

Equilibrium Selection in Multistage Congestion Games for Real-Time Streaming

Giovanni Rossi
Dept. of Computer Science,
University of Bologna
Mura Anteo Zamboni 7, 40127
Bologna, Italy
giorossi@cs.unibo.it

Stefano Ferretti
Dept. of Computer Science,
University of Bologna
Mura Anteo Zamboni 7, 40127
Bologna, Italy
sferrett@cs.unibo.it

Gabriele D'Angelo
Dept. of Computer Science,
University of Bologna
Mura Anteo Zamboni 7, 40127
Bologna, Italy
gdangelo@cs.unibo.it

ABSTRACT

We model peer-to-peer real-time streaming as a multistage congestion game with strategies consisting of time-sequences of network nodes (or peers), and identify a large set of strategy profiles through which the whole content reaches all peers in a minimum number of time-periods. In order to make these profiles sustainable as equilibria, we provide strategy restriction mechanisms implementing those equilibria where both streaming duration and congestion are minimized. Their functioning is exemplified in a simple simulation environment. The potential and social cost of equilibrium without strategy restrictions are also investigated.

Keywords

Equilibrium analysis, Selective and timely dissemination of information, Coordination in network communications, Distributed artificial intelligence.

1. INTRODUCTION

Real-time streaming is concerned with those many situations where some media content is produced and distributed in time. Specifically, one *new* content unit is periodically generated by a source (or broadcaster) and made available to a set of interested users. Given the real-time generation of contents, a main issue is their timely dissemination. It is widely accepted that this can be addressed with the creation of specifically designed overlay networks composed of the nodes interested in the contents. In these networks, nodes (or peers) collaborate toward data dissemination by forwarding the received data. This Peer-to-Peer (P2P) based overlay network can be designed in different ways. In any case, three main concerns are: *i*) the overall load due to forwarding activity should be evenly shared among the participants; *ii*) the time needed for full dissemination should be minimized; and *iii*) the protocol should be fair in that the expected time needed to receive the whole content should be the same across all peers.

In recent times, game theory is proving very useful for modeling communication systems and dynamic distributed environments. In this paper we model real-time streaming scenarios in terms of multistage congestion games. Roughly speaking, in congestion games there is a set of facilities and players have to choose what facilities to use. Each facility provides some common utility level to all its users, which depends on the number of users, and finally each player gets the sum over used facilities of the utility levels that these latter provide. In particular, the larger the number of players who use a facility, the lower the common utility level that they attain from that facility (monotonicity). Congestion games are potential games, thereby possessing pure-strategy (possibly strong) equilibrium [16]. When using congestion games for modeling real-time streaming, facilities are players themselves: at any stage of the game, some content distribution over peers prevails, and therefore each peer faces a facility set containing all other participants with some additional content units (i.e. possible data forwarders).

Our modeling of real-time streaming in terms of congestion games allows the analysis to distinguish between the number of stages needed to fully disseminate the whole content (or streaming length) on the one side, and stage-wise congestion on the other. In addition, and most importantly, it enables to identify a strategy restriction mechanism which at each stage prevents peers from asking certain content units, given the prevailing content distribution over the population. With such restrictions, at equilibrium both streaming length and stage-wise congestion are minimized. This is detailed by means of a distributed algorithm that we develop for implementing the proposed equilibrium selection method.

The paper is organized as follows: in Section 2 the background is presented. In Section 3 we introduce the needed basic concepts and notation. In Section 4, the problem of multimedia live streaming is presented as a multistage congestion game. In Section 5 we focus on how to distribute the whole content over the whole population in a minimum number of stages. In Section 6, we compute the maximum streaming length (or worst case) induced by (pure-strategy) equilibria. In the following Section 7, we show that a simple strategy restriction mechanism allows to minimize, at equilibrium, both streaming length and stage-wise congestion. In Section 8 the described model is translated into a distributed algorithm to be executed independently on each peer; its functioning is evaluated in Section 9. Some remarks conclude the paper (Section 10).

2. BACKGROUND

Current approaches to multimedia streaming provide methods for the construction of a P2P, adaptive overlay network allowing to distribute contents across peers [5]. Collaborative behavior may be achieved in different ways by suitably designing peer interaction so to avoid free-riding [13]. This yields fair protocols, enabling to rapidly distribute content units produced by broadcasters. In one direction, BitTorrent philosophy led to design *tit-for-tat*-based policies, thereby guaranteeing that every participant contributes to satisfying others' needs [4]. Along another route, P2P streaming may resort to a pseudo-random partner selection in a gossip scheme where peers have no incentive to deviate from the protocol [10]. In general, the issue may be dealt with by modeling multicast protocols as non-cooperative games [8] whose equilibrium outcomes display (socially) desirable features.

