A Game Theoretic Comparison of TCP and Digital Fountain Based Protocols

Luis López

Universidad Rey Juan Carlos (Spain)

Antonio Fernández

Universidad Rey Juan Carlos (Spain)

Vicent Cholvi

Universitat Jaume I, Campus del Riu sec, 12071 Castellón (Spain)

Abstract

In this paper we analyze a novel paradigm of reliable communication which is not based on the traditional timeout-and-retransmit mechanism of TCP. Our approach, which we call FBP (Fountain Based Protocol), consists of using a digital fountain encoding which guarantees that duplicate packets are almost impossible. By using Game Theory, we analyze the behavior of TCP and FBP in the presence of congestion. We show that hosts using TCP have an incentive to switch to an FBP approach, obtaining a higher goodput. Furthermore, we also show that a Nash equilibrium occurs when all hosts use FBP (i.e., when FBP hosts act in an absolutely selfish manner injecting packets into the network as fast as they can and without any kind of congestion control approach). At this equilibrium, the performance of the network is similar to the performance obtained when all hosts comply with TCP. Regarding the interaction of hosts using FBP at different rates, our results show that the Nash equilibrium is reached when all hosts send at the highest possible rate, and, as before, that the performance of the network in such a case is similar to the obtained when all hosts comply with TCP.

Key words: Protocol Analysis, Digital Fountain Codes, Game Theory.

⋆ A preliminary version of this paper appeared in the Proceedings of the IEEE Symposium on Computers and Communications, ISCC 2005. This work was partially supported by the Spanish Ministry of Science and Technology under Grants No. TSI2006-07799, No. TSI2004-02940 and No. TIN2005-09198-C02-01, by Bancaixa under Grant No. P1-1B2003-37 and by the Comunidad de Madrid under Grant No. S-0505/TIC/0285.

1 Introduction

Congestion control in communication systems has been an important and largely studied issue. Since many communication systems in our days are based on the principle of sharing common resources (e.g., routers, communication links) among different users, one of the main objectives of congestion control schemes is to establish rules to guarantee that the common resources are used optimally and shared fairly among users. However, most of these schemes require end-users to behave in a cooperative way. Users have to respect some "socially responsible" rules. For instance, the TCP (which is, by far, the most widely used protocol) congestion control scheme is voluntary in nature and critically depends on end-user cooperation. Indeed, TCP congestion control algorithms [1–6] voluntarily reduce the sending rate upon receiving some congestion signal such as ECN [7], packet loss [8–10], or source quench [11]. Such congestion control schemes are successful because all the end-users cooperate and voluntarily reduce their sending rates upon detection of congestion.

Nevertheless, it is currently impossible to guarantee that end-users will not act in a selfish manner. If they use TCP, this means that they will never reduce their sending rates even in the presence of congestion. As it has been shown in [12,13], if this happens and users overload the network, the total throughput of the network drops. This happens since most Internet routers use a drop-tail FIFO (First In First Out) scheduling discipline, and users can obtain more network bandwidth by transmitting more packets per unit of time. (With this policy, the more packets a user sends the more resources it gets.) Thus, the optimal strategy for each user is strongly suboptimal for the network as a whole.

Among the different techniques that can be used to evaluate the impact of selfish users, one of the most popular is *Game Theory* [14,15]. Game theory is a tool for analyzing the interaction of decision makers with conflicting interests. Roughly speaking, a *game* has three components: a set of players, a set of possible actions for each player, and a set of utility functions mapping action profiles into real numbers. In our case, the *game players* are the users and the congestion control schemes establish the *game rules*. Each player has a strategy, which establishes the traffic that it injects into the network.

The behavior of the TCP protocol has already been addressed with a game-theoretic approach by several authors. Some of the most remarkable works in this field are the ones carried out by Nagle [12,16], and Garg et al. [17]. Both of them show that evil (selfish) behavior leads to disaster and propose solutions based on creating incentive structures in the systems that discourage this behavior. Nagle suggests replacing the single FIFO queue associated to each outgoing link with multiple queues, one for each source host, which are served in a round-robin fashion. Garg et al. introduce a novel and sophisticated scheduling discipline called RIS (Rate Inverse Scheduling) that punishes evil behavior and rewards cooperation, in such a way that the resulting Nash equilibrium [1] leads to a fair allocation of resources. Both solutions require a significant (sometimes huge) per-packet processing, which might be impractical in many realistic applications (as in Internet core routers, for example). Another interesting work based on slightly different ideas is the one carried out by Akella et al. [13]. In this paper, a combination of analysis and simulations is carried out trying to characterize the performance of TCP in the presence of selfish users. The study covers different variations of TCP (Reno, SACK, etc.) and buffer management policies (Drop Tail, RED, etc.), showing that the most recent variations of TCP may become very inefficient in the presence of selfish behavior. Nevertheless, they show that a novel stateless buffer scheduling discipline called CHOKE [18], which does not require per-packet processing, may be useful in restoring the Nash equilibrium efficiency. There are other interesting proposals related to problems similar to this [19–22]. In all cases, these works show the potential applications of Game Theory within the problem of congestion control and routing in packet networks.

The abovementioned problem has a closer analogue with the, so called, Tragedy of the Commons [23] problem in economics. In this problem, each individual can improve her own position by using more of a free resource, but the total amount of the resource degrades as the number of users increases. Historically, this analysis was applied to the use of common grazing lands, but it also applies to such diverse resources as air quality and time-sharing systems. In general, experience indicates that multiplayer systems with this type of instability tend to go into serious trouble. To understand precisely what a Tragedy of the Commons is, we need first to observe that, in the context of Game Theory, players choose their strategy in a selfish way trying to maximize their benefit. If the system gets into a state in which no player has an incentive to unilaterally change its strategy we say that the system has reached the Nash equilibrium. In this context, a game is a Tragedy of the Commons when (i) there is always an incentive for a new player to become evil (this guarantees that the Nash equilibrium is reached when all players are evil) and (ii) the final benefit for evil players in the Nash equilibrium is under the initial benefit of fair players when all players collaborate. This definition guarantees the essential ingredient of a Tragedy of the Commons: if players behave in a selfish way, the Nash equilibrium will be reached, and hence, the benefit of the defectors will always be less than the initial reward of the fair players. Hence, all players lose. In the context of network protocols, it has been observed by

[1] An important concept in game theory is the Nash equilibrium. In our context, a Nash equilibrium is a scenario where no selfish user has incentive to unilaterally deviate from its current state. Clearly, being in a Nash equilibrium means that we are in a stable state in the presence of selfish users.

