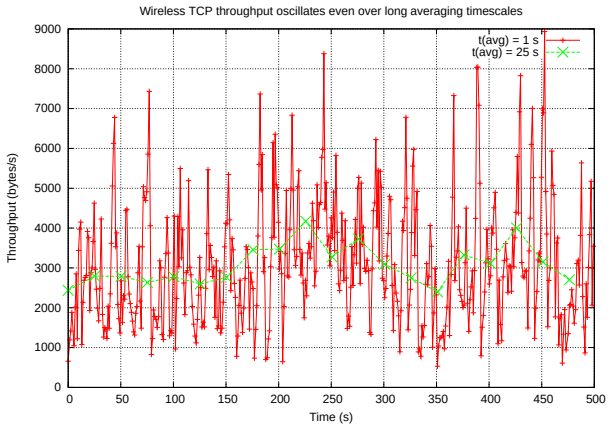

Fig. 9. Wireless TCP throughput is prone to oscillations, even over long time scales.

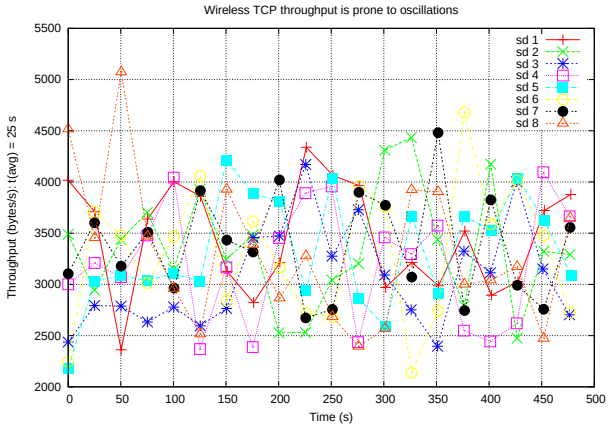

Fig. 10. Wireless TCP throughput oscillates for all flows.

as:

$$\text{nv} = \frac{\sigma_x^2}{E[X^2]},$$

where

$$E[X] = \frac{\sum_{i=1}^{n} x_i}{n},$$

$$\sigma_x^2 = \frac{\sum_{i=1}^{n} (x_i - E[X])^2}{n},$$

and

$$E[X^2] = \frac{\sum_{i=1}^{n} x_i^2}{n}.$$

We note that nv is related to the coefficient of variation cv as:

$$cv = \sqrt{\frac{nv}{1 - nv}}.$$

The normalization ensures that nv is always between zero and one, and allows us to compare oscillatory behavior across flows and scenarios, even if the magnitudes of the individual flow throughputs are not comparable. We computed this normalized variance on a per-flow basis for each run of the experiment. The normalized variance, averaged over all runs, is shown in Table 2 for the counterexample in Figure 2 and is consistently two orders of magnitude higher for wireless compared to the wired case. This confirms that the oscillatory behavior is a result of the negative interaction between wireless interference and TCP in the cyclic structure of the counterexample, and not an artifact of the simulation settings used.

This oscillatory behavior is a consequence of the multiple unfair equilibria in the wireless environment. These unfair equilibria are pretty close to each other, and a small disturbance causes the system to jump between these equilibria. This leads us to believe that the oscillatory behavior of TCP and wireless interference will depend on how many (possibly unfair) equilibrium points there are in the system. In the wired case, there is just one (fair) equilibrium point, and this is not an issue.

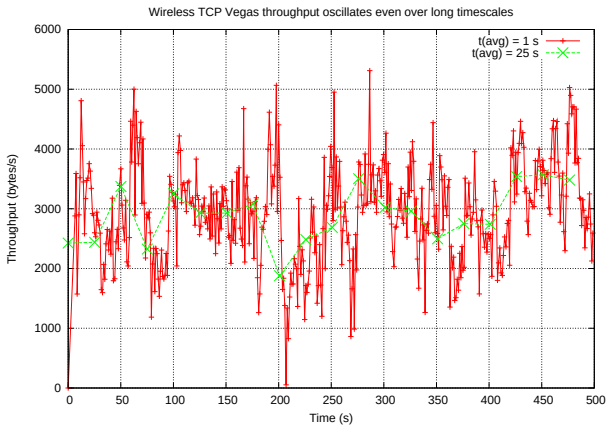

Fig. 11. Wireless Vegas throughput also oscillates.