Among non-cooperative approaches, potential games [12] include congestion ones as a sub-class. Routing problems probably constitute the main networking issue which is fruitfully framed in terms of congestion (or, more generally, potential) games. Broadly speaking, in basic routing problems modeled as games there is a given network one of whose nodes is the common destination. Then, each player is assigned to some origin node, and has to reach the destination through a route (or origin-destination path) provided by the network. Each arc induces a cost, which is greater the greater the number of players who choose a route using that arc, so that players selfishly aim at minimizing their own route cost. This game can be enriched in different ways; for example, it may be assumed to be repeated [1], so to exploit best response dynamics [2, 3] or more. In addition, for certain cases even strong equilibrium has been analyzed in detail [6, 7, 18]. Finally, from another perspective, important results have also been derived through cooperative game-theoretical approaches [15].

Congestion game modeling of live streaming crucially has to take into account that the game intrinsically must be multistage: for given numbers of content units and peers, any course of the streaming is fully characterized by a time-sequence of directed trees, each spanning all nodes, and specifying how each content unit reaches each peer from the broadcaster. Note that what game is actually played at each stage depends on what content distribution over peers does prevail at that stage. In fact, a main concern in this dynamic setting is streaming length. More specifically, each player aims at receiving all the content as fast as possible, but how long the whole dissemination will actually take depends on how peers distribute their requests to receive at each stage. This is a typical coordination issue, where selective and timely dissemination of content units deeply affects the overall performance of the system. Our analysis focuses on strategy restriction mechanisms enabling selfish agents to reach overall coordination through optimizing behavior. More specifically, in this paper the focus is on costs measured in terms of (expected) streaming length or duration, and thereby congestion seems most suitably measured by counting, at each stage, how many requests for additional content reach each peer and the source.

Finally, participants are assumed to fully cooperate, in that they all satisfy one request whenever they receive some valid one (see below). In fact, a main tool for limiting free-riding is the capability to tag those who do not cooperate, so

that everybody can recognize them. This is especially true in dynamic environments, where each participant has to interact one-to-one with several others, several times, as in our case. In this respect, the proposed method constructs a new P2P network for each content unit to be distributed, and thereby each peer is expected to interact with all other peers several times during the whole course, therefore incentive-based schemes are applicable [14].

3. PRELIMINARIES

We assume stream production to occur over a finite time-sequence $t = 0, 1, \dots, T$. Starting at time $t = 0$, a *source* provides one new unit c^t of content at the beginning of each time-period (or stage/round) $t \rightarrow t + 1$, until the end T is reached, while n *peers* aim at receiving all the content as soon as possible. This may be modeled as a *multistage game* [11] by means of *congestion game forms* [7], with player set $N = \{1, \dots, n\}$ and naturals $1, \dots, n$ being peers' identifiers. Also denote the source by 0 and let $N_0 = \{0, 1, \dots, n\}$.

A multistage game is identified by a *tree* each of whose nodes corresponds to a moment at which at least one player has to take action, and thereby *rooted* at the start, which here is when the source has just finished producing the very first content unit c^0 and peers begin to line-up in order to receive it. All paths from the start to some node referring to a fixed time t bijectively correspond to all distinct courses the game may take up to t . What nodes are actually reached, during any course, obviously depends on what actions players actually take. In *stochastic* games, for given per-node actions what successive node is actually reached depends on some random event. We assume *perfect information*: when asked to take action, at any time t , all players know exactly what node has been reached at t . The *leaves* of the tree correspond to outcomes, over which players have preferences. A *strategy*, for a player, specifies an (admissible) action to take at each node. The *rules* of our game are

Ru1: the source can send each unit $c^t, 0 \leq t \leq T$ over time period $t \rightarrow t + 1$ only (i.e. just after that unit is produced) and, additionally, to only one (i.e. some) peer;

Ru2: if a peer at the end t of any time-period $t - 1 \rightarrow t$ has already received units c^{t_1}, \dots, c^{t_h} , with $0 \leq t_1, \dots, t_h < t$, then in the following time-period $t \rightarrow t + 1$, or equivalently in the following *stage* or *round* t , this peer can send only one of such units and to one (i.e. some) other peer only;

Ru3: in any round, each peer may ask to receive some unit either from one other peer, or else from the source.