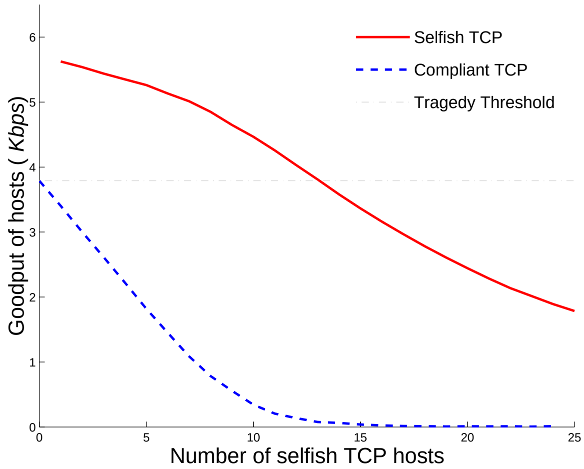

Fig. 1. This picture shows a typical Tragedy of the Commons scenario obtained by simulating with NS2 a single-bottleneck scenario with line capacities of $C = 100$ Kbps and a finite router buffer of 10 packets. Hosts can choose their strategy to be fair-cooperative (which comply with the TCP protocol) and evil-selfish (which implements a modified version of the TCP protocol in which the retransmission timeout is fixed to 0.4 seconds). The picture shows the throughput of fair and evil-selfish players as a function of the number N_e of selfish-evil hosts. As it can be observed, for any value of N_e, a given fair host always has an incentive to become evil. This implies that the Nash equilibrium is reached when all hosts are evil. Observe that the throughput in this equilibrium is remarkably smaller than the initial throughput of fair TCP hosts. Hence, the selfish strategy drives the game into a less efficient situation than the one obtained when all hosts cooperate. For this reason we say that a Tragedy of the Commons takes place.

several authors [12,13,24] that when hosts behave in a selfish manner and do not comply with the TCP congestion control mechanisms (for example, by using lower timeouts), a Tragedy of the Commons arises and the network throughput drops due to the presence of duplicate packets. This effect can be easily observed in Fig. 1, which presents a simulation of a system like the one shown in Fig. 2.

In this paper we compare, from a game theoretic point of view, TCP with a protocol based on digital fountain codes [25,26], which we call Fountain Based Protocol (FBP). The Digital Fountain approach has already been proposed as an appropriate mechanism for TCP-like reliable data transfer in multicast environments [27]. Moreover, suitable congestion control algorithms have been proposed to make these flows work in a TCP-friendly manner [28]. In this paper, we dig into these concepts, providing the following additional contributions:

- We propose an FBP for one-to-one reliable data transfer. This protocol is similar to UDP but uses fountain codes to avoid the presence of duplicate packets. Because of this, it does not require any type of packet retransmission mechanism. Contrary to UDP, FBP guarantees that the original source data will be correctly delivered, regardless of whether there are packet losses or not.

- Then, we establish a theoretical framework suitable for the analysis of this interaction of FBP and TCP. Under this framework, we show that users always have an incentive to switch from TCP to FBP. Furthermore, we validate the theoretical framework and results through simulations.
- We show that the Nash equilibrium of a network with a mixture of hosts using TCP and hosts using FBP is reached when all hosts behave in a selfish manner (by using FBP instead of TCP), but that this does not drive the network to a collapse. Moreover, we demonstrate that, in general, it does not even lead to a Tragedy of the Commons, since the throughput of hosts, even in the case where all of them act in a selfishly way, is no less than the throughput obtained when all host comply with the TCP protocol.
- Finally, we also study the interaction of hosts using FBP at different rates. Our results show that the Nash equilibrium is reached when all hosts send at the highest possible rate, and, as before, that this does not lead to a Tragedy of the Commons.

In the next section we present the details of the protocol FBP. In Section 3 we present the network model we use, with some analytical results under that model. In Section 5 we present simulations of the same network and compare them with the previous analysis. In Section 6 we analyze systems with only FBP hosts. Finally, in Section 7 we present some concluding remarks.

2 Protocols Based on Digital Fountain Codes

The basic principle behind the use of *digital fountain codes* [25,29,26] is conceptually simple. Roughly speaking, it consists of generating a stream of different encoded packets into the network, from which it is possible to reconstruct the source data. The key property is that the source data can be reconstructed from any subset of the encoded packets of (roughly) the same size as the source data. Such a concept is similar to ideas found in the seminal works of Maxemchuk [30] and Rabin [31].

A class of codes that satisfy the above mentioned property are classical *erasure codes*. Erasure codes generate additional redundant packets from the original k packets of the source data. Then, they guarantee that the source data can be recovered from any subset of $(1+\varepsilon)k$ packets ($1+\varepsilon$ is called the *decoding inefficiency*). Hence, they allow to tolerate packet losses during transmissions. For instance, one can use Reed-Solomon erasure codes [32], since they have the property that a decoder at the receiver can reconstruct the original source data whenever it receives any k of the transmitted packets (i.e., their decoding inefficiency is 1). However, the encoding and decoding processing times for such a class of codes are prohibitive.

Digital fountain codes can be seen as a kind of erasure codes with very fast encoding and decoding. Furthermore, the number of encoded packets that can be generated from the source data by using these codes is potentially limitless and does not need to be fixed ahead. That allows a digital fountain code to take source data consisting of k packets and produce as many encoding packets as needed to meet the user demand. The only drawback is that these codes have a decoding inefficiency a little larger than 1 (i.e., $\varepsilon > 0$).

Fountain Based Protocols use digital fountain codes to appropriately encode data to be transfered. Whenever a file has to be transmitted, a digital fountain encoder is used to continuously generate encoded packets. These packets are injected into the network, by the sender, at a given rate. On its turn, when the receiver has enough packets to reconstruct the source data, it sends a `stop` message to the sender. That is, the FBP does not require any kind of congestion control mechanism. Furthermore, it does not make use of packet retransmissions. The only "overhead" are the packets injected in the time interval since the receiver sends the `stop` message until the sender receives it. We note that, in order to increase performance in real scenarios, this simple protocol can be improved in a number of ways (see [33] for an overview regarding this issue).

For simplicity, in the next sections we will assume that the decoding inefficiency of the used codes is 1. In subsequent sections, we will analyze the effects and consequences of having $\varepsilon > 0$. Current implementations of digital fountain codes can guarantee an inefficiency of about 1.054 [29] and even less than that [25,26] (up to 1.02). We will also assume that the rate at which senders inject packets is constant (i.e., it is CBR).

3 A model for the interaction between TCP and FBP

To understand the interaction between TCP and FBP, we use the traditional single-bottleneck problem, in which a communication line is shared between N different hosts, as depicted in Fig. 2.

In our analysis, we assume that time is discrete and structured as a sequence of consecutive rounds, where each round is a group of $S \geq N$ consecutive slots. All communication lines are assumed to have the same capacity, fixed to one packet per slot, and all packets have the same size. Hosts are assumed to be greedy (i.e., they always wish to send new packets to the destination). The router is assumed to have a finite buffer so that, when congestion occurs

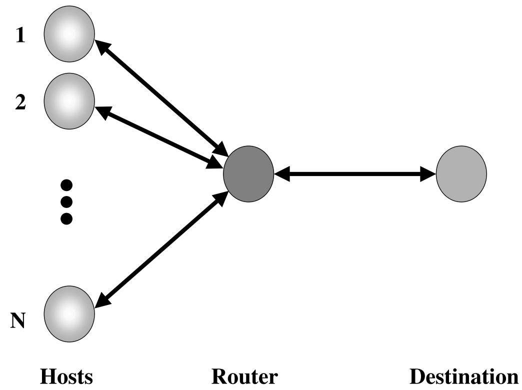

Fig. 2. The picture shows the interconnecting topology of the network we use in the proposed model. N hosts try to access a common communication link through a router. This router has a finite buffer which drops packets when it is full. All lines have the same capacity C.

and the buffer is full, new incoming packets are dropped.

