HAL
open science

TCP and Network Coding: Equilibrium and Dynamic Properties

Hamlet Jesse Medina-Ruiz, Michel Kieffer, Beatrice Pesquet-Popescu

To cite this version:

Hamlet Jesse Medina-Ruiz, Michel Kieffer, Beatrice Pesquet-Popescu. TCP and Network Coding: Equilibrium and Dynamic Properties. 2013 International Symposium on Network Coding (NetCod), Jun 2013, Calgary, Cameroon. pp.1-6, 10.1109/netcod.2013.6570821 . hal-00819495

HAL Id: hal-00819495

https://centralesupelec.hal.science/hal-00819495v1

Submitted on 1 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TCP and Network Coding Equilibrium and Dynamic Properties

Hamlet Medina Ruiz*†, Michel Kieffer*†‡, and Béatrice Pesquet-Popescu†

*L2S - CNRS - SUPELEC - Univ Paris-Sud, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France
†Institut Télécom, Télécom ParisTech and CNRS LTCI
Signal and Image Processing Department, 46 rue Barrault, 75634 Paris Cedex 13, France
‡Institut Universitaire de France

***Abstract*—This paper analyzes the impact on the TCP-Reno congestion control mechanism of a network coding (NC) layer inserted in the TCP/IP stack. A multi-source multi-link model is considered to study the equilibrium and dynamic properties of the TCP-NC protocol with RED as active queue management mechanism. The existence and uniqueness of some network equilibrium is demonstrated and characterized in terms of average throughput, loss rate, and queue length. Global stability is proved in absence of forward delay, and the effects of the NC redundancy factor and of the delay on the local stability of TCP-NC-RED is studied around the equilibrium point. Results provided by the proposed model are compared to those obtained by simulation for N sources sharing a single link. TCP-NC-RED becomes unstable when delay or capacity increases, as TCP-Reno does. Its stability region is characterized as a function of the redundancy factor.**[1]

***Index Terms*- congestion control, network coding, queue management, stability.**

I. Introduction

In [11], a new TCP-friendly protocol is proposed that interfaces network coding (NC) with TCP by introducing a new coding layer in the TCP/IP stack between TCP and IP. The idea is to benefit from NC to improve TCP throughput. In this layer TCP segments are encoded at the sender and decoded at the receiver. In [10] the authors additionally describe several important practical aspects of incorporating NC below TCP-Reno [1], such as the TCP-compatible sliding window code, and new rules for acknowledging bytes to the TCP layer. Redundant packets are transmitted by the sender to mask non-congestion losses, *e.g.*, losses on the wireless part of the network, from TCP congestion control. Simulation results in [10], [11] show an improvement in TCP throughput for a constant value of the redundancy factor ρ. The purpose of ρ is to match the rate at which data are obtained at the receiver to the sending rate of TCP. If ρ is too small, the losses are not masked from TCP and the NC decoding probability decreases. On the other hand if ρ is too large, the congestion may increase, reducing the throughput. Thus, an adaptation mechanism for ρ has been proposed in [8].

This paper characterizes the effect of NC on the equilibrium and local stability of TCP-Reno. The equilibrium or average throughput has been characterized in [4], where a framework is proposed to predict TCP-NC throughput when Vegas is used as TCP congestion control. Here, we consider the tools from control and optimization theory used to characterize TCP [2], [5], [7], [12] to study TCP-NC as a distributed primal-dual algorithm that maximizes some aggregate utility function.

In what follows, TCP-NC is briefly described in Section II, details about the network model and properties of the TCP-NC protocol are given in Sections III and IV, respectively. Our model is evaluated in Section V and some conclusions and futures research directions are drawn in Section VI.

II. Principles of the TCP-NC protocol

The TCP-NC protocol introduces a NC layer between the TCP and IP layers in the TCP/IP protocol stack [11]. The sender stores packets generated by TCP in a *coding buffer*. For every arrival from TCP, it transmits ρ random linear combinations of the most w recently arrived packets in the coding buffer. To convey information about the linear combination, an NC header is added to the coded packets, which contains information about the coefficients used to mix the packets, the first and last byte SEQ of each packet involved in the mix, and the first byte that has not been acknowledged. The original packets remain in the coding buffer until an appropriate TCP ACK arrives from the receiver side.

On the receiver side, upon receiving a packet containing a new linear combination, the decoder places it in the decoding buffer, appends the corresponding coefficients to the decoding matrix (which contains the coefficients used to mix already received packets), and performs Gaussian elimination. This process helps determining whether the received packet contains new information. Then the receiver sends a TCP cumulative ACK to the sender requesting the first unseen packet in order, like in conventional TCP. The Gaussian elimination may result in a new packet being decoded. In this case the decoder delivers this packet to the receiver TCP. Any ACK generated by the receiver TCP is suppressed and not sent to the sender. These ACKs may be used for managing the decoding buffer. An important point is that the introduction of the new NC layer does not require any change in the basic features of TCP. For more detail, see [10].

