Internet congestion control model

GABRIELA MIRCEA
Faculty of Economic Sciences
West University of Timişoara
ROMANIA
gabriela.mircea@fse.uvt.ro , http://www.fse.uvt.ro

Abstract: - In this paper, we consider an Internet model with n access links, which respond to congestion signals from the network, and study the bifurcation of such a system. By choosing the gain parameter as a bifurcation parameter, we prove that a Hopf bifurcation occurs. The stability of bifurcating periodic solutions and the direction of the Hopf bifurcation are determined by applying the normal form theory and the center manifold theorem. Finally a numerical example is given to verify the theoretical analysis.

Key-Words: Internet model, Hopf bifurcation, feedback delay, numerical simulation

1 Introduction

Congestion control mechanisms and active queue management schemes (AQM) for Internet have extensively studied. In [4], a stability condition was provided for a single proportionally fair congestion controller with delayed feedback.

In this paper, we consider an Internet model with n $(n \geq 3)$ link and single source, which can be formulated as a:

$$\dot{x}_i(t) = k(w - ax_i(t-\tau)p(x_i(t-\tau) - b\sum_{\substack{j=1 \\ j\neq i}}^{n} x_j(t-\tau)p(x_i(t-\tau))),\ i = 1,...,n \quad (1)$$

where $x_i(t)$ is the sending rate of the source i at time t, k is a positive gain parameter, τ is the sum of forward and return delays, w is a target (set – point), and the congestion indication function $p(\cdot)$ is increasing, nonnegative, and not identically zero, which can be viewed as the probability that a packet at the source receives a "mark"- a feedback congestion indication signal.

According to the existing theoretical results [2] on stability analysis of system (1), the rate control algorithm cannot always guarantee a stationary sending rate. We will show in this paper that a Hopf bifurcation will occur when the positive gain parameter k passes through a critical value, i.e, a family of periodic solutions will bifurcate out from the equilibrium. On the one hand, the bifurcations, which involve emergence of oscillatory behaviors, may provide an explanation for the parameter sensitivity observed in practice; and on the other hand, if we understand more about the bifurcation behaviors of the Internet congestion control systems, we can apply the existing effective bifurcation control methods to achieve some desirable system behaviors that benefit congestion controls. The rest of the paper is organized as follows. In section 2, we study the existence of the Hopf bifurcation. Based on the normal form theory and the center manifold theorem introduced in [3], the formulas for the determining the properties of the Hopf bifurcating periodic solutions are derived in section 3. In section 4, a numerical example is given to demonstrate the theoretical analysis. Finally, conclusions are drawn in section 5.

2 Existence of the Hopf bifurcation in the Internet model

Without loss of generality, we assume that the function $f(x) = xp(x)$ is a nonlinear function and its third –order continuous derivative exists, and that the delayed differential equations (1) is supplemented with an initial conditions of the form:

$$x(\theta) = \varphi(\theta), \theta \in [-\tau, 0], \quad (2)$$

where:

$x(\theta) = (x_1(\theta),..., x_n(\theta))^T$ $\varphi(\theta) = (\varphi_1(\theta),...,\varphi_n(\theta))^T$

and $\varphi_i(\cdot), i = 1,...,n$ is assumed to be a real – valued function continuous on $[-\tau, 0]$. The symmetric equilibrium point of (1) is $X^* = (x^*,...,x^*)^T$, where x^* satisfies:

$$w = (a + (n-1)b) f(x^*) \quad (3)$$

The linearized equation of (1) in x^* is:

$$\dot{u}(t) = -k\rho_1 Bu(t-\tau) \quad (4)$$

where $u(t) = (u_1(t),...,u_n(t))^T$, $\rho_1 = f'(x^*)$

and $B = \begin{pmatrix} a & b & ... & b \\ b & a & ... & b \\ b & b & ... & a \end{pmatrix}$.

The characteristic equation of (4) is:

$$(\lambda + k\rho_1(a+(n-1)b)e^{-\lambda\tau})(\lambda + k\rho_1(a-b)e^{-\lambda\tau})^{n-1} = 0 \quad (5)$$

The analysis of the characteristic equation (5) follows.

Theorem 1:

i) when $b > 0, a > 0, a > b, k\rho_1 > 0$ and $\tau = 0$, the equilibrium point of (1), be $X^* = (x^*,...,x^*)^T$ is asymptotically stable.

ii) when $b > 0$, $a > 0$, $\rho_1 > 0$, the characteristic equation (5) has pure imaginary roots $\lambda = \pm i\omega$, $\omega > 0$, if and only if :

$$k = k_p = \frac{(2p+1)\pi}{2\tau\rho_1(a+(n-1)b)}, \ p = 0,2,4,... \quad (6)$$

iii) when the gain parameter k passes through the value k_0, there is a Hopf bifurcation of system (1) at its equilibrium x^*.

3 The direction and stability of bifurcating periodic solutions

In this section, we study the direction, stability and the period of the bifurcating periodic solutions in system (1). The method we use is based on the normal form theory and the center manifold theorem introduced in [2]. For notational convenience, let $k = k_0^* + \mu$. Then $\mu = 0$ is the Hopf bifurcation value for (1). In this section, we assume that the function $f \in C^3(R)$. Expanding the vector field in equations (1) into first, second, third order around x^*, and defining $u(t) = x(t) + x^*$, then equations (1) can be rewritten as:

$$\dot{u}(t) = -k_0\rho_1 BF_1(u(t-\tau)) + F(u(t-\tau)) + O(|u(t-\tau)|^4) \quad (7)$$

where

$$\begin{aligned} F(u(t-\tau)) &= -\frac{1}{2}k_0\rho_2 BF_2(u(t-\tau)) - \frac{1}{6}k_0\rho_3 BF_3(u(t-\tau)), \\ F_1(u(t-\tau)) &= (u_1(t-\tau),...,u_n(t-\tau))^T, \\ F_2(u(t-\tau)) &= (u_1(t-\tau)^2,...,u_n(t-\tau)^2)^T, \\ F_3(u(t-\tau)) &= (u_1(t-\tau)^3,...,u_n(t-\tau)^3)^T, \\ \rho_1 &= f'(x^*), \rho_2 = f''(x^*), \rho_3 = f'''(x^*), \\ B &= \begin{pmatrix} a & b & ... & b \\ b & a & ... & b \\ b & b & ... & a \end{pmatrix}. \end{aligned} \quad (8)$$

