# Stabilized Vegas

Hyojeong Choe and Steven H. Low
CS and EE, Caltech, Pasadena CA 91125
http://netlab.caltech.edu/FAST

IEEE Infocom, San Francisco, CA, April 2003

*Abstract*— **We show that the current TCP Vegas algorithm can become unstable in the presence of network delay and propose a modification that stabilizes it. The stabilized Vegas remains completely source-based and can be implemented without any network support. We suggest an incremental deployment strategy for stabilized Vegas when the network contains a mix of links, some with active queue management and some without.**

## I. INTRODUCTION

TCP Vegas was introduced in 1994 [6] as an alternative to TCP Reno. Unlike Reno (or its variants such as NewReno and SACK), that uses packet loss as a measure of congestion, Vegas uses queueing delay as a measure of congestion [17]. Vegas introduces a new congestion avoidance mechanism that corrects the oscillatory behavior of AIMD. While the AIMD algorithm induces loss to learn the available network capacity, a Vegas source adjusts its sending rate to keep a small number of packets buffered in the routers along the path. Provided there is enough buffering, a network of Vegas sources will stabilize around a proportionally fair equilibrium and packet loss will be eliminated; see [17] for details. In this paper, we study the stability of this equilibrium in the presence of network delay, motivated by two lines of recent research.

First, extensive experimental results have been conducted to compare the performance of Vegas and Reno, e.g., [6], [1], [8]. Its dynamic and fairness properties have also been studied in [4], [19], [5], but these papers consider only a single bottleneck link and network delay is not accounted for in the study of its dynamics. Optimization based models are used in [20], [17] to analyze a general network of Vegas. In particular, it is shown in [17], [15] that any TCP/AQM (active queue management) protocol can be interpreted as carrying out a distributed primal-dual algorithm over the Internet to maximize aggregate utility, and a user's utility is (often implicitly) defined by its TCP algorithm; see also [10], [16], [18], [20], [13]. These models mostly focus on the equilibrium structure, and do not adequately deal with network delay. To complement this series of work, we use here a multi-link multi-source model, described in Section II, that explicitly includes heterogeneous forward and backward delays to analyze the linear stability of Vegas around an equilibrium. In comparison with previous analytical work, we sacrifice global (nonlinear) dynamics in order to understand the effect of delay on stability of Vegas, and how to stabilize it.

Second, this paper is motivated by the stability theory for linear distributed and delayed system recently developed in [21], [23], [24], [22], [12]. In particular, a TCP/AQM algorithm is designed in [21] that maintains linear stability for *arbitrary* network delays and capacities. It is in the class of "dual" algorithms of [16], [3] that use static source algorithms, and it employs a sophisticated scaling with respect to network delays and capacities to achieve high utilization and fast response without compromising stability. This form of arbitrarily scalable stability, however, dictates a specific source utility function and hence fairness in rate allocation. By introducing a slower timescale dynamics into the source algorithm, the TCP/AQM of [21] is extended in [22] to track *any* utility function, or fairness, on a slow timescale, provided there is a known bound on network delays.

The main insight from this series of work is to scale down source responses with their own round trip times and scale down link responses with their own capacities, in order to keep the gain over the feedback loop under control. It turns out that the (implicit) link algorithm of Vegas has exactly the right scaling with respect to capacity as used in [21], [22]; see [17]. This built-in scaling with capacity makes Vegas potentially scalable to high bandwidth, in stark contrast to Reno and its variants. The source algorithm of Vegas, however, has a different scaling from those in [21], [22] with respect to delay. We prove in Section III a sufficient stability condition that suggests that Vegas can become unstable at large delay. In Section IV we propose a small modification to stabilize it. We describe an incremental deployment strategy in Section V that would allow Vegas sources to work with a mix of routers, some implementing a queue-clearing AQM and some not. Finally, we present simulation results in Section VI to compare the dynamics of Vegas and its stabilized version.

In contrast to Reno and its variants, Vegas seems particularly well-suited for high speed networks. Reno and its variants, with RED, become unstable as network capacity increases [9], [14]. It also must maintain an exceedingly small loss probability in equilibrium that is difficult to reliably use for control. Vegas, on the other hand, scales correctly with capacity. Moreover, while the equilibrium queueing delay can be excessive at low capacity, it is reduced as capacity increases. Other problems, such as error in propagation delay estimation due to queues and rerouting [19], [17], may be less severe at high capacity, as buffers clear more frequently.

## II. Network model

A network is modeled as a set of $L$ links (scarce resources) with finite capacities $c = (c_l, l \in L)$. They are shared by a set of $N$ sources indexed by $r$. Each source $r$ uses a set of links defined by the $L \times N$ routing matrix

$$R_{lr} \quad = \quad \begin{cases} 1 & \text{if source } r \text{ uses link } l \\ 0 & \text{otherwise} \end{cases}$$

Associated with each link $l$ is a congestion measure $p_l(t)$ we will call 'price'; as we will see below, $p_l(t)$ is the scaled queueing delay at link $l$. Each source $r$ maintains a rate $x_r(t)$ in packets/sec. In this paper, we are mainly concerned with linearized model around an equilibrium, so we denote the equilibrium forward delay from source $r$ to link $l$ by $\tau_{lr}^{\rightarrow}$ and the equilibrium backward delay from link $l$ to source $r$ by $\tau_{lr}^{\leftarrow}$. At time $t$, we assume source $r$ observes the aggregate price in its path

$$q_r(t) \quad := \quad \sum_l R_{lr} p_l(t - \tau_{lr}^{\leftarrow}) \qquad (1)$$

and link $l$ observes the aggregate source rate

$$y_l(t) \quad := \quad \sum_r R_{lr} x_r(t - \tau_{lr}^{\rightarrow}) \qquad (2)$$

Let $T_r$ denote the equilibrium round trip time. We assume that

$$\tau_{lr}^{\rightarrow} + \tau_{lr}^{\leftarrow} = T_r, \qquad \forall l \in L$$

Then [17] models TCP Vegas, with its associated queue management, as the following dynamical system:[1]

$$\dot{p}_l(t) \quad = \quad \begin{cases} \frac{1}{c_l}(y_l(t) - c_l) & \text{if } p_l(t) > 0 \\ \frac{1}{c_l}(y_l(t) - c_l)^+ & \text{if } p_l(t) = 0 \end{cases} \qquad (3)$$

$$\dot{x}_r(t) \quad = \quad \frac{1}{T_r^2(t)} \,\text{sgn}\left(1 - \frac{x_r(t) q_r(t)}{\alpha_r d_r}\right) \qquad (4)$$

where $(z)^+ = \max\{0, z\}$, sgn$(z)$ = 1 if $z > 0$, $-1$ if $z < 0$, and 0 if $z = 0$. Here, $\alpha_r$ is a Vegas protocol parameter, and $d_r$ is the round trip propagation delay of source $r$. Price $p_l(t)$ is the queueing delay at link $l$ and $q_r(t)$ is the end-to-end queueing delay of source $r$ (see [17]). Round trip time of source $r$ is defined as

$$T_r(t) \quad := \quad d_r + q_r(t)$$

with the equilibrium value $T_r$ defined in (II).

An interpretation of Vegas algorithm is that each source $r$ adjusts its rate (or window) to maintain $\alpha_r d_r$ number of its own packets buffered in the queues in its path. The link algorithm (3) is automatically carried out by the buffer process. The source algorithm (4) increments or decrements the *window* by 1 packet per round trip time, according as the number $x_r(t)q_r(t)$ of packets buffered in the links is smaller or bigger than $\alpha_r d_r$. In equilibrium, $x_r^* q_r^* = \alpha_r d_r$, and the unique equilibrium rates $x^* := (x_r^*, r = 1, \ldots, N)$ maximize aggregate utility $\sum_r U_r(x_r)$ subject to link capacity constraints, with the utility functions (see [17] for details)

$$U_r(x_r) \quad = \quad \alpha_r d_r \log x_r$$

Hence Vegas achieves weighted proportional fairness [10].

