



**HAL**  
open science

## Extending the ad hoc horizon in dense 802.11 networks using fountain codes

Amel Ksentini, Tijani Chahed

► **To cite this version:**

Amel Ksentini, Tijani Chahed. Extending the ad hoc horizon in dense 802.11 networks using fountain codes. ICSNC 2009: 4th International Conference on Systems and Networks Communications, Sep 2009, Porto, Portugal. pp.63 - 67, 10.1109/ICSNC.2009.88 . hal-01355072

**HAL Id: hal-01355072**

**<https://hal.science/hal-01355072v1>**

Submitted on 22 Aug 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.

# Extending the Ad Hoc Horizon in Dense 802.11 Networks Using Fountain Codes

Amel Ksentini

*Ecole Supérieure de Communications de Tunis (Sup'com)*

*Ariana, Tunisia*

*Email: ksentini.amel@gmail.com*

Tijani Chahed

*Institut TELECOM; TELECOM et Management SudParis*

*Evry, France*

*Email: tijani.chahed@it-sudparis.eu*

**Abstract**—Ad hoc networks are one operational mode of IEEE802.11 Wireless Local Area Networks (WLANs) in which users form themselves the network and act as routers to one another. When the density of such a network is high, capacity becomes limited by the interference resulting from the collisions between simultaneously transmitted packets. Interference gets even worse in the case of reliable transport based on TCP as data packets interfere with the acknowledgements sent back in the reverse direction. In this context, the term Ad Hoc Horizon refers to the number of nodes and hops beyond which TCP performance is no longer satisfactory in terms of throughput. We investigate in this work this Horizon in the presence of a transport scheme based on Fountain Codes instead of TCP. Fountain codes are rateless, low complexity codes, which allow for open-loop reliable transport, without acknowledgements in the reverse direction; this would in turn decrease interference and increase the Horizon. Indeed, our results show a Horizon almost twice as large as that obtained with TCP.

**Keywords**—Ad hoc networks; Ad Hoc Horizon; TCP; Fountain Codes; interference

## I. INTRODUCTION

The IEEE802.11 standard defines two modes of communication: infrastructure and ad hoc. The former is composed of one Access Point (AP) and a set of (mobile) users that access the network through the AP. The ad hoc mode refers to a network that is formed by the users themselves, i.e., users act as routers to one another.

Ad hoc networks may be deployed following two scenarios: a low density one, termed Delay Tolerant Network (DTN) [1], with a low number of users per unit area which makes connectivity intermittent. Capacity, in this case, depends mainly on the users mobility pattern, the content replication method as well as the routing protocol. The dense scenario refers to a high number of nodes per unit area with transmission ranges typically larger than the distances between the nodes. In this case, capacity is generally limited by interference which results from collisions that may take place between simultaneously transmitted packets on the shared medium, managed, at the MAC layer, by a so-called Distributed Coordination Function (DCF), based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The denser the network, the higher the interference.

Interference is further exacerbated in the case of TCP. The latter relies on a closed-loop control mechanism wherein the

good receipt of data packets at the receiver is acknowledged through the transmission of ACKnowledgements (ACKs) in the reverse direction, back to the sender. TCP data packets, ACKs as well as possibly some control packets transmitted simultaneously collide, get lost and need thus to be retransmitted.

One way to reduce collisions, and hence interference, would be to reduce the number of packets that the system transports. If we want to operate at the same level of useful load, i.e., not to reduce the number of data packets, we would need to suppress ACKs and thus TCP altogether. In this case, reliability can be assured by using an open-loop mechanism, based on coding, at the expense, among others, of sending an extra amount of packets. This is the rationale behind erasure codes such as Reed-Salomon [3] or Tornado codes [4]. The main shortcoming in this case is complexity which renders these coding schemes impractical when the number of packets to be sent is large.

Fountain Codes (FCs) [5] are, on the contrary, low complexity, rateless codes, which operate as follows: the file to be sent is first decomposed into input symbols, which may be of any length. An integer is then retrieved at random from a distribution, called the degree distribution; a typical one being the Solitron [6]. This integer determines the number of symbols that would be summed up together, using XOR operations on the bit level, into one output symbol. The receiver then receives these output symbols, reads in the header their degrees and proceeds to decoding the input symbols using reverse XOR operations. The term *fountain* refers to the fact that the only thing that the receiver needs to be able to reconstruct the original input symbols is to receive a minimum number of *any* output symbols. The cost of using FCs in terms of load is mainly due to the number of encoded packets which needs to be larger than the initial number of packets that constitute the original file.