The operators A and A^* associated to the equations (4) are given by:

$$A(\mu)\varphi(\theta) = \begin{cases} \dfrac{d\varphi(\theta)}{d\theta}, \theta \in [-\tau, 0] \\ -k_0(\mu)\rho_1 B\varphi(\tau), \theta = 0 \end{cases}, \quad (9)$$

$\varphi \in C^1([-\tau,0], R^n)$,

$$A^*(\mu)\varphi^*(s) = \begin{cases} -\dfrac{d\varphi^*(s)}{ds}, s \in [0,\tau] \\ -k_0(\mu)\rho_1 B\varphi^*(\tau), s = 0 \end{cases}, \quad (10)$$

$\varphi^* \in C^1([-\tau,0], R^{n*})$.

For $\varphi \in C([-\tau,0], C^n)$ and $\varphi^* \in C([-\tau,0], C^{n*})$ we define a bilinear form by:

$$(\varphi^*, \varphi) = \overline{\varphi}^*(0)\varphi(0) - \int_{\theta=-\tau}^{\theta}\int_{s=0}^{\theta} \overline{\varphi}^*(s-\theta)B\delta(\theta+\tau)\varphi(s)ds \quad (11)$$

where δ is the Dirac delta function.

In order to determine the Poincare normal form of the operator A, we need to calculate the eigenvector $\phi(\theta)$ of A associated with the eigenvalue $i\omega$ and the eigenvector $\phi^*(\theta)$ of A^* associated with the eigenvalue $i\omega$. We can easily verify that for $\omega_0 = \omega(0) = \dfrac{\pi}{2\tau}$,

$$\phi(\theta) = e^{i\omega_0\theta}h, \overline{\phi}^*(\theta) = e^{-i\omega_0\theta}h, \quad \theta \in [-\tau, 0], \ h = (1,1,...,1)^T \quad (12)$$

are the eigenvectors of $A(0)$ associated with $\pm i\omega_0$ and

$$\psi(s) = \frac{2}{n(1+\pi i)}e^{i\omega_0 s}h, \overline{\psi}(s) = \frac{2}{n(1+\pi i)}e^{-i\omega_0 s} \quad h, s \in [0,\tau], \quad (13)$$

are the eigenvectors of $A^*(0)$ associated with $\pm i\omega_0$.

From (11), we have:

$$(\psi(s), \phi(\theta)) = 1, (\overline{\psi(s)}, \phi(\theta)) = 0,$$
$$(\psi(s), \overline{\phi}(\theta)) = 0, (\overline{\psi}(s), \overline{\phi}(\theta)) = 1,$$
$$s \in [0,\tau], \theta \in [-\tau, 0].$$

In what follows, we will continue the ideas and use the notations in [4].

Let

$$u(t-\tau) = z(t)e^{-i\omega_0\tau}h + \overline{z}(t)e^{i\omega_0\tau}h + \frac{1}{2}w_{20}(-\tau)z^2(t) + w_{11}(-\tau)z(t)\overline{z}(t) + \frac{1}{2}w_{02}(-\tau)z^2(t)$$

where

$w_{20}(-\tau) \in C^n, w_{11}(-\tau) \in R^n, w_{02}(-\tau) = \overline{w}_{20}(-\tau) \in C^n$ and $z(t) = x(t) + iy(t)$, $x(t), y(t) \in R.$

Replacing in (8), we have:

$$F(u(t-\tau)) = \frac{1}{2}F_{20}z(t)^2 + F_{11}z(t)\bar{z}(t) + \frac{1}{2}F_{02}\bar{z}(t)^2 + \frac{1}{2}F_{21}z(t)^2\bar{z}(t) \quad (14)$$

where

$$\begin{aligned} F_{20} &= k_0\rho_2(a+(n-1)b)h, \\ F_{11} &= -k_0\rho_2(a+(n-1)b)h, \\ F_{02} &= k_0\rho_2(a+(n-1)b)h, \\ F_{21} &= -k_0\rho_2 iB(w_{20}(-\tau)-2w_{11}(-\tau)+ \\ &\quad k_0\rho_3(a+(n-1)b))h. \end{aligned} \quad (15)$$

Theorem 2:

i) The local center manifold $W^c_{loc}(0)$ of system (7), in its point of origin, contains the elements $\tilde{\varphi} \in C^1([-\tau,0], R^n)$ given by:

$$\tilde{\varphi}(\theta) = z\phi(\theta) + \bar{z}\bar{\phi}(\theta) + \frac{1}{2}z^2 w_{20}(\theta) + z\bar{z}w_{11}(\theta) + \frac{1}{2}\bar{z}^2 w_{02}(\theta), \theta \in [-\tau, 0] \quad (16)$$

where $z = x_1 + iy_1, (x_1, y_1) \in V_1 \subset R^2$, and

$$\begin{aligned} w_{20}(\theta) &= -\frac{g_{20}}{i\omega_0}e^{i\omega_0\theta}h - \frac{\bar{g}_{02}}{3i\omega_0}e^{-i\omega_0\theta}h + 2e^{i\omega_0\theta}E_1, \\ w_{11}(\theta) &= -\frac{g_{11}}{i\omega_0}e^{i\omega_0\theta}h - \frac{\bar{g}_{11}}{i\omega_0}e^{-i\omega_0\theta}h + E_2, \end{aligned} \quad (17)$$

and

$$\begin{aligned} g_{20} &= -g_{11} = g_{02} = \frac{2k_0\rho_2}{1-\pi i}(a+(n-1)b), \\ g_{21} &= \frac{2k_0(a+(n-1)b)}{n(1-\pi i)}(\rho_2 ih^T(w_{20}(-\tau)-2w_{11}(-\tau))-\rho_3 n), \\ E_1 &= k_0\rho_2(a+(n-1)B)(k_0\rho_1 B-2i\omega I)^{-1}h, \\ E_2 &= -k_0\rho_2(a+(n-1)B)(-k_0\rho_1 B)^{-1}h. \end{aligned} \quad (18)$$