The link algorithm of [21], [22] is the same as (3) of Vegas, except that, there, $c$ is a virtual capacity that is strictly less than real link capacity in order to clear the queue in equilibrium. There, $p_l(t)$ can be interpreted as "virtual" queueing delay at a link fed by the same input but drained at the virtual capacity. As shown in [21], [22], scaling down $\dot{p}_l$ by $1/c_l$ is what gives delay, real or virtual, the right scaling with respect to network capacity. In Section V, we will explain how stabilized Vegas can be incrementally deployed in a network with a mix of links, generating both real and virtual queueing delays.

## III. Stability of Vegas

The Vegas source algorithm (4) is discontinuous. This can cause oscillation around the equilibrium. The original Vegas algorithm prevents oscillation by enlarging the equilibrium point to a set: source rate $x_r(t)$ (or window) is unchanged as soon as the number $x_r(t)q_r(t)$ of packets buffered in the links enters the range $[\alpha_r d_r, \beta_r d_r]$ with $\alpha_r < \beta_r$. It is however hard to control fairness with $\alpha_r < \beta_r$ [5]. Here, as in [17], we take $\alpha_r = \beta_r$.

In this section, we present a continuous approximation of the Vegas algorithm (4), derive a sufficient condition for linear stability based on this approximation. The condition suggests that a network of Vegas can become unstable when delay

[1] The model in [17] is discrete-time and ignores feedback delay in the interconnection defined by (1) and (2).

increases. In the next section, we will indeed propose to use (a stabilized version of) this continuous function to replace (4) in order to prevent oscillation due to discontinuity.

### A. Approximate model

Note that

$$\operatorname{sgn}(z) \quad \backsimeq \quad \frac{2}{\pi}\tan^{-1}(\eta z)$$

The approximation becomes exact in the limit as $\eta \to \infty$. Hence, consider the following approximation of (4):

$$\begin{aligned} \dot{x}_r(t) \quad &:= \quad f_r(x_r(t), q_r(t)) \\ &= \quad \frac{2}{\pi}\frac{1}{T_r^2(t)} \tan^{-1}\eta\left(1 - \frac{x_r(t)q_r(t)}{\alpha_r d_r}\right) \end{aligned} \tag{5}$$

where again $T_r(t) = d_r + q_r(t)$.

Consider the equilibrium point $(x^*, p^*)$. The rates $x^*$ are unique since the $\log$ utility function is strictly concave. Suppose the routing matrix $R$ has full rank, so that the equilibrium prices $p^* = (p_l^*, l = 1, \ldots, L)$ are also unique. Moreover, assume that only bottleneck links are included in the model so that $p_l^* > 0$ for all $l$. In equilibrium, the source rate $x_r^*$ and aggregate price $q_r^*$ satisfy

$$x_r^* q_r^* \quad = \quad \alpha_r d_r$$

Linearize around the equilibrium point,

$$\begin{cases} x_r(t) = x_r^* + \delta x_r(t) \\ q_r(t) = q_r^* + \delta q_r(t) \end{cases}$$

Then, to first order,

$$\dot{x}_r \;=\; \delta\dot{x}_r \;=\; \left.\frac{\partial f_r}{\partial x_r}\right|_* \delta x_r(t) + \left.\frac{\partial f_r}{\partial q_r}\right|_* \delta q_r(t)$$

where

$$\begin{aligned} \left.\frac{\partial f_r}{\partial x_r}\right|_* \quad &= \quad -\frac{2\eta}{\pi}\frac{1}{x_r^* T_r^2} \\ \left.\frac{\partial f_r}{\partial q_r}\right|_* \quad &= \quad -\frac{2\eta}{\pi}\frac{1}{q_r^* T_r^2} \end{aligned}$$

where we have used $T_r = d_r + q_r^*$. Hence, in Laplace domain, we have

$$\delta x_r(s) = -\frac{x_r^*}{q_r^*}\frac{a_r}{sT_r + a_r}\,\delta q_r(s) \tag{6}$$

where

$$a_r \quad = \quad \frac{2\eta}{\pi}\frac{1}{x_r^* T_r} \tag{7}$$

At the links, the equilibrium points $(y_l^*, p_l^*)$ satisfy $y_l^* = c_l$. Linearizing the link algorithm (3) around the equilibrium

$$\begin{cases} y_l(t) = y_l^* + \delta y_l(t) \\ p_l(t) = p_l^* + \delta p_l(t) \end{cases}$$

we have, to first order,

$$\begin{aligned} \delta\dot{p}_l \quad &= \quad \left.\frac{\partial g_l}{\partial p_l}\right|_* \delta p_l(t) + \left.\frac{\partial g_l}{\partial y_l}\right|_* \delta y_l(t) \\ &= \quad \frac{1}{c_l}\delta y_l(t) \end{aligned}$$

and its Laplace transform

$$\delta p_l(s) = \frac{1}{c_l s}\delta y_l(s) \tag{8}$$

In summary, the linearized model of Vegas is described by (6), (7) and (8). To simplify notation, we assume without loss of generality for the rest of the paper that all sources have the same target queue length, $\alpha_r d_r = \alpha$ for all $r$ (otherwise, take $\alpha$ to be the minimum $\alpha_r d_r$ in the stability results that follow).

### B. Stability

Following [21], we can express the error equations of (1)–(2) in matrix form, in Laplace, as

$$\delta y(s) \quad = \quad R(s)\delta x(s) \tag{9}$$

$$\delta q(s) \quad = \quad \operatorname{diag}\{e^{-sT_r}\}R^T(-s)\delta p(s) \tag{10}$$

where

$$R_{lr}(s) = \begin{cases} e^{-s\tau_{\overrightarrow{ls}}} & \text{if } R_{lr} = 1 \\ 0 & \text{otherwise} \end{cases}$$

The routing matrix $R(0) = R$ determines static relationship between equilibrium values, i.e.,

$$y^* = R(0)x^*, \qquad q^* = R^T(0)p^* \tag{11}$$

Given any finite positive $a$, let $\theta(a)$ be the unique solution in $(0, \pi/2)$ of

$$\theta\tan\theta \quad = \quad a \tag{12}$$

as a (strictly increasing) function of $a$.

We will characterize the stability of Vegas first in terms of maximum window size and then in terms of minimum queueing delay. The results below say that Vegas is linearly stable if the equilibrium window size is sufficiently small (Theorem 1), or equivalently, if the equilibrium queueing delay is sufficiently large (Corollary 2).

*Theorem 1:* Suppose for all $r$, $k_0 T_r \geq \max_r T_r$ for some $k_0$. Let $M$ be an upper bound on the number of links in the path of any source, $M \geq \max_r \sum_l R_{lr}$. The Vegas model described by (3) and (5) is locally asymptotically stable around the equilibrium point $(x_r^*, y_l^*, p_l^*, q_r^*)$ if

$$\max_r \; x_r^* T_r \, \operatorname{sinc}\theta\left(\frac{\hat{\eta}}{x_r^* T_r}\right) \quad < \quad \frac{\alpha}{M k_0^2}$$

where $\hat{\eta} := 2\eta/\pi$ and $\operatorname{sinc}\theta = \sin\theta/\theta$.