Examples of FCs are Luby Transform Codes (LTCs) [8], Raptor Codes (RCs) [9], which add one pre-coding step to LTC and use a mean degree equal to 3, and Online Codes [10] which are a special, asymmetric case of RCs. FCs have been standardized as application layer Forward Error Correction (FEC) codes for Multimedia Broadcast and Multicast Services (MBMS) by 3GPP [11] and Digital Video Broadcast (DVB) by ETSI [12].

Our aim in this work is to investigate the comparative performance of dense ad hoc networks using FC-based, open-loop, reliable transport versus TCP. We will specifically determine the extent of gain, in terms of the extra number of hops and nodes that can be simultaneously active and still yield a satisfactory performance, in the case of Fountain code-Based Transport (FBT).

The rest of this paper is organized as follows. In Section II, we describe the system of our focus, highlight its main components as described in [2] and introduce FBT operation and cost therein. In Section III, we detail our simulation setting and show our results for the comparative performance of FC-based transport versus TCP. Eventually, Section IV concludes the paper.

## II. SYSTEM

In order to study the system performance as a function of density *in a controlled manner*, we make use of the beam-star topology suggested in [2] and shown in Figure 1. As can be seen from the figure, this topology is composed of one central node acting as a source or destination for all sessions in the network and a (variable) number of beams originating from this central node. The end node of each beam will act as a source or destination in relationship with the central node. The overall communication pattern consists thus of parallel sessions, one on each beam.

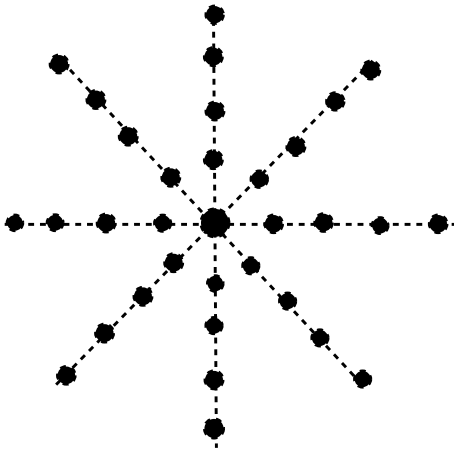


Figure 1. Beam-star topology

We consider, in this work, the case where file sizes are *finite*. This is different from [2] where flows were assumed of infinite length. The reason behind this choice is that, as we compare the performance between FC-based transport and TCP, we need to explicitly quantify the impact of the overhead of FCs in terms of the extra packets that need to be sent from source to destination, and which can be calculated as follows.

Let  $k$  be the original number of packets that constitute the file to be transmitted and let  $n$  be the total number of packets

that need to be received at the receiver so that it can decode the original content. We have  $n = k(1 + \epsilon)$ , where  $\epsilon$ , termed the decoding efficiency, ranges typically between 10 and 100%, depending on the specific implementation [7]. Please note that when the receiver receives the  $n$  symbols, decoding is successful with a probability equal to  $(1 - \delta)$  where  $\delta$  is upper-bounded by  $2^{-k\epsilon}$ . This means that larger file sizes and/or higher values of  $\epsilon$  make the decoding probability even larger. As of complexity, it may be as low as linear in the number of coded symbols.

Eventually, the performance metric in [2] was defined at the packet level in relationship to the number of no-progress intervals where TCP transmission was frozen, because of interference, for a duration larger than 3 seconds. This metric was next used to define the concept of Ad Hoc Horizon as the maximum number of nodes and hops that a dense ad hoc network can handle and still obtain a satisfactory TCP performance, in terms of the ration of no-progress intervals. Since we monitor performance at the flow level, our performance metric will be the (mean) transfer time of the entire file. The notion of Ad Hoc Horizon, quantified next for the cases of FCs versus TCP, refers now to the trend of this mean file transfer time and to the *jumps* that are observed when the number of nodes and hops exceeds a certain threshold or Horizon.