With these rules, spreading the whole content (or time-sequence of units) $C ON = \{c^0, c^1, \dots, c^t, \dots, c^T\}$ over the whole peer set surely requires much P2P exchange, as the source sends each unit precisely once, to only one peer, and next the unit has to be P2P-exchanged exactly $n - 1$ times. For turning this into a game, the first step is determining the players, which clearly are the peers. Next, the nodes in the game tree have to be identified: a time-indexed node C^t , referring to time t , is a $n + 1$ -set $C^t = \{C_0^t, C_1^t, \dots, C_n^t\}$ where each $C_i^t \subseteq \{c^0, c^1, \dots, c^t\}$ specifies what content units up to t peer $i \in N_0$ has received. Hence, a game tree node C^t may be looked at as a $0 - 1$ -matrix with i, t' -entry $C_i^t(t') \in \{0, 1\}$ defined by $C_i^t(t') = 1$ if $c^{t'} \in C_i^t$ (i.e. if in node C^t peer/source $i \in N_0$ has unit $c^{t'}$) and 0 otherwise. Notice that, given Ru1, the source is constrained to provide at each time only the current unit: $C_0^t(t') = 1$ if $t' = t$

and 0 otherwise for all $0 \leq t \leq T$, while $C_0^t(t') = 0$ for all $t > T, t' \geq 0$. For each $i \in N$, a strategy specifies, for each game tree node C^t , some $j \in N_0$ from whom to ask, in round t , a content unit $c^{t'}$, $0 \leq t' \leq t$. These strategies are finite sequences as long as some upper bound $T_* < \infty$ on duration (in stages) of the game, over all conceivable courses, exists.

In the sequel, this worst-case (or longest) streaming length at equilibrium T_* is observed in terms of the *streaming tree*, whose vertex set N_0 contains all peers together with the source, and through which each content unit spreads over the whole peer population. In fact, as already outlined, the streaming tree is a nested time-sequence of *sub-streaming trees*, one for each content unit. Specifically, the whole streaming, from an ex post perspective, is identified by the $T+1$ trees, with peers as vertexes, describing how each of the $T+1$ content units actually reaches the whole population. In each of these trees edges are directed and, in particular, the in-degree is 1 for each vertex (apart from the source). That is, while in general peers may well forward a content unit several times (to several different other peers, in different rounds), they receive each content unit only once.

Let \mathcal{N}^t denote the set of all game tree nodes referring to time $t \geq 0$. That is, \mathcal{N}^t is the family of all $(0-1)$ -matrices specifying content distributions over peers that may be reached along some game course up to t , when the above main constraint Ru1-3 applies: in any round each peer can forward at most one unit and can also submit at most one request for receiving. In our model, a strategy A^i for peer $i \in N$ has form $A^i : \mathcal{N} \rightarrow N_0$, where $\mathcal{N} = \bigcup_{0 \leq t \leq T_*} \mathcal{N}^t$,

and with $A^i(C) = j$ denoting the one (i.e. peer or source) $j \in N_0$ from whom i asks to receive at game tree node $C \in \mathcal{N}$. Hence $j \in \{0, \dots, i-1, i+1, \dots, n\}$, although we interpret $A^i(C) = i$ as one possible way (see below) in which peer i at node C does not ask to receive any content, from anybody. Thus, all game tree nodes are unified into a single set \mathcal{N} , independently from what different game courses let them prevail, and at what time. The resulting analysis provides behavioral rules according to which players respond to any realized content distribution over the population, at any time. The main concern is what game course length (or number of rounds needed to distribute the whole content over all peers) may be the outcome of selfish, node-wise optimization by players.

4. CONGESTION GAMES AND FORMS

In a congestion game form there is a set N of players and a set M of facilities, and each player $i \in N$ has a set $\Sigma^i \subseteq 2^M$ of strategies, where 2^M is the (power) set of all subsets of M . Usually, M is the edge set of a graph, and each player $i \in N$ has to reach a destination v_i^d from an origin v_i^o . Then, the set Σ^i of strategies for i contains all (edges of) $v_i^o - v_i^d$ -paths.

A *congestion game form* $F = (N, M, \Sigma^1 \times \dots \times \Sigma^n)$ identifies a whole class of congestion games, each obtained by specifying the payoffs $\pi^i : \Sigma \rightarrow \mathbb{R}_+$ of players $i \in N$, where $\Sigma = \Sigma^1 \times \dots \times \Sigma^n$. Profile $A = \{A^1, \dots, A^n\} \in \Sigma$ of strategies identifies *congestion vector* $\sigma(A) = \{\sigma_a(A) : a \in M\}$ specifying how many players have each facility $a \in M$ in their strategy A^i . That is, $\sigma_a(A) = |\{i \in N : a \in A^i\}|$. The game is *monotone* when each $a \in M$ has an associated utility function $u_a : \mathbb{Z}_+ \rightarrow \mathbb{R}_+$ satisfying $u_a(k) < u_a(k')$ whenever $k > k'$, and each $i \in N$ gets a payoff given by the sum over all the chosen facilities $a \in A^i$ of the corresponding util-

ity: $\pi^i(A) = \sum_{a \in A^i} u_a(\sigma_a(A))$. Finally, a congestion game form (and any game derived from it) is *symmetric* when the strategy set is the same across players: $\Sigma^1 = \dots = \Sigma^n$ [7].

