

TCP and Network Coding: Equilibrium and Dynamic Properties

Hamlet Medina Ruiz, Michel Kieffer, Beatrice Pesquet-Popescu

▶ **To cite this version:**

Hamlet Medina Ruiz, Michel Kieffer, Beatrice Pesquet-Popescu. TCP and Network Coding: Equilibrium and Dynamic Properties. IEEE/ACM Transactions on Networking, 2016, 24 (4), pp.1935-1947. 10.1109/TNET.2015.2477349 . hal-01260537

HAL Id: hal-01260537
https://hal.science/hal-01260537

Submitted on 22 Jan 2016

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 - CNRS LTCI, 46 rue Barrault, 75634 Paris Cedex 13, France

***Abstract*—This paper analyzes the impact on the stability of the TCP-Reno congestion control mechanism when a network coding (NC) layer is inserted in the TCP/IP stack. A model of the dynamics of the TCP-NC protocol combined with random early detection (RED) as active queue management mechanism is considered to study the network equilibrium and stability properties. The existence and uniqueness of an equilibrium point 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 are studied around the equilibrium. The fairness of TCP-NC with respect to TCP-Reno-like protocols is also studied. A version of TCP-NC with adaptive redundancy factor (TCP-NCAR) is also introduced. Results provided by the proposed model are compared with 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, but also when the redundancy factor increases. Its stability region is characterized as a function of the redundancy factor. If TCP-NC and TCP-Reno share the same links, TCP-NC is fair with TCP-Reno-like protocols when no redundancy is added. Simulations show that TCP-NCAR is able to compensate losses on the wireless parts of the network.**
[1]

***Index Terms*- congestion control, network coding, queueing analysis, stability analysis.**

I. Introduction

In [24], a TCP-fair protocol was proposed that interfaces network coding (NC) [1], [7], [13] with TCP by introducing a new coding layer between TCP and IP. The idea is to benefit from NC to improve TCP throughput. In this layer, TCP segments are network coded at the sender and network decoded at the receiver. In [23], several practical aspects are described such as the TCP-compatible sliding window within which packets are network coded or new rules for acknowledging bytes at the TCP layer.

In TCP-NC [24], a fixed amount of redundant network-coded packets may be transmitted by the sender to mask part of the losses occurring, *e.g.*, on the wireless parts of the network. An improvement in TCP throughput for a constant value of the NC redundancy factor ρ is shown experimentally in [23], [24]. The role of ρ is to match the rate at which data are obtained at the receiver to the sending rate of TCP. When ρ is too small, some losses are not masked from TCP and the NC decoding probability decreases. When ρ is too large, the congestion may increase, reducing the throughput.

Moreover, the possible side effects of a fixed redundancy introduced at the NC layer on the TCP congestion control mechanism have been overlooked, particularly stability issues and TCP fairness. The stability region of plain TCP congestion control mechanisms (Vegas or Reno) is affected by the value of RED parameters, bottleneck link capacities, and round trip times [17], [18]. In TCP-NC, the value of ρ has also a significant impact, as illustrated by Figures 4 and 5 in Section VI. A protocol originally stable characterized by small oscillations in the TCP congestion window and in the queue length for $\rho = 1$ may become unstable when $\rho = 1.2$.

This paper analyzes the effect of the NC layer on the stability of TCP and its fairness with TCP. Losses due to wireless links in the network are taken into account. To the best of our knowledge, no characterization of the average throughput of TCP-NC (with TCP-Reno), of its fairness, or of its stability has been done previously. This paper characterizes the effect of NC on the equilibrium and local stability of TCP-Reno. For that purpose, tools from control and optimization theory used to characterize TCP [9], [14], [18], [26] are considered to study TCP-NC as a distributed primal-dual algorithm that maximizes some aggregate utility function. A consequence of this study is that ρ should be properly adjusted. A variant of TCP-NC with adaptive redundancy (TCP-NCAR) is thus introduced, improving preliminary results in [21]. The adaptation of ρ is based on an estimation of packets dropped and packets backlogged as in TCP-Vegas [4].

Tools from control and optimization theory are considered in [5] to study the problem of congestion control when NC is employed to multicast flows. Using these tools the authors propose two sets of decentralized congestion controllers that can be implemented in a distributed manner. They work at the transport layer to adjust the source rate of each user and at the network layer to carry out NC. The interface between TCP and NC layers [24] is not considered in [5].

Among the previous works, [11] analyzes the improvement in average throughput provided by TCP-NC compared to plain TCP. A framework is proposed to predict the average throughput when TCP-Vegas [4] is used as TCP congestion control. In [23], [24] the average queue size is studied when TCP-NC is used. The analysis assumes nodes with infinite capacity buffers, but does not consider congestion control mechanism and propagation delay. In the same work, the fairness is only studied based on simulations.

[1]This work was partly supported by DIM-LSC SWAN and by ICODE. M. Kieffer is partly supported by the Institut Universitaire de France. Some parts of this work were presented at IEEE NETCOD 2012 and 2013.

TCP-NC is briefly described in Section II. The network model and the properties of TCP-NC are presented in Sections III and IV, respectively. TCP-NCAR is introduced in Section V. The considered models are applied to study several sources sharing a single bottleneck link in Section VI. Conclusions and future research directions are drawn in Section VII. Notations are presented in Table I.

TABLE I
SYMBOLS USED IN THIS PAPER

α_ℓ	exponential weighting factor for RED at link ℓ
b_ℓ	queue length at link ℓ
$\underline{b}_\ell$, $\overline{b}_\ell$, $\overline{p}_\ell$, γ_ℓ	RED parameters at link ℓ
$\lvert\mathcal{B}\rvert$	cardinality of set $\mathcal{B}$
c_ℓ	capacity of link ℓ
d_i	round trip propagation delay of source i
F_i	TCP dynamic of source i
G_ℓ	link dynamic at link ℓ
L	total number of links
$\mathcal{L}$	set of links
$\mathbf{L}_{NC}$	close-loop gain of NC
$\mathbf{L}_{TCP}$	close-loop gain of TCP
N	total number of sources
p^{c}_ℓ	congestion measure at link ℓ
p^{e}_ℓ	packet loss probability at link ℓ
ψ_i	decreasing rate per NACKs of source i in TCP-Reno-like protocols
q_i	aggregate congestion measure of user i
$\mathbf{R}$	routing matrix
R_{ACKs}	rate at which ACKs are received with TCP-NC
R_{NACKs}	rate at which NACKs are estimated with TCP-NC
ρ_i	redundancy factor for source i
$\mathcal{S}$	set of sources
τ_i	RTT for user i
$\tau^{\mathrm{b}}_{\ell i}$	backward delay from link ℓ to source i
$\tau^{\mathrm{f}}_{\ell i}$	forward delay from source i to link ℓ
U_i	utility function of source i
w_i	TCP congestion window size of source i
w_{m}	size of the NC coding window
x_i	rate for source i
ξ_i	increasing rate per ACKs of source i in TCP-Reno-like protocols
y_ℓ	aggregate input rate at link ℓ
$(\cdot)^*$	equilibrium value of $(\cdot)$

II. TCP-NC PROTOCOL

TCP-NC introduces a NC layer between the TCP and IP layers of the TCP/IP protocol stack [24]. The sender stores packets generated by TCP in a *coding buffer*. For each arrival from TCP, it transmits ρ random linear combinations of the w_{m} most recently arrived packets in the coding buffer. An NC header is added to the each network-coded packet, which contains information about the coefficients used to mix the TCP packets, the first and last byte sequence number (SEQ) of each packet involved in the mix, and the first byte that has not been acknowledged [23]. The original packets remain in the coding buffer until an appropriate TCP ACK arrives from the receiver side.

On the receiver side, upon reception of a packet containing a linear combination of TCP packets, 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 on this matrix. 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 plain TCP. Gaussian elimination may result in a new packet being decoded, in which case, the decoder delivers this packet to the TCP receiver. Any ACK generated by the TCP receiver is suppressed without being sent to the sender. These ACKs may be used for managing the decoding buffer. For more details, see [23].

III. DYNAMIC MODEL

A model of TCP-NC/RED is proposed here to study the equilibrium and dynamic properties of the TCP-NC protocol. A first non-linear model is used to determine its equilibrium properties in absence of forward delay using tools from optimization theory [9], [15], [26]. The model is then linearized around its equilibrium to study the effect of delay and of the NC redundancy factor on the stability of the system.

A. Network Model

Using the model introduced in [15], the network is represented by a set $\mathcal{L}$ of L links, indexed by $\ell = 1, \ldots, L$, with finite capacities c_ℓ. They are shared by a set $\mathcal{S}$ of S sources indexed by $i = 1, \ldots, S$. Each source i uses a subset $\mathcal{L}_i \subseteq \mathcal{L}$ of links to convey information through the network to some receivers. The sets $\mathcal{L}_i$ define an $L \times S$ routing matrix $\mathbf{R}$, with $R_{\ell i} = 1$ if $\ell \in \mathcal{L}_i$ and $R_{\ell i} = 0$ otherwise, see Figure 1.

Source i has an associated TCP congestion window size $w_i(t)$ and an NC redundancy factor ρ_i. As in [15], we assume that source i has access to the *aggregate price* or loss $q_i(t)$ of all the links in its route. Here, $q_i\,(t)$ represents losses due to transmission errors on the channels and to congestion[2]. Let $p^{\mathrm{e}}_\ell\,(t)$ and $p^{\mathrm{c}}_\ell\,(t)$ be the probabilities of packet losses due respectively to channel transmission errors and to congestion on link ℓ. The value of $p^{\mathrm{e}}_\ell\,(t)$ may vary in time and may be relatively large for wireless links, whereas it is negligible for wired links. The evolution of the congestion measure $p^{\mathrm{c}}_\ell\,(t)$ is described by adaptive queue management (AQM) algorithms. Assuming that $p^{\mathrm{e}}_\ell\,(t)$ and $p^{\mathrm{c}}_\ell\,(t)$ are small, the price (or loss probability) of the link ℓ is given by

$$p_\ell\,(t) = p^{\mathrm{e}}_\ell\,(t) + p^{\mathrm{c}}_\ell\,(t)\,. \tag{1}$$

The average round-trip time (RTT) $\tau_i(t)$ for source i is evaluated as

$$\tau_i\,(t) = d_i + \sum_\ell R_{\ell i} \frac{b_\ell\,(t)}{c_\ell}, \tag{2}$$

where d_i is the round-trip propagation delay assumed to be constant and $b_\ell(t)$ is the queue length at link ℓ. As in [15], a common time t for all the queues is considered. This avoids considering the fact that queuing delays at link ℓ depend themselves on previous queuing delays in the path. The price to be paid is an impossibility to consider transients at a time-scale smaller than the RTT. Queuing delays in the backward

[2]Losses due to packets which are linear combinations of already received packets may be represented by an additional channel loss on the last link to the receiver. This probability decreases with the size of field size in which NC operations are performed [19], [23].

path from the destination to source i are neglected in (2): congestion is assumed to occur only in the forward path from source i to the destination.