Hence, we have a traditional Game Theory problem in which N different players (the hosts) compete for a common resource (the shared line and the router buffer) trying to obtain the maximum yield (the goodput). In this context, we assume that our hosts are free to choose between two different strategies when transmitting their packets. The first strategy is to comply with a given communication protocol suitable to solve the congestion problem of the single-bottleneck link. This protocol must be designed to fairly share the resources among the hosts. For this reason, following the traditional Game Theory notation, we say that players obeying this protocol are *fair*. On the other hand, hosts can adopt a different strategy consisting of sending packets as soon as they are ready and not complying with any given protocol designed to avoid congestion. These hosts will be called *evil* because they do not obey the established rules guaranteeing fairness in the game. Observe that, in a realistic communication environment, hosts using TCP could be considered as fair, while hosts using FBP would be evil because they do not take into account any congestion control mechanism. Thus, we say that TCP is an ordering protocol (in the sense that it enforces a set of fixed and known rules), while FBP is a disordered protocol (in the sense that it does not enforce any coordination).

Following, we describe the two protocols we will use:

The ordering protocol (TCP) The ordering protocol we use emulates the two main characteristics of TCP: resource sharing and congestion control. On one hand, it assigns one fixed exclusive slot to each host within each round, in which the host is allocated to send packets. Then, when all hosts use this ordering protocol, they transmit one packet per round, and this packet does not compete with any other to enter the router buffer. On the other hand, it implements a basic timeout-and-retransmit mechanism to control congestion. With this purpose, we introduce an acknowledgment scheme so that, whenever the destination receives a packet, it immediately generates an `ack`, which is sent back to the corresponding host in the subsequent time slot.

The disordered protocol (FBP) By using this protocol, hosts use some kind of digital fountain encoding which guarantees that duplicates are not possible and all packets reaching the destination are useful. As it has been said previously, a single `stop` message is sent at the end of the whole file transfer to indicate the sender that the transmission has ended. As a first approximation, we consider that the size of the files being exchanged is very large, and hence we disregard `stop` messages. We consider that hosts that use FBP transmit on all slots of the round with a given probability p. For simplicity, we assume that the value of p is the same for all hosts.

Observation 1 *Before continuing, we wish to remark that our model is not a totally realistic scenario where TCP and FBP could be competing. Actually, it is optimistic when estimating the fair (TCP based) yield, and pessimistic for the evaluation of the evil rates. The optimistic behavior occurs since our simplified ordering protocol does not react in any way when packets are lost, while current TCP implementations react to congestion by decreasing its offered load. In turn, the pessimistic behavior of FBP occurs since the decrease in the offered load of the TCP-based hosts would imply a higher probability for evil packets to get into the router buffer. Therefore, in our subsequent analysis, we will be using a scenario that penalizes FBP against TCP. In Section 5 this behaviour will be substantiated by means of experimental evaluation.*

4 Analysis of the TCP–FBP Interaction

From the previous section, our communication scheme is based on rounds of S slots, with two kinds of slots. First, N_f fair slots (F-slots), where one fair host always transmits and N_e evil hosts transmit with probability p. Second, $S - N_f$ evil slots (E-slots) where N_e evil hosts transmit with probability p.

Before we proceed with the analysis, we note that, as it has been shown in [24], in scenarios where at least one of the hosts does not use any kind of congestion control mechanism, with high priority the router buffer is always full. Then, in that congested situation, only one packet can enter the buffer in each time slot, because only one packet gets out of it in that interval. Therefore, the probability of a given evil host with selfishness degree p to get a packet in the congested buffer in an E-slot can be easily calculated. To do so, just note that if we consider a particular evil host, the probability that the other $N_e - 1$ send $i - 1$ packets to the router is given by a binomial distribution of the form $\binom{N_e-1}{i-1} p^{i-1}(1-p)^{N_e-i}$. As the considered host sends itself a packet with probability p, we have i packets trying to enter the router with probability $\binom{N_e-1}{i-1} p^{i}(1-p)^{N_e-i}$. Given that the router admission policy is fair, if there are i packets trying to occupy the single free buffer position, any of them can get to it with probability $1/i$. Hence, summing for all possible values of i, we have:

$$p_E^e(N_e, p) = \sum_{i=1}^{N_e} \frac{1}{i} \binom{N_e - 1}{i - 1} p^i (1-p)^{N_e - i}. \tag{1}$$

The probability of an evil host to get a packet into the buffer in an F-slot can be calculated in the same way, just noting that an additional fair host sends its packet with probability 1

$$p_F^e(N_e, p) = \sum_{i=1}^{N_e} \frac{1}{i+1} \binom{N_e - 1}{i - 1} p^i (1-p)^{N_e - i}. \tag{2}$$

With these results, we can evaluate the transmission rate for evil hosts R_e. Since we assume evil hosts use FBP, then all packets arriving to the destination (all packets getting into the router buffer) are useful. So,

$$R_e(N, N_e, p, S) = \frac{(S - N_f) p_E^e(N_e, p) + N_f p_F^e(N_e, p)}{S}. \tag{3}$$

For fair hosts the result is similar. First, the probability of a fair host to get its packet into the buffer in its F-slot is

$$p_F^f(N_e, p) = \sum_{i=0}^{N_e} \frac{1}{i+1} \binom{N_e}{i} p^i (1-p)^{N_e - i}. \tag{4}$$

Now, taking into account that fair hosts do not get packets into the buffer in E-slots, we can evaluate the transmission rate for fair hosts R_f. Namely,

$$R_f(N, N_e, p, S) = \frac{p_F^f(N_e, p)}{S}. \tag{5}$$

From the analysis of the transmission rates for evil and fair hosts (Equation 3 and 5), we can derive some interesting results.

Property 1 An optimal protocol controlling the congestion is just as good as letting all hosts to send their FBP packets as fast as possible.

PROOF. To prove this property, we analyze the form of the transmission rates for the two extreme situations. Namely, when all hosts are fair ($N_e = 0$) and when all hosts are evil $N_e = N$). The interesting fact is to remark that if $p = 1$ then $R_e(N, N, 1, S) = \frac{1}{N} \geq R_f(N, 0, 0, S)$, which means that the best goodput obtained when all hosts use an unordered protocol is over the one obtained when they try to access the common resource in an ordered way.

The key issue to understand why this happens is to observe that, when using FBP, all packets arriving to the destination are useful and duplicates are not possible. Many authors have remarked [13,16,24] that when using a timeout-and-retransmit based approach (as the one of TCP), if congestion and flow control algorithms are not respected by the hosts, the global throughput of the network drops due to the presence of duplicates, which are retransmitted when timeouts occur. Nevertheless, when using FBP, no duplicates are present and no timeouts are needed to ensure that the network is not collapsed by them.

Property 2 The Nash equilibrium of the game is reached when all hosts are evil.