P2P streaming systems may be approached through congestion games with facilities being players themselves: every strategy profile $A = (A^1, \dots, A^n)$ has an associated *congestion matrix* $\sigma(A) = \{\sigma_C^i(A) : C \in \mathcal{N}, i \in N_0\}$, where

$$\sigma_C^i(A) = |\{j \in N : i = A^j(C), C_i \not\subseteq C_j\}|$$

is the number of peers who ask to receive from $i \in N_0$ some content that *this latter has but they miss*. That is to say, at any reached game tree node $C \in \mathcal{N}$, if a peer asks to receive from someone who has no additional content, then such a request is simply ignored by the system: it causes null congestion. A request is *valid* if it contributes to congestion.

Denote by $\kappa = |\mathcal{N}|$ the whole number of game tree nodes. A strategy A^i for a peer $i \in N$ can be regarded as a point $A^i \in N_0^\kappa$, as it specifies somebody (although possibly with no additional content) to ask from at each node $C \in \mathcal{N}$ that may be reached. Hence, the corresponding congestion game

$$\text{form is } F = \left(N, N_0^\kappa, \underbrace{N_0^\kappa \times \dots \times N_0^\kappa}_n \right).$$

Players' payoffs $\pi^i : N_0^\kappa \rightarrow \mathbb{R}_+$ are assumed to consist of a sum over nodes of some (possibly 0) utility or *per-node* payoff received at each $C \in \mathcal{N}$. This utility depends on the prevailing content distribution (which is precisely what the game tree node C specifies), and on the profile $A^1(C), \dots, A^n(C)$ of *per-node* strategies that players choose at node C . Hence, per-node payoffs received by peers $i \in N$ may depend on congestion, which here is the number of other peers $i' \in N$ with the same (valid) per-node strategy $A^{i'}(C) = A^i(C)$. This models real-time streaming in terms of congestion games with facilities being pairs (j, C) , where $j \in N_0$ is either a peer or the source and $C = C^t$ is a game tree node or content distribution that may prevail at some time t .

In a simplest form¹ for $\pi^i : N_0^\kappa \rightarrow \mathbb{R}_+$ ($i \in N$), the payoff $\pi^i(A)$ of any given strategy profile $A \in N_0^\kappa$ to a peer $i \in N$ is the sum, over all conceivable game tree nodes $C \in \mathcal{N}$, of the values taken by utility u_{C_j} , which in turn depends only on congestion $\sigma_C^j(A)$, that is,

$$\pi^i(A) = \sum_{C \in \mathcal{N}} \sum_{\substack{j \in N_0 : C_j \not\subseteq C_i \\ A^i(C) = j}} u_{C_j} \left(\sigma_C^j(A) \right). \quad (1)$$

For any strategy profile A , each peer $i \in N$ gets a utility at each node $C \in \mathcal{N}$ which depends exclusively on the number $\sigma_C^j(A)$ of those with the same (valid) per-node strategy.

A profile $A = (A^1, \dots, A^n)$ is Pareto-optimal if there is no profile $B = (B^1, \dots, B^n)$ such that $\pi^i(B) \geq \pi^i(A)$ for all $i \in N$, with strict inequality for at least one i . Hence, from an aggregate perspective, Pareto-optimal profiles are efficient: there is no chance of improving someone's payoff without deteriorating someone else's one. Congestion games allow for neat conditions under which desirable properties, such as Pareto-optimality and strength of equilibrium, attain. In fact, in symmetric and monotone such games, these properties depend on the structure of the union $\Sigma^U = \bigcup_{i \in N} \Sigma^i$ of

¹Further payoff functions, possibly also depending on the difference between the sender's and the receiver's contents (and hence P2P-specific [9]), are considered in [17].

strategy spaces. In particular, on whether a *bad configuration* appears or not. Formally, Σ^U displays a bad configuration when there are three strategies $X, Y, Z \in \Sigma^U$ and two facilities $x, y \in M$ such that $x \in X \not\supseteq y$ and $x \notin Y \supseteq y$ but $x \in Z \supseteq y$. Thus, two facilities give rise to a bad configuration if there are strategies in Σ^U which use one of them but not the other, and there is also a strategy in Σ^U which uses both of them. The latter never occurs if Σ^U consists of singletons [7, pp. 87-88]. As the name suggests, it is desirable that no bad configuration exists. Here facilities are players themselves, although any fixed player corresponds to two distinct facilities when referring to two distinct game tree nodes. By Ru3 above, strategies are time-sequences of singletons, and hence the safe case applies.