Then, the average rate in the time interval $[t-\tau_i(t), t]$ (at the output of the TCP layer) for the i-th source is

$$x_i(t) = \frac{w_i(t)}{\tau_i(t)}. \tag{3}$$

The aggregate rate at link ℓ is the sum of the delayed source rates. It has to account for ρ_i and is expressed as

$$y_\ell(t) = \sum_i R_{\ell i}\rho_i x_i\left(t - \tau^{\mathrm{f}}_{\ell i}(t)\right), \tag{4}$$

where $\tau^{\mathrm{f}}_{\ell i}$ is the forward delay from source i to link ℓ. In (4), the fact that the arrival rate from source i at link ℓ may be smaller due to packets lost on earlier links in the time interval $\left[t-\tau^{\mathrm{f}}_{\ell i}, t\right)$ is not taken into account, as done in [5], [15], [17], [25], [26] for losses due to congestion. Thus, $y_\ell(t)$ represents an upper bound of the rate that may have to be carried by link ℓ. The upper bound is reached in absence of losses on previous links.

The *price*

$$q_i(t) = q^{\mathrm{e}}_i(t) + q^{\mathrm{c}}_i(t), \tag{5}$$

where $q^{\mathrm{e}}_i(t) = \sum_\ell R_{\ell i} p^{\mathrm{e}}_\ell\left(t-\tau^{\mathrm{b}}_{\ell i}(t)\right)$ and $q^{\mathrm{c}}_i(t) = \sum_\ell R_{\ell i} p^{\mathrm{c}}_\ell\left(t-\tau^{\mathrm{b}}_{\ell i}(t)\right)$ represent the sum of delayed aggregate losses due to channel transmission errors and to congestion, respectively. To get (5), one assumes that $q^{\mathrm{e}}_i(t)$ and $q^{\mathrm{c}}_i(t)$ are small.[3] These prices are delayed by $\tau^{\mathrm{b}}_{\ell i}(t)$, which is the backward delay from link ℓ to source i. Link ℓ observes the aggregate flow rate $y_\ell(t)$ and is assumed to know the price $p^{\mathrm{c}}_\ell(t)$, which is controlled by its AQM algorithm.

TCP algorithms adjust the i-th source rate using some function F_i of $x_i(t)$, $q_i(t)$, and ρ_i (to account for the presence of the NC layer) as follows

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

AQM algorithms adjust the ℓ-th link price using some function G_ℓ of $p^{\mathrm{c}}_\ell(t)$ and $y_\ell(t)$ as follows

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

AQM algorithms are usually unaware of $p^{\mathrm{e}}_\ell(t)$. This is why G_ℓ depends only on $p^{\mathrm{c}}_\ell(t)$ and $y_\ell(t)$. F_i depends on the considered TCP algorithm (Reno, Vegas, etc.) and G_i depends on the AQM mechanism (FIFO, REM, RED, *etc.*) [15], [18]. The interaction between the control at sources and the control at links results in a feedback system, described in Figure 1.

This network model is used in Section III-B to derive the dynamics of TCP-NC combined with RED-AQM.

[3]This assumption may be difficult to satisfy for networks with wireless links with large p^{e}_ℓ. Nevertheless, considering mixed wired-wireless networks with one or two wireless links on each source-destination path (typically, for mobile video streaming), even when $q^{\mathrm{e}}_i(t) = 0.5$ and $q^{\mathrm{c}}_i(t) = 0.1$, (5) provides a conservative estimation of $q_i(t) = 0.6$ whereas in reality it is $q_i(t) = 1-(1-0.5)(1-0.1) = 0.55$, resulting in an error less than 10%. This remains acceptable from a control perspective.

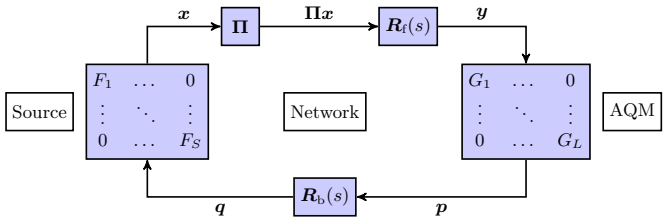


Fig. 1. Interactions between the TCP and the AQM mechanisms

B. TCP-NC/RED-AQM Model

We focus on the congestion avoidance phase of TCP-Reno. At time t, packets are sent by the NC layer of source i to the receiver at a rate $\rho_i x_i(t)$. A fraction of these packets is acknowledged at an average rate $\mathbb{E}[R_{\text{ACKs}}]$ and each ACK increments the size $w_i(t)$ of the TCP congestion window by $1/w_i(t)$. Lost packets or packets containing dependent linear combinations are detected by the reception of duplicate ACKs (DUPACKS) [6], [8]. These DUPACKs allow to deduce negative acknowledgments (NACKs) for some packets, even if NACKs are not explicitly transmitted. NACKs, occurring at an average rate of $\mathbb{E}[R_{\text{NACKs}}]$, lead to a reduction of $w_i(t)$ by $\beta w_i(t)$. The constant β is the rate at which $w_i(t)$ is decreased each time there are losses (in TCP-Reno, $\beta = 1/2$). As a result, the dynamics of the congestion window, during the congestion avoidance phase of TCP-Reno with NC, may be described as

$$\dot{w}_i(t) = \frac{1}{w_i(t)}\mathbb{E}\left[R_{\text{ACKs}}\right] - \beta w_i(t)\mathbb{E}\left[R_{\text{NACKs}}\right]. \tag{8}$$

Accounting for the effects of the NC layer, one has

$$\mathbb{E}\left[R_{\text{NACKs}}\right] = \sum_{k=(\rho_i-1)x^{\mathrm{d}}_i(t)+1}^{\rho_i x^{\mathrm{d}}_i(t)} \left[k-(\rho_i-1)x^{\mathrm{d}}_i(t)\right] \times \binom{\rho_i x^{\mathrm{d}}_i(t)}{k} \left(q_i(t)\right)^k \left(1-q_i(t)\right)^{\rho_i x^{\mathrm{d}}_i(t)-k}, \tag{9}$$

where $x^{\mathrm{d}}_i(t) = x_i(t-\tau_i(t))$, see Appendix A. In the congestion avoidance phase, (9) can be approximated as

$$\mathbb{E}\left[R_{\text{NACKs}}\right] \approx \max\left\{0, q_i(t)\,x^{\mathrm{d}}_i(t) - (\rho_i-1)\left(1-q_i(t)\right)x^{\mathrm{d}}_i(t)\right\}, \tag{10}$$

see Appendix B. The term $x^{\mathrm{d}}_i(t)q_i(t)$ in (10) is common to TCP-Reno and represents the rate at which packets are lost when there is no redundancy. The term $(\rho_i-1)\left(1-q_i(t)\right)x^{\mathrm{d}}_i(t)$ is the rate at which NC compensates losses by sending redundant packets.

Out-of-order delivery effects occurring in TCP-Reno are difficult to capture with a fluid model approximation. In TCP-NC, these effects do not appear, since independent linear combinations are acknowledged. The evolution of the congestion window of TCP-NC may thus be more precisely described with the fluid model approximation than that of TCP-Reno.

The rate at which ACKs are received is

$$\mathbb{E}[R_{\text{ACKs}}] = x^{\mathrm{d}}_i(t) - \mathbb{E}[R_{\text{NACKs}}]. \tag{11}$$

Combining (8), (10), and (11), the evolution of the congestion window controlled by TCP-NC (6) becomes

$$\dot{w}_i(t) = \frac{x_i^{\mathrm{d}}(t)}{w_i(t)} - \left(\frac{1}{w_i(t)} + \beta w_i(t)\right) \max\{0, q_i(t)\, x_i^{\mathrm{d}}(t) - (\rho_i - 1)(1 - q_i(t))\, x_i^{\mathrm{d}}(t)\}. \quad (12)$$

To study the evolution of the queue length in each link ℓ, we consider RED as AQM mechanism. With RED-AQM, $b_\ell(t)$ evolves as $\dot{b}_\ell(t) = (y_\ell(t) - c_\ell)^+_{b_\ell(t)}$, where $(a)^+_z = a$ if $z > 0$, and $(a)^+_z = \max(0, a)$ if $z = 0$, see [15]. 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 $p^{\mathrm{c}}_\ell(t)$ at link ℓ is given by

$$p^{\mathrm{c}}_\ell(t) = m_\ell(r_\ell(t)) = \begin{cases} 0, & r_\ell(t) \le \underline{b}_\ell, \\ \gamma_\ell(r_\ell(t) - \underline{b}_\ell), & \underline{b}_\ell < r_\ell(t) < \overline{b}_\ell, \\ \eta_\ell r_\ell(t) - (1 - 2\overline{p}_\ell), & \overline{b}_\ell < r_\ell(t) < 2\overline{b}_\ell, \\ 1, & r_t(t) \ge 2\overline{b}_\ell, \end{cases} \quad (14)$$

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

IV. Properties of TCP-NC, constant ρ_i

This section analyzes the equilibrium and some 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) is easier to characterize by considering the underlying optimization problem associated to TCP-NC-AQM. The aggregate utility of the users is maximized subject to link capacity constraints [9]. To prove the global stability of the algorithm, we assume the absence of forward delay, $\tau^{\mathrm{f}}_{\ell i} = 0$. The local stability is studied by linearizing the system around the equilibrium as a function of the RTT and of the constant redundancy factor. To study the equilibrium and dynamic properties of TCP-NC, we assume in what follows that $\mathbf{p}^{\mathrm{e}*}$ is constant with components equal to $p^{\mathrm{e}*}_\ell$.

A. Equilibrium and utility maximization framework

We consider the utility maximization framework introduced in [14], [18]. We show that, even in presence of channel losses, the equilibrium source rate vector $\mathbf{x}^*$ and the equilibrium price vector $\mathbf{p}^{\mathrm{c}*}$ of (6) and (7) are the primal and dual optimal solutions to an equivalent utility maximization problem.

1) Utility maximization in presence of redundancy and channel losses: The equilibrium points $(x_i^*, w_i^*, q_i^*, \tau_i^*)$ and $(p^{\mathrm{c}*}_\ell, y_\ell^*)$ of user i and link ℓ satisfy

$$\begin{cases} F_i(x_i^*, q_i^*, \rho_i) = 0, \\ G_\ell(p^{\mathrm{c}*}_\ell, y_\ell^*) = 0. \end{cases} \quad (15)$$

A solution of $F_i(x_i^*, q_i^*, \rho_i) = 0$ in (6) defines a relation between q_i^* and x_i^* parametrized in ρ_i, expressed as

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

where, for most TCP protocols, f_i is a positive, strictly monotone decreasing function with respect to q_i^* [15]. In Section IV-A2, we show that this remains valid for TCP-NC. The rate x_i^* may also be interpreted as the solution of a utility maximization problem

$$x_i^* = \arg\max_{x_i} U_i(x_i, \rho_i) - \rho_i x_i q_i \quad (17)$$

$$= \arg\max_{x_i} U_i(x_i, \rho_i) - q_i^{\mathrm{e}} \rho_i x_i - q_i^{\mathrm{c}} \rho_i x_i, \quad (18)$$

where the utility is defined as

$$\frac{\partial U_i}{\partial x_i}(x_i, \rho_i) = \rho_i f_i^{-1}(\rho_i, x_i). \quad (19)$$

The properties of f_i guarantee that U_i has a positive decreasing derivative with respect to x_i and is therefore monotone increasing and strictly concave in x_i.