## III. SIMULATIONS

We run our simulations on NS2. Unless otherwise indicated later in the text, Table I summarizes the parameters used in the simulations.

IEEE 802.11 bandwidth	2Mbps
Transmission range	250m
Interference range	550m
Routing protocol	AODV-UU 0.8, local repair enabled
Simulation run	300sec
Number of repetitions	6
Inter-node distance	100m

Table I  
SIMULATION PARAMETERS

### A. TCP versus FBT

We first compare, respectively, in Figures 2 and 3, the performance, in terms of mean file transfer time, between TCP and FBT (with  $\epsilon = 0.1$ ), as a function of an increasing number of hops and beams, for a file size equal to 50 packets.

We observe that the Ad Hoc Horizon is only 2 hops and 12 nodes with TCP whereas it extends to 5 hops and 24 nodes with FBT. This is mainly due to the fact that with FBT, no ACKs need to be sent in the opposite direction and so the load in the network, in terms of units of data, and hence interference, decreases by almost half (assuming one ACK for every TCP packet).

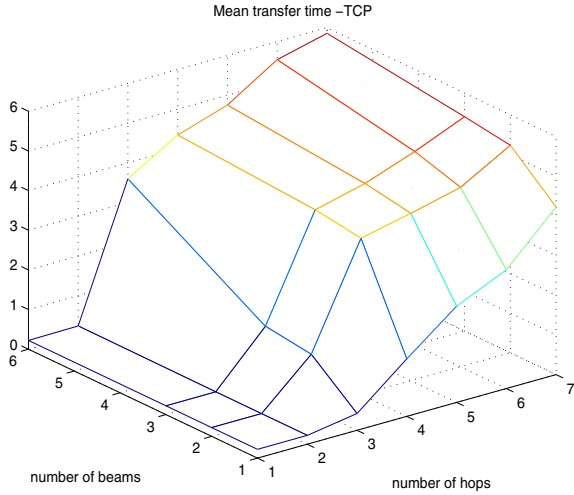


Figure 2. TCP mean transfer time - 50 packets

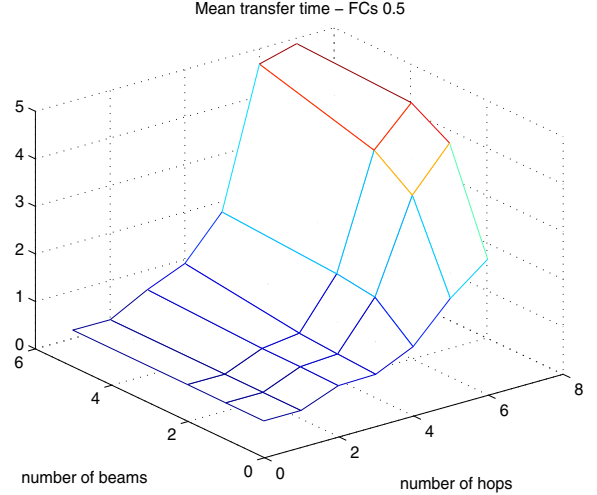


Figure 4. FBT mean transfer time -  $\epsilon = 0.5$

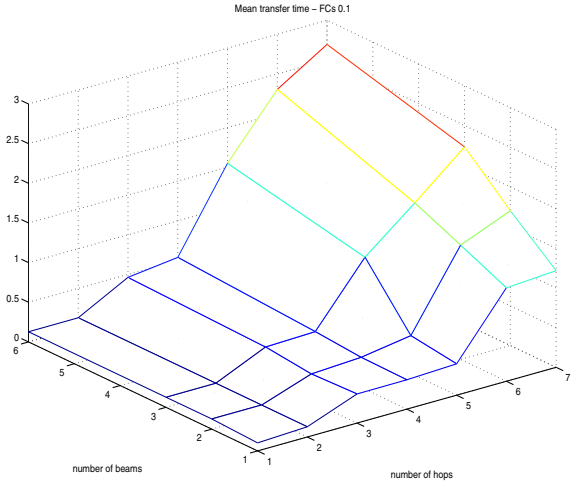


Figure 3. FBT mean transfer time - 50 packets

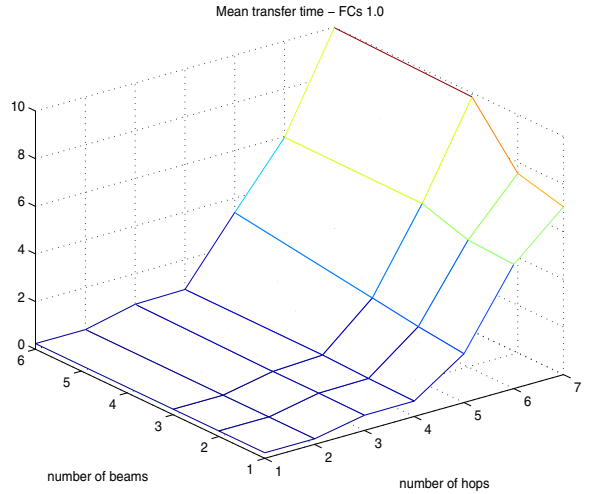


Figure 5. FBT mean transfer time -  $\epsilon = 0.5$