*Proof:* See Section III-C below. ■

Note that since $\theta(\cdot)$ is strictly increasing and $\operatorname{sinc}(\cdot)$ is strictly decreasing, the left-hand side of the stability condition in Theorem 1 is strictly increasing in the window size $x_r^* T_r$. Hence the stability condition imposes a limit on the maximum window size.

Since $x_r^* q_r^* = \alpha$, this condition directly translates into one on queueing delay. The left-hand side of the following corollary is strictly increasing in $q_r^*/T_r$, implying a lower bound on queueing delay.

*Corollary 2:* Suppose for all $r$, $k_0 T_r \geq \max_r T_r$ for some $k_0$. Let $M$ be an upper bound on the number of links in the path of any source, $M \geq \max_r \sum_l R_{lr}$. The Vegas model described by (3) and (5) is locally asymptotically stable around the equilibrium point $(x_r^*, y_l^*, p_l^*, q_r^*)$ if

$$\min_r \frac{q_r^*/T_r}{\operatorname{sinc}\theta\left(\frac{\hat{\eta}}{\alpha}\cdot\frac{q_r^*}{T_r}\right)} \quad > \quad M k_0^2$$

where $\hat{\eta} := 2\eta/\pi$ and $\operatorname{sinc}\theta = \sin\theta/\theta$.

The next result shows that the stability condition cannot be satisfied when there are more than one (bottleneck) link in a source's path.

*Corollary 3:* The stability condition cannot be satisfied if a source has more than one link, i.e., if there is an $r$ with $R_{lr} = 1$ for more than one $l$.

*Proof:* The conditions in Theorem 1 and Corollary 2 are the same, so we will work with Corollary 2. Since $\theta(\cdot) < \pi/2$, $\operatorname{sinc}\theta(\cdot) > 2/\pi$. By definition $k_0 \geq 1$. Hence the stability condition in Corollary 2 implies

$$\min_r \frac{q_r^*}{T_r} \quad > \quad \frac{2M}{\pi}$$

If $M \geq 2$, then the right-hand side is bigger than 1. Yet, the left-hand side cannot exceed 1 since $T_r = d_r + q_r^*$. ■

We emphasize that the stability condition is only sufficient in the multiple link case. It is however both necessary and sufficient in the single-link-homogeneous-source case. We now illustrate, for this case, the stability region and the effect of protocol parameter $\alpha = \alpha_r d_r$.

**Example 1: Single link with homogeneous sources** $(c, d, N)$ Consider a single link of capacity $c$ shared by $N$ homogeneous sources with round trip propagation delay $d$. For this case, $a_r = a_0$, $T_r = T_0$, and $\omega_r = \omega_0$ for all sources $r$ in the proof of Theorem 1; see Section III-C. This implies that the stability condition ($M = 1$ and $k_0 = 1$)

$$\frac{q_r^*/T_r}{\operatorname{sinc}\theta\left(\frac{\hat{\eta}}{\alpha}\cdot\frac{q_r^*}{T_r}\right)} \quad > \quad 1 \qquad \text{for all } r \tag{13}$$

is both necessary and sufficient. Note that the equilibrium quantities $q_r^*$ and $T_r$ depend on the target queue length $\alpha$. To get insight on the effect of protocol parameter $\alpha$ on stability, we look at a simpler condition.

As noted above, since $\operatorname{sinc}\theta(\cdot) > \frac{2}{\pi}$, a *necessary* condition is

$$\frac{q_r^*}{T_r} \quad > \quad \frac{2}{\pi} \qquad \text{for all } r$$

Since $T_r = d + q_r^*$ and $q_r^* = \alpha/x_r^* = \alpha N/c$ by symmetry, this condition is equivalent to

$$cd \quad < \quad \left(\frac{\pi}{2} - 1\right)\alpha N \tag{14}$$

Hence, a necessary condition for Vegas stability is that the bandwidth delay product be small. Moreover the stability region is larger with larger target queue length $\alpha$ or number $N$ of sources. ■

## C. Proof of Theorem 1

The proof proceeds in three steps. First, we follow the argument of [23], [22] to show that the Nyquist trajectories of the loop gain matrix is contained in the convex hull of $N$ complex functions of $j\omega$. Second, we show that at large enough $\omega$ when at least one of these functions has a phase lag of $-\pi$, all of them are contained in the unit circle, under an appropriate condition, and hence cannot encircle $-1$ in the complex plane. Third, we show that this condition is the one in the theorem.

**Step 1:** Using the linearized equations (6), (8), (9), (10), the return ratio seen at the source is described as :

$$\operatorname{diag}\left(a_r \frac{x_r^*}{q_r^*}\right)\operatorname{diag}\left(\frac{e^{-sT_r}}{sT_r + a_r}\right) R^T(-s)\operatorname{diag}\left(\frac{1}{c_l s}\right) R(s)$$

For stability, it suffices to show that the eigenvalues of this function does not encircle $-1$ in the complex plain for $s = j\omega$, $\omega \geq 0$. The set of eigenvalues is identical to that of

$$L(j\omega) = \operatorname{diag}\left(\frac{k e^{-j\omega T_r}}{j\omega T_r\,(j\omega T_r + a_r)}\right)\hat{R}^T(-j\omega)\hat{R}(j\omega) \tag{15}$$

where

$$k \quad = \quad \frac{M a_r T_r}{q_r^*} \quad = \quad \frac{2\eta M}{\pi\alpha} \tag{16}$$

and

$$\hat{R}(j\omega) := \operatorname{diag}\left(\sqrt{\frac{1}{c_l}}\right) R(j\omega)\operatorname{diag}\left(\sqrt{\frac{x_r^*}{M}}\right)$$

From the lemma of [23], the spectrum of $L(j\omega)$ satisfies

$$\begin{aligned}
&\sigma\left(L(j\omega)\right) \\
&= \quad \sigma\left(\text{diag}\left(\frac{ke^{-j\omega T_r}}{j\omega T_r\left(j\omega T_r + a_r\right)}\right)\hat{R}^T(-j\omega)\hat{R}(j\omega)\right) \\
&\subseteq \quad \rho\left(\hat{R}^T(-j\omega)\hat{R}(j\omega)\right)\cdot \\
&\qquad \text{co}\left(0\bigcup\left\{\frac{ke^{-j\omega T_r}}{j\omega T_r\left(j\omega T_r + a_r\right)},\ r = 1,\ldots,N\right\}\right)
\end{aligned}$$

where co$(\cdot)$ above denotes the convex hull of the $N$ eigentrajectories and the origin.

The spectral radius of $\hat{R}(j\omega)$ satisfies

$$\begin{aligned}
&\rho\left(\hat{R}^T(-j\omega)\hat{R}(j\omega)\right) \\
\leq\ & \left\|\text{diag}\left(\frac{1}{M}\right)R^T(-j\omega)\,\text{diag}\left(\frac{1}{c_l}\right)R(j\omega)\,\text{diag}(x_r^*)\right\|_\infty \\
\leq\ & \left\|\text{diag}\left(\frac{1}{M}\right)R^T(-j\omega)\right\|_\infty\cdot \\
& \left\|\text{diag}\left(\frac{1}{y_l^*}\right)R(j\omega)\,\text{diag}\left(x_r^*\right)\right\|_\infty \\
=\ & 1
\end{aligned}$$

since, by (11), all the absolute row sums are equal to 1. Hence,

$$\sigma\left(L(j\omega)\right) \quad\subseteq\quad \text{co}\left(0\bigcup\left\{\frac{ke^{-j\omega T_r}}{j\omega T_r\left(j\omega T_r + a_r\right)},\ r = 1,\ldots,N\right\}\right)$$

Let

$$\Lambda_r(j\omega) \quad := \quad \frac{ke^{-j\omega T_r}}{j\omega T_r\left(j\omega T_r + a_r\right)}$$

We now show that under the condition of the theorem, at no $\omega$ will the convex combination of $\Lambda_r(j\omega)$ encircle the critical point $-1$.