In (18), the loss probabilities q_i^{e} are uncontrolled. The role of q_i^{c} is to coordinate the individual sources, so that the x_is maximize the aggregate utility under link rate constraints

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \geqslant 0} \sum_{i \in \mathcal{S}} \left(U_i(x_i, \rho_i) - q_i^{\mathrm{e}} \rho_i x_i\right) \text{ subject to } \mathbf{R\Pi x} \leqslant \mathbf{c}, \quad (20)$$

where $\mathbf{x}$ is the source rate vector, $\mathbf{\Pi}$ is an $S \times S$ redundancy factor matrix, with $\Pi_{ii} = \rho_i$ and $\Pi_{ij} = 0$ for $i \neq j$. The main difference with [9] is that each term of the aggregate utility in (20) incorporates the price due to random losses on the links. The constraint in (20) formalizes the fact that for each link ℓ, the (upper bound of the) aggregate rate y_ℓ should not exceed the capacity c_ℓ. These constraints may be quite conservative when the loss probability is high.

An optimal rate vector $\mathbf{x}^*$ exists since the objective function U_i is continuous. The feasible solution set defined by $\mathbf{R\Pi x} \leqslant \mathbf{c}$ remains compact in presence of redundancy factors. The optimal rate vector $\mathbf{x}^*$ is unique since U_i is strictly concave. As in [18], $\mathbf{x}$ and $\mathbf{p}^{\mathrm{c}}$ are interpreted as primal and dual variables, and $(\mathbf{F}, \mathbf{G}) = (F_i, G_\ell)_{i \in \mathcal{S}, \ell \in \mathcal{L}}$ as distributed primal-dual algorithms to solve the primal problem (20) and its Lagrangian

$$\mathcal{L}(\mathbf{p}^{\mathrm{c}}) = \max_{x_i \geqslant 0} \sum_{i \in \mathcal{S}} U_i(x_i, \rho_i) - q_i^{\mathrm{e}} \rho_i x_i - \sum_{\ell \in \mathcal{L}} p^{\mathrm{c}}_\ell \left[\sum_i R_{\ell i} \rho_i x_i - c_\ell\right],$$
$$= \sum_{i \in \mathcal{S}} \max_{x_i \geqslant 0} \left[U_i(x_i, \rho_i) - q_i \rho_i x_i\right] + \sum_{\ell \in \mathcal{L}} p^{\mathrm{c}}_\ell c_\ell. \quad (21)$$

We can use the projected gradient method [2], [3] to solve the dual problem

$$\mathbf{p}^{\mathrm{c}*} = \arg\min_{\mathbf{p}^{\mathrm{c}} \geqslant \mathbf{0}} \mathcal{L}(\mathbf{p}^{\mathrm{c}}). \quad (22)$$

In this case, the components of the gradient of (21) are

$$\frac{\partial \mathcal{L}(\mathbf{p}^{\mathrm{c}})}{\partial p^{\mathrm{c}}_\ell} = \frac{\partial}{\partial p^{\mathrm{c}}_\ell} \sum_{i \in \mathcal{S}} \max_{x_i \geqslant 0} \left[U_i(x_i, \rho_i) - q_i \rho_i x_i\right] + \frac{\partial}{\partial p^{\mathrm{c}}_\ell} \sum_\ell p^{\mathrm{c}}_\ell c_\ell,$$
$$= \sum_{i \in \mathcal{S}} \max_{x_i \geqslant 0} \left[\frac{\partial U_i(x_i, \rho_i)}{\partial x_i} - q_i \rho_i\right] \frac{\partial x_i}{\partial p^{\mathrm{c}}_\ell} - \sum_i \frac{\partial q_i}{\partial p^{\mathrm{c}}_\ell} \rho_i x_i + \frac{\partial}{\partial p^{\mathrm{c}}_\ell} \sum_\ell p^{\mathrm{c}}_\ell c_\ell,$$
$$= c_\ell - \sum_i R_{\ell i} \rho_i x_i = c_\ell - y_\ell. \quad (23)$$

where we use the fact that the unique maximizer of (21) satisfies $\frac{\partial U_i(x_i,\rho_i)}{\partial x_i} = \rho_i q_i$ and that $\frac{\partial q_i}{\partial p_\ell^c} = R_{\ell i}$, see (5). The solution of (22) is thus such that for all ℓ, one has either $p_\ell^{c*} > 0$, leading to $y_\ell^* = c_\ell$, or $p_\ell^{c*} = 0$, leading to $y_\ell^* < c_\ell$. Then, any AQM algorithm with a dynamics leading to the satisfaction of these constraints can solve (21), see [16], [22]. For example, RED is an AQM algorithm satisfying this target.

When $\rho_i = 1$, the solution of (20) becomes that of the aggregate utility maximization problem with random losses studied in [12] using a penalty function approach. Additionally, if $q_i^e \approx 0$, the solution is that of the standard network utility maximization problem described in [9], [15], [16], [22].

2) Global stability in absence of forward delay, $\tau_{\ell i}^f = 0$: Neglecting the forward delay, from the equilibrium of (12) obtained for $\dot{x}_i = 0$, one deduces the aggregate price at equilibrium seen by user i

$$q_i^* = \frac{1 + (\rho_i - 1)\left(1 + \beta\left(x_i^*\tau_i^*\right)^2\right)}{\rho_i\left(1 + \beta\left(x_i^*\tau_i^*\right)^2\right)}, \tag{24}$$

see Appendix C. Using (24), the utility function of TCP-NC is obtained by integration of (19) with respect to x_i^*

$$U_i(x_i) = \frac{1}{\sqrt{\beta}\tau_i}\tan^{-1}\left(\sqrt{\beta}\tau_i x_i\right) + (\rho_i - 1)\, x_i. \tag{25}$$

This utility function differs from that found for TCP-Reno in [9] by the addition of the linear term, which increases with the redundancy of the NC layer for the i-th source. Nevertheless, $U_i(x_i)$ in (25) is still concave and yields a unique equilibrium point. The average throughput is found by considering again the equilibrium of (12) and by solving $\dot{x}_i(t) = 0$ for x_i^*

$$x_i^* = \frac{1}{\tau_i^*}\sqrt{\frac{(1 - q_i^*)\,\rho_i}{\beta\left(1 - (1 - q_i^*)\,\rho_i\right)}}. \tag{26}$$

To study the global stability of TCP-NC, we consider the following function introduced in [22]:

$$V(\mathbf{p}) = \sum_\ell (c_\ell - y_\ell^*)\, p_\ell + \sum_i \int_{q_i^*}^{q_i} \left(x_i^* - \left(U_i'\right)^{-1}(\rho_i \sigma)\right) d\sigma. \tag{27}$$

For TCP with RED-AQM, (27) is a Lyapunov function if the corresponding TCP utility function of each user i is strictly concave, see [22]. When considering TCP-NC, U_i given by (25) is strictly concave. Thus (27) is still a Lyapunov function for our case, *i.e.*, $V(\mathbf{p}) \geqslant 0$, $\dot{V}(\mathbf{p}) \leqslant 0, \forall \mathbf{p} \geqslant 0$. By the Lyapunov theorem one deduces that the system is globally stable in the absence of forward delay [10].

3) Interaction of TCP-Reno-like flows and TCP-NC flows: Additive-increase multiplicative-decrease (AIMD) algorithms increase the rate $x_i(t)$ by $\xi_i(x_i(t))$ on each positive acknowledgment, decrease it by $\psi_i(x_i(t))$ on each loss, and react to the loss congestion measure $p_\ell^c(t)$. Thus, the source dynamic (6) of TCP-Reno like algorithms may be written as

$$\dot{x}_i(t) = (1 - q_i(t))\, x_i^d(t)\xi_i(x_i(t)) - q_i(t)x_i^d(t)\psi_i(x_i(t)). \tag{28}$$

With standard TCP-Reno

$$\xi_i(x_i(t)) = \frac{1}{x_i(t)\tau_i^2(t)}, \tag{29}$$

and

$$\psi_i(x_i(t)) = x_i(t)/2. \tag{30}$$

A source rate control is said to be TCP-fair if its equilibrium rate coincides with that obtained by TCP-Reno. The equilibrium of such algorithms is found by setting $\dot{x}_i(t) = 0$ in (28). Thus, an algorithm with parameters (ξ_i, ψ_i) [14] is TCP fair if and only if

$$\frac{\xi_i(x_i^*)}{\xi_i(x_i^*) + \psi_i(x_i^*)} = q_i^*. \tag{31}$$

In order to analyze the fairness of TCP-NC with TCP-Reno, the corresponding parameters (ξ_i, ψ_i) for TCP-Reno (29), (30), and the equilibrium of TCP-NC (24) are substituted in (31) leading to

$$\frac{2}{2 + \left(x_i^*\tau_i^*\right)^2} = \frac{2 + (\rho_i - 1)\left(2 + (x_i^*\tau_i^*)^2\right)}{\rho_i\left(2 + (x_i^*\tau_i^*)^2\right)}, \tag{32}$$

which is true only for $\rho_i = 1$. This analysis shows that TCP-NC is TCP-Reno fair in absence of redundancy introduced by the NC layer, which reduces the decoding probability in presence of network congestion or channel losses. A similar analysis can be done to verify that TCP-NC is not fair with different values of (ξ_i, ψ_i) corresponding to different TCP-Reno-like flavors.