ii) The solution of equation (1) around x^*, is given by

$$x_i(t) = 2x(t) + r_{20}(x(t)^2 - y(t)^2) + r_{11}(x(t)^2 + y(t)^2) - 2i_{20}x(t)y(t) + x^*, i = 1,...,n \quad (19)$$

where $(x(t)y(t))$ is the solution of the systems:

$$\begin{aligned} \dot{x}(t) &= -\omega_0 y(t) + \frac{1}{2}(R_{20}+2R_{11}+R_{02})x(t)^2 - \frac{1}{2}(R_{20}-2R_{11}+R_{02})y(t)^2 \\ &\quad + (I_{02}-I_{20})x(t)y(t) + \frac{1}{2}R_{21}x(t)(x(t)^2+y(t)^2) - \frac{1}{2}I_{21}y(t)(x(t)^2+y(t)^2) \\ \dot{y}(t) &= -\omega_0 x(t) + \frac{1}{2}(I_{20}+2I_{11}+I_{02})x(t)^2 - \frac{1}{2}(I_{20}-2I_{11}+I_{02})x(t)^2 \\ &\quad + (R_{20}-R_{02})x(t)y(t) + \frac{1}{2}R_{21}y(t)(x(t)^2+y(t)^2) - \frac{1}{2}I_{21}x(t)(x(t)^2+y(t)^2) \end{aligned} \quad (20)$$

with initial condition

$x(0) = \mathrm{Re}(\psi,\varphi), y(0) = \mathrm{Im}(\psi,\varphi)$ and

$R_{ij} = \mathrm{Re}(g_{ij}), I_{ij} = \mathrm{Im}(g_{ij}), i, j = 0,1,2,$

$r_{20} = \mathrm{Re}(v_{20}), r_{11} = \mathrm{Re}(v_{11})$

and

$$\begin{aligned} v_{20} &= -\frac{g_{20}}{i\omega_0} - \frac{\bar{g}_{02}}{3i\omega_0} + \\ &\quad \frac{\rho_2(a+(n-1)b)(k_0\rho_1(a-b)-2i\omega_0)}{(n-1)k_0\rho_1^2 - (k_0\rho_1 a - 2i\omega_0)(k_0\rho_1(a+(n-2)b)-2i\omega_0)} \\ v_{11} &= -\frac{g_{11}+\bar{g}_{11}}{i\omega_0} + \\ &\quad \frac{\rho_2(a+(n-1)b)(b-a)}{\rho_1((n-1)b^2 - a(a+(n-2)b))} \end{aligned} \quad (21)$$

Therefore we have the formulas to compute the following parameters:

$$\begin{aligned} \mu_2 &= -\frac{\mathrm{Re}\,C(0)}{\mathrm{Re}\,M(0)}, \\ T_2 &= -\frac{\mathrm{Im}\,C(0) + \mu_2 \,\mathrm{Im}\,M(0)}{\omega_0}, \\ \beta_2 &= 2\,\mathrm{Re}\,C(0) \end{aligned} \quad (22)$$

where

$$\begin{aligned} C(0) &= \frac{i}{2\omega}(g_{20}g_{11} - 2|g_{11}|^2 - \frac{1}{3}|g_{02}|^2) + \frac{g_{21}}{2}, \\ M(0) &= \left.\frac{d\lambda(\tau)}{d\tau}\right|_{\lambda=i\omega} = \frac{\omega_0 k_0 \rho_1(a+(n-1)b)}{1-\pi i} \end{aligned} \quad (23)$$

Now we can state the main results of this section.

Theorem 3:

In the parameter formulas (22), μ_2 determines the direction of the Hopf bifurcation: if $\mu_2 > 0 (< 0)$, then the Hopf bifurcation is supercritical (subcritical) and the bifurcating periodic solutions exist for $k > k_0^* (< k_0)$; β_2 determines the stability of the bifurcating periodic solution: the solutions are orbitally stable (unstable) if $\beta_2 < 0 (> 0)$; and T_2 determines the period of the bifurcating periodic solutions: the period increases (decreases) if $T_2 > 0 (< 0)$.

4 A numerical example: proportionally – fair congestion controller with REM

Random Early Marking (REM) marks a pachet with probability $(1-e^{-mW})$ if it finds a workload w already present in the virtual queue, where m is a positive constant. Using a reflected Brownian motion approximation [1], this can be viewed as a mechanism with the following marking function:

$p(x)=\dfrac{m\sigma^2 x}{m\sigma^2 x+2(c-x)}$. Here, σ^2 denotes the variability of the traffic at the packet level and C is the capacity of the virtual queue. We assume $m\sigma^2=0.5$ and let the capacity of the virtual queue be 1Mbps and the round trip delay be 40 ms. Let one round trip time be the unit of time. If the packets are 1000 bytes each, then the virtual queue capacity can be equivalently expressed as $c=5$ packets per time unit. Let the increase parameter $w=1$ and $f(x)=xp(x)$ where $p(x)=\dfrac{x}{20-3x}$. The equilibrium rate can be found by solving $x^* p(x^*)=1$. If $a=1, b=0.5, n=10$ yielding $x^*=1.65360$. With Theorem 1, we can determine that $k_0=1.114862$, $\omega_0=1.57079$. It following from (22) $\mu_2=0.5831$ (supercritical), $T_2=-0.8750$ (decreasing), $\beta_2=-1.4389$ (stable). These calculations prove that the system equilibrium x^* is stable when $k<k_0$, the critical value $k_0=1.114862$, x^* loses its stability and a Hopf bifurcation occurs, i.e, a family of periodic solutions bifurcate out from x^*.

With the help of a program in Maple 8, the following values were obtained for $x^*=1.65360$ as well as the following graphics: Fig.1 represents the orbit $(t, x(t))$, Fig.2 represents the graphics $(x(t), x(t-\tau))$ and Fig.3 represents the cycle orbit on the extended center manifold $W^c_{loc}(0)\times R$.