### B. Effect of coding efficiency $\epsilon$

We now take larger values for  $\epsilon$  and plot, in Figures 4 and 5, the mean transfer times corresponding to values of  $\epsilon$  equal to 0.5 and 1.0, respectively, again, as a function of the number of hops and beams.

We observe that, in both cases, the Ad Hoc Horizon has almost the same values as previously. For  $\epsilon = 1$  and a large number of hops ( $> 5$ ), the performance of TCP may nevertheless turn out to be better. This large value of  $\epsilon$  is however not typical.

### C. Impact of file size

We now consider a small file size, equal to 5 packets, and show, in Figures 6 and 7, the mean download time for TCP and FBT with  $\epsilon = 30\%$ , respectively.

We observe that the Ad Hoc Horizon with TCP has the same shape as with larger file size but the values for the mean transfer time are so small in this case that even FBT's Horizon has almost the same shape as well, albeit a little bit more flat when the number of beams is small.

### D. Impact of distance

We now turn to the performance of TCP versus FBT as a function of the distance between the nodes. Indeed, the overall performance may degrade as the distance between nodes increases, as it adds more path loss. But, on the other hand, an increase in distance may enhance the overall system performance as it lowers the interference.

We show, in Figures 8 and 9, the obtained performance for TCP and FBT with  $\epsilon = 30\%$ , respectively, for several