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

So far we studied the equilibrium of the TCP-NC model. In this section, the TCP-NC-RED dynamic model is linearized in order to study the effect of the delay and of the redundancy factor around the equilibrium. Considering that $q_i^* > 0$ at equilibrium (see Appendix C), (8) can be expressed as

$$\begin{aligned}\dot{w}_i(t) = \frac{x_i^d(t)}{w_i(t)} - \left(\frac{1}{w_i(t)} + \beta w_i(t)\right) \\ \times \left(x_i^d(t) q_i(t) - (\rho_i - 1)\left(1 - q_i(t)\right) x_i^d(t)\right).\end{aligned} \tag{33}$$

After some manipulations, (33) becomes

$$\begin{aligned}\dot{w}_i(t) = \rho_i x_i(t - \tau_i(t))\left(1 - q_i(t)\right)\frac{1}{w_i(t)} \\ - \beta x_i(t - \tau_i(t))\left(1 - (1 - q_i(t))\,\rho_i\right) w_i(t).\end{aligned} \tag{34}$$

From Section IV-A, assuming that $\mathbf{R}$ has full rank, there is a unique equilibrium (w_i^*, q_i^*, τ_i^*). Expressing (34) as a function of the congestion window with (3), one obtains

$$\begin{aligned}\dot{w}_i(t) = \frac{\rho_i w_i(t - \tau_i^*)\left(1 - q_i(t)\right)}{\tau_i(t - \tau_i^*)}\frac{1}{w_i(t)} \\ - \frac{\beta w_i(t - \tau_i^*)\left(1 - (1 - q_i(t))\,\rho_i\right) w_i(t)}{\tau_i(t - \tau_i^*)},\end{aligned} \tag{35}$$

where $\tau_i^* = d_i + \sum_\ell R_{\ell i}\frac{b_\ell^*}{c_\ell}$ is the equilibrium RTT, and b_ℓ^* denotes the equilibrium queue length. Let $\delta w_i(t) = w_i(t) -$

w_i^*. By linearizing (35) around the equilibrium (26), we obtain the linearized window dynamics

$$\delta\dot{w}_i(t) = -\frac{2\beta\left(1-\left(1-q_i^*\right)\rho_i\right)w_i^*}{\tau_i^*}\delta w_i\left(t\right) - \frac{\rho_i}{\tau_i^*}\left(1+\beta\left(w_i^*\right)^2\right)\delta q_i\left(t\right), \quad (36)$$

Inserting (26) in (36), one gets

$$\begin{aligned}\delta\dot{w}_i(t) &= -\frac{2\beta\left(1-\left(1-q_i^*\right)\rho_i\right)w_i^*}{\tau_i^*}\delta w_i\left(t\right) \\ &\quad - \frac{\rho_i}{\tau_i^*}\left(1+\frac{\left(1-q_i^*\right)\rho_i}{\left(1-\left(1-q_i^*\right)\rho_i\right)}\right)\delta q_i\left(t\right) \\ \delta\dot{w}_i(t) &= -\frac{2\beta\left(1-\left(1-q_i^*\right)\rho_i\right)w_i^*}{\tau_i^*}\delta w_i\left(t\right) \\ &\quad - \frac{\rho_i}{\left(1-\left(1-q_i^*\right)\rho_i\right)\tau_i^*}\delta q_i\left(t\right). \end{aligned} \quad (37)$$

Considering that $\delta p_\ell\left(t\right) = p_\ell\left(t\right) - p_\ell^*$, $\delta p_\ell^{\mathrm{c}}\left(t\right) = p_\ell^{\mathrm{c}}\left(t\right) - p_\ell^{\mathrm{c}*}$, $\delta b_\ell\left(t\right) = b_\ell\left(t\right) - b_\ell^*$, and by linearizing (14) around the equilibrium and assuming that $\underline{b}_\ell < r_\ell\left(t\right) < \overline{b}_\ell$, one obtains the linearized price dynamics

$$\delta\dot{p}_\ell^{\mathrm{c}}\left(t\right) = -\alpha_\ell c_\ell \delta p_\ell^{\mathrm{c}}\left(t\right) + \alpha_\ell c_\ell \gamma_\ell \delta b_\ell\left(t\right). \quad (38)$$

To linearize the link dynamics when considering RED, the effects of non-bottleneck links are ignored. Adapting the AQM model of [15] to the NC context, the dynamics of the queue length $b_\ell\left(t\right)$ is given by

$$\dot{b}_\ell\left(t\right) = \sum_i R_{\ell i}\frac{\rho_i w_i\left(t-\tau_{\ell i}^{\mathrm{f}*}\right)}{d_i + \sum_k R_{ki} b_k\left(t-\tau_{\ell i}^{\mathrm{f}*}\right)/c_k} - c_\ell, \quad (39)$$

where $\tau_{\ell i}^{\mathrm{f}*}$ is the forward equilibrium delay from source i to link ℓ. The linearized queue length dynamics around the equilibrium RTT is

$$\begin{aligned}\delta\dot{b}_\ell\left(t\right) &= \sum_i R_{\ell i}\frac{\rho_i \delta w_i\left(t-\tau_{\ell i}^{\mathrm{f}*}\right)}{\tau_i^*} \\ &\quad - \sum_i\sum_k R_{\ell i}R_{ki}\frac{\rho_i w_i^*}{\left(\tau_i^*\right)^2 c_k}\delta b_k\left(t-\tau_{\ell i}^{\mathrm{f}*}\right). \end{aligned} \quad (40)$$

In the Laplace domain, (37), (38), and (40) become

$$\delta\mathbf{w}\left(s\right) = -\left(s\mathbf{I}+\mathbf{D}_1\right)^{-1}\mathbf{D}_2\mathbf{R}_{\mathrm{b}}^T\left(s\right)\delta\mathbf{p}\left(s\right), \quad (41)$$

$$\delta\mathbf{p}^{\mathrm{c}}\left(s\right) = \left(s\mathbf{I}+\mathbf{D}_3\right)^{-1}\mathbf{D}_4\delta\mathbf{b}\left(s\right) \quad (42)$$

$$\delta\mathbf{p}\left(s\right) = \delta\mathbf{p}^{\mathrm{c}}\left(s\right) \quad (43)$$

$$\delta\mathbf{b}\left(s\right) = \left(s\mathbf{I}+\mathbf{R}_{\mathrm{f}}\left(s\right)\mathbf{D}_5\mathbf{R}^T\mathbf{D}_6\right)^{-1}\mathbf{R}_{\mathrm{f}}\left(s\right)\mathbf{D}_7\delta\mathbf{w}\left(s\right), \quad (44)$$

where $\delta\mathbf{w}\left(s\right) = \left(\delta w_1\left(s\right),\ldots,\delta w_{\mathcal{S}}\left(s\right)\right)^T$, $\delta\mathbf{p}\left(s\right) = \left(\delta p_1\left(s\right),\ldots,\delta p_{\mathcal{L}}\left(s\right)\right)^T$, $\delta\mathbf{b}\left(s\right) = \left(\delta b_1\left(s\right),\ldots,\delta b_{\mathcal{L}}\left(s\right)\right)^T$, and $\mathbf{D}_k$, $k=1,\ldots,7$, are diagonal matrices defined as

$$\mathbf{D}_1 = \mathrm{diag}\left(\frac{2\beta\left(1-\left(1-q_i^*\right)\rho_i\right)w_i^*}{\tau_i^*}\right),$$

$$\mathbf{D}_2 = \mathrm{diag}\left(\frac{\rho_i}{\left(1-\left(1-q_i^*\right)\rho_i\right)\tau_i^*}\right)$$

$$\mathbf{D}_3 = \mathrm{diag}\left(\alpha_\ell c_\ell\right), \mathbf{D}_4 = \mathrm{diag}\left(\alpha_\ell c_\ell \gamma_\ell\right)$$

$$\mathbf{D}_5 = \mathrm{diag}\left(\frac{\rho_i w_i^*}{\left(\tau_i^*\right)^2}\right), \mathbf{D}_6 = \mathrm{diag}\left(\frac{1}{c_\ell}\right), \mathbf{D}_7 = \mathrm{diag}\left(\frac{\rho_i}{\tau_i^*}\right).$$

In (41)-(44), $\mathbf{R}_{\mathrm{f}}(s)$ and $\mathbf{R}_{\mathrm{b}}(s)$ are the forward and backward equilibrium delay matrices, given by $\left(\mathbf{R}_{\mathrm{f}}\left(s\right)\right)_{\ell i} = e^{-\tau_{\ell i}^{\mathrm{f}*}s}$ if $\ell\in\mathcal{L}_i$ and $\left(\mathbf{R}_{\mathrm{f}}\left(s\right)\right)_{\ell i} = 0$ otherwise, $\left(\mathbf{R}_{\mathrm{b}}\left(s\right)\right)_{\ell i} = e^{-\tau_{\ell i}^{\mathrm{b}*}s}$ if $\ell\in\mathcal{L}_i$ and $\left(\mathbf{R}_{\mathrm{b}}\left(s\right)\right)_{\ell i} = 0$ otherwise. Using (41)-(44) and a procedure similar to that used in [15], the loop gain $\mathbf{L}_{NC}\left(s\right)$ defined by $\delta\mathbf{p}\left(s\right) = \mathbf{L}_{NC}\left(s\right)\delta\mathbf{y}\left(s\right)$ is given by

$$\begin{aligned}\mathbf{L}_{NC}\left(s\right) &= \mathbf{R}_{\mathrm{f}}\left(s\right)\mathbf{D}_7\left(s\mathbf{I}+\mathbf{D}_1\right)^{-1}\mathbf{D}_2\mathbf{R}_{\mathrm{b}}^T\left(s\right)\left(s\mathbf{I}+\mathbf{D}_3\right)^{-1} \\ &\quad\times\mathbf{D}_4\left(s\mathbf{I}+\mathbf{R}_{\mathrm{f}}\left(s\right)\mathbf{D}_5\mathbf{R}^T\mathbf{D}_6\right)^{-1}. \end{aligned} \quad (45)$$

where $\delta\mathbf{y}\left(s\right) = \mathbf{R}_{\mathrm{f}}\left(s\right)\mathbf{D}_7\delta\mathbf{w}\left(s\right)$. $\mathbf{L}_{NC}\left(s\right)$ may be used to characterize the stability.

V. Adapting the redundancy factor ρ_i

Ideally, the redundant packets added by the NC layer should mask to the TCP layer only wireless channel losses. In the proposed algorithm, ρ_i is increased if the wireless links are dropping packets and is decreased when congestion appears, allowing TCP to detect it and to reduce its rate.

The TCP Vegas loss predictor [20] is used to estimate the congestion state, in conjunction with the number of DUPACKs generated by the receiver to estimate the network load.

1) Vegas loss predictor: TCP Vegas uses a loss predictor to decide whether the network is congested. It is based on rate estimators [4],which help estimating the amount of backlogged packets in the buffer of the bottleneck link. We propose to use the Vegas loss predictor at the NC layer to determine when the network experiences congestion.

As in [4], let $BaseRTT$ to be the RTT of a TCP segment when the connection is not congested (in practice the minimum of all measured RTTs, which is reset after a timeout). $BaseRTT$ serves as a reference to derive the expected throughput the network can accommodate. To estimate the cause of the packet losses, the number of queued packets Q^{v} is calculated as

$$Q^{\mathrm{v}} = \left(\rho\frac{w}{BaseRTT} - \rho\frac{w}{RTT}\right)BaseRTT, \quad (46)$$

where w is the size of the congestion window, and ρ accounts for the redundancy added by the NC layer. To estimate the network load, (46) is compared with two thresholds, ς_1 and ς_2. If $Q^{\mathrm{v}} \geq \varsigma_2$, the network is considered congested. If $Q^{\mathrm{v}} \leq \varsigma_1$, few packets are backlogged and the network is considered not congested. Finally if $\varsigma_1 < Q^{\mathrm{v}} < \varsigma_2$, the predictor assumes that the network state is the same as in the previous estimation. Typical values used in Unix implementations are $\varsigma_1 = 1$ and $\varsigma_2 = 3$. See [20] for more details.