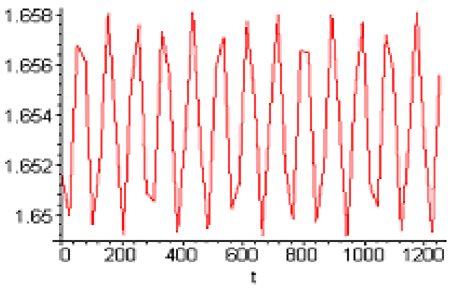

Fig.1 The orbit $(t, x(t))$

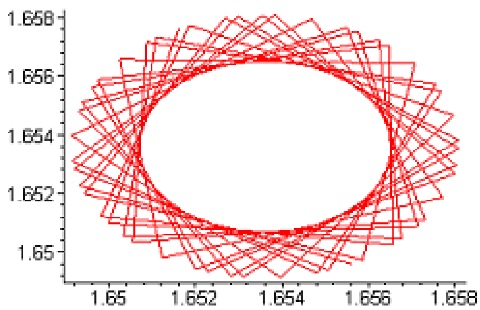

Fig.2 The orbit $(x(t), x(t-\tau))$

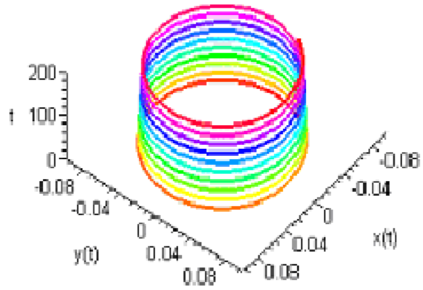

Fig.3 The dynamics of the system on $W_c(0)\times R$

 # Sequence of Program in Maple 11 for Internet congestion control model.

```
> a:=1.;b:=0.5;n:=10;tau:=1.;w:=1.;

>f1(x):=x*(0.06*x0.2*x^2+0.05*x^3):f2(x):=x^2/(2
0-3*x):

> eq:=(a+(n-1)*b)*f2(x)-w; sols:= [solve (eq,x)];
x0:=sols[2];

>D1:=diff(f2(x),x):D2:=diff(f2(x),x,x):D3:=diff(f2(x
),x,x,x):ro1:=eval(D1,[x=x0]);ro2:=eval(D2,[x=x0]);
ro3:=eval(D3,[x=x0]);k0:=evalf(Pi)/(2*tau*ro1*(a+(
n1)*b));omega:=evalf(Pi)/(2*tau);

> g20:=2*k0*ro2*(a+(n-1)*b) /(1evalf(Pi)*I) :g11:
=-2*k0*ro2*(a+(n-1)*b)/(1-evalf(Pi)*I) :g02:= 2*
k0*ro2* (a+(n-1)*b)/(1-evalf(Pi)*I): g21:=-2*k0
*(a+(n-1)*b) /(1-evalf(Pi)*I) *(ro2*I*((3*g20-
conjugate(g02) +3* (g11 +conjugate(g11))) /(3*
omega) +ro2*(a+(n-1)*b) *(b-a)/(ro1((n-1)*b^2-
a*(a+(n-2)*b)))-ro2* (a+(n-1)*b)*(k0*ro1 *(a-b)-
2*I*omega)/((n-1)*k0*ro1^2*b^2-(k0*ro1 *a-2*
omega*I) * (k0* ro1*(a+(n-2)*b)-2*omega *I)))-
ro3):

> v20:=-g20/(omega*I)- conjugate (g02) /(3*
omega*I) +ro2* (a+(n-1)*b) * (k0*ro1 *(a-b)-
2*omega*I)/ ((n-1)*k0*ro1^2*b^2-(k0* ro1*a-2*
omega*I)*(k0*ro1*(a+(n-2)*b)-2* omega *I))
:v11:=(g11+conjugate(g11))/(omega*I)+ro2*(a+(n-
1)*b)*(b-a)/(ro1*((n-1)*b^2-a*(a+(n-2)*b))) :
w20:= (3*g20-conjugate(g02)) /(3*omega) -(ro2
*(a+(n-1)*b)*(k0*ro1*(a-b)-2*omega*I)/((n-1) *k0
*ro1^2*b^2-(k0*ro1*a-2*omega*I) *(k0*ro1
*(a+(n-2)*b) -2*omega*I))) :w11 := (g11+
conjugate(g11)) /omega+ro2* (a+(n-1)*b) *(b-a)/
(ro1*((n-1)*b^2-a*(a+(n2)*b))):

> M:=omega*k0*ro1*(a+(n-1)*b)/(1-tau*I): C1:=
(g20*g11-2*abs(g11)^2 -abs(g02)^2/3) *I/ (2*
```

omega) +g21/2: mu2:=Re(C1)/Re(M); T2:=-(Im(C1)+mu2*Im(M))/omega;beta2:=2*Re(C1);

>r20:=Re(v20):i20:=Im(v20):r11:=Re(v11):i11:=Im(v11):r220:=Re(w20):i220:=Im(w20):r211:=Re(w11):i211:=Im(w11):R20:=Re(g20):R11:=Re(g11):R02:=Re(g02):I20:=Im(g20):I11:=Im(g11):I02:=Im(g02):R21:=Re(g21):I21:=Im(g21):

>F1(x(t),y(t)):=omega*y(t)+(R20/2+R11+R02/2)*x(t)^2- (I20-I02)* x(t)*y(t)-(R20/2-R11 +R02/2) *y(t)^2+R21*x(t)*(x(t)^2+y(t)^2)/2-I21* y(t) *(x(t)^2+y(t)^2)/2: F2(x(t),y(t)) :=omega *x(t) +(I20/2+I11+I02/2) *x(t)^2-(I20/2-I11+I02/2) *y(t)^2 +(R20R02)*x(t) *y(t)+R21*y(t)* (x(t)^2 +y(t)^2)/2+I21*x(t)*(x(t)^2+y(t)^2)/2:F3(x(t),y(t)) := 2*x(t)+r20*(x(t)^2-y(t)^2)- 2*i20*x(t) *y(t) +r11* (x(t)^2+y(t)^2)+x0: F4(x(t) ,y(t)) :=2*x(t)* cos(omega*tau)+2*y(t)*sin(omega*tau)+r220*(x(t)^2-y(t)^2) -2*i220*x(t) *y(t)+r211 *(x(t)^2 +y(t)^2)+x0:
with(DEtools):