**Step 2:** Define $a_0 = \min_r a_r$ and $T_0 = \max_r T_r$. Let $\omega_r$, $r = 0, 1, \ldots, N$, be the value in $(0, \pi/2)$ that satisfies

$$\omega_r T_r \tan \omega_r T_r \quad = \quad a_r, \qquad r \geq 0 \tag{17}$$

Clearly $\omega_0 \leq \omega_r$ for all $r$. Here $\omega_r$, $r \geq 1$, is the critical frequency when the eigenvalue $\Lambda_r(j\omega)$ has a phase lag of $-\pi$. Hence, for $\omega < \omega_0 \leq \omega_r$, the convex combination of $\Lambda_r(j\omega)$ cannot encircle $-1$ because phase$(\Lambda_r(j\omega)) > -\pi$ for all $r$. We now show that, for $\omega \geq \omega_0$, all $\Lambda_r(j\omega)$ are in the unit circle and hence their convex combination cannot encircle $-1$ either.

For $\omega \geq \omega_0$, since $k_0 T_r \geq T_0$, we have

$$\begin{aligned}
|\Lambda_r(j\omega)| \quad &= \quad \frac{k}{\omega T_r\ \sqrt{\omega^2 T_r^2 + a_r^2}} \\
&\leq \quad \frac{kk_0^2}{\omega_0 T_0\sqrt{\omega_0^2 T_0^2 + a_0^2}} \\
&\leq \quad \frac{kk_0^2}{a_0}\cdot\frac{1}{\omega_0 T_0}\cdot\frac{a_0}{\sqrt{\omega_0^2 T_0^2 + a_0^2}} \\
&= \quad \frac{kk_0^2}{a_0}\cdot\text{sinc}\,\theta(a_0)
\end{aligned}$$

where the last equality follows from (17) and the definition of $\theta(\cdot)$ in (12). Hence a sufficient stability condition is $|\Lambda_r(j\omega)| < 1$ for all $r$, $\omega \geq \omega_0$, or:

$$\frac{\text{sinc}\,\theta(a_0)}{a_0} \quad < \quad \frac{1}{kk_0^2}$$

**Step 3:** Substituting $k$, from (16), and $a_0$:

$$k \;=\; \frac{\hat{\eta}M}{\alpha} \qquad \text{and} \qquad a_0 \;=\; \min_r a_r \;=\; \frac{\hat{\eta}}{\max_r x_r^* T_r}$$

into the above condition, we have

$$\max_r\ x_r^* T_r\ \text{sinc}\,\theta\left(\frac{\hat{\eta}}{\max_r\ x_r^* T_r}\right) \quad < \quad \frac{\alpha}{Mk_0^2}$$

Since $\text{sinc}\,\theta\left(\hat{\eta}(x_r^* T_r)^{-1}\right)$ is strictly increasing in $x_r^* T_r$, we have

$$\begin{aligned}
&\left(\max_r\ x_r^* T_r\right)\ \text{sinc}\,\theta\left(\frac{\hat{\eta}}{\max_r\ x_r^* T_r}\right) \\
&\quad = \quad \max_r\left\{x_r^* T_r\ \text{sinc}\,\theta\left(\frac{\hat{\eta}}{x_r^* T_r}\right)\right\}
\end{aligned}$$

hence the stability condition in the theorem. ■

## IV. Stabilized Vegas

In this section, we propose a PD (proportional differential) controller at each source to stabilize a network of Vegas sources. We modify the (approximate) Vegas algorithm (5) into

$$\dot{x}_r \quad = \quad \frac{w}{T_r^2(t)}\cdot\tan^{-1}\left(\eta_r(t)\Delta_r(t)\right) \tag{18}$$

or

$$\dot{x}_r = \begin{cases} \min\left[\frac{w}{T_r^2(t)}\left(e^{\eta_r(t)\Delta_r(t)} - 1\right), \frac{\ln 2}{T_r(t)}x(t)\right], & \text{if } \Delta_r(t) > 0 \\ \max\left[\frac{w}{T_r^2(t)}\left(1 - e^{-\eta_r(t)\Delta_r(t)}\right), -\frac{\ln 2}{T_r(t)}x(t)\right], & \text{otherwise} \end{cases} \qquad (1$$

where $T_r(t) = d_r + q_r(t)$, $\Delta_r(t) = 1 - \frac{x_r(t)q_r(t)}{\alpha_r d_r} - \kappa_r(t)\dot{q}_r(t)$ and

$$\kappa_r(t) \quad = \quad \frac{1}{a}\cdot\frac{T_r(t)}{q_r(t)} \tag{20}$$

$$\eta_r(t) \quad = \quad \frac{\mu a}{w}\cdot x_r(t)T_r(t) \tag{21}$$

Here, the parameter $w$ determines the maximum change in window size per round trip time (for the original Vegas, the maximum change is 1 packet per round trip time). The parameters $a > 0$ and $\mu \in (0,1)$ are to be chosen to ensure stability (see below). The overall gain parameter $\eta_r(t)$ is proportional to the current window size: the larger the window, the more aggressive the response. The gain $\kappa_r(t)$ on the differential term is proportional to the ratio of round trip time to end-to-end queueing delay of source $r$, and serves as a normalization to $\dot{q}_r(t)$. The additional differential term $\kappa_r(t)\dot{q}_r(t)$ anticipates the future of $q_r(t)$. Without this term, source rate $x_r(t)$ will be increased if the number $x_r(t)q_r(t)$ of packets buffered in the links is small compared with $\alpha_r d_r$. With this term, even when $x_r(t)q_r(t)$ is small, the source may decrease its rate if prices are rapidly growing, i.e. if $\dot{q}(t)$ is large. We note that a differential term is also used in the link algorithm of [11], motivated by an optimal control formulation of AQM design.

Both (18) and (19) have the same equilibrium point as the original Vegas, and both linearize to the same first-order equations

$$\delta \dot{x}_r = \left.\frac{\partial f_r}{\partial x_r}\right|_* \delta x_r(t) + \left.\frac{\partial f_r}{\partial q_r}\right|_* \delta q_r(t) + \left.\frac{\partial f_r}{\partial \dot{q}_r}\right|_* \delta \dot{q}_r(t)$$

where

$$\left.\frac{\partial f_r}{\partial x_r}\right|_* = -\frac{\mu a}{T_r}$$
$$\left.\frac{\partial f_r}{\partial q_r}\right|_* = -\frac{\mu a x_r^*}{T_r q_r^*}$$
$$\left.\frac{\partial f_r}{\partial \dot{q}_r}\right|_* = -\frac{\mu x_r^*}{q_r^*}$$

Its Laplace transform is

$$\delta x_r(s) = -\frac{\mu x_r^*}{q_r^*}\left(\frac{sT_r + a}{sT_r + \mu a}\right)\delta q_r(s) \qquad (22)$$

We have chosen $\eta_r$ and $\kappa_r$ so that the lead-lag compensator in (22) for all sources have a common zero $a$ and pole $\mu a$. In contrast, the algorithm of [22] allows $\mu_r$ to depend on $r$, corresponding to unrestricted choice of utility functions. Hence, we need a slightly different stability proof from [22].