2) Adaptive algorithm for ρ: Each time an ACK is received, Q^{v} is evaluated using (46). If $Q^{\mathrm{v}} < \varsigma_1$, and some DUPACKs have been received, packets have been dropped mainly due to channel transmission errors and not to buffer overflows in the router queues. In this situation, ρ_i is increased to mask the losses. If $Q^{\mathrm{v}} \geq \varsigma_2$, ρ_i is decreased. Initially ρ is set to a value ρ_0 that takes into account the loss in throughput due to the finiteness of the field in which NC operations are performed. These operations are summarized in Algorithm 1.

Algorithm 1 NC redundancy factor adaptation

$\varsigma_1 = 1$, $\varsigma_2 = 3$, $DUPACKs = 0$, $Q^{\mathrm{v}} = 0$, $\rho = \rho_o$, $M = 3$, update $BaseRTT$, RTT {Initialization}
while TCP connection is not closed **do**
  while $w_i \geq$ slow-start-threshold **do**
    for every ACK **do**
      if ACK is a duplicate **then**
        $DUPACKs = DUPACKs + 1$
      else
        $DUPACKs = 0$
      end if
      if ACK of packet has been marked to update Q^{v} **then**
        update Q^{v} using (46)
      end if
      if $Q^{\mathrm{v}} \geq \varsigma_2$ **then**
        $\rho \leftarrow \max(\rho_o, \rho - a_2/w_i)$ {congested network}
      else if $Q^{\mathrm{v}} \geq \varsigma_1$ **then**
        if $DUPACKs \geq M$ **then**
          $\rho \leftarrow \rho + a_2/w_i$ {non congested network}
        end if
      end if
      if $DUPACKs \geq M$ **then**
        $\rho \leftarrow \rho + a_1$ {if there are losses, increase ρ}
      end if
    end for
  end while
  update $BaseRTT$, RTT, $\rho \leftarrow \rho_o$
end while

In Algorithm 1, we take $a_2 > a_1$ to reduce ρ faster in presence of losses due to congestion, independently of the magnitude of losses due to channel transmission errors.

The implementation of Algorithm 1 in the NC layer requires some minor changes to the TCP-NC protocol described in [23]. They are described in [21].

VI. Model verification and stability region

In this section we evaluate the properties of the linearized model (45) and the effect of the delay, of the channel losses, and of the redundancy factor on the stability region of TCP-NC. In all simulations, the size of the coding window is taken as $w_{\mathrm{m}} = 2$ and NC is performed in $GF\left(2^8\right)$.

A. Identical sources sharing a common bottleneck link

Consider N identical sources sharing a single link $|\mathcal{L}| = 1$ with capacity c, packet loss probability $p^{\mathrm{e}*}$, RED as AQM, equal round-trip propagation delay $d_i = d$, and equal redundancy factors $\rho_i = \rho$. The routing matrix is $\mathbf{R} = [1, \ldots, 1]$, with N columns. From (41), the transfer function between the i-th congestion window and the aggregate price is

$$\delta w_i(s) = \frac{\rho}{p^* z(\rho)} \frac{e^{-\tau^{\mathrm{b}*} s}}{\tau^* s + 2\beta z(\rho) p^* w^*} \delta p(s), \tag{47}$$

where $z(\rho) = \frac{1-(1-p^*)\rho}{p^*}$, and $p^* = p^{\mathrm{e}*} + p^{\mathrm{c}*}$. The equilibrium RTT, forward delay, and backward delay are τ^*, $\tau^{\mathrm{f}*}$, and $\tau^{\mathrm{b}*}$ respectively. The equilibrium congestion window is given by $w^* = \frac{\tau^* c}{\rho N}$. The equilibrium price

$$p^* = \frac{1 + (\rho - 1)\left(1 + \beta (w^*)^2\right)}{\rho\left(1 + \beta (w^*)^2\right)}, \tag{48}$$

is obtained from (24). Using (41)-(43), the transfer function between the aggregate price and the congestion window is

$$\delta p(s) = \frac{\alpha c \gamma}{s + \alpha c} \frac{N \rho e^{-\tau^{\mathrm{f}*} s}}{\tau^* s + e^{-\tau^{\mathrm{f}*} s}} \delta w_i(s) \tag{49}$$

Using (44), (47), and (49), the loop transfer function of the system is

$$L_{NC}(s) = \frac{\alpha c \gamma}{s + \alpha c} \frac{1}{\tau^* s + e^{-\tau^{\mathrm{f}*} s}} \frac{N \rho^2 e^{-\tau^* s}}{z(\rho) p^* (\tau^* s + 2\beta z(\rho) p^* w^*)}. \tag{50}$$

The first factor of (50) is due to the queue length 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 (50) by taking $\rho = 1$

$$L_{TCP}(s) = \frac{\alpha c \gamma}{s + \alpha c} \frac{1}{\tau^* s + e^{-\tau^{\mathrm{f}*} s}} \frac{N e^{-\tau^* s}}{p^* (\tau^* s + 2\beta p^* w^*)}. \tag{51}$$

The close-loop system is stable if and only if the loop function (50) does not encircle the point $(-1, 0)$ as s crosses the closed D contour in the complex plane. The pure delay term in (50) 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 various packet loss probabilities $p^{\mathrm{e}*}$, $N = 20, 30, \ldots, 60$ sources, link capacities $c = 8, 9, \ldots, 15$ packets/ms, and equal propagation delays $d = 50, 55, 60, \ldots, 100$ ms. A constant packet size $\kappa = 1000$ 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^*, equal to the sum of the propagation delay d and the equilibrium queuing delay. For all evaluations, the RED parameters considered in [15] are $\bar{b}_1 = 540$ packets, $\underline{b}_1 = 40$ packets, $\alpha_1 = 10^{-4}$, $\overline{p}_1 = 0.1$, leading to $\gamma_1 = 20 \times 10^{-6}$.

The previous simulation scenario is implemented in OPNET using FTP persistent sessions, see Figure 2. TCP-Reno-NC and RED with explicit congestion notification (ECN) marking are used. The topology consists of a network with $N = 20$ or $N = 30$ FTP clients, sharing a single bottleneck link with capacity $c = 9$ packets/ms, with a fixed packet size κ. In Figure 2, the cloud is used to represent channel losses on some wireless links. Ethernet is used at the MAC layer. The start time of all the FTP-NC sessions follows a uniform distribution between 0 and 10 s, and the pseudo-random seed used for all

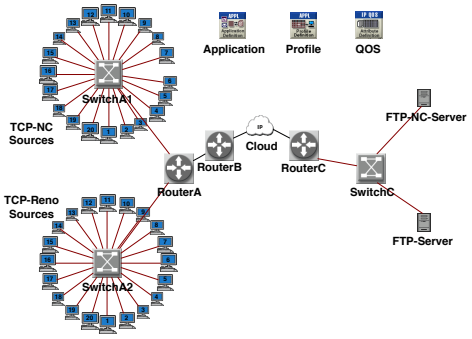
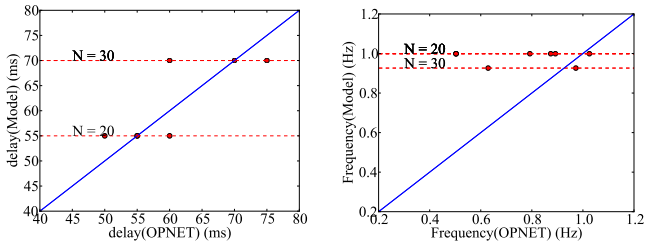


Fig. 2. Topology used in the simulations

the random timer values for message retransmission calculations in the TCP/IP stack and in Ethernet is changed between realizations. For each (N, c) pair, the round trip propagation delay d is taken in $\{50, 55, 60, \ldots, 100\}$ ms. The trajectories of the queue length and of the TCP congestion window are examined to determine the critical delay $d_S(N, c)$ at which the system transits from stability to instability. The fundamental frequency of oscillation $f_S(N, c)$ is obtained from the FFT of the queue length trajectory.

1) Stability region without losses due to transmission errors: Figure 3 shows the results of our model predictions and of the simulation results obtained from OPNET when losses due to transmission errors are not considered. Figure 3(a) shows the critical delay $d_M(N, c)$ computed using (50), and the critical delay $d_S(N, c)$ obtained from OPNET simulations. Each data point corresponds to a simulation realization for a particular value of (N, c). Simulation points on a given dotted line have been obtained for the same value of N, but for different random start times of the TCP-NC session. The solid line is the geometric locus where all points should be if the critical frequency predicted from the model would perfectly agree with the simulation results. Figure 3(b) represents the corresponding $f_S(N, c)$ and $f_M(N, c)$. It can be observed from Figure 3 that our predictions are acceptable, since the resolution for the delay is 5 ms.

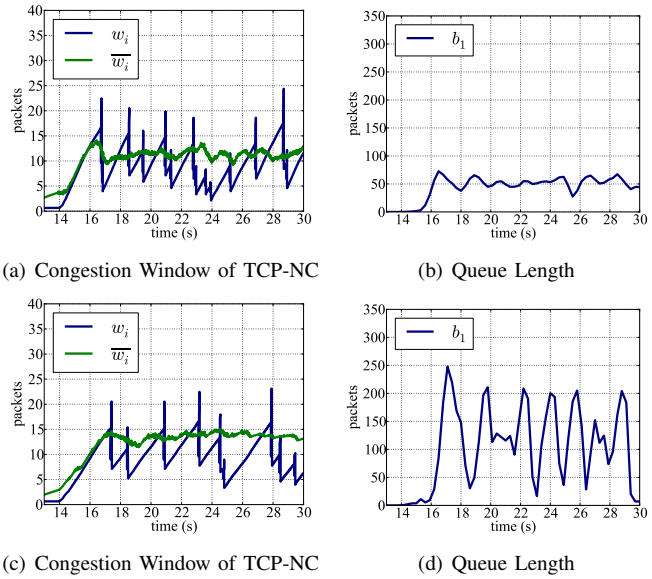


(a) Round Trip propagation delay (ms) at critical frequency.

(b) Critical Frequency (Hz)

Fig. 3. Model vs OPNET simulation for $N = 20$ and $N = 30$ sources sharing a single bottleneck link, with RED as AQM, and bottleneck link capacity $c = 9$ packets/ms

In Figures 4 and 5, only losses due to congestion are considered. Figures 4(a), 4(b), 4(c), and 4(d), show the trajectories of the size of the congestion window and of the queue length. These results are obtained when $N = 20$ sources share a link of capacity $c = 9$ packets/ms, each one with a round-trip propagation delay $d = 60$ ms. In Figure 4, $\rho_i = 1$. Figure 4(a) shows the size of an individual window and the average window size over 100 realizations for the 20 sources, both as a function of time. In this case, small oscillations may be seen in Figure 4(b) for the average window size and in the queue length: the protocol is in a stable regime. This can be predicted from (50) by plotting the Nyquist diagram, see Figure 5(a).