If $\pi^i(A^{-i}, A^i) \geq \pi^i(A^{-i}, B^i)$ for all $i \in N$ and $B^i \in N_0^\kappa$, where $A^{-i} \in N_0^{\kappa(n-1)}$ is a $n-1$ profile for peers $j \in N \setminus i$ as well as $A^i \in N_0^\kappa$ is a strategy for peer i , then (A^{-i}, A^i) is an equilibrium. In particular, $A = (A^1, \dots, A^n)$ is a strong equilibrium if for no coalition $\emptyset \neq S \subseteq N$ is there a choice of $B^i \in N_0^\kappa$ for coalition members $i \in S$ such that $\pi^i(B^S, A^S) > \pi^i(A)$ for all coalition members $i \in S$, where (B^S, A^S) denotes the profile in which each $i \in S$ chooses B^i and each $j \in S^c = N \setminus S$ chooses A^j . In words, no non-empty coalition can deviate from strong equilibrium profiles and thereby strictly increase the payoffs of *all* its members.

When considering the implications of strong equilibrium for $S = N$, one gets rather similar conditions as those identifying Pareto-optimal profiles. In fact, as strategies are time-sequences of singletons, the model provided thus far yields a symmetric monotone congestion game with no bad configuration, where therefore the set of strong equilibria is non-empty, coincides with the set of equilibria and, generically, is (weakly) included in the set of Pareto-optimal profiles [7].

Given payoff functions $\pi^i : N_0^{\kappa n} \rightarrow \mathbb{R}_+$ for peers $i \in N$, any function $P : N_0^{\kappa n} \rightarrow \mathbb{R}$ is a potential if for all $i \in N$, all $n-1$ -profiles $A^{-i} \in N_0^{\kappa(n-1)}$ and all pairs $A^i, B^i \in N_0^\kappa$

$$[P(A^{-i}, A^i) - P(A^{-i}, B^i)][\pi^i(A^{-i}, A^i) - \pi^i(A^{-i}, B^i)] \geq 0.$$

A potential is *exact* when the two differences within square parentheses are equal. In words, a potential takes values on strategy profiles, and for any such a profile and unilateral deviation from it, the deviating player's payoff and the potential itself change in the same direction. Exactness means that these changes are equal.

Claim: with payoffs (1), an exact potential is (for $A \in N_0^{\kappa n}$)

$$P(A) = \sum_{C \in \mathcal{N}} \sum_{j \in N_0} \sum_{1 \leq k \leq \sigma_C^j(A)} u_{C_j}(k). \quad (2)$$

Proof: fix $i \in N$ and $A^i, B^i \in N_0^\kappa$ as well as a $n-1$ -profile $A^{-i} \in N_0^{\kappa(n-1)}$ of strategies for peers $j \in N \setminus i$. Consider congestion matrices $\sigma(A)$ and $\sigma(B)$ associated, respectively, with $A = (A^{-i}, A^i)$ and $B = (A^{-i}, B^i)$. Their elements $\sigma_C^j(A)$ and $\sigma_C^j(B)$ ($j \in N_0$) differ only if $C_j \not\subseteq C_i$ and either $A^i(C) = j \neq B^i(C)$ or $A^i(C) \neq j = B^i(C)$. In the former case $\sigma_C^j(A) = \sigma_C^j(B) + 1$, while in the latter $\sigma_C^j(A) = \sigma_C^j(B) - 1$. Then, $P(A) - P(B) =$

$$\sum_{C \in \mathcal{N}} \left(\sum_{\substack{j \in N_0: C_j \not\subseteq C_i \\ A^i(C)=j \neq B^i(C)}} u_{C_j}(\sigma_C^j(A)) - \sum_{\substack{j \in N_0: C_j \not\subseteq C_i \\ A^i(C) \neq j = B^i(C)}} u_{C_j}(\sigma_C^j(B)) \right)$$

$$= \pi^i(A) - \pi^i(B). \bullet$$

For monotone congestion games with no bad configuration, (2) is a strong potential, any of whose maximizers is a strong equilibrium [7, theorem 5.2].

5. ON MINIMAL STREAMING LENGTH

One indicator of streaming efficiency is simply the number of rounds needed to spread the whole content CON over the whole peer set N . Assume the number of peers is a power of 2, that is, $n = 2^m$ for some natural m . Under our assumptions Ru1-3, any content unit can spread over the whole population no faster than through $m+1$ (consecutive) rounds. For example, in round 0 the very first content unit c^0 will go from the source to some peer i_1^0 . In round 1 it will go from i_1^0 to some other peer $i_2^0 \in N \setminus i_1^0$. In round 2 it will go from i_1^0, i_2^0 to two distinct other peers $i_3^0, i_4^0 \in N \setminus \{i_1^0, i_2^0\}$, and so on, doubling the forwards in each round, until in round m (which is the $m+1$ -th round that this unit c^0 circulates) exactly half of the population sends the content unit to the other half through one-to-one matching. A crucial fact from now on is that *all* the $T+1$ content units, for any T , can spread over the whole peer set in exactly $m+1$ rounds. This can be observed in terms of the different possibilities for building the whole streaming tree (see above). In particular, if a generic content unit c^t reaches everybody in $m+1$ rounds, then the number of peers who send (and therefore also the number of those who receive) this content unit c^t in round $t+k$ is 2^{k-1} for $k = 1, \dots, m$.