[1]This work has been partly supported by DIM-LSC SWAN.

III. Dynamic Model

In this section we develop a model of TCP-NC/RED to study the equilibrium and dynamic properties of the TCP-NC protocol. We start with a non-linear model, study its equilibrium properties in absence of forward delay using optimization theory [2], [6], [12], and then linearize the model around its equilibrium to study the effect of the delay and of the redundancy factor ρ on the stability of the system.

A. Network Model

The model introduced in [6] is used to describe the network. A network is represented by a set of $|\mathcal{L}| = L$ links, indexed by $\ell = 1, \ldots, L$, with finite capacities c_ℓ. They are shared by a set of $|\mathcal{S}| = S$ sources indexed by i. Each source i uses a set $\mathcal{L}_i \subseteq \mathcal{L}$ of links to convey information through the network to some receiver. The set $\mathcal{L}_i$ defines an $L \times S$ routing matrix R, with $R_{\ell i} = 1$ if $\ell \in \mathcal{L}_i$ and $R_{\ell i} = 0$ otherwise. Following the notation in [6], each source i has an associated TCP congestion window size $w_i(t)$, a NC redundancy factor ρ_i, and an aggregate congestion measure or *aggregate price* $q_i(t)$. Here the redundancy factor ρ_i is introduced to account for NC layer. Each link ℓ has a congestion measure or *price* $p_\ell(t)$ at time t.

Define the round trip time (RTT) $\tau_i(t)$ for source i as

$$\tau_i(t) = d_i + \sum_\ell R_{\ell i} \frac{b_\ell(t)}{c_\ell}, \tag{1}$$

where d_i is the round-trip propagation delay assumed to be constant and $b_\ell(t)$ is the backlog (queue length) at link ℓ. The rate for the i-th source is

$$x_i(t) = \frac{w_i(t)}{\tau_i(t)}. \tag{2}$$

The aggregate input rate at link ℓ is the sum of the delayed source rates

$$y_\ell(t) = \sum_i R_{\ell i} x_i(t - \tau^f_{\ell i}), \tag{3}$$

where $\tau^f_{\ell i}$ is the forward delay from source i to link ℓ. The active queue management (AQM) [6] models describe how the congestion measure $p_\ell(t)$ at link ℓ is updated. The congestion control problem involves a continuous interaction between TCP algorithms implemented in the sources, and AQM algorithms implemented in the links. The aggregate congestion measure of user i (chosen here as the end-to-end loss probability) is

$$q_i(t) = \sum_l R_{\ell i} p_\ell(t - \tau^b_{\ell i}). \tag{4}$$

It is the sum of delayed link loss probabilities, where $\tau^b_{\ell i}$ is the backward delay from link ℓ to source i. The RTT time is the sum of the forward and backward delays

$$\tau_i(t) = \tau^f_{\ell i}(t) + \tau^b_{\ell i}(t). \tag{5}$$

As in [5], we assume that at each link the congestion measure is modeled through a price mapping function m_ℓ that maps the queue length $b_\ell(t)$ at link ℓ to a price $p_\ell(t)$.

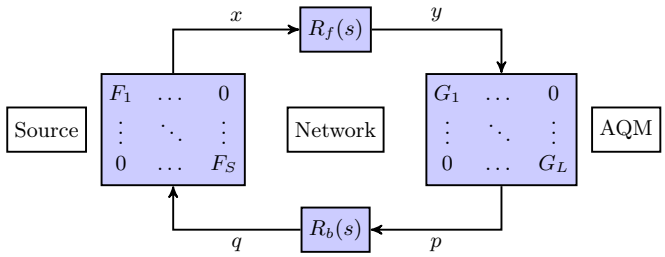


Fig. 1. Interactions between the TCP and the AQM mechanisms

Source i can observe only its own rate $x_i(t)$ and its aggregate congestion measure $q_i(t)$. Link ℓ observes only its own local congestion $p_\ell(t)$ and the aggregate flow rate $y_\ell(t)$.

TCP algorithms adjust the i-th source rate using some function F_i based on $q_i(t)$ and ρ_i (to account for the presence of the NC layer). The dynamics of the source rate is given by

$$\dot{x}_i(t) \quad = \quad F_i\left(x_i(t), q_i(t), \rho_i\right), \tag{6}$$

for all sources i.

AQM algorithms adjust link prices according to some function G_ℓ based on the aggregate link rate on link ℓ. In the network side, the evolution of the congestion of link ℓ is

$$\dot{p}_\ell(t) \quad = \quad G_\ell\left(p_\ell(t), y_\ell(t)\right). \tag{7}$$

F_i depends on the considered TCP algorithm (Reno, Vegas, etc.) and G_i depends on the AQM mechanism (FIFO, REM, RED, etc.) [6], [7]. The interaction between the control at sources and the control at links results in a feedback system, see Figure 1. This network model is used in Section III-B to derive the dynamics of TCP-NC combined with RED-AQM.