PROOF. For the proof, let us assume that $p > \frac{2}{N+1}$. Then, it follows that $p > \frac{i+1}{Ni+n+1}$ for all $i \in \{1, ..., N\}$ and for all $n \in \{0, ..., N\}$. This can be seen by assuming a worst case ($n = 0$), and by observing that the inequality holds for $i = 1$ and that the expression on the right strictly decreases with i. The inequality can also be written as

$$\frac{n+1}{i} + \frac{N-n-1}{i+1} > \frac{1}{ip}. \tag{6}$$

Now, we define $f_i = \binom{n}{i-1} p^i (1-p)^{n+1-i}$. Observe that f_i is always positive. In this situation, we can multiply Eq. (6) by f_i without changing the inequality. Hence, summing all the inequalities for all i gives

$$\sum_{i=1}^{n+1} \frac{n+1}{i} f_i + \frac{N-n-1}{i+1} f_i > \sum_{i=1}^{n+1} \frac{1}{ip} f_i.$$

Observe that substituting f_i, making a change of variables in the second part of the inequality, dividing by N and recovering the original expressions of p_E^e, p_F^e and p_F^f from Eqs. (1,2,4), this can be written as

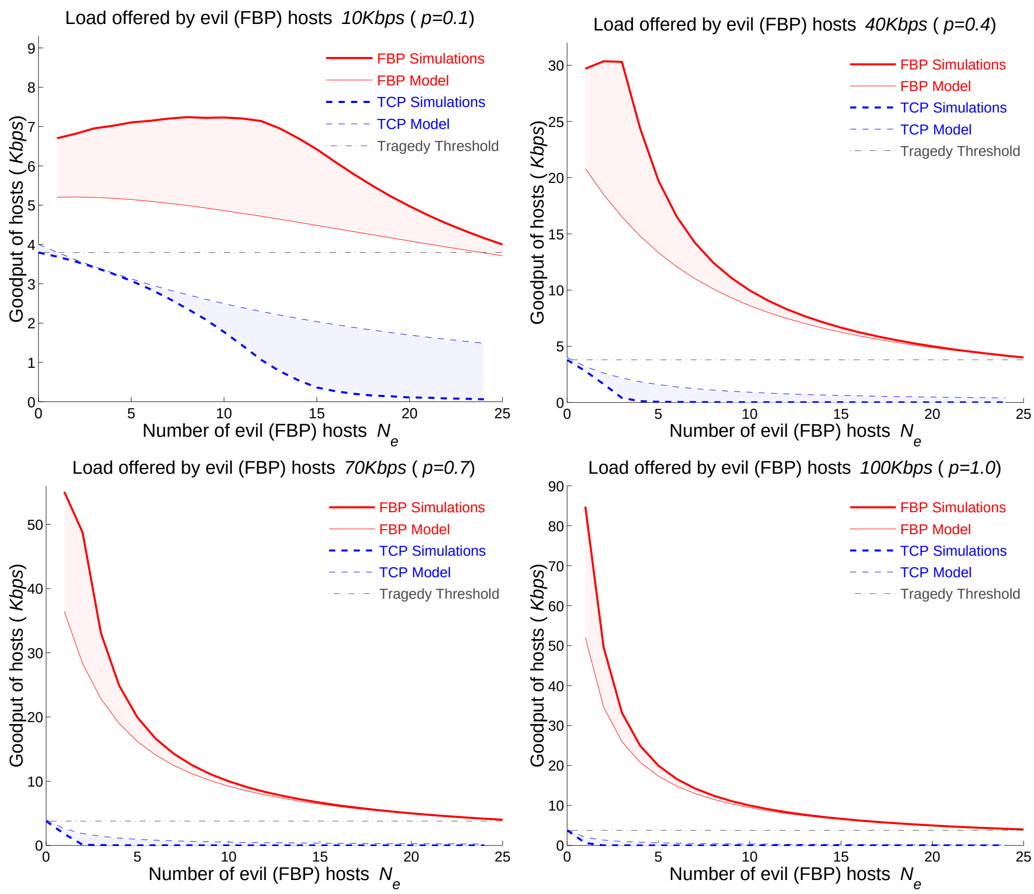

Fig. 3. The figure represents the goodput of evil (FBP) and fair (TCP) hosts as a function of the number of evil hosts N_e for four different values of evil selfishness. The shaded region represents the error between the theoretical model and the simulations. The horizontal dashed line indicates the simulated goodput of the TCP hosts when no evil players are present. All pictures have been calculated for $N = S = 25$.

$$\frac{(n+1)p_E^e(n+1,p) + (N-n-1)p_F^e(n+1,p)}{N} > \frac{p_F^f(n,p)}{N}, \tag{7}$$

which using Eqs. (3, 5) is equivalent to $R_e(N, n+1, p, S) > R_f(N, n, p, S)$. Then $R_e(N, N_e+1, p, S) > R_f(N, N_e, p, S)$ for all $N_e \in \{0, ..., N-1\}$. Therefore, in any given situation, a fair host always has an incentive to become evil.

5 Simulations for the TCP-FBP interaction

The model we have just present allows understanding some key issues in the interaction between TCP and FBP. Nevertheless, to gain a deeper insight into the TCP/FBP competition, we have carried out a number of simulations using a slightly modified version of the NS2 simulator. For these, we consider that all communication lines have a fixed capacity of $C = 100$ Kbps, with delays of 1 ms and a router buffer of 10 packets. Fair hosts have been modeled using standard one-way TCP agents. FBP hosts have been implemented using modified UDP agents. In both cases, agents are greedy.

We use the goodput (including headers) as the measurement of the information transmitted by each player. The traffic of the TCP hosts has been implemented using the usual FTP application of NS2 (which assumes that the file being transmitted is infinite). Fountain traffic has been implemented with CBR generators with the *random_* bit set (uniform distribution). This randomization is necessary to guarantee that the router does not benefit any of the hosts when dropping packets. (If a pure CBR is used, there may be time patters making some hosts more likely to introduce their packets into the router.) The buffer management policy is drop-tail and the scheduling discipline is FIFO. All the simulation results presented in this paper have been averaged for 50 executions of the simulation scenario. Each execution has been run for a simulated time of $30,000$.

Note that it is possible to establish a direct parallelism between the TCP based hosts of the simulations and the fair hosts of the analytical model because both comply with a set of ordered rules which try of optimize the utilization of the shared resource avoiding congestion. In the same way, the evil hosts of the analytical model can be assimilated as the FBP (CBR-UDP) hosts of the simulations. In this case, the selfishness probability p can be easily calculated as the utilization of the corresponding line (the ratio between the offered load of the evil CBR source, λ_e, and the total capacity of the communication line C). For instance, since $C = 100$ Kbps, an evil host with $p = 0.5$ would correspond to an FBP agent using a CBR source of $\lambda_e = 50$ Kbps.

5.1 Optimal decoding inefficiency

The results of the simulations, as well as the predictions of the simplified mathematical model when considering optimal decoding inefficiency (presented above), have been depicted in Fig. 3.