*Theorem 4:* Suppose for all $r$, $k_0 T_r \geq \max_r T_r$ for some $k_0$. Let $M$ be an upper bound on the number of links in the path of any source, $M \geq \max_r \sum_l R_{lr}$. For any given $a > 0$ and $\mu \in (0,1)$, the modified Vegas model described by (3) and (18)–(21) is locally asymptotically stable around the equilibrium point $(x^*, y^*, p^*, q^*)$ if

$$\max_r \; x_r T_r \; < \; \frac{\alpha\phi}{\mu k_0 M}\sqrt{\frac{\phi^2 + \mu^2(k_0 a)^2}{\phi^2 + (k_0 a)^2}} \qquad (23)$$

or equivalently, if

$$\min_r \frac{q_r^*}{T_r} \; > \; \frac{\mu k_0 M}{\phi}\sqrt{\frac{\phi^2 + (k_0 a)^2}{\phi^2 + \mu^2(k_0 a)^2}} \qquad (24)$$

where

$$\phi = \tan^{-1}\frac{2\sqrt{\mu}}{1-\mu}$$

and $\alpha = \alpha_r d_r$ is the common target queue length.

*Proof:* The proof proceeds in two steps. First, we follow the argument of [23], [22] to show that the Nyquist trajectories of the loop gain matrix is contained in the convex hull of $N$ complex functions of $j\omega$. Second, we show that at large enough $\omega$ when at least one of these functions has a phase lag of $-\pi$, all of them are contained in the unit circle, under the conditions in the theorem, and hence cannot encircle $-1$ in the complex plane.

**Step 1:** Using the linearized equations (22), (8), (9) and (10), the return ratio seen at the sources can be written as:

$$\text{diag}\left(\frac{\mu x_r^*}{q_r^*}\right)\,\text{diag}\left(\frac{sT_r + a}{sT_r + \mu a}e^{-sT_r}\right) R^T(-s)\,\text{diag}\left(\frac{1}{c_l s}\right) R(s)$$

At $s = j\omega$, the set of eigenvalues is identical to that of

$$L(j\omega) = \text{diag}\left(\frac{\mu M T_r}{q_r^*}\right)\cdot$$
$$\text{diag}\left(\frac{e^{-j\omega T_r}}{j\omega T_r}\cdot\frac{j\omega T_r + a}{j\omega T_r + \mu a}\right)\hat{R}^T(-j\omega)\hat{R}(j\omega)$$

where

$$\hat{R}(j\omega) = \text{diag}\left(\sqrt{\frac{1}{c_l}}\right) R(j\omega)\text{diag}\left(\sqrt{\frac{x_r^*}{M}}\right) \qquad (25)$$

Using (25) and (11), the usual argument gives

$$\rho\left(\hat{R}^T(-j\omega)\hat{R}(j\omega)\right) \leq 1$$

The lemma from [23] then implies that all eigenvalues of $L(j\omega)$ are in the convex hull:

$$\text{co}\left\{0 \bigcup \left\{\frac{MT_r}{q_r^*}\cdot\Lambda(j\omega T_r)\right\},\quad r = 1,\ldots,N,\;\right\} \qquad (26)$$

where

$$\Lambda(j\omega T_r) := \mu\cdot\frac{e^{-j\omega T_r}}{j\omega T_r}\cdot\frac{j\omega T_r + a}{j\omega T_r + \mu a}$$

Note that $\Lambda(\cdot)$ is independent of $r$. By the generalized Nyquist stability criterion [7], the system is stable if the set in (26) does not encircle $-1$.

**Step 2:** Let $\omega_r$ be the critical frequency at which the phase $\angle\Lambda(j\omega_r T_r)$ is $-\pi$ for sources $r$:

$$\omega_r T_r - \angle\frac{j\omega T_r + a}{j\omega T_r + \mu a} = \frac{\pi}{2}$$

Without loss of generality, we can assume $T_1 \geq T_r$ for all $r$. Then $\omega_1 \leq \omega_r$ for all $r$ since $\omega_r T_r = \omega_1 T_1$ for all $r$. Thus, at $\omega \leq \omega_1$, the convex hull of (26) cannot encircle $-1$. At $\omega \geq \omega_1$, the set in (26) does not encircle $-1$ if, for all $r$,

$$\frac{MT_r}{q_r^*} \cdot |\Lambda(j\omega T_r)| \quad < \quad 1 \tag{27}$$

We now show that this is implied by the conditions in the theorem.

For $\omega \geq \omega_1$, we have $\omega T_r \geq \omega_1 T_r \geq \omega_1 T_1 / k_0$. Notice that the magnitude

$$|\Lambda(j\omega T_r)| \quad = \quad \frac{\mu}{\omega T_r} \sqrt{\frac{(\omega T_r)^2 + a^2}{(\omega T_r)^2 + \mu^2 a^2}}$$

is a strictly decreasing function of $\omega T_r$. Hence for all $r$

$$\begin{aligned} |\Lambda(j\omega T_r)| \quad &\leq \quad \left| \Lambda \left( j\omega_1 \frac{T_1}{k_0} \right) \right| \\ &= \quad \frac{\mu k_0}{\omega_1 T_1} \sqrt{\frac{(\omega_1 T_1)^2 + (ak_0)^2}{(\omega_1 T_1)^2 + \mu^2 (ak_0)^2}} \\ &\leq \quad \frac{\mu k_0}{\phi} \sqrt{\frac{\phi^2 + (ak_0)^2}{\phi^2 + \mu^2 (ak_0)^2}} \end{aligned}$$

The last inequality follows from Lemma 5 below which implies that, for all $r$,

$$\omega_r T_r \quad \geq \quad \frac{\pi}{2} - \tan^{-1} \frac{1-\mu}{2\sqrt{\mu}} \quad = \quad \phi$$

where $\phi$ is defined in the theorem.

The condition (24) in the theorem then guarantees that (27) holds. Since $x_r^* q_r^* \;=\; \alpha$, the conditions (23) and (24) are equivalent. Hence the proof is complete with the following lemma. ■

*Lemma 5:* Let

$$\Lambda(j\omega T_r) \quad = \quad \mu \cdot \frac{e^{-j\omega T_r}}{j\omega T_r} \cdot \frac{j\omega T_r + a}{j\omega T_r + \mu a}$$

where $0 < \mu < 1$ and $a > 0$. Then, for all $r$, $\omega \geq 0$,

$$\angle \Lambda(j\omega T_r) \quad \geq \quad -\omega T_r - \frac{\pi}{2} - \; \tan^{-1} \frac{1-\mu}{2\sqrt{\mu}}$$

*Proof:* Let

$$\begin{aligned} h_r(\omega) \quad &:= \quad \angle \frac{j\omega T_r + a}{j\omega T_r + \mu a} \\ &= \quad \tan^{-1} \left( \frac{\omega T_r}{a} \right) - \tan^{-1} \left( \frac{\omega T_r}{\mu a} \right) \end{aligned}$$

Then

$$h_r'(\omega) \quad = \quad \frac{(1-\mu)((\omega T_r)^2 - a^2 \mu) a}{((\omega T_r)^2 + a^2)((\omega T_r)^2 + \mu^2 a^2)}$$

Since $\mu \in (0,1)$, it can be checked that the solution, $\omega_r^* = \frac{a\sqrt{\mu}}{T_r}$, of $h'(\omega) = 0$ minimizes the phase $h_r(\omega)$. Hence

$$\angle h_r(\omega) \quad \geq \quad \tan^{-1} \sqrt{\mu} - \tan^{-1} \frac{1}{\sqrt{\mu}} \quad =: \quad \psi$$

which is independent of $r$. Moreover

$$\tan \psi \quad = \quad \frac{\sqrt{\mu} - \frac{1}{\sqrt{\mu}}}{2} \quad = \quad -\frac{1-\mu}{2\sqrt{\mu}}$$

Therefore

$$\begin{aligned} \angle \Lambda(j\omega T_r) \quad &= \quad -\omega T_r - \frac{\pi}{2} + \angle h_r(\omega) \\ &\geq \quad -\omega_r T_r - \frac{\pi}{2} - \tan^{-1} \frac{1-\mu}{2\sqrt{\mu}} \end{aligned}$$

and hence the lemma follows.