> dsys := {diff(x(t),t)=F1(x(t),y(t)), diff(y(t),t) =F2(x(t),y(t)), x(0)=-0.001, y(0)=0.002}:

> dsol := dsolve(dsys,numeric):

> plots[odeplot](dsol,[[t,F3(x(t),y(t)) ,color=red]] ,0..250*tau, title= "Fig.1. The orbit(t, x(t)) ") ; plots[odeplot](dsol,[[F3(x(t),y(t)),F4(x(t),y(t)) , color=red]] ,0..250*tau,title= "Fig.2.The Poincare Application (x(t),x(t-tau)) ");

> OSCEqns := [diff(x(t),t) =-omega*y(t)+ (R20/2 +R11+R02/2)*x(t)^2-(I20-I02)* x(t)*y(t)-(R20/2-R11+R02/2)* y(t) ^2+R21 *x(t) *(x(t)^2 +y(t)^2) /2-I21*y(t) *(x(t) ^2+y(t)^2)/2, diff(y(t),t)= omega * x(t)+ (I20/2+I11+I02/2) *x(t)^2-(I20/2-I11+I02/2)*y(t)^2+(R20-R02) *x(t) *y(t)+R21* y(t)*(x(t)^2+y(t)^2)/2+I21*x(t)*(x(t)^2+y(t)^2)/2]:
DEplot 3d (OSCEqns, {x(t),y(t)}, t=0..40*tau,[[x(0)=-0.05, y(0)=0.05]],x=-0.1..0.1, y=0.1..0.1, scene=[x(t),y(t),t],stepsize=0.05, linecolour=t, title= "Fig.3.The dynamics of the system on Wc(0)XR");

5 Congestion Control

Originally, Cerf and Kahn assumed that the retransmissions mechanism would be rarely used.
This belief was based on the experience with the early ARPANET, which consisted of fairly homogeneous set of links and hosts, interconnected in a well-thought out fashion, and with excess capacity. TCP constants were well tuned to this particular network. However, this assumption broke down as soon as the network technologies used in the Internet became more heterogeneous, and its growth made careful and controlled design impossible. As a result, congestion started to be a serious problem, and retransmissions became more frequent.

TCP`s congestion control mechanisms were first implemented in 1987. Since then, they have undergone several changes, as more became known about their performance in various types of networks. In this section, we present the mechanisms implemented in TCP, discuss their evolution, and describe the differences between the various versions.

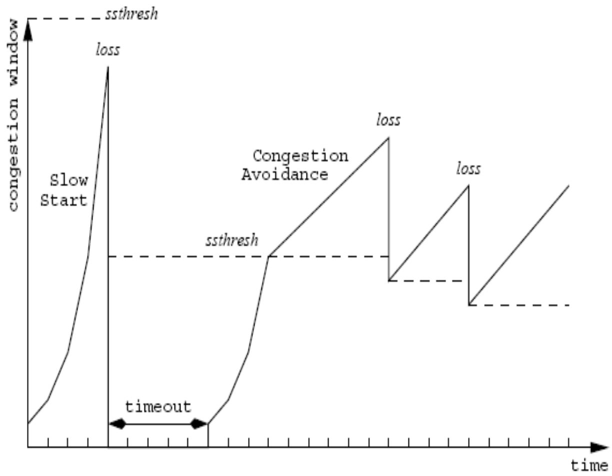

Fig.4. TCP congestion control mechanism in action

6 Active Queue Mangement

In this section we discusse active queue management mechanisms which have been specifically proposed to handle Transmission Control Protocol (TCP) traffic in the network.

The Random Early Detection (RED) mechanism is currently the recommended active queue management mechanism for the Internet. The goal of RED is to provide early feedback to end hosts about congestion at a router port, through dropping packets before the port buffer is actually full. Arriving packets are dropped based on the average queue occupancy computed using an Exponential Weighed Moving Average (EWMA) according to a random drop function. At each packet arrival, if the queue is non-empty, the average queue size is update as follows:

$$q_{avg} = (1-w)q_{avg} + wq \qquad (24)$$

where q_{avg} is the average queue size, w is the

EWMA weight, and q is the current instantaneous queue length. If the queue is empty, the average queue size is exponentially decayed depending on the time since the queue went idle, as follows:

$$q_{avg} = (1-w)^m q_{avg} \quad (25)$$

where $m = \dfrac{idle\ time}{typical\ packet\ \ transmission\ time}$

estimates the number of typical size packet (e.g. 500bytes) that would have transmitted during the time where the queue was idle. The computed average queue size is used to determine the probability with which the packet should be dropped.

Parameter	Recomanded value	Comments
w	0.002	Set to small value to filter instantaneous variations
C	5 packets	Set depending on desired queue size
max_{thresh}	15 packets	Should be three times min_{thres}
max_p	0.1	Originally recomanded 0.02, later found to be too small. This sets a target loss rate of 10 %.

Table 1: Recomanded settings of RED parameters

The drop probability is 0 when the average queue size is below a threshold (min_{thresh}). Then, it is a simple linear function which increases from to as the average queue size increases from to another threshold (max_{thresh}). When the average queue size is larger than, the drop probability is set to 1. However, these settings have been shown to give poor performance in many situations. In fact, RED is notoriously hard to configure to improve over the performance of drop tail queues.

The EWMA filter has for purpose to take into account the history of queue occupancy as opposed to the instantaneous queue size. This allows occasional short bursts to be admitted, while ensuring that the average queue size is small. The random drop function is designed to distribute the loss among connections in proportion to their bandwidth. Thereby, RED is supposed to address problems which were observed in simulations with Tail-drop queues, namely: a bias against bursty traffic, and the synchronization of connections caused by simultaneous packet loss. Not only has the ability of RED to address these problems been questioned, but also their existence in the Internet is being challenged by recent findings. In particular, synchronization occurs when a limited number of long transfers share a bottleneck, and congestion causes all the connections to backoff. However, it is nonexistent when a large number of random size transfers are present.