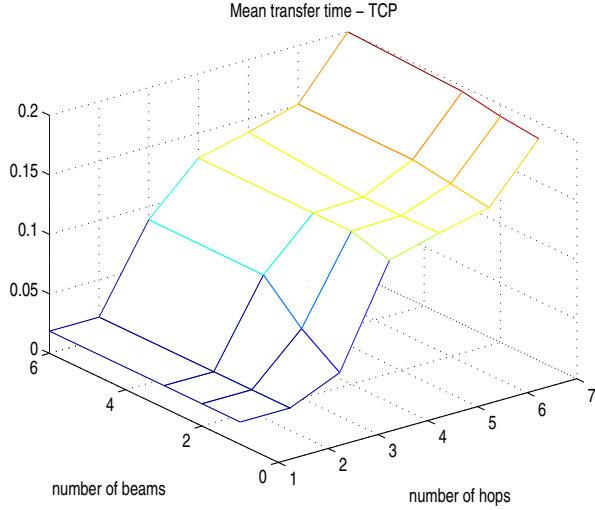


Figure 6. TCP mean transfer time - 5 packets

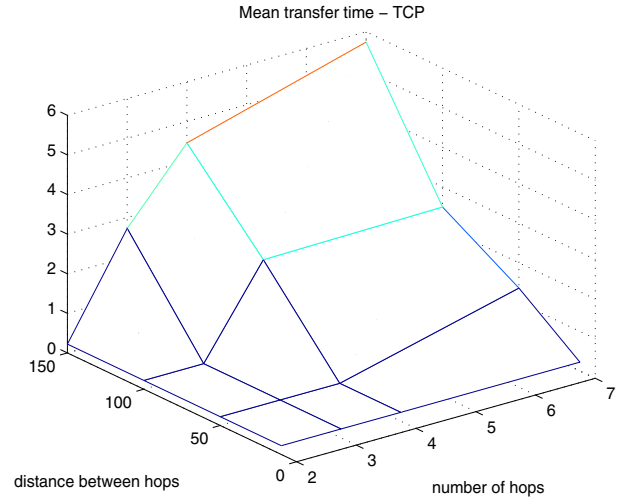


Figure 8. TCP mean transfer time

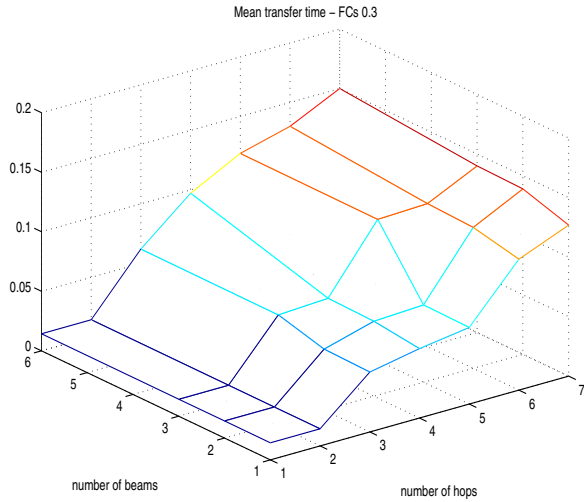


Figure 7. FBT mean transfer time - 5 packets

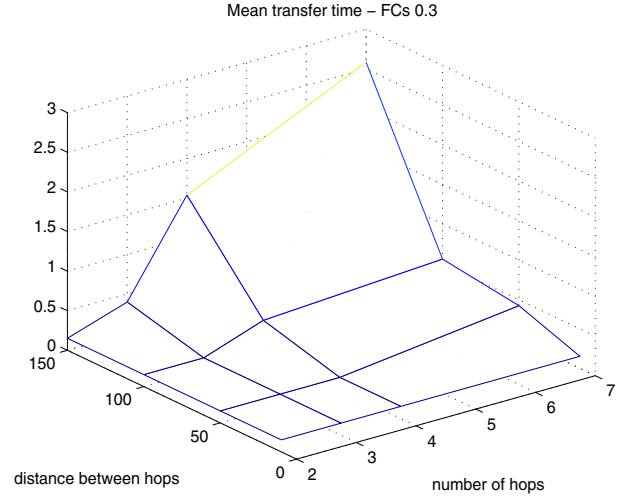


Figure 9. FBT mean transfer time

values of distances, ranging between 10 and 150 meters.

We observe that the Horizon varies a lot with distance, in favor of FBT, and that in both cases, the overall performance decreases as distance increases which means that the effect of path loss is larger than that of interference.

### E. Unfairness

Unfairness refers to a worst-case scenario in which one or a subset of sessions/beams can monopolize the resources at the expense of others which would experience long starvation duration [2]. It can be assessed using the classical maxmin fairness index  $f$  given by [13]:

$$f = \frac{(\sum_{i=1}^B x_i)^2}{B \sum_{i=1}^B x_i^2} \quad (1)$$

where  $B$  is the number of beams and  $x_i$  is the ratio of the file size divided by the corresponding transfer time on beam  $i$ ,  $i = 1, \dots, B$ . The index varies between 0 and 1; a value of zero means that all sessions are able to progress in a fair manner, a value of 1 means that one session monopolizes all the bandwidth.

We plot, in Figures 10 and 11, the values of  $(1 - f)$  that we obtain for TCP and FBT ( $\epsilon = 0.3$ ), respectively.

We observe that with TCP, fairness is within 3 hops and 9 nodes Horizon whereas, with FBT, it extends to 6 hops and 18 nodes.

## IV. CONCLUDING REMARKS

We investigated, in this work, the performance of FC-based open-loop reliable transport in the context of dense