■

We remark on the implications of Theorem 4. In the case of homogeneous round trip time, i.e. $k_0 = 1$, the stability (24) becomes

$$M \max_r \frac{T_r}{q_r^*} \quad < \quad \frac{\theta}{\mu} \sqrt{\frac{\theta^2 + \mu^2 a^2}{\theta^2 + a^2}} \tag{28}$$

$M$ is a bound on the number of *bottleneck* links in the path of a source. It is typically much less than 10. $\frac{q_r^*}{T_r}$ is the ratio of round trip queueing delay and entire round trip time ; both quantities are available at a Vegas source. For the current network, this ratio seems to be small (less than 5) for long delay routes. Hence, a choice of design parameters $a$ and $\mu$ that guarantees that the right-hand side of the stability condition exceeds 100 seems safe. From Fig. 1, this requires small $a$ and $\mu$ (e.g., $a = 0.01$ and $\mu = 0.001$). Recall the definition of (20) and (21): $kappa_r(t) = (T_r(t)/q_r(t))/a$ and $\eta_r(t) = \mu a(x_r(t) T_r(t)/w)$. A small $a$ implies a large $kappa_r(t)$, which means that the stabilized Vegas reacts more aggressively to price change $\dot{q}_r(t)$. A small $\mu a$ implies a small $\eta$, which means that the slope of (18) around equilibrium is small, yielding a smoother overall gain. For the heterogeneous round-trip case, i.e. $k_0 > 1$, a smaller $a$ than in the homogeneous is required to guarantee stability.

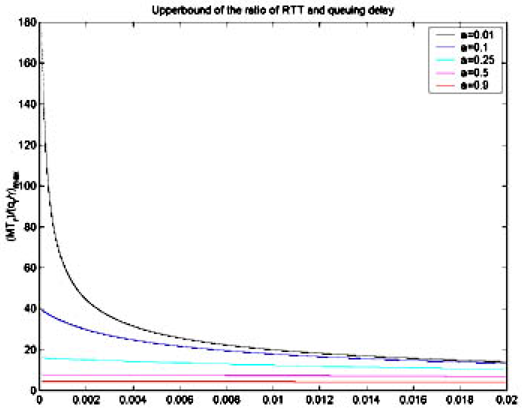


Fig. 1. The upper limit of $\frac{MT_r}{q_r^*}$.

**Example 2: Single link with homogeneous sources** $(c, d, N)$ For a direct comparison with the original Vegas stability, we consider the same setup as in Example 1: a single link of capacity $c$ shared by $N$ homogeneous sources with round trip propagation delay $d$. For this case, the sufficient condition of Theorem 4 with ($M = 1$ and $k_0 = 1$) is simplified to

$$q_r^*/T_r \quad > \quad \frac{\mu}{\phi}\sqrt{\frac{\phi^2 + a^2}{\phi^2 + \mu^2 a^2}} \qquad \text{for all } r$$

Since $T_r = d + q_r^*$ and $q_r^* = \alpha/x_r^* = \alpha N/c$ by symmetry, this condition is equivalent to

$$cd \quad < \quad \left(\frac{\phi}{\mu}\sqrt{\frac{\phi^2 + \mu^2 a^2}{\phi^2 + a^2}} - 1\right)\alpha N \qquad (29)$$

Hence, like the original Vegas, it also has a larger stability region with larger queue length $\alpha$ or number $N$ of sources. Furthermore, given $\alpha$ and $N$, stabilized Vegas can choose a small ($a > 0$, $\mu \in (0, 1)$) such that the right-hand side of (29) can be larger than that of (14) for stability of the original Vegas. This is illustrated in Fig. 2 where the stability regions in (14) and (29), for Vegas and stabilized Vegas respectively, are plotted, with $(a, \mu) = (0.5, 0.015)$ and $\alpha = 20$ packets, $N = 100$ sources. ■

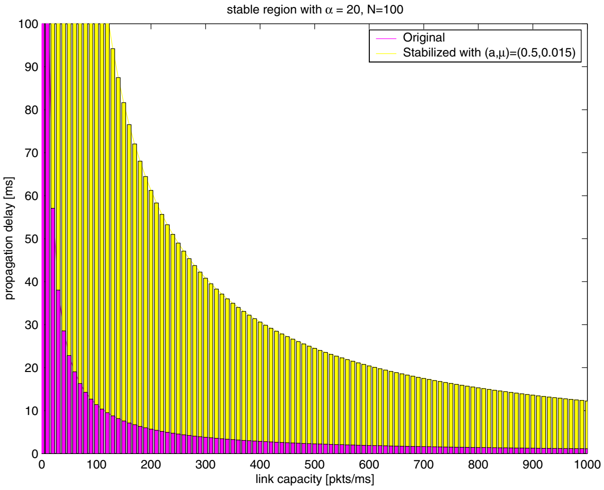


Fig. 2. Stability regions of Examples 1 and 2: single-link shared by homogeneous sources

The stability condition is only sufficient. Indeed, less conservative values can be used for $a$ and $\mu$. For instance, the Nyquist plots of $(MT_r/q_r^*) \cdot \Lambda(j\omega T_r)$ for $MT_r/q_r^* = 100$ are shown in Fig. 3, for the scenario in Example 2 with $a = 0.1$ and $\mu \in [0.001, 0.015]$. Even though these $a$ and $\mu$ values do not satisfy the stability condition of Theorem 4, the network is indeed stable, as shown by the Nyquist plots.

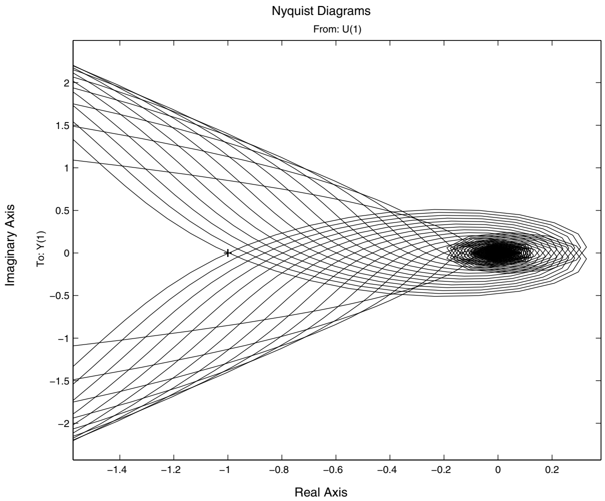


Fig. 3. Nyquist Stability of $\Lambda(j\omega T_r)$ for $\frac{MT_r}{q_r^*} = 100$ with $a = 0.1, \mu = [0.001 : 0.015]$

## V. Discussion: implementation and deployment

The most attractive feature of TCP Vegas is its suitability for high speed large delay networks. In this regime, window size is large and TCP Reno or its variants must maintain an extremely small loss probability (e.g., on the order $10^{-10}$) in equilibrium. Using such a small probability reliably is a great challenge.