(a) Congestion Window of TCP-NC

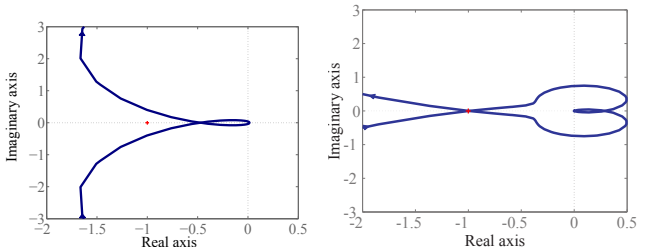
(b) Queue Length

(c) Congestion Window of TCP-NC

(d) Queue Length

Fig. 4. Scenario without channel transmission errors. Congestion window trajectory and queue trajectory for $N = 20$, $c = 9$ packets/ms, $p_1^{e*} = 0$, $d = 60$ ms, and $\rho_i = 1$ (top) or $\rho_i = 1.2$ (bottom)

When $\rho_i = 1.2$, Figure 4(c) shows oscillations on the size of an individual window and the average window size both as a function of time. Oscillation with larger amplitude are observed in Figure 4(d) compared to those in Figure 4(b), which indicates that the protocol is in an unstable regime. The corresponding Nyquist diagram in Figure 5(b) encircles the point $(-1, 0)$, leading indeed to an unstable closed-loop system. Figures 4 and 5 illustrates the fact that increasing ρ may transit a protocol from an stable to an unstable regime.



(a) $\rho_i = 1$

(b) $\rho_i = 1.2$

Fig. 5. Scenario without channel transmission errors. Nyquist diagrams for $N = 20$, $c = 9$ packets/ms, $p_1^{e*} = 0$, and $d = 60$ ms

Figure 6 shows the stability region for $\rho_i = \rho = 1.2$ obtained from (50). For each value of N, the critical delay $d_M(N, c)$ is plotted versus the capacity c of the common link. Each curve separates the stable region, below the curve, from

the unstable one, above it. TCP-Reno (solid) and TCP-NC (dotted) become unstable when the delay or capacity becomes large. The size of the stability region increases with the number of users sharing the link, as with standard TCP-Reno without NC [15]. The reduction in the stability region of TCP-NC compared to that of the standard TCP-Reno is a result of the redundancy factor.

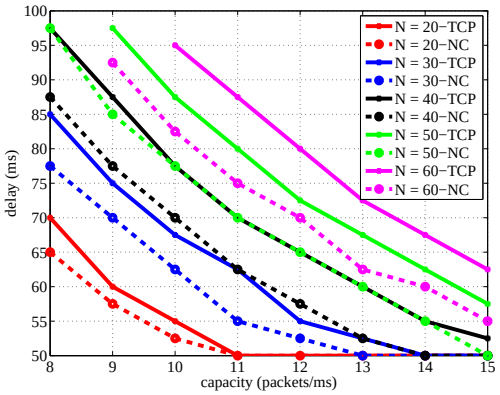
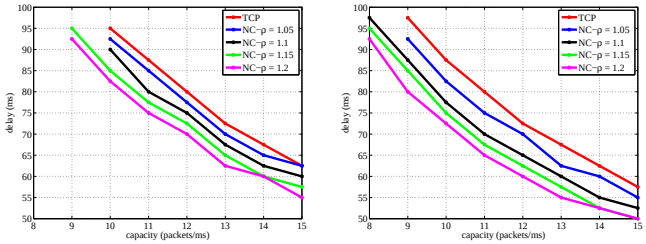


Fig. 6. Scenario without channel transmission errors. Critical stability region of TCP-Reno and TCP-NC with $p_1^{e*} = 0.0$ and $\rho_i = 1.2$



(a) $p_1^{e*} = 0.0$ (b) $p_1^{e*} = 0.1$

Fig. 7. Critical stability region of TCP-NC with $N = 60$ and different values of ρ_i and p_1^{e*}

2) Stability region with losses due to both transmission errors and to congestion : Figure 7 shows the stability region for TCP-Reno and TCP-NC obtained from (50) with $N = 60$, and different homogenous redundancy factors ρ_i. Figure 7(a) is for losses only due to congestion, $p_1^{e^*} = 0.0$. Figure 7(b) is for $p_1^{e^*} = 0.1$. In both cases, one can observe a reduction of the size of the stability region for TCP-NC as ρ_i increases.

Now, one would expect, when wireless losses are exactly compensated by the added redundancy, to have a stability region close to that without losses and without added redundancy. Nevertheless, to combat losses, more packets are injected in the network, potentially increasing the congestion, especially when wireless losses occur far from the sources. The model (4) cannot take into account the thinning of the source rates due to losses and provides therefore an upper bound of the rate on the various links. This overestimation is reflected in the decrease of the stability region when the redundancy increases, independently of the presence of losses on the wireless part of the network.

Figure 8 shows the Nyquist diagram of $L_{NC}(j\omega)$ for various p^{e*}. These simulations use $N = 60$, $c = 12$ packets/ms, $d = 65$ ms, and $\rho = 1.1$. TCP-NC transits from stability to

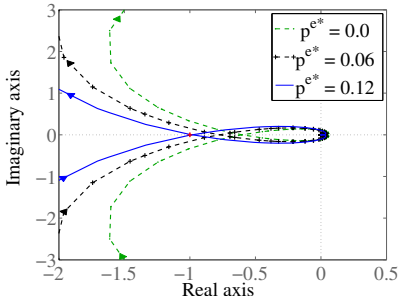


Fig. 8. Nyquist diagram for $N = 60$, $c = 12$ packets/ms, $\rho_i = 1.1$, $d = 65$ ms, and different values of p_1^{e*} when losses are due to both channel transmission errors and to congestion

instability when p^{e*} increases. In this case, a system originally stable with $p^{e*} = 0$ or $p^{e*} = 0.06$, becomes unstable for $p^{e*} \geqslant 0.12$. Similarly, Figure 9 shows the stability region of TCP-NC for $N = 60$, $\rho = 1.1$, and different values of p_1^{e*}. The size of the stability region of TCP-NC also decreases for increasing values of p_1^{e*}.

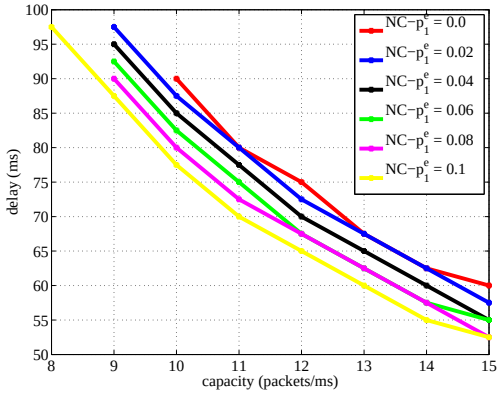


Fig. 9. Losses due to both channel transmission errors and to congestion. Critical stability region of TCP-NC with $\rho_i = 1.1$, $N = 60$ and different values of p_1^{e*}

The reduction on the local stability region of TCP and TCP-NC due to channel transmission errors p_1^e around the equilibrium point may be explained as follows. For a fixed value of ρ, the equilibrium queue length b_1^* decreases as p_1^e increases. Then, because of (14), p_1^{c*} decreases for increasing values of p_1^{e*}, leading to decreasing values of q^*. Due to the fact that f in (16) is a monotonic decreasing function, $x^* = f(\rho, q^*)$, disturbances around a smaller value of q^* causes TCP-NC to produce larger variations around an increased x^*. These oscillations, amplified by ρ, explain why the system reaches faster the instability regime. Thus, the size of the local stability region decreases for increasing values of p_1^e and ρ, as shown in Figures 6 to 9.

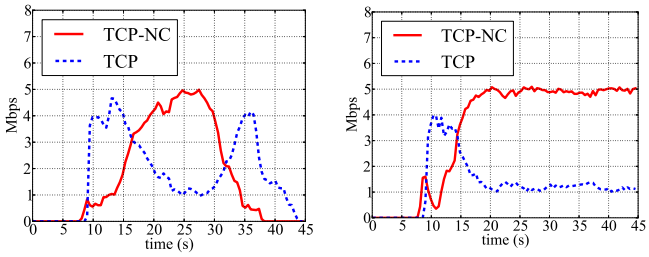
B. Interaction between TCP-Reno and TCP-NC flows

Consider N identical sources per protocol sharing a single bottleneck link with capacity c. Each source uploads a 10 MB file to a server using FTP. In the simulation we consider $N = 1$

to $N = 5$ sources per protocol, $c = 10$ Mbits/s, and $d_i = 40$ ms. TCP-Reno sources transfer the file to a common FTP server and TCP-NC sources to a FTP-NC server. Again RED with ECN marking is used as AQM with the parameters used in Section VI-A.

In a first simulation, $\rho = 1.2$ is used for all TCP-NC sources and $p_1^{e*} = 0$. Ethernet is used at the MAC layer, and the start time of all FTP-NC sessions follows a uniform distribution between 0 and 5 s. Results are averaged over 100 realizations. Figures 10(a) and 10(b) represent the average trajectory of the rate per protocol with $N = 1$ and $N = 5$ sources/protocol, respectively.

Most of the time, the TCP-NC flows consumes more bandwidth than TCP-Reno flows, showing the lack of fairness of TCP-NC. The NC layer masks part of the RED losses before reaching congestion, which improves the throughput compared to plain TCP Reno. Increasing the number of users N in the network increases the stability of the closed-loop system, as shown in the throughput trajectories in Figures 10(a) and 10(b), and in the stability region represented in Figure 7(a).



(a) $N = 1$ source per protocol (b) $N = 5$ sources per protocol

Fig. 10. Scenario without channel transmission errors; Average rate allocation per protocol , $\rho = 1.2$, $p_1^{e*} = 0$, $d = 40$ ms, and $c = 10$ Mbps

Table II provides the average TCP throughput (time and ensemble average) for different values of N. TCP-NC flows always consume more bandwidth than TCP-Reno flows, a consequence of the use of redundancy to mask losses, as previously explained.

TABLE II
INTERFAIRNESS BETWEEN TCP AND TCP-NC

Flows/protocol N	Average TCP Throughput TCP (Mbps)	TCP-NC (Mbps)	Total (Mbps)
1	1.83	2.11	3.94
2	1.38	3.31	4.69
3	1.37	3.37	4.74
4	1.33	3.42	4.75
5	1.29	3.45	4.74

In a second set of simulations, losses due to both transmission errors and to congestion are considered. $N = 5$ flows per protocol are considered and $\rho = 1.1$. The other parameters are the same as before. Several situations are considered, namely transmission errors on the flows sent by TCP-NC sources and TCP-Reno sources. We consider the cases of transmission errors on the flows (*i*) before or (*ii*) after the bottleneck link. This distinction is difficult to analyze theoretically using the tools of Section IV, since the thinning of the rate due to losses is not fully taken into account. For all scenarios, losses due to the finiteness of the field size are also considered. Since NC operations are in $GF\left(2^8\right)$, dependent linear combinations among $w_m = 2$ packets are obtained with a probability $1/2^8 \simeq 4.10^{-3}$. The additional loss probability is added to the aggregate loss probability of the link reaching each sink.