RED has been the subject of a large number of performance studies, which produced a number of variants on the original scheme. For example, a simulation study found that RED does not result in a balance sharing of the bandwidth. In particular, by distributing the packet loss across all connections, it penalizes “fragile” flows, e.g. flows with long round trip times and/or small windows, which cannot efficiently recover from loss. This work proposes the use of a scheme, called Flow RED (FRED), which keeps of the buffer usage of active flows. The drop function applied to each flow is then made to depend on its buffer usage. In addition, experimental work has shown that RED does not perform better than Tail-drop. For a large number of connections, the router queue length was found to stabilize around the max_{thres}, which means that a RED queue behaves like a Tail-drop queue with a size equal to that threshold. Conversely, when in under-utilization of the link. A self-configuring Adaptive RED gateway (ARED) was proposed to address these limitations. ARED dynamically modifies as the average queue size changes. Thus, when the queue size falls below min_{thresh}, is decreased (e.g., divided by 3), and when the queue size exceeds max_{thresh}, max_p is increased (e.g., multiplied by 2). Otherwise, is not changed. This scheme attempts to keep queue size between min_{thresh} and max_{thresh}, where the random drop operates as designed, and is claimed to avoid the problems above.

A typical shortcoming with RED performances studies has been the focus on large file transfers and network oriented performance measures. This leaves a gap in our understanding of the performance of RED in the Internet, given the predominance of Web traffic in the Internet. Not until recently did the results of a study of short transfers become available. This study consisted of an experimental setup with a foxed network topology, and a large number of simulated HTTP users. The main observations were that RED had minimal effect on HTTP response times, and that these times were largely insensitive to RED parameters, unless the link is very highly loaded (more than 90 %). This high load range was the only where RED could improve the performance compared to Tail-drop. However, the improvements were obtained at the expense of long-lived connections, and involved an exhaustive trial and error process. In, extensive simulations are used compare the *user-perceived* performances of applications when RED and Drop Tail queues are used in the network. The results show no compelling evidence to support the claim that RED improves on the performances of Drop Tail.

7 TCP Apllication Characteristics and Requirements

TCP applications account for the large majority of today`s Internet traffic. A measurement study of several Internet backbone links conducted in 1997 found that close to 95% of traffic is carried by TCP, with about 75% consisting of HTTP (Web) traffic alone. These measurements have since been corroborated by other studies, such as, which confirm the preponderance of TCP traffic in the Internet. While the proportion of traffic using UDP may increase as streaming multimedia applications get deployed, TCP traffic is expected to retain the majority share for the near future. In fact, firewalls are often configured to filter UDP traffic, forcing all applications (including streaming multimedia) to use TCP for transport.

While the term "quality of service" is commonly used in reference to real-time, streaming application traffic (e.g., voice and video), the predominance of data applications in today`s Internet, and their importance in our daily life, behoove us to ensure an appropriate quality of service for these applications as well. Indeed, stringent quality requirements are not restricted to voice and video applications. Until recently, the norm has been to consider that TCP applications do not have such requirements, and are content with "qualitative" rather than "quantitative" service. While this used to be true to a certain extent for the main "traditional" data applications, namely file transfer and Email, this can no longer be satisfactory as an approach to servicing all of today's data applications. Clearly, there is now a wide variety of popular data applications, which differ considerably in their characteristics, requirements and importance to the users. Moreover, it is important to take into account that the Internet has evolved from experimental network, where user expectations are modest, to a commercial network where paying users have increasingly higher expectations. This has placed higher de-facto requirements for all applications, including traditional ones, such as Email, which is now widely used as a critical communication tool. In addition, the nature and importance of the transactions for many of today's data applications require fast response time. For example, business transactions over the Web (e.g., stock trading), remote login and interactive data applications in general, have requirements which go beyond reliability to include low transaction delay. Finally, the attractive properties of the TCPIP protocol suite, and the availability of inexistence equipment and trained personnel are driving it into new applications areas which require very high performance, such as strong area network (SANs). Although TCP normally provides adequate performance to the different applications, we all experienced the severe quality degradation that befalls interactive TCP applications, such as Telnet and the Web, during network congestion episodes. Indeed, interactive data applications have requirements comparable to those of real-time applications, and therefore need similar care in the network. For example, the interaction of TCP's reliability and congestion control mechanisms with packet loss in the network was shown to result in poor performance for such applications. As a first step towards insuring optimal user perceived performance of TCP applications at all times, it is therefore important to understand the requirements of such applications, their working details, and their behavior in the current "best effort" Internet. In the following sections, we first summarize the results of traffic measurement studies, which shed some light on TCP traffic composition, and provide models for the general traffic characteristics of the different applications. Then, we attempt to classify the different popular applications into interactivity classes according to their delay requirements. Finally, we describe in detail the characteristics and requirements of three popular applications, namely Telnet, Web and FTP, which are representative of the different classes of interactivity.

The composition of traffic per application is shown in Table 2, based on measurements from the MCI Internet backbone, published in. As indicated earlier, the largest portion belongs to the Web application (75%), with server generated traffic amounting to about 68% of the total, owing to the asymmetric nature of the Web client-server interaction. However, not all HTTP traffic is interactive. Indeed, HTTP is used both for the transfer of interactive Web pages and, as an alternative to FTP, for the transfer of large documents and multimedia files over the Internet. Unfortunately, the figure above does not show how this portion is divided among interactive Web transfers and non-interactive ones. However, recent measurements seem to indicate that a limited number of long transfers (e.g., 20% of flows) account for the majority (e.g., 60%) of the traffic. In addition, these measurements show that the emergence of new applications, such as peer-to-peer file sharing, can significantly alter the breakdown of traffic by application on some newsgroups and Telnet, which generate a small but measurable amount of traffic.