Vegas on the other hand has two advantages, both stem from the use of delay as a measure of congestion. First, its implicit link algorithm has a built-in scaling with respect to network capacity, which together with stabilized source algorithm, can potentially scale to much larger bandwidth delay product. Second, each measurement of delay by a source provides a much finer-grained estimate of congestion than the binary-valued loss or marking does. As capacity scales up, it is easy to scale at the source to maintain the strength of the congestion signal (delay) by scaling up the $\alpha$ parameter.

The problem that delay may be excessive in the low bandwidth regime in order for Vegas to reach equilibrium is much less severe in the high bandwidth regime. Moreover, problem with error in propagation delay estimation and persistent congestion [19], [17] is also eased with high capacity, as buffers empty more frequently. Though there are other issues with using delay for congestion control, it seems that unless ECN is widely deployed, these problems are less fundamental than the intrinsic difficulty of reliably using an extremely small loss probability for control. Further study is required to resolve these issues.

We now describe a viable strategy for stabilized Vegas to work with incremental deployment of new AQM and ECN. The link algorithm in Vegas computes the queueing delay as follows:

$$\dot{p}_l(t) \quad = \quad \frac{1}{c_l}(y_l(t) - c_l) \qquad (30)$$

The division by $c_l$ is what gives Vegas the built-in scalability with network capacity (see proofs of Theorems 1 and 4). As discussed in [17], Vegas exploits the buffer process to automatically compute congestion prices, at the expense of having to maintain a non-zero queueing delay (these are Lagrange multipliers for the utility maximization problem Vegas is implicitly solving).

The link algorithm in the scalable scheme of [21], [22] uses the same expression as in (30) except that instead of real link capacity $c_l$, a virtual link capacity that is slightly smaller (say, 95% of $c_l$) is used to explicitly compute the price $p_l(t)$. The advantage of using a virtual queue is that while the prices still converge to their non-zero values, the real queue will be cleared in equilibrium. Since queues are now empty, queueing delay can no longer serve as a feedback signal. ECN marking must be used to explicitly feedback the prices, e.g., using REM [2], [21], [22].

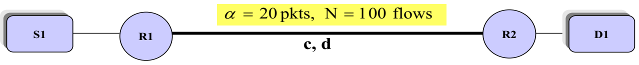
Imagine now a network with both types of links, one does not use ECN nor perform AQM to clear their queues and one does. The first type maintains a queue but does not mark, while the second type has no queueing delay but sends a stream of marks to the sources. A source observes two types of feedback signals: aggregate queueing delay from type 1 links and aggregate prices from type 2 links (after REM estimation). Not only do these two signals not interfere with each other, their sum yields precisely the total price in the path of the source! Hence, by observing both signals and summing them, the source automatically obtains the necessary information for its control, without having to know the type or number of links in its path. As more and more links convert to AQM with ECN, the source algorithm needs no upgrade. The only effect is that queueing delay steadily decreases.

## VI. Simulation results

We have conducted an extensive set of ns-2 simulations to study the behavior of Vegas and stabilized Vegas algorithms with different number of sources, homogeneous and heterogeneous sources, single and multiple bottleneck links, in dynamic scenarios, with two-way traffic, and with mice traffic. Due to space limit, we will report only the simplest set of results with a single bottleneck link, that validates the theory in the previous sections and illustrates the suitability of stabilized Vegas in high speed large delay networks where window size, in packets, is large. Other simulation results are included in the journal version of the paper.

We simulate the scenarios of Examples 1 and 2 with a single bottleneck link shared by $N$ homogeneous sources. The simulations use the usual dumb-bell topology with $N$ sources connected to $N$ destinations via two routers. The access links between sources (or destinations) and their router are non-bottlenecks with zero latency, and the link between the two routers is the only bottleneck link with capacity $c$ Gbps (or pkts/ms) with a fixed packet size of 1 KBytes. The latency between the routers is $d/2$ ms. Routers do FIFO with Droptail and queue capacity is set to $40$K packets so the possibility of packet loss is negligible. All sources start to send packet simultaneously.

We fix $\alpha = 20$ packets and $N = 100$ flows, and vary $c$ and $d$. We present three sets of simulations, with the three different $(c, d)$s on the region of Fig. 2, as shown in Fig. 4.

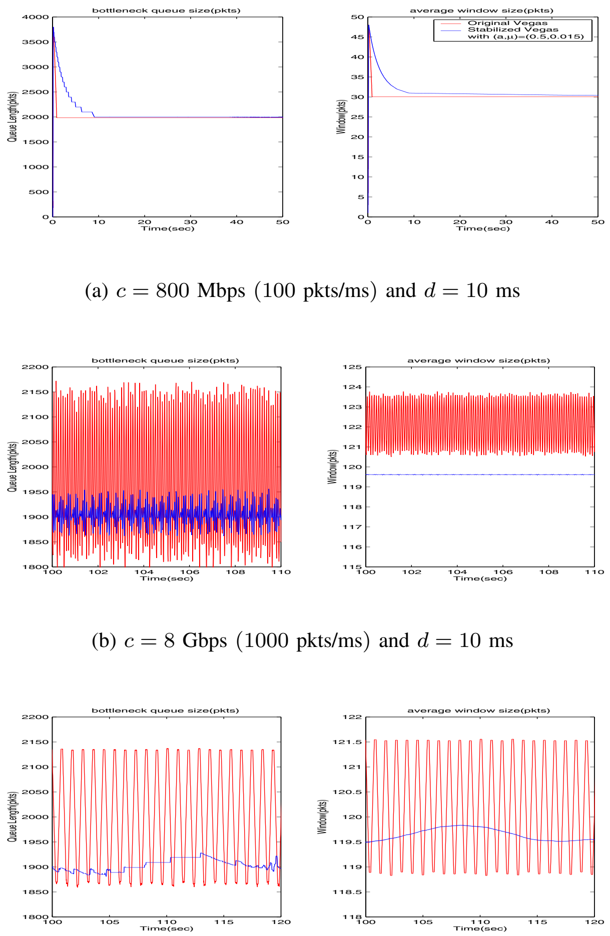


| | capacity<br>c [Gbps]<br>(pkts/ms) | propagation<br>delay<br>d [ms] | expected<br>queue length<br>αN [pkts] | expected<br>window size<br>$\alpha + \frac{cd}{N}$ [pkts] |
|---|---|---|---|---|
| (a) | 0.8 (100) | 10 | 2000 | 30 |
| (b) | 8.0 (1000) | 10 | 2000 | 120 |
| (c) | 0.8 (100) | 100 | 2000 | 120 |

Fig. 4. Network topology with single bottleneck and homogeneous sources : $(\alpha, N) = (20$ pkts, $100$ flows). For stabilized Vegas, $(a, \mu) = (0.5, 0.015)$.

Simulation (a) is for small capacity and delay in the intersection of both stability regions. Simulation (b) scales up the capacity by 10 times, and simulation (c) scales up the delay used in (a) by 10 times. Both (b) and (c) are outside the stability region of the original Vegas, but still in the stability region of stabilized Vegas. For both the original Vegas and Stabilized Vegas algorithms, we set the target queue length of each flow $\alpha = 20$ pkts and $N = 100$ flows. The last two columns of Fig. 4 show the equilibrium queue length and the equilibrium window size calculated from [17].

The simulation results are shown in Fig. 5. The first plot of each case shows the total queue length buffered in the bottleneck link. The second plots are the average window size averaged over the $N$ sources. As expected, the original Vegas exhibits instability in cases (b) and (c), where the stabilized Vegas remains stable.