If *all* content units must reach everybody in $m+1$ rounds, then whenever a peer receives a unit c^t in round $t+k$ it must forward c^t for the remaining rounds $t+k+k'$, where $k' = 1, \dots, m-k$. Hence, this peer for these latter rounds cannot receive units to be further forwarded. That is to say, in any round $t+k+k'$, with $k' = 1, \dots, m-k$, if this peer receives some unit, then such a unit must be $c^{t+k+k'-m}$, in which case this round $t+k+k'$ is precisely the $m+1$ -th (i.e. last) one in which this unit $c^{t+k+k'-m}$ circulates. In other words, the peer must be among those 2^{m-1} who are the last ones to receive that unit. This not only is feasible, but can be obtained through many different streaming trees².

Definition: profile $A \in N_0^{\kappa n}$ is *deterministic* if

(I) for all $i \in N$ both the following hold:

(a) $|\{C \in \mathcal{N} : A^i(C) = j, C_j \not\subseteq C_i\}| = T+1$,

(b) $A^i(C) = j, C_j \not\subseteq C_i \Rightarrow |C_j \setminus C_i| = 1$ for all $C \in \mathcal{N}$.

Hence, each peer makes exactly $T+1$ valid requests to receive (which under Ru1-3 clearly is the minimum number of valid such requests needed to receive all the $T+1$ content units c^0, c^1, \dots, c^T), and therefore receives some (distinct) content unit every time a valid request is made. Also, the $T+1$ valid requests made by any peer i are all addressed, each at a different game tree node, to someone who at that node has precisely one additional content unit.

The name *deterministic* is due to the assumption that transitions from one game tree t -node $C^t \in \mathcal{N}^t, t \geq 0$ to $t+1$ -nodes $C^{t+1} \in \mathcal{N}^{t+1}$ are stochastic: a generic strategy profile A does not yield a unique game course, but a probability distribution over game courses. Whatever its

²Although in all of them the sub-streaming tree for each unit must result from a suitable permutation of vertexes/peers, thereby always reproducing the same essential condition that for each unit c^t the number of senders doubles after each round $t+h$ for $h = 1, \dots, m$, vanishing afterward.

form, an underlying probabilistic model essentially decides who gets what when multiple peers i_1, \dots, i_k ask to receive from a common $j \in N_0$ such that $C_j^t \not\subseteq C_{i_{k'}}^t, 1 \leq k' \leq k$. As peers (and the source) can forward at most one unit per round, such a model has to select precisely one peer $i_{k'} \in \{i_1, \dots, i_k\}$ and one unit $c^{t'} \in C_j^t \setminus C_{i_{k'}}^t$ to be received by the former. Deterministic profiles actually allow to ignore the underlying probabilistic model (which is intended to be complex and mostly unknown), because the transition from any node to successive ones becomes deterministic.

Deterministic profiles put probability 1 on one game course and probability 0 on all other courses. Accordingly, consider the unique content distribution over peers (or game tree node) reached at t by deterministic profile $A \in N_0^{kn}$. In particular, denote it by $C^t(A) = \{C_0^t(A), C_1^t(A), \dots, C_n^t(A)\}$.

Definition: a deterministic profile $A \in N_0^{kn}$ is *fastest streaming* if for all $0 < k \leq t \leq T_*$

(II) $|\{i \in N : C_i^t(A) \ni c^{t-k}\}| = \min\{2^m, 2^{k-1}\}$.

These profiles spread each content unit c^t over $2^0 = 1$ peer in round t , over (new) $2^0 = 1$ peer in round $t+1$, over (new) $2^1 = 2$ peers in round $t+2$, and so on, until (new and final) 2^{m-1} peers receive unit c^t in round $t+m$, which is the $m+1$ -th (i.e. final) round where this unit circulates. Let \mathcal{A}^* be the set of fastest streaming profiles. As $\sum_{0 \leq h < k} 2^h = 2^k - 1$, summing newly reached peers across these $m+1$ rounds yields that each unit c^t reaches the whole population in $m+1$ rounds. That is, the whole peer set is covered at the end of round $t+m$, as $2^0 + 2^0 + 2^1 + \dots + 2^{m-1} = 2^0 + 2^m - 1 = 2^m$. In addition, in these profiles $A \in \mathcal{A}^*$ peers $i \in N$ receive each and every time t they ask something from someone (i.e. some $j = A^t(C^t)$ with $C_j^t \not\subseteq C_i^t$). Given Ru1-3, these conditions just listed are rather demanding, and one may well wonder whether $\mathcal{A}^* \neq \emptyset$ at all. Accordingly, $|\mathcal{A}^*|$ is now determined.