The first thing we see is that our observation about the analytical model is correct. That is, the theoretical curve for fair hosts is optimistic and it remains always over the real goodput of the TCP hosts, and the one of evil hosts is pessimistic and stays all the time under the real FBP results. This confirms the validity of our arguments, in Observation 1, about the analytical model.

Furthermore, it can be seen that an optimal protocol controlling the congestion is just as good as letting all hosts to send their FBP packets as fast as possible

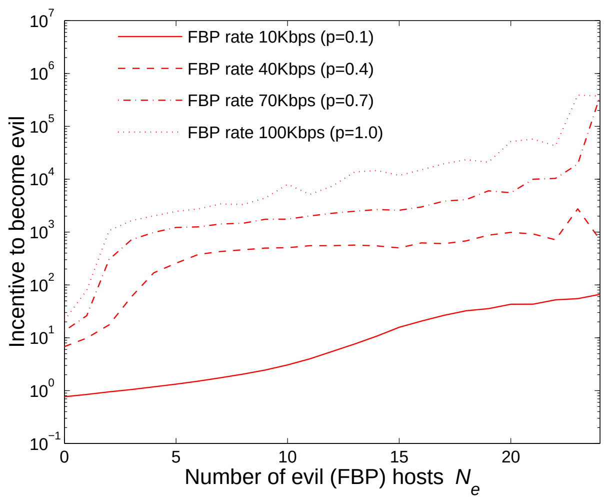

Fig. 4. Incentive to become evil as a function of the number of evil hosts N_e for different values of the load offered by FBP hosts. The simulation conditions are identical to the ones described in Fig. 3.

in a selfish manner and without any kind of control. As we explained previously using the mathematical model (Property 1), this means that fair hosts always have an incentive to become evil, because in any possible situation the most rational strategy is to use FBP. Fig. 4 shows the *incentive* hosts have to become evil for different values of N_e, where incentive is defined as

$$\frac{R_e(N, N_e + 1, p, S) - R_f(N, N_e, p, S)}{R_f(N, N_e, p, S)}.$$

Finally, the TCP (fair) rate when N_e hosts are evil (for any value of N_e) is always under the FBP (evil) rate when one more host becomes evil. This confirms that, as explained using the mathematical model (Property 2), the Nash equilibrium is reached when all hosts are evil ($N_e = N$). This feature can be observed more clearly in Fig. 5, where we have represented the simulated Nash equilibrium goodput and the simulated cooperative goodput for 10 different values of the load injected by the FBP hosts (10 different values of p). This means that, in this particular game, the selfish equilibrium is slightly more efficient than the global cooperation of TCP. Hence, we can claim that the Tragedy of the Commons is not present, at least under the assumptions we have accepted.

5.2 *Suboptimal decoding inefficiency*

For simplicity, in the previous sections it has been assumed that the decoding inefficiency of the used codes is 1. However, in a real situation, $\varepsilon > 0$, with

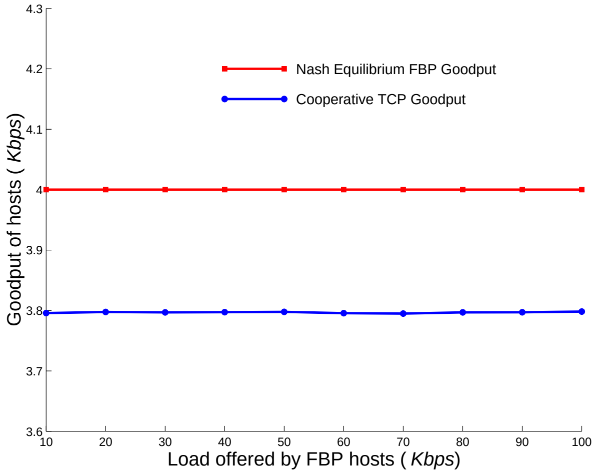

Fig. 5. Simulated FBP Nash equilibrium goodput (solid line with squares) and the simulated TCP cooperative goodput (solid line with circles) for 10 different values of the load offered by FBP hosts. The simulation conditions are identical to the ones described in Fig. 3.

typical values for ε in the range of $[0.02, 0.05]$. In this context, when the value of ε increases, the FBP (evil) goodput decreases in a factor of $1+\varepsilon$ with respect to the best case situation described previously. The question that arises immediately is whether the same conclusions we described previously would be obtained when the FBP hosts do not behave so optimally.

Here, we will study how the situation changes with ε. In Fig. 6 we have represented the goodput as a function of p for 4 different values of ε. The values have been normalized with respect to the TCP cooperative throughput, where there are not evil hosts. We can see that, for reasonable values of ε (smaller than approximately 0.05), the Nash equilibrium is slightly more efficient than the TCP solution, while for higher values of ε, it is slightly under the TCP throughput, and we have a (not very tragic) Tragedy of the Commons.

Hence, we show that it is possible to obtain a Nash equilibrium in the system that is less efficient than the TCP cooperative situation if the value of ε increases. That is, the system may fall in a Tragedy of the Commons. However, in contrast with the results presented in [24,13,16], the tragedy is well bounded and the network would never collapse. The performance of the system is guaranteed to be very close to the value obtained in the TCP cooperative situation, at least for typical values of the parameter ε. This occurs since TCP does not have an efficiency of 100% in the utilization of the line.

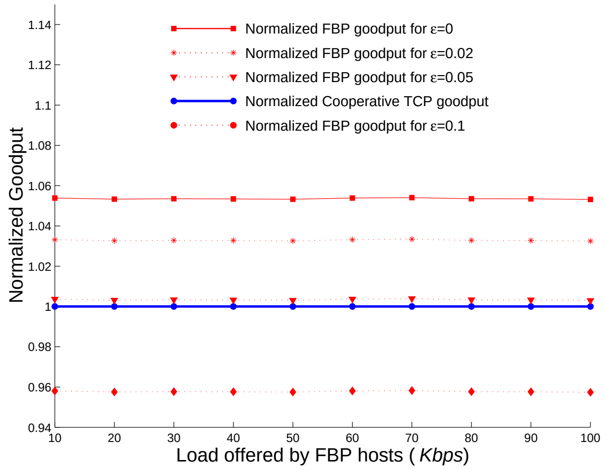

Fig. 6. Normalized goodputs for FBP as a function of the load offered by FBP hosts. A solid thick line with circles plots the normalized only-TCP goodput and a solid thin line with squares plots the normalized Nash equilibrium goodput for $\varepsilon = 0$. The dotted lines represent the normalized Nash equilibrium goodput for $\varepsilon > 0$: the stars indicate $\varepsilon = 0.02$, the triangles $\varepsilon = 0.05$, and the diamonds $\varepsilon = 0.1$. The simulation conditions are identical to the ones described in Fig. 3.

5.3 Fast lines and high delays

In our analysis, we have implicitly accepted that the RTT of packets is low and that we do not have high speed communication lines. Observe that, in the mathematical model, large values of RTT or high speed communication lines decrease the throughput within a real TCP scenario because once the whole sender window has been transmitted, a host must wait either the arrival of an `ack` or a timeout to be able to transmit something again. Therefore, this assumption implies that we have been considering the best TCP (fair) performance that can be found in a real scenario.