5.3 How prevalent are the instabilities?

It is interesting to investigate if our counterexamples are merely pathological, or if the TCP congestion control mechanism is indeed prone to wireless instabilities over a wider range of topologies and flow patterns. Preliminary

Table 2
Normalized variance of TCP throughput

Network	TCP	Normalized variance (averaged over all runs and flows)
Wired	Tahoe	1.381×10^{-4}
	Reno	9.530×10^{-5}
	Vegas	3.784×10^{-5}
Wireless	Tahoe	2.358×10^{-2}
	Reno	2.824×10^{-2}
	Vegas	2.821×10^{-2}

investigations indicate that the latter is indeed the case, i.e., instabilities are quite prevalent. We randomly generated 104 wireless network topologies and traffic flow patterns and measured the normalized throughput variance (at the 5 second timescale) on a per-flow basis. We computed the normalized throughput variance for a scenario as the average value of the per-flow normalized throughput variances. We repeated the experiment for the analogous wired topologies and traffic patterns. In Figure 12, we show a plot of the normalized throughput variance over all scenarios for wireless and wired networks. The horizontal lines represent the corresponding values of the normalized throughput variance for the counterexample in Figure 2 (averaged over all runs). It is clear that the oscillatory behavior of the counterexample is fairly typical in wireless networks; 36% of the scenarios are worse than the counterexample; and the oscillatory behaviour of 97% of the scenarios is within an order of magnitude of the counterexample in Figure 2.

On the other hand, TCP is fairly stable for wired networks over a wired range of scenarios; wireless TCP is on average 2614 times worse than wired TCP. These numbers indicate that the oscillatory behavior of wireless TCP is a fairly widespread problem and not merely restricted to a pathological class of counterexamples.

6 RELATED WORK

Congestion control has been studied extensively in wireline networks using an optimization-based approach. Given utility functions that capture users' value for bandwidth, Kelly et al [9] suggested that the objective of congestion control is to allocate rates to users so as to maximize the sum of user utilities. This resource allocation optimization problem can be decomposed and solved in a distributed manner using primal approaches [2] [10] and dual approaches

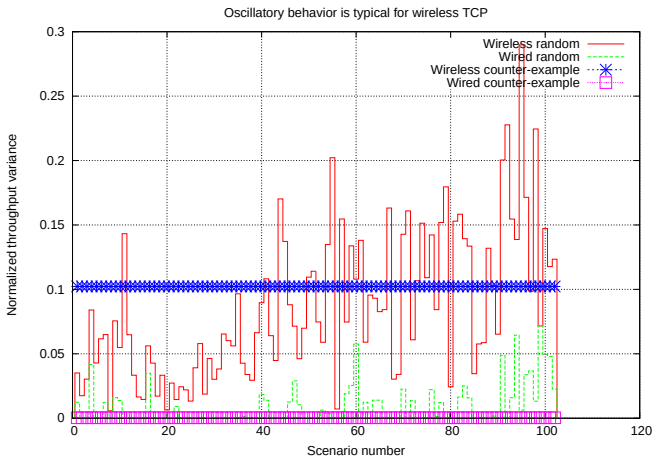

Fig. 12. Wireless oscillatory throughput behavior is typical.

[11] with guaranteed convergence properties. The individual utility functions that each individual TCP variant implicitly optimizes can then be derived using this framework [3]. These approaches can be implemented using delay or one-bit ECN [4] congestion feedback from the network.

In wireless networks, TCP performance suffers for a variety of reasons. For example, TCP cannot distinguish random wireless losses from congestion; the common solution is to split the TCP connection at the base station into two separate connections, one for wireless and one for wired [12], [13]. In multi-hop wireless ad-hoc networks, TCP can interact negatively with the network layer and mistake topology variation due to mobility and the consequent route fluctuations for congestion. This causes unnecessary TCP retransmissions and exponential backoffs and can be prevented by using route failure notification mechanisms like TCP-F [14] and ELFN [15].

When TCP is used over the IEEE 802.11 MAC protocol, wireless contention can cause the MAC retry count to be exceeded and result in packet drops. This results in a bad interaction between TCP and the MAC layer: contention-related losses dominate and cause excessive congestion window build-up with too many in-flight packets competing with each other for the wireless medium [16], resulting in TCP throughput degradation. An extensive measurement study has shown that clamping the TCP sender window to $\lceil \frac{3}{2}n \rceil$ (where n is the number of hops traversed by the flow) results in order of magnitude improvement in delays [17].

The negative interaction of TCP and wireless interference at the MAC layer can also result in severe unfairness [18]. (Interestingly, the 8 node ring topology in our counterexample was considered in [18]; however, they studied the average TCP throughput and thus, apparently did not observe the oscillatory dynamics of wireless TCP.) As our differential equation simulations indicate, this divergence of TCP congestion control from the "fair equilibrium" is a nec-

essary consequence of the local instability of the "fair equilibrium", as proved in Examples 1 and 2. Neighborhood RED [19] modifies the reported congestion price using neighbor interference information to improve fairness.

The wireless interference constraints can also be modeled as a conflict graph [20] [21]. Recently, these interference models have been incorporated into the network utility maximization approach to congestion control. Such cross layer approaches to joint congestion control and scheduling in wireless multi-hop networks attempt to separate the rate control and wireless scheduling problems using dual decomposition [22] [23] [24] [25] . Our work considers a fairly realistic congestion control mechanism that only uses path-based feedback and proves formally that wireless interference can make this mechanism unstable. Our counterexample in Figure 2 shows why Internet congestion control cannot be extended to wireless networks in a straightforward manner without considering the underlying scheduling problem. This indicates that wireless interference must be taken into account in the congestion pricing mechanism and provides a motivation for such cross-layer approaches.