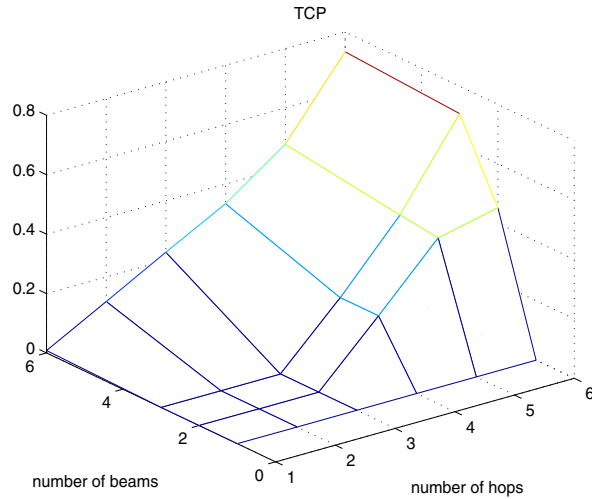


Figure 10. Unfairness - TCP

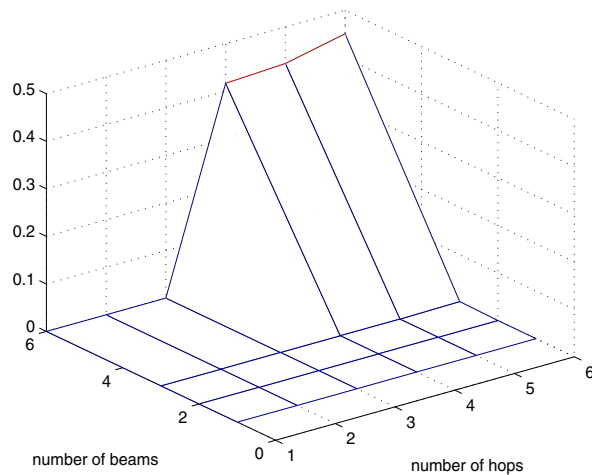


Figure 11. Unfairness - FBT

ad hoc networks where capacity is limited by interference resulting from collisions between simultaneously transmitted packets.

We have shown that FBT generally outperforms TCP, and this in all typical cases of operation. The gain comes from reduced interference as TCP transmits packets in one direction and ACKs in the reverse direction which collide with each other as well as with packets and ACKs from other users. FBT sends only packets, without ACKs, albeit in a larger number than the original one.

Last, it should be also noted that FCs have a cost, notably in terms of complexity, implementation as well as CPU utilization. This is beyond the scope of the present work.

## ACKNOWLEDGMENT

This work was supported by Institut Telecom project FORWIN.

## REFERENCES

- [1] E. Altman and F. De Pellegrini, *Forward Correction and Fountain Codes in Delay Tolerant Networks*, INFOCOM'2009, Rio de Janeiro, April 2009.
- [2] C. Tschudin, E. Osipov, *Estimating the Ad Hoc Horizon for TCP over IEEE 802.11 Networks*, Med-Hoc-Net'2004, Bodrum, June 2004.
- [3] I. S. Reed, G. Salomon, *Polynomial Codes over Certain Finite Fields*, Journal Soc. Indust. Applied Math., vol. 8, pp. 300-304, 1960.
- [4] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, *Efficient Erasure Correction Codes*, IEEE Transactions on Information Theory, Special Issue on Codes and Graphs and Interactive Algorithms, February 2001.
- [5] J. Byers, M. Luby, M. Mitzenmacher, *A digital fountain approach to reliable distribution of bulk data*, ACM Sigcomm'98, 1998.
- [6] E. Hytti, T. Tirronen and J. Virtamo, *Optimal Degree Distribution for LT Codes with Small Message Length*, in INFOCOM'2007, 26th IEEE International Conference on Computer Communications, pp. 2576-2580, Anchorage, April 2007.
- [7] D. J. C. MacKay, *Fountain Codes*, Cavendish laboratory, University of Cambridge.
- [8] M. Luby, *LT Codes*, IEEE-FOCS, pp. 271-282, 2002.
- [9] M. A. Shokrollahi, *Raptor Codes*, IEEE International Symposium on Information Theory, July 2003.
- [10] P. Maymounkov, *Online Codes*, Technical report TR2002-833, New York University, October 2002.
- [11] 3GPP Technical Specification Group Services and System Aspects, *Multimedia Broadcast/Multicast Services, Architecture and Functional Description*, 3GPP TS 23.246 V6.9.0, December 2005.
- [12] ETSI DVB TM-CBMS1167, *IP datacast over DVB-H: Content Delivery Protocols Draft Technical Specification*, September 2005.
- [13] R. Jain, D. Chiu, W. Hawe, *A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems*, DEC Technical Report 301, 1984.