B. TCP-NC/RED-AQM Model

To describe the dynamics of TCP-NC, we focus on the congestion avoidance phase of TCP-Reno, in which a large continuous flow set up by TCP spends most of the time. We derive the fluid model approximations from the packet level description of the TCP-NC protocol and study its dynamics. At time t, $\rho_i x_i(t)$ is the rate at which packets are sent to the receiver. A fraction of these packets is acknowledged at a rate $\mathbb{E}[R_{\text{ACKs}}]$ and each ACK increments the size $w_i(t)$ of the TCP congestion window by $1/w_i(t)$. Negative acknowledgments, at an average rate of $\mathbb{E}[R_{\text{NACKs}}]$, return back to the sender, each one reducing $w_i(t)$ by $\beta w_i(t)$. The constant β is the rate at which $x_i(t)$ is decreased each time there are losses (typically $\beta = 1/2$). The dynamic of the source rate during the congestion avoidance phase of TCP-Reno with NC may be described as follows

$$\dot{x_i}(t) = \frac{1}{w_i(t)} \mathbb{E}[R_{\text{ACKS}}] - \beta w_i(t) \mathbb{E}[R_{\text{NACKS}}] \tag{8}$$

Accounting for the effects of the NC layer, $\mathbb{E}[R_{\text{NACKS}}]$ is

$$\begin{aligned}\mathbb{E}[R_{\text{NACKS}}] = x_i(t-\tau_i(t))q_i(t) - \\ (\rho_i - 1)(1-q_i(t))x_i(t-\tau_i(t)) - \\ \sum_{k=0}^{(\rho_i-1)x_i(t-\tau_i(t))-1} [k-(\rho_i-1)x_i(t-\tau_i(t))] \times \\ \begin{pmatrix} \rho x_i(t-\tau_i(t)) \\ k \end{pmatrix} (q_i(t))^k (1-q_i(t))^{\rho_i x_i(t-\tau_i(t))-k} \end{aligned} \quad (9)$$

The first term in (9) is the rate at which packets are lost; this term appears also in TCP-Reno without NC. The second and third terms in (9) are related to NC. The second is the rate at which NC compensate losses, and the third expresses the fact that $x_i(t)$ packets/s are enough to perform network decoding at the receiver. The rate at which ACKs are received is $\mathbb{E}[R_{\text{ACKS}}] = \rho_i x_i(t-\tau_i(t)) - \mathbb{E}[R_{\text{NACKS}}]$. In congestion avoidance phase, $x_i(t)q_i(t) \geq (1-q_i(t))(\rho_i-1)x_i(t)$, and (9) can be approximated as

$$\mathbb{E}[R_{\text{NACKS}}] \approx x_i(t-\tau_i(t))q_i(t) - (\rho_i-1)(1-q_i(t))x_i(t-\tau_i(t)) \quad (10)$$

Combining (8) and (10), the evolution of source rates under the control of TCP-NC (6) may be rewritten as

$$\begin{aligned}\dot{x}_i(t) = x_i(t-\tau_i(t))\left(\rho_i - 1 + (1-q_i(t))\rho_i\right)\frac{1}{w_i(t)} - \\ \beta x_i(t-\tau_i(t))\left(1-(1-q_i(t))\rho_i\right)w_i(t). \end{aligned} \quad (11)$$

To study the evolution of the queue length in each link ℓ, we consider random early detection (RED) as AQM mechanism. With RED-AQM, $b_\ell(t)$ evolves as follows [6]

$$\dot{b_\ell}(t) = (y_\ell(t) - c_\ell)^+_{b_\ell(t)} \quad (12)$$

where $(a)^+_z = a$ if $z > 0$, and $(a)^+_z = \max(0, a)$ if $z = 0$. RED averages the instantaneous queue length $b_\ell(t)$ by an exponentially weighted average $r_\ell(t)$, given by

$$\dot{r_\ell}(t) = -\alpha_\ell c_\ell (r_\ell(t) - b_\ell(t)) \quad (13)$$

for some constant $0 < \alpha_\ell < 1$. Given $r_\ell(t)$, the local congestion measure $p_\ell(t)$ at link ℓ is given by

$$p_\ell(t) = m_\ell(r_\ell(t)) = \begin{cases} 0, & r_\ell(t) \leq \underline{b}_\ell, \\ \gamma_\ell(r_\ell(t) - \underline{b}_\ell), & \underline{b}_\ell < r_\ell(t) < \overline{b}_\ell, \\ \eta_\ell r_\ell(t) - (1-2\overline{p}_\ell), & \overline{b}_\ell < r_\ell(t) < 2\overline{b}_\ell, \\ 1, & r_t(t) \geq 2\overline{b}_\ell, \end{cases} \quad (14)$$

where $\underline{b}_\ell$, $\overline{b}_\ell$, $\overline{p}_\ell$, are the parameters of the RED algorithm. Morover, $\gamma_\ell = \overline{p}_\ell/(\overline{b}_\ell - \underline{b}_\ell)$ and $\eta_\ell = (1-\overline{p}_\ell)/\underline{b}_\ell$.