If we increase the RTT or the speed of the lines, the TCP goodput will fall. Hence, in all these cases, the Nash equilibrium will represent an even more efficient option than the all-TCP case. This can be easily observed in Fig. 7, where we show a situation similar to the one of Fig. 3, but where the speed of the communication lines has been increased to $C = 100$ Mbps. As it can be noticed, when the delay of the lines increases, the initial TCP goodput decreases, while the FBP Nash equilibrium remains constant. Note that for the same delay of 1 ms used in Fig. 3 the TCP goodput quickly degrades with the increase in the number of evil hosts.

Taking into account the results obtained in this section, the essential aspect that must be remarked is that the introduction of FBP drives the system to a Nash equilibrium where TCP disappears. Furthermore, such an equilibrium has an efficiency that can be slightly over or slightly under the one obtained

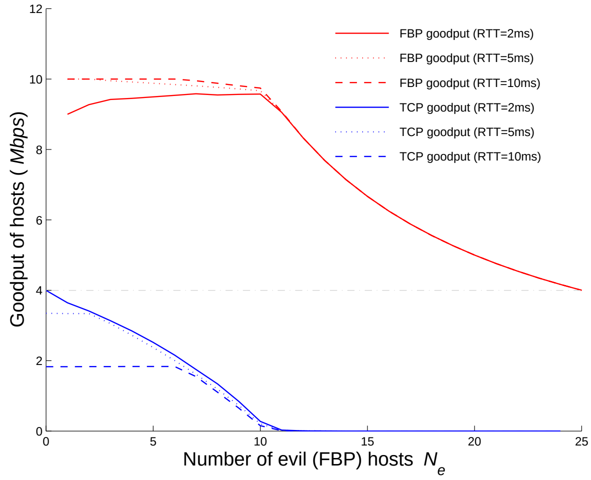

Fig. 7. Goodput for the FBP evil and TCP fair hosts as a function of the number of evil hosts N_e in the single-bottle-neck scenario of Fig. 2. The capacity of the lines has been fixed to $C = 100$ Mbps. The offered load of FBP hosts is 10 Mbps.

by using only TCP in most real situations, but never drives the system into a collapse.

5.4 Effects of finite data files

Since the relationship between bandwidth (BW), latency (measured as RTT) and size of target data (D) is essential to evaluate the real goodput of the FBP protocol, it is clear that, given the current definition of the protocol, all packets arriving after the last stop (ACK) signal is emitted by the receiver are useless (basically because the data has fully been decoded by that time). Given that, at Nash equilibrium, the sender emits packets at maximum rate, it can be easily demonstrated that the product $BW \cdot RTT$ corresponds to useless data. Hence, the utility of the link can be evaluated as $U = D/(D + BW \cdot RTT)$ being D the size of the original data we want to transmit (we do not consider the decoding inefficiency ϵ, which is discussed in another section of the paper). With this equation in mind, it is clear that the results provided in the paper are only applicable when D is much larger than $BW \cdot RTT$. In some cases, this could restrict the number of applications for which FBP can be of real use, but it is undoubtedly that there are many scenarios where that condition is fulfilled; for example, in the transmission of large video files (p2p applications, video on demand, etc), D is usually in the range of some hundreds of megabytes, while the $BW \cdot RTT$ product is rarely over one megabyte with current Internet access capabilities (ADSL or similar).

6 Congestion and fairness in FBPs

In the previous sections we have evaluated systems in which TCP and FBP hosts coexist. In this section we analyze systems with only FBP hosts. Our objective is to explore the situation when an FBP host has a choice between sending packets at a low rate and sending packets at a faster rate. Hence, in our system we are going to have two classes of FBP hosts. *Slow* hosts will send packets at a rate λ_{slow} (bits/second), while *fast* hosts will send packets at a rate $\lambda_{fast} > \lambda_{slow}$. For simplicity we will assume that $N\lambda_{fast} \geq C$, which implies that when all hosts are fast the bottleneck link is fully used.

In order to analyze this system, we observe that the behavior of the router can be approximated by that of a queueing system M/M/1/K, where the buffer of the router can hold $K-1$ packets. The arrival rate at this queue is $\lambda = N_e\lambda_{fast} + (N-N_e)\lambda_{slow}$ and the service rate is $\mu = C$. Then, if we define $\rho = \frac{\lambda}{\mu}$, using traditional queuing theory, we obtain that the transmission rate of the bottleneck link is

$$\lambda' = \lambda(1-p_K),$$

where

$$p_K = \begin{cases} \frac{1-\rho}{1-\rho^{K+1}}\,\rho^K \text{ when } \rho \neq 1 \\ 1/(K+1) \text{ when } \rho = 1 \end{cases} \tag{8}$$

Hence, the goodput for a slow host is

$$T_{slow} = \lambda_{slow}(1-p_K), \tag{9}$$

while the goodput for a fast host is

$$T_{fast} = \lambda_{fast}(1-p_K). \tag{10}$$

As we did in the previous section, we have used NS2 to simulate a system with $N = 25$ hosts with link capacities of $C = 100$ Kbps. In all the experiments, hosts use, for different values of λ_{fast}, UDP-CBR packet generators with randomization. Figures 8 and 9 present the results of the simulations compared with the queueing theory approach for four different values of λ_{fast}, both when $\lambda_{slow} = C/N$ (Figure 8) and when $\lambda_{slow} < C/N$ (Figure 9). As it can be readily seen, the theoretical models fit very nicely the results obtained by simulation. The small differences have to do with the assumption that packets have exponentially distributed lengths.

In these figures, it can be observed that, as in the previous sections, the goodput of slow hosts when there are N_e fast hosts is always smaller than the goodput of fast hosts when there are $N_e + 1$ fast hosts (for any $N_e < N$).