Consider a generic t such that $m \leq t \leq T$. For any $A \in \mathcal{A}^*$, in round t there are exactly $m+1$ content units $c^t, c^{t-1}, \dots, c^{t-m}$ being distributed across the whole population, out of which precisely m (i.e. $c^{t-1}, c^{t-2}, \dots, c^{t-m}$) are sent by some peers to some other peers, while one unit (i.e. c^t) is sent from the source to some suitably chosen peer. Hence, in this round t each peer is a receiver (of some unit $c^{t-k}, 0 \leq k \leq m$). Conversely, only $2^m - 1$ peers also send (units $c^{t-k}, 1 \leq k \leq m$), as the source forwards c^t .

Conditions (I) and (II) may be turned into a useful recursive method for establishing, for any course of the game reached up to any time $t \geq 0$ (and thus applying since the very beginning), how to proceed in round t in order to have a A -induced streaming for some $A \in \mathcal{A}^*$. In fact, all scheduling priorities can be captured by the following main constraint: for any $t \geq 0$, if in the previous round a peer has received and/or forwarded some unit that will have to be forwarded in round $t+1$ as well, then in this round t this peer cannot receive any unit that will also have to be forwarded in round $t+1$. Let S_t^k be the subset of peers who *send* unit c^{t-k} in round t ($1 \leq k \leq m$), and R_t^k be the subset of peers who *receive* unit c^{t-k} in round t ($0 \leq k \leq m$).

Recurrence:

Re1 - if $i \in S_{t-1}^k$ or $i \in R_{t-1}^k$ for some $k \leq m-2$

Re2 - or $t \geq m$ and $i \notin S_{t-1}^k$ for all $k \geq 1$,

Re3 - then $i \notin R_t^k$ for all $k \leq m-1$.

Consider that any unit $c^{t'}$ has to be (still) forwarded in round $t+1$ (i.e. over $t+1 \rightarrow t+2$) when $t'+m \geq t+1$. Now focus on a generic unit c^{t-1-k} that a peer is either sending

or else forwarding in round $t-1$. Letting $t' = t-1-k$ we have that this unit will have to be forwarded in round $t+1$ if $t-1-k+m \geq t+1$, that is, if $k \leq m-2$. If this is the case, then the peer cannot now (i.e. in round t) receive any unit c^{t-k} such that $t-k+m \geq t+1$ or $k \leq m-1$. On the other hand, Re2 entails precisely that when we reach any round $m \leq t \leq T+m$ all peers who still have not received unit c^{t-m} are matched with the other half of the population, so to ultimately receive such a unit, that will no longer be distributed throughout the whole streaming.

We now proceed to counting all streaming trees that satisfy this recurrence Re1-3. Recall that for naturals $a \geq b$ product $[a]_b = a(a-1)(a-2) \dots (a-b+1)$ is the *falling factorial*. Our enumerative concern is with situations where $T > m$, and the whole streaming evolves in *three phases*:

Ph1: comprehends all initial rounds $t = 0, 1, \dots, T_1$ where at least one peer does not receive any content unit,

Ph2: comprehends all rounds $t = T_1 + 1, T_1 + 2, \dots, T_1 + T_2$ where each peer receives a unit,

Ph3: comprehends all rounds $t > T_1 + T_2$ where some peer does not receive any unit but some other peer receives one.

Claim: the number of fastest streaming profiles is $|\mathcal{A}^*| =$

$$\left(\prod_{t=0}^{m-1} [2^m - 2^t + 1]_{2^t} \right) \Phi(T, m) \left(\prod_{t=1}^m [2^{m-1}]_{2^{m-1}-2^{t-1}} \right)$$

where $\Phi(T, m) = (2^{m-1}!)^{2T+2-m}$.

Proof: the needed counting procedure (detailed in [17]) may be sketched as follows. There are three phases, out of which the first and final ones display an excess demand of content units, while demand equals supply in the central phase.