IV. Properties of TCP-NC, constant redundancy

In this section we characterize some equilibrium and dynamic properties of TCP-NC. To study the equilibrium of a network characteristic (average throughput, loss rate, delay, etc.) under the TCP-NC-AQM control, we will use the fact that the equilibrium of (6) and (7) can be easily characterized by studying its underlying optimization problem, where we maximize the aggregate utility of the users subject to link capacity constrains [2]. To prove the global stability of the algorithm we assume the absence of forward delay, $\tau^f_{\ell i} = 0$. The local stability is studied by linearizing the system around the equilibrium as a function of the round-trip time and the redundancy factor.

A. Equilibrium and utility maximization framework

Let $(x_i^*, w_i^*, q_i^*, \ldots, p_\ell^*)$ denotes the equilibrium point of user i and link ℓ. It satisfies

$$\begin{cases} 0 &= F_i(x_i^*, q_i^*, \rho_i) \\ 0 &= G_\ell(p_\ell^*, y_\ell^*) \end{cases} \quad (15)$$

We consider the utility maximization framework introduced in [5], [7], where the equilibrium source rate vector x^* and the equilibrium price vector p^* of (6)-(7) are the primal and dual optimal solutions to an equivalent utility maximization problem. A fixed point of (6) defines an implicit relationship between x_i^* and q_i^*, expressed as

$$q_i^* = f_i(x_i^*, \rho_i) \quad (16)$$

where f_i is a positive, strictly monotone decreasing function [6]. This assumption allows to introduce the duality between the optimal solution of a utility maximization problem and the equilibrium of (6) and (7) by defining U_i, the utility function of user i as $\frac{\partial U_i(x_i^*)}{\partial x_i^*} = f_i(x_i^*)$. Using the assumptions on f_i, one deduces that U_i has a positive decreasing derivative and is therefore monotone increasing and strictly concave.

Now, consider the problem of maximizing the aggregate utility formulated in [2]

$$\begin{aligned} \max_{x \geq 0} \quad & \sum_i U_i(x_i) \\ \text{subject to} \quad & Rx \leq c \end{aligned} \quad (17)$$

where the constraint says that at each link ℓ the aggregate rate y_ℓ does not exceed the capacity. An optimal rate vector x^* exists since the objective function U_i is continuous and the feasible solution set is compact [6]. It is unique since U_i is strictly concave.

The key to understand the equilibrium of (6)-(7) is to regard the vectors x and p as primal and dual variables, respectively, and $(F, G) = (F_i, G_\ell, i \in \mathcal{S}, \ell \in \mathcal{L})$ as distributed primal-dual algorithms to solve the primal problem (17) and its Lagrangian dual

$$\min_{p \geq 0} \sum_i \max_{x_i \geq 0} (U_i(x_i) - x_i q_i) + \sum_\ell p_\ell c_\ell. \quad (18)$$

Hence the dual variable is a precise measure of congestion in the network. The dual problem has an optimal solution since the primal problem is feasible. The equilibrium points (x^*, p^*) of (6) and (7) are interpreted as the optimal solution of the primal-dual problem, and that (F, G) iterates on both the primal and dual variables to solve both problems. The rate vector x^* is optimal if and only if there exists $p^* \geq 0$ such that

$$\frac{\partial U_i(x_i^*)}{\partial x_i^*} = q_i^* = \sum_\ell R_{\ell i} p_\ell^* \quad (19)$$

and $y_\ell^* = \sum_i R_{\ell i} x_i^* \leq c_\ell$, with equality if $p_l \geq 0$. The last condition is the complementary slackness conditions. This condition indicates that all bottlenecks are fully utilized at equilibrium or that the AQM should match the input rate to link capacity to maximize the utilization at every bottleneck link. Any AQM algorithm that stabilizes queues possesses this property, and generates a Lagrange multiplier p^* that solves the dual problem, as is the case for classical TCP without NC [2], [6].

In summary, the utility U_i of source i is completely determined by the source dynamics F_i, *i.e.*, by the TCP algorithm. The link dynamics G_i guarantees complementary slackness condition, and the prices p^* play the role of the Lagrange multipliers of the dual problem. The strict concavity of U_i guarantees the existence and uniqueness of the optimal rates x^* of (17) and the global convergence of the algorithms, and p^* is unique when R has full row rank. Thus, (6) and (7) solve the utility maximization problem given by (17) in a distributed way. Moreover, various TCP-AQM protocols can be modeled as different distributed primal-dual algorithms (F, G) to solve the primal optimization problem (17) and its dual (18).