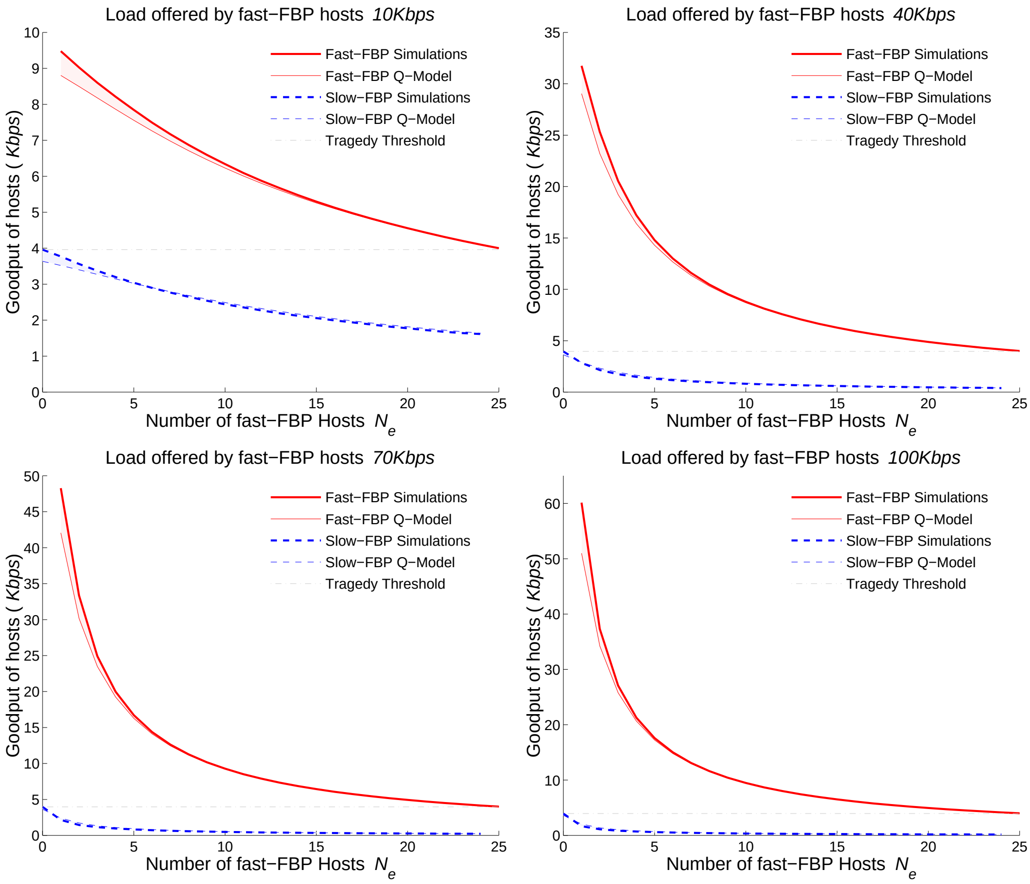

Fig. 8. Goodput of fast and slow hosts as a function of the number of fast hosts N_e for four different values of evil selfishness when $\lambda_{slow} = C/N = 4$ Kbps. The thick lines represent results from the simulations, the thin lines are predictions from the theoretical model based on queuing theory. The shaded region represents the error between the theoretical model and the simulations. The horizontal dashed line indicates the simulated throughput of the slow hosts when no fast hosts are present. All pictures have been calculated for $N = 25$ hosts.

Hence, slow hosts always have an incentive to increase their sending rate. Furthermore, this effect is more remarkable when $\lambda_{slow} < C/N$.

Another observation is that there is never a Tragedy of the Commons. In the Nash equilibrium (which is reached when $N_e = N$) all hosts evenly share the resources like in the all-slow case, all obtaining a goodput of C/N. This implies that if the aggregation of slow rates fills the bottleneck link (i.e., $\lambda_{slow} \geq C/N$), the Nash equilibrium yields the same goodput as the all-slow case. However, if the slow rate is below C/N, the Nash equilibrium presents a goodput larger than the all-slow case.

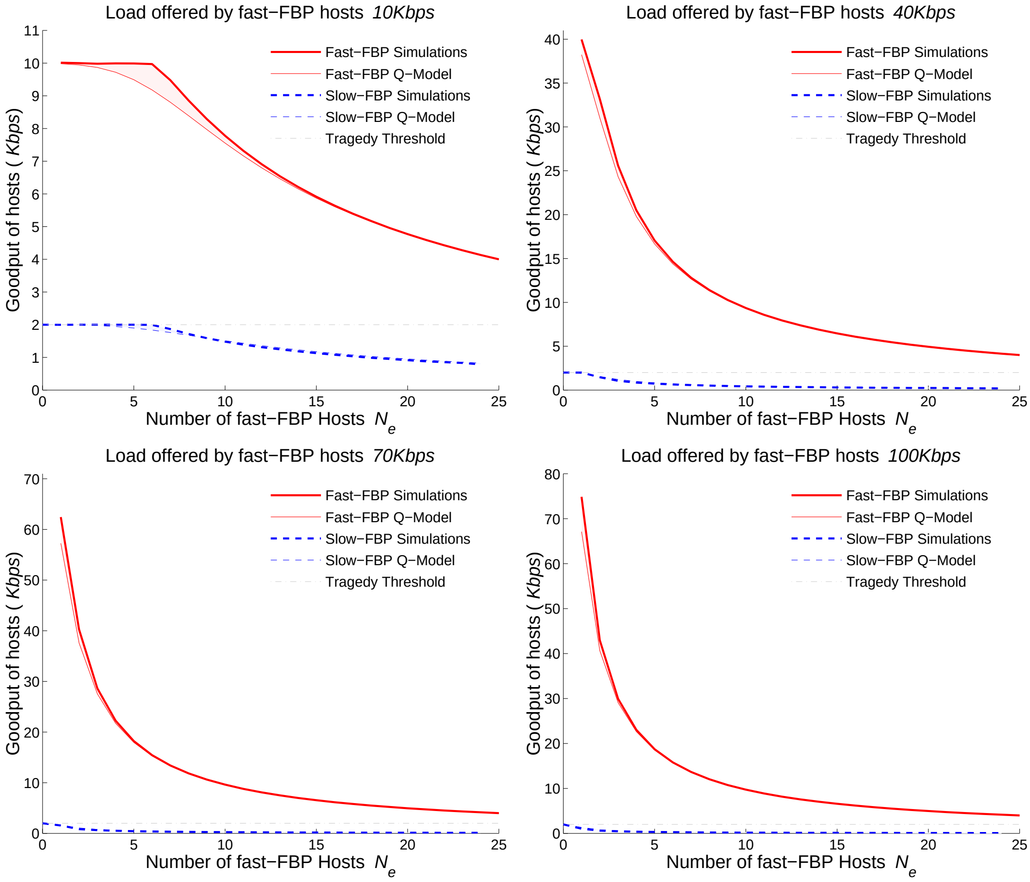

Fig. 9. Goodput of fast and slow hosts as a function of the number of fast hosts N_e for four different values of evil selfishness when $\lambda_{slow} = 2$ Kbps $< C/N$. The thick lines represent results from the simulations, the thin lines are predictions from the theoretical model based on queuing theory. The horizontal dashed line indicates the simulated goodput of the slow hosts when no fast hosts are present. All pictures have been calculated for $N = 25$ hosts.

7 Concluding Remarks

In this paper we have analyzed a novel paradigm of reliable communication which is not based on the traditional timeout-and-retransmit mechanism of TCP. Our approach, which we call FBP (Fountain Based Protocol), consists of using a digital fountain encoding which guarantees that all received packets are useful. Using Game Theory, we analyzed the behavior of TCP and FBP in the presence of congestion and show that two main characteristics arise. First, in this scenario, any given host using TCP has an incentive to switch to an FBP approach obtaining a higher throughput. This guarantees the Nash equilibrium to be reached when all hosts use FBP. Second, we showed that, at this equilibrium, the performance of the network is similar (may be slightly over or slightly under) the performance obtained when all hosts comply with TCP. This latter claim holds even when FBP hosts act in an absolutely selfish manner injecting packets into the network as fast as they can and without any kind of congestion control mechanism.