In fact, TCP has been reported to oscillate over long timescales and converge to strange fractal attractors even in a single bottleneck dumbbell topology wired network [7]. It seems that such behavior is strongly related to the bottleneck link's buffer size and can be avoided by using buffers that are large enough to absorb oscillations in offered load due to congestion window adaptation [8]. In order to completely eliminate oscillatory effects due to small buffers, we set the buffer sizes to be much larger than the bandwidth-RTT product in our simulations. The persistence of oscillatory behavior over very long timescales in wireless networks even in this setting confirms our intuition that these oscillations are a manifestation of the negative interaction between TCP and wireless interference.

Finally, it has been suggested that TCP produces pseudo-self similar behavior, with high variability over timescales upto hundreds of RTTs which disappears over longer time scales [26] [27]. We have not yet examined whether the oscillatory phenomenon in wireless networks is of this pseudo self-similar nature, or whether wireless TCP congestion control in fact produces truly self-similar behavior.

7 CONCLUSION

The broadcast nature of wireless networks causes transmissions from neighboring nodes to interfere with each other, and can cause congestion control mechanisms to lose stability vis-a-vis the desired fair equilibrium allocation of rates. We have presented example wireless networks and flow patterns that ex-

hibit such a loss of stability. These counterexamples suggest that TCP cannot be used unmodified in wireless multi-hop networks if stability considerations are important. Our work thus provides a proof of necessity for a cross layer re-design of TCP+MAC for wireless networks, taking interference effects into account.

ACKNOWLEDGEMENTS

We are grateful to Prof. Tom Seidman for helpful discussions in setting up the problem. Xue Liu's feedback helped remove a bug in an earlier version of the paper. The anonymous reviewers provided invaluable feedback in revising the paper.

References

[1] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, 1993.

[2] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," in *Journal of the Operational Research Society*, vol. 49, 1998.

[3] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 525–536, 2003.

[4] K. Ramakrishnan, S. Floyd, and D. Black, "RFC 3168 - the addition of Explicit Congestion Notification (ECN) to IP," 2001.

[5] F. Xue, L. L. Xie, and P. R. Kumar, "The transport capacity of wireless networks over fading channels," *IEEE Trans. on Information Theory*, vol. 51, no. 3, pp. 834–847, Mar 2005.

[6] R. M. Gray, "Toeplitz and circulant matrices: A review," 2002. [Online]. Available: http://www-ee.stanford.edu/ gray/toeplitz.pdf

[7] A. Veres and M. Boda, "The chaotic nature of TCP congestion control." in *Proceedings of IEEE INFOCOM*, 2000, pp. 1715–1723.

[8] A. Gilbert, Y. Joo, and N. McKeown, "Congestion control and periodic behaviour," in *LANMAN Workshop*, 2001.

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

[10] S. Kunniyur and R. Srikant, "A time scale decomposition approach to adaptive ECN marking," in *Proceedings of IEEE INFOCOM*, 2001, pp. 1330–1339.

[11] S. H. Low and D. E. Lapsley, "Optimization flow control — I: basic algorithm and convergence," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[12] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," *ACM Wireless Networks*, vol. 1, no. 4, 1995.

[13] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," *15th International Conference on Distributed Computing Systems*, 1995.

[14] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving TCP performance in ad-hoc wireless networks," in *International Conference on Distributed Computing Systems*, 1997, pp. 472–479.

[15] G. Holland and N. H. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Mobile Computing and Networking*, 1999, pp. 219–230.

[16] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," in *Proceedings of IEEE INFOCOM*, 2003.

[17] V. Kawadia and P. R. Kumar, "Experimental investigations into TCP performance over wireless multihop networks," in *Proceedings of Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, 2005.

[18] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," 1999.

[19] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP fairness in ad hoc wireless networks using neighborhood RED," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking.* New York, NY, USA: ACM Press, 2003, pp. 16–28.

[20] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking.* New York, NY, USA: ACM Press, 2003, pp. 66–80.

[21] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking.* New York, NY, USA: ACM Press, 2000, pp. 87–98.

[22] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proceedings of IEEE CDC*, 2004.

[23] A. Eryilmaz and R. Srikant, "Joint congestion control, routing and MAC for stability and fairness in wireless networks," in *IEEE Journal on Selected Areas in Communication*, 2006.

[24] M. Neely and E. Modiano, "Fairness and optimal stochastic control for heterogeneous networks," in *Proceedings of IEEE INFOCOM*, 2005.

[25] L. Chen, S. Low, and J. Doyle, "Joint congestion control and media access control design for ad hoc wireless networks," in *Proceedings of IEEE INFOCOM*, 2005.

[26] D. R. Figueiredo, B. Liu, V. Misra, and D. Towsley, "On the autocorrelation structure of TCP traffic," *Comput. Networks*, vol. 40, no. 3, pp. 339–361, 2002.

[27] L. Guo, M. Crovella, and I. Matta, "How does TCP generate pseudo-self-similarity?" in *MASCOTS*, 2001, pp. 215–223.