(a) $c = 800$ Mbps $(100$ pkts/ms$)$ and $d = 10$ ms

(b) $c = 8$ Gbps $(1000$ pkts/ms$)$ and $d = 10$ ms

(c) $c = 800$ Mbps $(100$ pkts/ms$)$ and $d = 100$ ms

Fig. 5. The queue length and the average window size near equilibrium ($\alpha = 20$ pkts, $N = 100$).

In order to see the performance of the stabilized Vegas for high speed large delay networks where window size can be large, we use a much smaller number of flows, $N = 3$, with $(c, d) = (3.2 \text{ Gbps}, 100 \text{ ms})$. We also set $\alpha = 400$ packets , so that about 3% of propagation delay is allowed as the queuing delay. Here the equilibrium window size calculated from [17] is about $W^* = 13,700$ packets per flow. As shown in Fig. 6, the stabilized Vegas can almost achieve the equilibrium window size in less than 10 ms, while the linear increment of the original Vegas is too far slow. The stabilized Vegas also has good steady-state.

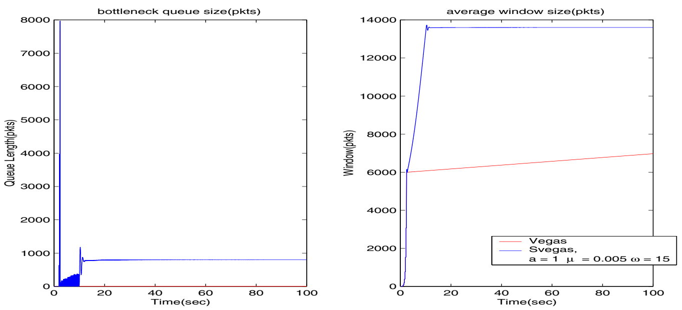


Fig. 6. Fast convergence for high speed network $c = 3.2$ Gbps, $d = 100$ ms and $(\alpha, N) = (400 \text{ pkts}, 3)$.

## VII. Conclusion

In this paper, we have presented a detailed analysis of Vegas stability in a general multi-link multi-source setup with heterogeneous forward and backward delays. We have derived a stability condition that suggests that Vegas can be unstable in the presence of delay. We have proposed a small modification that stabilizes it in the presence of large network delays.

Vegas is particularly attractive for high speed network because of its built-in scalability with network capacity. In the high bandwidth regime, the potential problem with persistent congestion of Vegas is alleviated. Moreover, it avoids the intrinsic difficulty of having to control based on extremely small loss probability, as Reno must. Despite these advantages, there are issues associated with delay-based congestion control that must be resolved, especially incremental deployment. We have described one aspect of this: how Vegas source can work gracefully as the network migrates to an ECN-based AQM.

**Acknowledgments:** We are grateful to John Doyle, Fernando Paganini, Jiantao Wang, Zhikui Wang, for helpful discussions. We acknowledge the support of Korean Government through BK21 Project, NSF through grant ANI-0113425.

## References

[1] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan. Evaluation of TCP Vegas: emulation and experiment. In *Proceedings of SIGCOMM'95*, 1995.

[2] Sanjeewa Athuraliya, Victor H. Li, Steven H. Low, and Qinghe Yin. REM: active queue management. *IEEE Network*, 15(3):48–53, May/June 2001. Extended version in *Proceedings of ITC17*, Salvador, Brazil, September 2001. `http://netlab.caltech.edu`.

[3] Sanjeewa Athuraliya and Steven H. Low. Optimization flow control with Newton–like algorithm. *Journal of Telecommunication Systems*, 15(3/4):345–358, 2000.

[4] Thomas Bonald. Comparison of TCP Reno and TCP Vegas via fluid approximation. In *Workshop on the Modeling of TCP*, December 1998. `http://www.dmi.ens.fr/\%7Emistral/tcpworkshop.html`.

[5] C. Boutremans and J. Y. Le Boudec. A note on the fairness of TCP Vegas. In *Proceedings of International Zurich Seminar on Broadband Communications*, pages 163–170, February 2000.

[6] Lawrence S. Brakmo and Larry L. Peterson. TCP Vegas: end-to-end congestion avoidance on a global Internet. *IEEE Journal on Selected Areas in Communications*, 13(8):1465–80, October 1995. `http://cs.princeton.edu/nsg/papers/jsac-vegas.ps`.

[7] C. A. Desoer and Y. T. Yang. On the generalized Nyquist stability criterion. *IEEE Trans. on Automatic Control*, 25:187–196, 1980.

[8] U. Hengartner, J. Bolliger, and T. Gross. TCP Vegas revisited. In *Proceedings of IEEE Infocom*, March 2000.

[9] Chris Hollot, Vishal Misra, Don Towsley, and Wei-Bo Gong. A control theoretic analysis of RED. In *Proceedings of IEEE Infocom*, April 2001. `http://www-net.cs.umass.edu/papers/papers.html`.

[10] Frank P. Kelly, Aman Maulloo, and David Tan. Rate control for communication networks: Shadow prices, proportional fairness and stability. *Journal of Operations Research Society*, 49(3):237–252, March 1998.

[11] Ki Baek Kim and Steven H. Low. Analysis and design of AQM based on receding horizon control in stabilizing TCP. Caltech Technical Report, caltechCSTR:2002.009, March, 2002. Submitted for publication, 2002.

[12] S. Kunniyur and R. Srikant. A time-scale decomposition approach to adaptive ECN marking. *IEEE Transactions on Automatic Control*, June 2002.

[13] Srisankar Kunniyur and R. Srikant. End–to–end congestion control schemes: utility functions, random losses and ECN marks. In *Proceedings of IEEE Infocom*, March 2000. `http://www.ieee-infocom.org/2000/papers/401.ps`.

[14] S. H. Low, F. Paganini, J. Wang, S. A. Adlakha, and J. C. Doyle. Dynamics of TCP/RED and a scalable control. In *Proc. of IEEE Infocom*, June 2002. `http://netlab.caltech.edu`.

[15] Steven H. Low. A duality model of TCP and queue management algorithms. In *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management (updated version)*, September 18-20 2000. `http://netlab.caltech.edu`.

[16] Steven H. Low and David E. Lapsley. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, December 1999. `http://netlab.caltech.edu`.

[17] Steven H. Low, Larry Peterson, and Limin Wang. Understanding Vegas: a duality model. *J. of ACM*, 49(2):207–235, March 2002. `http://netlab.caltech.edu`.

[18] L. Massoulie and J. Roberts. Bandwidth sharing: objectives and algorithms. In *Infocom'99*, March 1999. `http://www.dmi.ens.fr/\%7Emistral/tcpworkshop.html`.

[19] J. Mo, R. La, V. Anantharam, and J. Walrand. Analysis and comparison of TCP Reno and Vegas. In *Proceedings of IEEE Infocom*, March 1999.

[20] Jeonghoon Mo and Jean Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking*, 8(5):556–567, October 2000.

[21] Fernando Paganini, John C. Doyle, and Steven H. Low. Scalable laws for stable network congestion control. In *Proceedings of Conference on Decision and Control*, December 2001. `http://www.ee.ucla.edu/~paganini`.

[22] Fernando Paganini, Zhikui Wang, Steven H. Low, and John C. Doyle. A new TCP/AQM for stability and performance in fast networks. In *Proc. of 39th Annual Allerton Conference on Communication, Control, and Computing*, October 2002.

[23] Glenn Vinnicombe. On the stability of end-to-end congestion control for the Internet. Technical report, Cambridge University, CUED/F-INFENG/TR.398, December 2000.

[24] Glenn Vinnicombe. Robust congestion control for the Internet. submitted for publication, 2002.