1) Global stability in absence of forward delay, $\tau_{\ell i}^f = 0$: Neglecting the forward delay, from the equilibrium of (11) obtained for $\dot{x}_i = 0$, one deduces the average aggregate price seen by user i

$$q_i^* = \frac{\beta\left(\rho_i - 1\right)\left(x_i^* \tau_i^*\right)^2 + 1}{\rho_i\left(1 + \beta(x_i^* \tau_i^*)^2\right)} = f(x_i^*). \tag{20}$$

The utility function of TCP-NC using (19) and (20) is

$$U_i(x_i) = \frac{1 - (\rho_i - 1)}{\sqrt{\beta}\tau_i \rho_i} \tan^{-1}\left(\sqrt{\beta}\tau_i x_i\right) + \left(\frac{\rho_i - 1}{\rho_i}\right) x_i. \tag{21}$$

This utility function differs from the one found for TCP-Reno in [2] in the amplitude of the inverse tangent and in the addition of a linear term. Both are due to the effect of the NC layer. Nevertheless, the utility for TCP-NC-Reno is still concave and yields a unique equilibrium point. The average throughput is found by considering again the equilibrium of (11), $\dot{x}_i(t) = 0$, leading to

$$x_i^* = \sqrt{\frac{\rho_i - 1 + (1 - q_i^*)\rho_i}{\beta\left(1 - (1 - q_i^*)\rho_i\right)}} \frac{1}{\tau_i^*}. \tag{22}$$

To study the global stability of TCP-NC, consider the following function used in [9]

$$V(p) = \sum_\ell \left(c_\ell - y_\ell^*\right) p_\ell + \sum_i \int_{q_i^*}^{q_i} \left(x_i^* - \left(U_i'\right)^{-1}(\sigma)\right) d\sigma. \tag{23}$$

One can show that (23) is a Lyapunov function, even when considering NC, $V(p) \geq 0$, $\dot{V}(p) \leq 0, \forall p_\ell \geq 0$. By the Lyapunov theorem one deduces that the system is globally stable in the absence of forward delay [3].

B. Dynamics of TCP-NC-RED: linearized model

So far we studied the equilibrium point of the TCP-NC model. In this section, the TCP-NC-RED equations are linearized to study the effect of the delay and of the redundancy factor around the equilibrium point. Assuming R has full rank, there is a unique equilibrium (w_i^*, q_i^*, τ_i^*). Expressing (11) as a function of the congestion window, one obtains

$$\dot{w}_i(t) = \frac{w_i(t - \tau_i^*)\left(\rho_i - 1 + (1 - q_i(t))\rho_i\right)}{\tau_i(t - \tau_i^*)} \frac{1}{w_i(t)} - \frac{\beta w_i(t - \tau_i^*)\left(1 - (1 - q_i(t))\rho_i\right) w_i(t)}{\tau_i(t - \tau_i^*)}, \tag{24}$$

where b_ℓ^* denotes the equilibrium queue lengths and $\tau_i^* = d_i + \sum_\ell R_{\ell i} \frac{b_\ell^*}{c_\ell}$ the equilibrium RTT. Let $\delta w_i(t) = w_i(t) - w_i^*$. By linearizing (24) around the equilibrium, we obtain the linearized window dynamics

$$\delta w_i(t) = -\frac{2\beta\left(1 - (1 - q_i^*)\rho_i\right) w_i^*}{\tau_i^*} \delta w_i(t) - \frac{\rho_i^2}{\left(1 - (1 - q_i^*)\rho_i\right)\tau_i^*} \delta q_i(t). \tag{25}$$

In a similar way, with $\delta p_\ell(t) = p_\ell(t) - p_\ell^*$ and by linearizing (14) around the equilibrium, we obtain the linearized price dynamics

$$\delta p_\ell(t) = -\alpha_\ell c_\ell \delta p_\ell(t) + \alpha_\ell c_\ell \gamma_\ell \delta b_\ell(t). \tag{26}$$

For the purpose of linearizing the link dynamics of RED, the non-bottleneck links can be ignored. The linearized queue length dynamics around the equilibrium RTT is given in [6]

$$\delta b_\ell(t) = \sum_i R_{\ell i} \frac{\delta w_i\left(t - \tau_{\ell i}^{f*}\right)}{\tau_i^*} - \sum_i \sum_k R_{\ell i} R_{ki} \frac{w_i^*}{(\tau_i^*)^2 c_k} \delta b_k\left(t - \tau_{\ell i}^{f*}\right), \tag{27}$$