TABLE III
INTERFAIRNESS BETWEEN TCP AND TCP-NC.

Flows/protocol $N = 5$	Average TCP Throughput TCP (Mbps)	TCP-NC (Mbps)	Total (Mbps)
Channel losses for both flows			
	0.28	4.12	4.40
Channel losses for flows generated by TCP-NC sources only			
before bottleneck	2.69	2.67	5.36
after bottleneck	2.57	2.93	5.50

Figure 11 shows the bandwidth allocation when transmission errors affect both TCP-Reno and TCP-NC flows ($p_1^{e*} = 0.05$). As in Figure 10, TCP-NC sources get more bandwidth than TCP-Reno sources because of the use of redundancy which masks transmission errors, see also Table III.

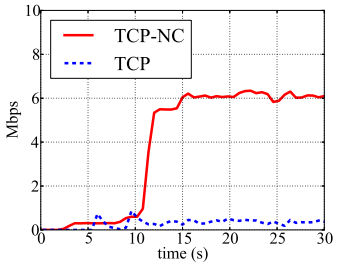
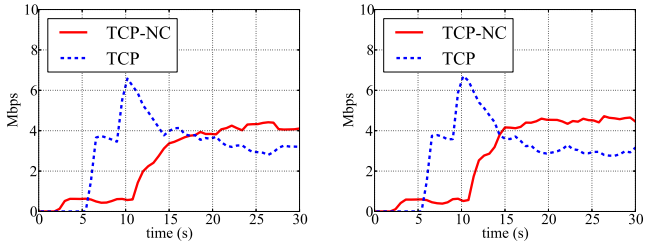


Fig. 11. Losses due to both channel transmission errors and to congestion for TCP and TCP-NC sources. Average rate allocation per protocol for $N = 5$ source per protocol, $\rho_i = 1.1$, $p_1^{e*} = 0.05$, $d = 40$ ms, and $c = 10$ Mbps

Figure 12 shows the results when transmission errors affect only flows generated by TCP-NC sources before or after the bottleneck link ($p^{e*} = 0.05$ in both cases). After some transients, TCP-NC sources use more bandwidth than plain TCP sources. The fairness is improved when losses are encountered before the bottleneck link (Figure 12(a)) compared to a situation where they are after (Figure 12(b)), see also Table III. When the channel losses are before the bottleneck link, the redundancy introduced by TCP-NC sources compensates more or less the losses due to transmission errors, but is not sufficient to mask losses due to RED. In this case, TCP-NC is almost fair. When the losses are after the bottleneck link, the flows of TCP-NC sources are not yet thinned. Losses introduced by RED have thus more impact on TCP-Reno flows than on TCP-NC flows, which may compensate part of them, resulting in a slightly degraded fairness.

C. Redundancy adaptation of TCP-NCAR

Consider now $N = 20$ sources running an upload FTP persistent session to a server using TCP-NCAR. The topology is represented in Figure 2. The sources share a single bottleneck link with capacity $c = 8$ packets/ms and round-trip propagation delay of $d = 1$ ms. The packet size is $\kappa = 1000$



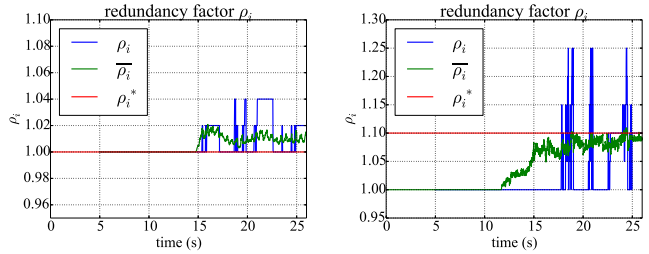
(a) Before bottleneck link (b) After bottleneck link

Fig. 12. Losses due to both channel transmission errors and to congestion; channel losses for TCP-NC sources. Average rate allocation per protocol for $N = 5$ sources per protocol, $\rho_i = 1.1$, $p_1^{e*} = 0.05$, $d = 40$ ms, and $c = 10$ Mbps

bytes, and all the routers in the topology use the same RED parameters already mentioned. The parameters of TCP-NCAR are $a_1 = a_2/2$ s/packets and $a_2 = 5.10^{-3}$ s.

Figure 13 shows the evolution of ρ_i for different values of p_1^{e*}. Figure 13(a), for which $p_1^{e*} = 0$, shows a single realization of ρ_i, the expected equilibrium value ρ^*, and an ensemble averages over 40 realizations for $\overline{\rho}_i$. One observes that $\overline{\rho}_i$ remains close to the expected equilibrium values $\rho^* = 1.0$. In Figure 13(b), $p_1^{e*} = 0.1$. The ensemble average $\overline{\rho}_i$ eventually approaches the expected equilibrium value $\rho^* = 1.1$.

These results confirms that TCP-NCAR is able to distinguish the cause of losses and to adjust ρ appropriately.



(a) $p_1^{e*} = 0.0$ (b) $p_1^{e*} = 0.1$

Fig. 13. Evolution of TCP-NCAR redundancy factor with $N = 20$, $a_2 = 5.10^{-3}$ s, $a_1 = a_2/2$ s/packets, $c = 8$ packets/ms and $d = 1$ ms

VII. 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 network involving TCP-NC-RED. The theoretical model and simulation results have shown that TCP-NC becomes unstable when the network scales up in delay and capacity, as standard TCP-Reno does, but additionally that the local stability region of TCP-NC reduces, compared to standard TCP-Reno, when the redundancy factor or channel losses increase. We developed a sufficient stability condition for the case of a single link with heterogeneous sources as a function of the delay and the redundancy factor per source, and the shape of the stability region of TCP-NC-RED was also presented. We proved that if TCP-NC and TCP-Reno share the same links, TCP-NC is TCP fair only when no redundancy is added by the source, and this fact was illustrated through simulations. Fairness is observed experimentally when both types of flows share only part of the links of a network and when the redundancy factor of TCP-NC is perfectly tuned to compensate losses due to channel transmission errors.

Refining the model (4) to better take into account the thinning of the rate due to losses would clearly be useful in this work. This goes beyond classical control-oriented network models, see [5], [15], [17], [25], [26]. This difficult study will be considered for future work.

We will also study the equilibrium and stability properties of TCP-NCAR extending results from [21].

References

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[4] L. Brakmo and L. Peterson, "TCP Vegas: end-to-end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, pp. 1465–1480, oct 1995.

[5] L. Chen, T. Ho, M. Chiang, S. Low, and J. Doyle, "Congestion control for multicast flows with network coding," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5908–5921, 2012.

[6] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno and SACK TCP," *SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 3, pp. 5–21, Jul. 1996. [Online]. Available: http://doi.acm.org/10.1145/235160.235162

[7] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, Jan. 2006. [Online]. Available: http://doi.acm.org/10.1145/1111322.1111337

[8] V. Jacobson, "Congestion avoidance and control," *SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, 1988.

[9] 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*, vol. 49, no. 3, pp. 237–252, 1998. [Online]. Available: http://dx.doi.org/10.2307/3010473

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

[11] M. Kim, M. Médard, and J. A. Barros, "Modeling network coded TCP throughput: a simple model and its validation," in *Proc. VALUETOOLS*, Cachan, France, 2011, pp. 131–140. [Online]. Available: http://dl.acm.org/citation.cfm?id=2151688.2151704

[12] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: Utility functions, random losses and ecn marks," in *In Proceedings of IEEE Infocom*, 2000, pp. 1323–1332.

[13] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371 –381, feb. 2003.

[14] S. Low, "A Duality Model of TCP and Queue Management Algorithms," *IEEE/ACM Transactions on Networking*, vol. 11, no. 4, pp. 525–536, 2003.

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

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

[17] S. H. Low, F. Paganini, J. Wang, and J. C. Doyle, "Linear stability of TCP/RED and a scalable control," *Computer Networks*, vol. 43, no. 5, pp. 633 – 647, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128603003049

[18] S. H. Low, L. L. Peterson, and L. Wang, "Understanding TCP Vegas: A Duality Model," *SIGMETRICS Perform. Eval. Rev.*, vol. 29, no. 1, pp. 226–235, Jun. 2001. [Online]. Available: http://doi.acm.org/10.1145/384268.378787

[19] D. E. Lucani, M. Médard, and M. Stojanovic, "Random linear network coding for time-division duplexing: field size considerations," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–6.

[20] F. Martignon and L. Fratta, *Loss Differentiation Schemes for TCP over Wireless Networks*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3375, pp. 586–599.

[21] H. Medina, M. Kieffer, and B. Pesquet-Popescu, "Redundancy adaptation scheme for network coding with TCP," in *Proc. IEEE Internation Symposium on Network Coding - IEEE NetCod 2012*, MIT, Cambridge MA., USA, Jun. 2012, pp. 49 – 54.

[22] S. Shakkottai and R. Srikant, "Network Optimization and Control," *Found. Trends Netw.*, vol. 2, no. 3, pp. 271–379, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1561/1300000007

[23] J. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network Coding Meets TCP: Theory and Implementation," *Proceedings of the IEEE*, vol. 99, no. 3, pp. 490 –512, March 2011.

[24] J. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros, "Network Coding Meets TCP," in *Proc. IEEE INFOCOM 2009*, April 2009, pp. 280–288.

[25] A. Tang, J. Wang, S. Hegde, and S. Low, "Equilibrium and fairness of networks shared by TCP Reno and Vegas/FAST," *Telecommunication Systems*, vol. 30, no. 4, pp. 417–439, 2005.

[26] A. Tang, X. Wei, S. H. Low, and M. Chiang, "Equilibrium of Heterogeneous Congestion Control: Optimality and Stability," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 844–857, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1109/TNET.2009.2034963

APPENDIX

A. Derivation of (9)

From here on, we neglect the probability of receiving dependent linear combinations due to the finiteness of the field in which NC operations are done.

To evaluate $\mathbb{E}[R_{\text{NACKs}}]$ we use the following approach. At time t, source i sends packets at the output of the NC layer at a rate $\rho_i x_i(t)$ packets/s. We assume that source i observes ACKs with a delay $\tau_i(t)$.Only $x_i(t)$ packets/s are needed to perform NC decoding and $(\rho_i-1)x_i(t)$ packets/s are redundant packets added to combat losses. Let $(Z_k)_{k\geq 0}$ be a Bernoulli process to indicate whether the k-th packet sent by user i has been dropped. The k-th packet is dropped with probability $q_i = \Pr(Z_k = 1)$, and successfully received with probability $1 - q_i = \Pr(Z_k = 0)$. Let M be the aggregate number of losses per second, $M = \sum_{k=1}^{\rho_i x_i^{\text{d}}(t)} Z_k$. Source i receives $v > 0$ NACKs when $M = (\rho_i-1)x_i^{\text{d}}(t)+v$. Using these assumptions, the rate of NACKs is

$$R_{\text{NACKs}} = \begin{cases} 0, & \text{if } M \leqslant (\rho_i - 1)\, x_i^{\text{d}}(t) \\ M - (\rho_i - 1)x_i^{\text{d}}(t), & \text{else.} \end{cases} \quad (52)$$

Taking the expectation of (52), one gets

$$\mathbb{E}\left[R_{\text{NACKs}}\right] = \sum_{k=(\rho_i-1)x_i^{\text{d}}(t)+1}^{\rho_i x_i^{\text{d}}(t)} \left[k - (\rho_i - 1)\, x_i^{\text{d}}(t)\right] \Pr\left(M = k\right), \quad (53)$$

where M, the sum of Bernoulli variables, follows a binomial distribution

$$\Pr(M = k) = \binom{\rho x_i^{\text{d}}(t)}{k} \left(q_i(t)\right)^k \left(1 - q_i(t)\right)^{\rho_i x_i^{\text{d}}(t) - k}, \quad (54)$$

with

$$\mathbb{E}[M] = \sum_{k=0}^{\rho_i x_i^{\text{d}}(t)} k \Pr(M = k) = \rho_i x_i^{\text{d}}(t) q_i(t). \quad (55)$$

Then (9) is obtained by combining (53) and (54).