Considering the traffic share of each application, it is evident from the average flow statistics shown in Table 2 that the characteristics of application flows differ considerably. For example, according to these figures, an average HTTP server flow lasts 12 seconds, and generates 10, 000 bytes. In contrast, an average Telnet

Flow lasts 10 to 20 times longer, but generates 2 to 5 times less traffic. A great deal of effort at characterizing the different TCP applications, based on real Internet measurements, has been spent over the last decade. While most have focused on Web traffic, for obvious reasons, several studies conducted before the explosion in WWW traffic, address the basic characteristics of the other popular applications and attempt to represent them with analytical models. These models capture the distributions of the random variables, such as bytes or packets transferred and session duration, associated with each application. Table 2 summarize this information. Most parameters have distributions that exhibit decaying tails, such as lg-normal, lg-extreme and Pareto, which means that very large values of these parameters are common.

	Share			Flow statistics		
Application	Bytes	Packets	Flows	Duration	Bytes	Packets
Web server	68%	40%	40%	12 sec.	10,000	16
Web client	7%	30%	35%	12 sec.	1,000	15
E-mail	5%	5%	3%	-	1,500-2,000	-
FTP data	5%	3%	1%	20-500 sec.	200,000	-
NNTP	2%	1%	1%	100-500 sec.	50K-300K	200-800
Telnet	1%	1%	1%	100-500 sec.	2,000-5,000	100
Other	6%	20%	19%	-	-	-

Table 2: Average traffic share and flow statistics per TCP application

The lg-normal distribution is defined as follows. A random variable X is said to have a l-normal distribution if the random variable $Y = \lg X$ has a normal distribution. The probability density function of the normal distribution with mean μ and variance σ^2 is:

$$P(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \qquad (26)$$

The lg-extreme distribution is defined as follows. A random X is said to have a lg-extreme distribution if the random variable $Y = \lg X$ has an extreme distribution. The cumulative distribution function $(F(y) = P\{Y \le y\}$ of the extreme distribution whith location parameter α and shape parameter β is as follows:

$$F(y) = e^{-e^{-\frac{(x-\alpha)}{\beta}}} \qquad (27)$$

The probability density function of the Pareto distribution with location parameter and shape parameter β is:

$$P(x) = \frac{\beta\alpha^{\beta}}{x^{\beta+1}} \qquad (28)$$

and its cumulative distribution function is:

$$F(x) = 1 - \left(\frac{\alpha}{x}\right)^{\beta} \qquad (29)$$

The Pareto distribution has infinite variance when $\beta \le 2$ and infinite mean when $\beta \le 1$.

Application	Variable	Model	Parameters
E-mail	Sender bytes	lg-normal	$\bar{x} = 2^{10}$, $\sigma_x = 2.75$
NNTP	Sender bytes	lg-normal	$\bar{x} = 11.5$, $\sigma_x = 3$
Telnet	Client bytes	lg-extreme	$\alpha = \lg 100$, $\beta = \lg 3.5$
	Server bytes	lg-normal	$\bar{x} = 4{,}500$, $\sigma_x = 7.2$
	Duration (seconds)	lg-normal	$\bar{x} = 240$, $\sigma_x = 7.8$

Table3: Summary of analytic models for TCP applications.

Component	Model	Parameters	Probability Density function
Transfer bytes-body	lg-normal	$\mu = 9.36, \sigma = 1.32$	$p(x) = \frac{1}{1.32x\sqrt{2\pi}} e^{-(\ln x - 9.36)^2/3.48}$
Transfer bytes-tail	Pareto	$\alpha = 133K, \beta = 1.1$	$p(x) = 1.1(133{,}000)^{1.1} x^{-2.1}$
Popularity	Zipf		
Request size bytes	Pareto	$\alpha = 1K, \beta = 1$	$p(x) = 1{,}000 x^{-2}$
No of embedded files	Pareto	$\alpha = 1, \beta = 2.43$	$p(x) = 2.43 x^{-3.43}$

Table4: Models and parameters for WWW traffic.

8 TCP Apllication Requirements

As shown in Tables 1 and 3, popular TCP applications have greatly different characteristics. These applications also have widely requirements, which can be classified along two axes: bandwidth and delay. Fig.5 attemps to illustrate the large spectrum of such requirements are applications such as remote login (Telnet and secure login ssh), where the generated traffic is of low bandwidth , but very low perpacket delays and required. At the opposite end are applications that generate large amounts of bulk traffic, with relaxed timing requirements, a typical example of which is system backups. Between these two extremes are applications which have low delay and moderate bandwidth requirements, such as Web downloads and Email. Other applications have low delay requirements along with moderate to high bandwidth requirements, such as real time gaming and remote graphical desktop access.

Clearly, the transactions associated with the different applications occur at different time scales. We classify TCP applications based on the level of interactivity they involve. For interactive TCP applications when bandwidth requirements are not satisfied, i.e. when the network is congested, packet loss typically occurs. Loss is then translated by TCP`s reliability mechanisms to delay in completing the transactions at hand. It is therefore possible to focus on the delay requirement alone when studying the performance of such applications. Delay refers to the time spent in one transaction the definition of which varies per applications.

Thus, an HTTP transaction consists of the download of a Web page, which might include several embedded components, requiring distinct transfers. Similarly, an FTP transaction consists of the transfer of one file from the server to the client or vice-versa. Naturally, an Email and NNTP transaction is considered to be the exchange of an email or news message between end users and the appropriate server, or between two servers.

Finally, a Telnet transaction time consists of the delay between typing a character at the terminal side, and the reception of the corresponding echo generated by the server.

In general, a faster response or task completion time is preferable regardless of the application. It could arguably be possible to provide very low response times to all applications by investing the necessary amount of resources in user nodes and in the network.

However, this approach can be cost prohibitive or even impossible in some situations, e.g. when the resources are naturally limited, such s on the wireless communication medium. Instead, an adequate user-perceived performance may be provided for all applications, at reasonable cot, by prioritizing according to their importance to the user and ensuring that delay requirements of the most important applications are satisfied.