Expressing (25), (26), and (27) in the Laplace domain, we get

$$\delta w(s) = -(sI + D_1)^{-1} D_2 R_b^T(s) \delta p(s) \tag{28}$$

$$\delta p(s) = (sI + D_3)^{-1} D_4 \delta b(s) \tag{29}$$

$$\delta b(s) = (sI + R_f(s) D_5 R^T D_6)^{-1} R_f(s) D_7 \delta w(s) \tag{30}$$

where the D_i are diagonal matrices, defined as

$$D_1 = \text{diag}\left(-\frac{2\beta(1-(1-q_i^*)\rho_i)w_i^*}{\tau_i^*}\right)$$

$$D_2 = \text{diag}\left(\frac{\rho_i^2}{(1-(1-q_i^*)\rho_i)\tau_i^*}\right)$$

$$D_3 = \text{diag}\left(\alpha_l c_l\right) \qquad D_4 = \text{diag}\left(\alpha_l c_l \gamma_l\right)$$

$$D_5 = \text{diag}\left(\frac{w_i^*}{(\tau_i^*)^2}\right) \qquad D_6 = \text{diag}\left(\frac{1}{c_l}\right) \qquad D_7 = \text{diag}\left(\frac{1}{\tau_i^*}\right)$$

and where $R_f(s)$ and $R_b(s)$ are the forward and backward delay matrices, with $[R_f(s)]_{\ell i} = e^{-\tau_{\ell i}^f s}$ if $\ell \in \mathcal{L}_i$ and $[R_f(s)]_{\ell i} = 0$ else, and $[R_b(s)]_{\ell i} = e^{-\tau_{\ell i}^b s}$ if $\ell \in \mathcal{L}_i$, and $[R_b(s)]_{\ell i} = 0$ alse.

Using (28)-(30), one can derive the loop gain $L(s) = \delta q(s)/\delta w(s)$ as

$$L(s) = R_f(s) D_7 (sI + D_1)^{-1} D_2 R_b^T(s) (sI + D_3)^{-1} \times D_4 (sI + R_f(s) D_5 R^T D_6)^{-1}. \tag{31}$$

which may be used to characterize stability.

V. Verification and stability region

In this section we present the properties of the linearized model (31) and the effect of the delay and of the redundancy on the stability region of TCP-NC.

Consider N identical sources sharing a single link $|\mathcal{L}| = 1$ with capacity c, RED as AQM, and equal redundancy factors $\rho_i = \rho$. In this case, dropping the subscripts from (28), the transfer function between the congestion window and the loss congestion measure is

$$\delta w(s) = \frac{\rho^2}{p^* z(\rho)} \frac{e^{-\tau^{b*} s}}{\tau^* s + 2\beta z(\rho) p^* w^*} \delta p(s), \tag{32}$$

where $z(\rho) = \frac{1-(1-p^*)\rho}{p^*}$. The equilibrium congestion window is given by $w^* = \frac{\tau^* c}{N}$, where τ^* is the equilibrium RTT. The equilibrium price p^*

$$p^* = \frac{\beta(\rho - 1)(w^*)^2 + 1}{\rho\left(1 + \beta(w^*)^2\right)},$$

is obtained from (20). From (28)-(29), the transfer function between the congestion measure and the congestion window is

$$\delta p(s) = \frac{\alpha c \gamma}{s + \alpha c} \frac{N e^{-\tau^{f*} s}}{\tau^* s + e^{-\tau^{f*} s}} \delta w(s) \tag{33}$$

Using (30), (32), and (33) the close-loop transfer function of the system is

$$L_{NC}(s) = \frac{\alpha c \gamma}{s + \alpha c} \frac{1}{\tau^* s + e^{-\tau^{f*} s}} \frac{N \rho^2 e^{-\tau^* s}}{z(\rho) p^* \left(\tau^* s + 2\beta z(\rho) p^* w^*\right)} \tag{34}$$

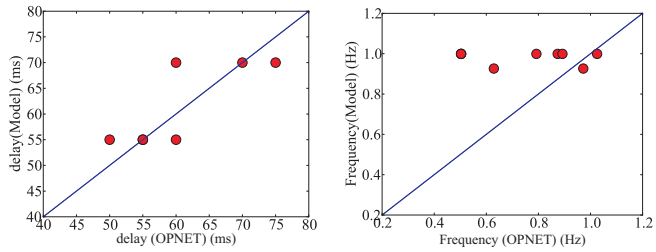
The first factor of (34) is due to the queue averaging (13), the second one describes the relation between the congestion window and the buffer size, and the third one illustrates the effect of TCP-NC. Considering a similar scenario without NC, the transfer function for the standard TCP-Reno is obtained from (34) by taking $\rho = 1$

$$L_{TCP}(s) = \frac{\alpha c \gamma}{s + \alpha c} \frac{1}{\tau^* s + e^{-\tau^{f*} s}} \frac{N e^{-\tau^* s}}{p^* \left(\tau^* s + 2\beta p^* w^*\right)} \tag{35}$$