B. Approximation of (9)

Two approximations for $\mathbb{E}\left[R_{\text{NACKs}}\right]$ will be evaluated to be used in (8). A first for small values of q_i, and a second for large values of q_i. To lighten notations, the subscript i indicating the source is omitted, as well as the dependency in t of $q_i(t)$, $x_i^{\text{d}}(t)$, and of $\tau_i(t)$.

For q small, the term with exponent $k = (\rho - 1)x^{\text{d}} + 1$ dominates in the sum of (9) and one obtains the following approximation

$$\mathbb{E}\left[R_{\text{NACKs}}\right] \approx \binom{\rho x^{\text{d}}}{(\rho - 1)x^{\text{d}} + 1} q^{(\rho-1)x^{\text{d}}+1}. \quad (56)$$

For q large, *i.e.*, close to 1, one may rewrite (53) as

$$\mathbb{E}\left[R_{\text{NACKs}}\right] = \sum_{k=0}^{\rho x^{\text{d}}} k \Pr(M = k) - (\rho - 1)x^{\text{d}} \times \left(1 - \sum_{k=0}^{(\rho-1)x^{\text{d}}} \Pr(M = k)\right) - \sum_{k=0}^{(\rho-1)x^{\text{d}}} k \Pr(M = k). \quad (57)$$

Substituting (55) in (57), one gets

$$\mathbb{E}\left[R_{\text{NACKs}}\right] = x^{\text{d}} q - (\rho - 1)x^{\text{d}}\left(1 - q\right) - \sum_{k=0}^{(\rho-1)x^{\text{d}}} \left(k - (\rho - 1)x^{\text{d}}\right) \Pr(M = k). \quad (58)$$

Using (54), (57) becomes

$$\mathbb{E}\left[R_{\text{NACKs}}\right] = x^{\text{d}} q - (\rho - 1)(1 - q)x^{\text{d}} - \sum_{k=0}^{(\rho-1)x^{\text{d}}} \left[k - (\rho - 1)x^{\text{d}}\right] \binom{\rho x^{\text{d}}}{k} q^k \left(1 - q\right)^{\rho x^{\text{d}} - k}. \quad (59)$$

Consider $q = 1 - \delta$, with δ small. In this regime, in the sum of (59), the term $k = (\rho - 1)x^{\text{d}} - 1$ dominates and one gets

$$E\left[R_{\text{NACKS}}\right] \approx x^{\text{d}}\left(1 - \delta\right) - \left(\rho - 1\right)\delta x^{\text{d}} + \binom{\rho x^{\text{d}}}{(\rho - 1)x^{\text{d}} - 1} \delta^{x^{\text{d}}+1}. \quad (60)$$

Since δ is small, the last term in (60) is neglected to obtain

$$E\left[R_{\text{NACKS}}\right] \approx x^{\text{d}} q - \left(\rho - 1\right)\left(1 - q\right) x^{\text{d}}. \quad (61)$$

To study the dynamic of the source rate (8), the approximation (56) is quite difficult to manipulate. When q is small, it is very likely that $E\left[R_{\text{NACKS}}\right] \approx 0$, and the term $\frac{1}{w_i(t)}\mathbb{E}\left[R_{\text{ACKs}}\right]$ dominates in (8). The approximation $E\left[R_{\text{NACKS}}\right] = 0$ is thus reasonable for small q. When q is close to 0, (61) becomes negative. Thus, using (61), the approximation (10) of (9) is obtained.

C. Equilibrium of (8)

To compute the equilibrium of (8), assume that at equilibrium, q_i^* is large enough to have

$$\mathbb{E}\left[R_{\text{NACKs}}\right] = x_i^* q_i^* - (\rho_i - 1)(1 - q_i^*)x_i^* \geqslant 0 \quad (62)$$

in (10). Then, at equilibrium, (12) becomes

$$\frac{x_i^*}{w_i^*} - \left(\frac{1}{w_i^*} + \beta w_i^*\right)\left(x_i^* q_i^* - (\rho_i - 1)\left(1 - q_i^*\right) x_i^*\right) = 0. \tag{63}$$

Using the fact that at equilibrium, $x_i^* = w_i^*/\tau_i^*$, one gets (24). Now, to check whether (62) is satisfied, one has to prove that $\rho_i q_i^* \geqslant (\rho_i - 1)$. Substituting q_i^* with (24), one obtains

$$\frac{1 + (\rho_i - 1)\left(1 + \beta\left(x_i^* \tau_i^*\right)^2\right)}{1 + \beta\left(x_i^* \tau_i^*\right)^2} \geqslant 1, \tag{64}$$

which is true provided that $\rho_i \geqslant 1$.

PLACE PHOTO HERE

Hamlet Medina Ruiz Hamlet Medina received the B.S. degree in Electronic Engineering from the University Dr Rafael Belloso Chacín, Venezuela, in 2005, the M. Sc. degrees in pattern recognition and statistical data analysis from Télécom-SudParis and the University of Versailles-Saint-Quentin-en-Yvelines, respectively, in 2010, and the Ph.D. degree in Physics from the University Paris-Sud, in 2014. He worked as a control process engineer for Petróleos de Venezuela (PDVSA), from 2005 till 2008. His research interests include network coding, control and optimization of networks and protocols, time series analysis, and machine learning. He is currently working as a Data Scientist at Criteo.

PLACE PHOTO HERE

Michel Kieffer (IEEE M'02 - SM'07) is a full professor in signal processing for communications at the Paris-Sud University and a researcher at the Laboratoire des Signaux et Systèmes, Gif-sur-Yvette. Since 2009, he is also part-time invited professor at the Laboratoire Traitement et Communication de l'Information, Télécom ParisTech, Paris.

His research interests are in signal processing for multimedia, communications, and networking, distributed source coding, network coding, joint source-channel coding and decoding techniques, joint source-network coding. Applications are mainly in the reliable delivery of multimedia contents over wireless channels. He is also interested in guaranteed and robust parameter and state bounding for systems described by non-linear models in a bounded-error context.

Michel Kieffer is co-author of more than 150 contributions in journals, conference proceedings, or books. He is one of the co-author of the book *Applied Interval Analysis* published by Springer-Verlag in 2001, (this book was translated in Russian in 2005) and of the book *Joint source-channel decoding: A crosslayer perspective with applications in video broadcasting* published by Academic Press in 2009. He serves as associate editor of Signal Processing since 2008 and of the IEEE transactions on Communications since 2012.

In 2011, Michel Kieffer became junior member of the Institut Universitaire de France.

PLACE PHOTO HERE

Béatrice Pesquet-Popescu received the engineering degree in telecommunications from the *Politehnica* Institute in Bucharest in 1995 (highest honours) and the Ph.D. thesis from the Ecole Normale Supérieure de Cachan in 1998. In 1998 she was a Research and Teaching Assistant at Université Paris XI and in 1999 she joined Philips Research France, where she worked during two years as a research scientist, then project leader, in scalable video coding. Since Oct. 2000 she is with Télécom ParisTech (formerly, ENST), first as an Associate Professor, and since 2007 as a Full Professor, Head of the Multimedia Group. She is also the Scientific Director of the UBIMEDIA common research laboratory between Alcatel-Lucent Bell Labs and Institut Mines Télécom.

Béatrice Pesquet-Popescu was appointed as IEEE Fellow at the age of 41 (youngest woman IEEE Fellow in France) and is a member of the FET Advisory Board of the European Commission for the H2020 program, as well as member of the Advisory Group on Part IV of Horizon 2020, *Spreading Excellence and Widening Participation* (2014-2016). She was an EURASIP BoG (Board of Governors) member between 2003-2010, and an IEEE Signal Processing Society IVMSP TC (2008-2013) and IDSP SC (2010-2015, Vice-Chair 2012) member and an MMSP TC associate member. In 2013-2014 she served as a Chair for the Industrial DSP Standing Committee. She was an IEEE SPS Multimedia Signal Processing (MMSP) TC member between 2006-2009. She is also a member of the IEEE Comsoc Technical Committee on Multimedia Communications (2009-2014, chair of the Awards Board, 2015). In 2008-2009 she was a Member at Large and Secretary of the Executive Subcommittee of the IEEE Signal Processing Society (SPS) Conference Board, and she was (2012-2014) a member of the IEEE SPS Awards Board and of the IEEE SPS Conference Board (2015-2016).

Beatrice Pesquet-Popescu served as an Editorial Team member for IEEE Signal Processing Magazine, as an Associate Editor for IEEE Trans. on Multimedia, IEEE Trans. on Circuits and Systems for Video Technology, IEEE Transactions on Image Processing, Area Editor for Elsevier Image Communication journal, Associate Editor for APSIPA Transactions on Signal and Information Processing, and Associate Editor for the Hindawi Int. J. Digital Multimedia Broadcasting journal and for Elsevier Signal Processing. She was a Technical Co-Chair for the PCS 2004 conference, and General Co-Chair for IEEE SPS MMSP2010, EUSIPCO 2012, and IEEE SPS ICIP 2014 conferences. Béatrice Pesquet-Popescu is a recipient of the *Best Student Paper Award* in the IEEE Signal Processing Workshop on Higher-Order Statistics in 1997, of the Bronze Inventor Medal from Philips Research and in 1998 she received a *Young Investigator Award* granted by the French Physical Society. In 2006, she was the recipient, together with D. Turaga and M. van der Schaar, of the IEEE Trans. on Circuits and Systems for Video Technology *Best Paper Award*. In April 2012, Usine Nouvelle cited her among the *100 who matter in the digital world* in France.

She holds 23 patents in video coding and has authored more than 320 book chapters, journal and conference papers in the field. She is a co-editor of the book *Emerging Technologies for 3D Video: Creation, Coding, Transmission, and Rendering*, Wiley Eds., 2013. She has (co-)directed 27 PhD students and was in 30+ PhD committee defense juries.

Her current research interests are in source coding, scalable, robust and distributed video compression, multi-view video, 3DTV, holography, sparse representations and convex optimisation.