The two above mentioned observations have direct implications in the context of the Internet. The first means that if FBP protocols are widely available for users, they will tend to employ them because they will obtain improved performance. Moreover, when more and more FBP hosts exist, the performance of the TCP players will decrease and the incentive to become evil will increase, possibly making that after some period of time all hosts become FBP. In this case, our second observation guarantees that the global performance of the network will not be under the original one which was obtained when only TCP hosts existed.

An aspect which merits some comments is the one relative to the architecture of current networks, which are designed to avoid congestion and to try to drop as few packets as possible. This fact could make the current Internet infrastructure to be seriously impaired by congestion if a large portion of users decides to switch to FBP. In this case, a new kind of routers would be necessary. This novel technology should be designed to work under extremely congested scenarios with communication lines being saturated to nearly 100% of their capacities most of the time. In this new situation buffering could have a limited utility, mainly contributing to increase network latencies.

Although these results seem promising, we wish to note that the FBP approach presents several aspects that should be taken into consideration. First, the analysis we have carried out has been done on the basis of large file transfers. However, this scenario can substantially change when considering other kind of communications requiring more interaction between the sender and the receiver. For example, several real time or multimedia applications (like Telnet) require small units to be continuously transferred. This scenario makes the FBP approach less practical, because, although duplicate packets cannot exist, it is possible that useless packets not containing additional information could flood the network before the appropriate stop message issued by the receiver arrives to the sender. Furthermore, in our analysis we have assumed that all packets arriving to the destination are useful. In reality, it could happen that a fast sender floods a slow receiver, which must drop packets. However, there are currently techniques that can be used to guarantee that receivers will not be saturated because of the fast sender rate (see for instance the mechanism used in [34]). Finally, another issue that deserves further attention is to analyze what happens if we consider energy consumption issues in battery powered devices (which would waste a lot of energy). In those scenarios, the energy consumption is important and the use of FBP could be a problem. In these cases, it would be necessary to control the sending rate to avoid wasting a lot of energy due to the loss of many packets.

References

[1] S. Floyd, The NewReno modification to TCP's fast recovery algorithm, IETF Request for Comments 2582.

[2] L. S. Brakmo, S. W. O'Malley, L. L. Peterson, TCP Vegas: New techniques for congestion detection and avoidance, in: Proceedings of ACM SIGCOMM, London, UK, 1994, pp. 24–35.

[3] M. Mathis, J. Mahdavi, Forward acknowledgement: refining TCP congestion control, in: Proceedings of ACM SIGCOMM, ACM Press, Palo Alto, California, United States, 1996, pp. 281–291.

[4] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, TCP selective acknowledgment options, IETF Request for Comments 2018.

[5] J. Mo, R. J. La, V. Anantharam, J. C. Walrand, Analysis and comparison of TCP Reno and Vegas, in: Proceedings of the IEEE INFOCOM, 1999, pp. 1556–1563.

[6] J. Postel, Transmission control protocol, IETF Request for Comments 793.

[7] K. Ramakrishnan, S. Floyd, A proposal to add Explicit Congestion Notification (ECN) to IP, IETF Request for Comments 2481.

[8] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (4) (1993) 397–413.

[9] V. Jacobson, Congestion avoidance and control, in: Proceedings of ACM SIGCOMM, ACM Press, 1988, pp. 314–329.

[10] D. Lin, R. Morris, Dynamics of random early detection, in: Proceedings of ACM SIGCOMM, ACM Press, Cannes, France, 1997, pp. 127–137.

[11] W. Prue, J. Postel, Something a host could do with source quench: The source quench introduced delay (SQuID)., IETF Request for Comments 1016.

[12] J. Nagle, On packet switches with infinite storage, IEEE Transactions on Communications 35 (4) (1987) 435–438.

[13] A. Akella, S. Seshan, R. Karp, S. S. , C. Papadimitriou, Selfish behavior and stability of the internet: A game-theoretic analysis of TCP, in: Proceedings of ACM SIGCOMM, ACM Press New York, 2002, pp. 117–132.

[14] D. Fudenberg, J.Tirol, Game Theory, MIT Press, 1991.

[15] G. Owen, Game Theory, Academic Press, 1995.

[16] J. Nagle, Congestion control in IP/TCP internetworks, IETF Request for Comments 896.

[17] R. Garg, A. Kamra, V. Khurana, A game-theoretic approach towards congestion control in communication networks, Computer Communication Review 32 (3) (2002) 47–61.

[18] R. Pan, B. Prabhakar, K. Psounis, CHOKE, A stateless active queue management scheme for approximating fair bandwidth allocation, in: Proceedings of the IEEE INFOCOM, IEEE Los Alamitos, 2000, pp. 942–951.

[19] H. Yaïche, R. R. M. , C. Rosenberg, A game theoretic framework for bandwidth allocation and pricing in broadband networks, IEEE/ACM Transactions on Networking 8 (5) (2000) 667–678.

[20] S. J. Shenker, Making greed work in networks: a game-theoretic analysis of switch service disciplines, IEEE/ACM Transactions on Networking 3 (6) (1995) 819–831.

[21] T. Roughgarden, T. Éva, How bad is selfish routing?, Journal of the ACM 49 (2) (2002) 236–259.

[22] C. Papadimitriou, Algorithms, games, and the internet, in: Proceedings of the thirty-third annual ACM symposium on Theory of computing, ACM Press, Hersonissos, Greece, 2001, pp. 749–753.

[23] G. Hardin, The tragedy of the commons, Science 162 (1968) 1243–1248.

[24] L. López, G. Rey, A. Fernández, S. Paquelet, A mathematical model for the TCP tragedy of the commons, Theoretical Computer Science 343 (1) (2005) 4–26.

[25] M. Luby, LT codes, in: 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.

[26] A. Shokrollahi, Raptor codes, To appear in IEEE Transactions on Information Theory.

[27] J. Byers, M. Luby, M. Mitzenmacher, A digital fountain approach to asynchronous reliable multicast, IEEE Journal on Selected Areas in Communications 20:8 (2002) 1528–1540.

[28] J. Byers, G. Horn, M. Luby, M. Mitzenmacher, W. Shaver, Flid-dl: Congestion control for layered multicast, IEEE Journal on Selected Areas in Communications 20:8 (2002) 1558–1570.

[29] J. Byers, M. Luby, M. Mitzenmacher, A digital fountain approach to the reliable distribution of bulk data, IEEE Journal on Selected Areas in Communications 20:8 (2002) 1528–1540.

[30] N. F. Maxemchuk, Dispersity routing, in: Proceedings of ICC, San Francisco CA, 1975, pp. 41–10, 41–13.

[31] M. O. Rabin., Efficient dispersal of information for security, load balancing, and fault tolerance, Journal of the Association for Computing Machinery 36 (2) (1989) 335–348.

[32] I. S. Reed, G. Solomon, Polynomial codes over certain finite fields, Journal of the Society for Industrial and Applied Mathematics 8 (2) (1960) 300–304.

[33] B. Raghavan, A. C. Snoeren, Decongestion control, in: Proceedings of the Fifth Workshop on Hot Topics in Networks (HotNets-V), ACM Press, 2006, pp. 61–66.

[34] H.Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: A transport protocol for real-time applications, IETF Request for Comments 3550.