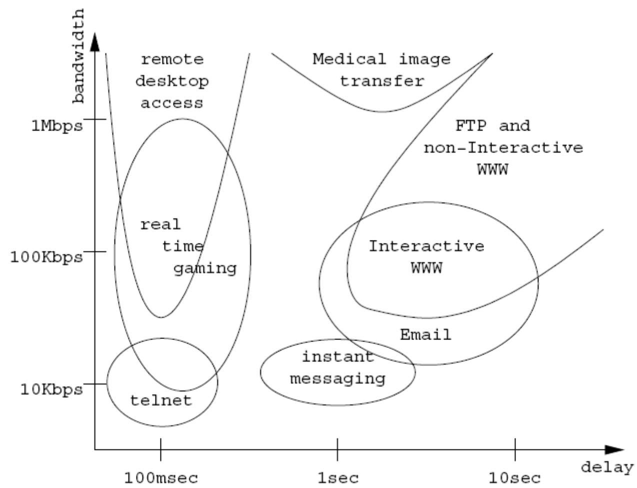

Fig.5. Approximate bandwidth and delay requirements for popular TCP applications

9 Conclusion

In this paper, an Internet congestion control system model with $n(n \geq 3)$ links and single source has been studied. By using the positive gain parameter as a bifurcation parameter, we have shown that a Hopf bifurcation occurs in such an Internet congestion control model, yielding a family of periodic orbits bifurcating out from the network equilibrium. Simulation results have verified and demonstrated the correctness of the theoretical results.

In this document, we presented TCP`s mechanisms for reliable data transfer, as well as the various versions of its congestion control mechanisms. In addition, we summarized the mian results obtained in the areas of TCP performance evaluation and active queue management. We also discussed the use of TCP by popular applications, namely Telnet, Web and FTP, and presented the characteristics of these applications. We closed with a summary of TCP modeling efforts and recommendations on the use of TCP in simulations.

In this document, our goal is to help readers working with TCP to avoid some of the errors, pitfalls and confusion that result from the large number of different versions and modifications of TCP.

We note that TCP is a highly fluid protocol, particularly when the details of its operation are considered. Many non-standard modifications and enhancements are independently added to the various popular implementations. In addition, given the complexity of the protocol, as well as some imprecision in the specifications, many implementors allow themselves the freedom to deviate from the standard behavior, in the benefit of simplicity or inter-operability with other existing implementations. Therefore the information contained in this document may not apply to every single TCP implementation or version.

References:

[1] Athuraliya S., Lapsley D.E, Low S.H, *Random early marking for Internet congestion control*. Proc. IEEE GLOBECOM nr. 3, pp. 1747-1752, 1999;

[2] Barford P., Crovella M., *Criticall Path Analysis of TCP Transactions,* IEEE Transactions on Networking, Vol. 9, Nr. 3, June 2001;

[3] Hassard B.D., Kazarinoff N.D., Wan Y.H., *Theory and applications of Hopf bifurcation*, Cambridge University Press, Cambridge, 1981;

[4] Kelly F.P., *Models for a self-managed Internet*, Philos. Trans., Roy Soc. A nr. 358, pg. 2335-2348, 2000;

[5] Mircea G., Neamțu M., Opriş D., *Dynamical systems in economics, mechanics, biology described by differential equations with time delay,* Mirton Publishing House, Timişoara, 2003;

[6] Mircea G., Neamțu M., Opriş D., *Hopf bifurcations for dynamical systems with time delay and application* , Mirton Publishing House, Timişoara, 2004;

[7] Mircea G., Neamțu M., Opriş D., *Systems of harmonic oscillators with time delay and applications,* 5th International Conference on Nonlinear Problems in Aviation and Aerospace "Mathematical Problems in Engineering and Aerospace Sciences" ICNPAA 2004, Cambridge Scientific Publishers Ltd., ISBN:1-904868-48-7, pp.475-485.

[8] Mircea G., Opriş D. *Nonlinear effects in a discrete-delay dynamic model for a stock market*, 8th International Congress on Insurance: Mathematics & Economics, June 14-16, Rome, 2004.

[9] G. Mircea, M. Muntean "*Nonlinear Analysis of Network Traffic*", The Central and East European Conference in Business Information Systems 2004, CECBIS 2004, Cluj-Napoca, Romania 2004, Ed. Risoprint, ISBN: 973-656-648-X, pp. 587-602.

[10] Mircea G., Muntean M.- *Numerical simulation for the Hopf bifurcation in TCP systems with continuous distributed network delays* – WSEAS Inernational Conference on EUROPEAN COMPUTING CONFERENCE, Athens, Greece, 2007, Springer Verlag Proceedings.

[11] Paxson V., Growth trends in Wide Area TCP Connections, in IEEE Network, Aug. 1994;

[12]Udriste C. – *Lagrangian Constructed from Hamiltonian Systems,* Proceedings of 9th WSEAS International Conference on MATHEMATICS & COMPUTERS IN BUSINESS & ECONOMICS, Bucuresti, Romania, June 24-26, 2008, pag. 30-34.

[13] Voicu M.C. – *Computational Implementation Methods used to Detect the Dynamics of Nonlinear Determinist Discrete Systems,* European Computing Conference, Vouliagmeni, Athens, Greece, September 25-27, 2007, Springer Verlag Proceedings.

[14] Voicu M.C. -*Analytical results and numerical simulations for k-order nonlinear discrete determinist exchange rate models*, Proceedings of 9th WSEAS International Conference on MATHEMATICS & COMPUTERS IN BUSINESS & ECONOMICS, Bucuresti, Romania, June 24-26, 2008, pag. 135-140.

[15] Voicu M., Mircea G.- *Algorithms for exploiting multidimensional database*s – WSEAS Transactions on Information Science and Applications, Issue 12, Volume 2, December 2005, ISSN 1790-0832, pp. 2176-2183.

[16] Yeom I., Narashinka R., – Modelling TCP behavior in a Differentialated Services Network, TAMU ECE Technical Report, May 1999

[17] Zhang L., Shenker S., Clark D. – *Observations on the Dynamics of a Congestion Control Alghorithm: The Effects of two-way Traffic*, in Proceedings of SIGCOMM, September 1991.

[18] Zhang L., Shenker S., Clark D. – Why TCP Timers Don't work well, in Proceedings of SIGCOMM'86, August 1986.