The closed-loop system is stable if and only if the close-loop function (34) does not encircle the point $(-1, 0)$ as s traverses the closed D contour in the complex plane. The pure delay term in (34) adds significant phase at frequencies of the order of $1/\tau^*$ and higher, and the loop gain at the crossover frequency will determine stability. To characterize the stability region, we examine the Nyquist plot of $L_{NC}(j\omega)$ considering $N = 20, 30, \ldots, 60$ sources, with link capacity $c = 8, 9, \ldots, 15$ packets/ms, and equal propagation delays $d = 50, 55, 60, \ldots, 100$ ms. A constant packet size of 1000 bytes is assumed. For each (N, c) pair, we determine the delay $d_M(N, c)$ at which the Nyquist plot intersects the real axis closest to -1. This is the delay at which the system with parameters (N, c) transits from stability to instability. For this delay the critical frequency $f_M(N, c)$ at which the phase of $L_{NC}(j\omega)$ equals $-\pi$ is also computed. Note that the computation of $L_{NC}(j\omega)$ requires the equilibrium RTT τ^*, which is equal to the sum the of propagation delay $d_M(N, c)$ and the equilibrium queuing delay. The queuing delays is computed from the equilibrium model. For all evaluations, the following RED parameters have been used: $\overline{b}_\ell = 540$ packets, $\underline{b}_\ell = 40$ packets, $\alpha_\ell = 10^{-4}$, and $\overline{p}_\ell = 0.1$.

The previous simulation scenario is implemented in OPNET using the FTP persistent sessions. The TCP-Reno-NC protocol and RED with explicit congestion notification (ECN) marking are used, as well as the same parameter values for N, c, d, and packet size. For each (N, c) pair, the queue and TCP congestion window trajectories are examined to determine the critical delay $d_S(N, c)$ at which the system transits from stability to instability. This delay is measured at 5 ms accuracy. The fundamental frequency of oscillation $f_S(N, c)$ is obtained from the FFT of the queue trajectory.

Figure 2 shows the results of our model predictions and of the simulation results obtained from OPNET. Figure 2(a) shows the critical delay $d_M(N, c)$ computed using (34), and the critical delay $d_S(N, c)$ obtained from OPNET simulations. Each data point corresponds to a particular value of (N, c). The solid lines is the geometric locus where all the points should be if the prediction from the model would perfectly agree with the simulation results. Figure 2(b) represents the corresponding critical frequencies $f_S(N, c)$ and $f_M(N, c)$. It can be observed from Figure 2 that our predictions are acceptable, since the resolution for the delay is 5 ms.



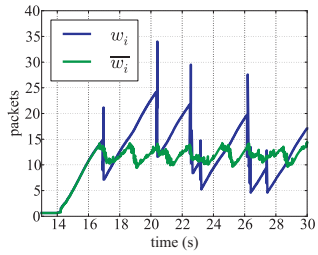
(a) Round Trip propagation delay (ms) at critical frequency

(b) Critical Frequency (Hz)

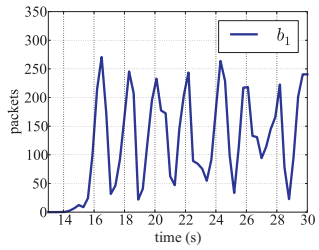
Fig. 2. Model vs OPNET simulation

Figures 3(a) and 3(b) show the OPNET simulation results corresponding to the trajectories of the sizes of the congestion window and of the queue when $N = 20$ sources share a link of capacity $c = 9$ packets/ms, with a round trip propagation delay $d = 50$ ms. Figure 3(a) shows the size of an individual window and the average window size for the 20 sources, both as a function of time. Oscillations are present in the individual window, and more important the average size shows a deterministic limit cycle, that is also present in the queue trace, Figure 3(b). In this case, the protocol is in unstable regime. This result can be predicted from the model (34) by plotting its Nyquist diagram for the considered scenario. Figure 3(c) shows that the Nyquist diagram encircles the point $-1 + j0$, leading to an unstable closed-loop system.

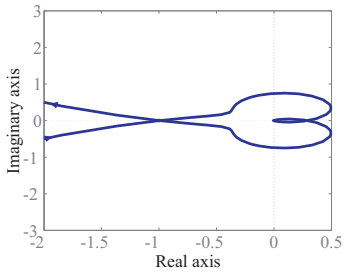
Figure 4 shows the stability region for the TCP-NC and TCP-Reno protocols using $\rho_i = \rho = 1.2$ for all users. For each N the critical delay $d_M(N, c)$ is plotted versus the capacity c of the common link. Each curve separates the stable



(a) Congestion Window of TCP-NC



(b) Queue Length



(c) Nyquist Diagram

Fig. 3. Congestion window trajectory, queue trajectory, and Nyquist stability diagram for $N = 20$, $c = 9$ packets/ms and $d = 50$ ms

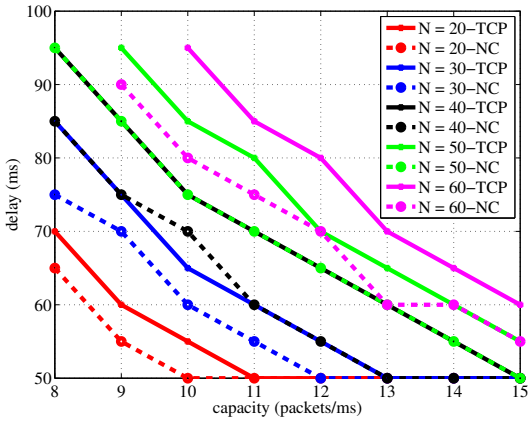


Fig. 4. Critical stability region of TCP-Reno and TCP-NC with $\rho_i = 1.2$

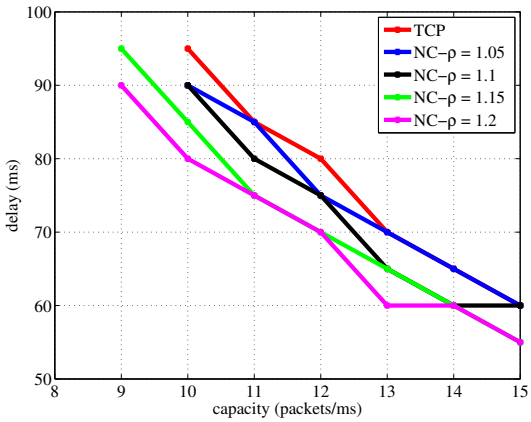


Fig. 5. Critical stability region of TCP-NC with $N = 60$ for various ρ

region, below the curve, from the unstable one, above it. The TCP-NC curve is dotted, and the TCP-Reno one is solid. TCP-Reno and TCP-NC become unstable when the delay or capacity become large, and the stability region increases with the number of users sharing the link, as with the standard TCP-Reno without NC [6]. The reduction in the stability region of TCP-NC with respect to the standard TCP-Reno is a result of the redundancy factor. Thus, increasing the redundancy factor increases instability.

Figure 5 shows the stability region for TCP-Reno and TCP-NC for redundancy factors $\rho_i = 1.05, 1.1, 1.15, 1.2$ when $N = 60$ users share the network. Similar to the last experiment, for each pair (ρ_i, c) the critical delay $d_M(N, c)$ is plotted versus the capacity c of the common link. Taking the stability region of TCP without NC as reference, one can observe a reduction of the size of the stability region for TCP-NC as ρ_i increases.

VI. Conclusions and Future Work

In this paper we have used tools from control theory to describe the dynamics of the TCP-NC protocol. We showed that TCP-NC solves a concave utility maximization problem, which yields a unique equilibrium point. We proposed a multi-link multi-source model that can be used to study the stability of a general TCP-NC-RED network. The theoretical model we developed and simulation results have shown that TCP-NC becomes unstable when the network scales up in delay and capacity, as the standard TCP-Reno does, but additionally the stability region of TCP-NC reduces, compared to standard TCP-Reno, when the redundancy factor increases. In future work we want to derive stability conditions for the original TCP-NC, and develop a new model to study global and local stability of TCP-NC with an adaptive redundancy factor.

References

[1] V. Jacobson. Congestion avoidance and control. *SIGCOMM Comput. Commun. Rev.*, 18(4):314–329, 1988.

[2] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *The Journal of the Operational Research Society*, 49(3):237–252, 1998.

[3] H. Khalil. *Nonlinear Systems*. Prentice Hall, Jan. 2002.

[4] M. Kim, M. Médard, and J. a. Barros. Modeling network coded tcp throughput: a simple model and its validation. In *Proc. VALUETOOLS*, pages 131–140, Cachan, France, 2011.

[5] S. Low. A Duality Model of TCP and Queue Management Algorithms. *Networking, IEEE/ACM Transactions on*, 11(4):525–536, 2003.

[6] S. Low, O. Paganini, and J. C. Doyle. Internet Congestion Control. *IEEE Control Systems Magazine*, 22:28–43, 2002.

[7] S. H. Low, L. L. Peterson, and L. Wang. Understanding TCP Vegas: A Duality Model. *J. ACM*, 49(2):207–235, Mar. 2002.

[8] H. Medina, M. Kieffer, and B. Pesquet-Popescu. Redundancy adaptation scheme for network coding with TCP. In *Proc. IEEE NETCOD*, 2012.

[9] S. Shakkottai and R. Srikant. Network Optimization and Control. *Found. Trends Netw.*, 2(3):271–379, Jan. 2007.

[10] J. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros. Network Coding Meets TCP: Theory and Implementation. *Proceedings of the IEEE*, 99(3):490 –512, march 2011.

[11] J. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros. Network Coding Meets TCP. In *INFOCOM 2009, IEEE*, pages 280–288, april 2009.

[12] A. Tang, X. Wei, S. H. Low, and M. Chiang. Equilibrium of Heterogeneous Congestion Control: Optimality and Stability. *IEEE/ACM Trans. Netw.*, 18(3):844–857, June 2010.