Ph1: (II) entails that in rounds $t = 0, 1, 2, \dots, m-1$ all needed receivers must be chosen among those who still have not received anything, yielding 2^t new peers involved in the streaming at each t . As $\sum_{0 \leq h < k} 2^h = 2^k - 1$, at time m (i.e. at the end of round $m-1$ and at the beginning of round m) all peers apart from one (i.e. $2^m - 1$) have received precisely one unit. The number of distinct ways to achieve this, or number of alternative streaming tree evolutions induced by profiles $A \in \mathcal{A}^*$ in rounds 0 to $m-1$ inclusive, is

$$\prod_{t=0}^{m-1} \binom{2^m - 2^t + 1}{2^t} 2^t! = \prod_{t=0}^{m-1} [2^m - 2^t + 1]_{2^t}.$$

Ph2: in the starting round m , a unique peer $i^* \in N$ received nothing in previous (initial) rounds. By rules Ru1-3 above, from now on until the end, in any round t peer i^* must be in the half of the population who receives from the other half precisely the unit c^{t-m-1} whose distribution terminates in that round, thereby never forwarding any unit. Also, in all successive rounds $t = m, m+1, \dots, T$ the set of senders contains the source together with $2^m - 1 = |N \setminus \{i^*\}|$ peers, and these latter are partitioned into m blocks with cardinalities $2^0, 2^1, \dots, 2^{m-1}$ whose members send respectively units $c^{t-1}, c^{t-2}, \dots, c^{t-m-1}$. Counting the number of distinct ways to match demand and supply in these central rounds $t = m, m+1, \dots, T$ (where they both involve 2^{m-1} participants, but twice) gives $(2^{m-1}!)^{2(T+1-m)}$ different evolutions available between round m and round T inclusive.

Ph3: in the first of final rounds $t = T+1, T+2, \dots, T+m$, a unique peer $i_{T+1}^* \neq i^*$ does not receive any unit. Hence, at time $T+2$ (i.e. at the end of round $T+1$) an additional peer i_{T+1}^* , like i^* , still misses units $c^{T+2-m}, c^{T+3-m}, \dots, c^T$.

In view of Re2, in round $T + 2$ this peer i_{T+1}^* , as a receiver, matches some c^{T+2-m} -sender. Till the end of the streaming this peer will never be among those who receive units to be further forwarded. The number of peers such as i_{T+1}^* (or i^*) increases by 2^{t-T-1} in each remaining round t until the end. In fact, in the last round $T + m$, half of the peer send (without receiving anything) what the other half receives. Repeating the same argument used above for phase Ph1, but in an opposite manner, between rounds $T + 1$ and $T + m$ inclusive the streaming tree may be checked to evolve in

$$\prod_{t=1}^m 2^{m-1}! \binom{2^{m-1}}{2^{m-1} - 2^{t-1}} (2^{m-1} - 2^{t-1})! =$$

$$= (2^{m-1}!)^m \prod_{t=1}^m [2^{m-1}]_{2^{m-1} - 2^{t-1}} \text{ different possible ways. } \bullet$$

Hence, many different streaming tree evolutions allow to spread the whole content over the whole peer population in a way such that each unit c^t reaches *new* 2^{k-1} peers in each round $t + k$ for $k = 1, \dots, m$. Also note that all of them satisfy condition (I.a), as at any node C and for any two peers $i, j \in N$ we have $|C_i \setminus C_j| \in \{0, 1\}$. Still, with payoffs given by (1) above, fastest streaming is not sustainable at equilibrium, because condition (I.b) is too demanding: selfish (and myopic) peers try to receive some unit in any round until they get the whole content.

6. WORST-CASE EQUILIBRIUM

Given the P2P setting, where peers always satisfy precisely one (randomly selected) valid request among those received, in each round the number of distributed units equals the number of those who are asked to forward through some valid request. At equilibrium such a number equals the number of those who (at the beginning of the round) have some units that someone else is missing.

Claim: the upper bound for equilibrium streaming length is $T_* = T + 2^m + 1$.

Proof: let $N_0^t \subseteq N_0, t \geq 0$ be the set containing all those who at t have some content that someone else still misses: $N_0^t = \{j \in N_0 : C_j^t \not\subseteq C_i^t \text{ for some } i \in N\}$. Also let $N^t \subseteq N, t \geq 0$ contain all those who at t still miss some unit: $N^t = \{i \in N : C_i^t \not\supseteq C_j^t \text{ for some } j \in N_0\}$. Note that $N = N^t, 0 \leq t \leq T$, because at t no peer has unit c^t yet, but the source $0 \in N_0$ does. Recall that players always prefer to make a valid request through a link where congestion is 1 rather than through one where congestion is > 1 (i.e. independently from what non-empty additional content is reachable through the links). On the other hand, at equilibrium each peer makes a valid request in each round as long as the whole content CON is not fully received. Hence,

$$|\{j \in N_0^t : \sigma_{C^t}^j(A) > 0\}| = \min\{|N_0^t|, |N^t|\} \quad (3)$$

for any node C^t that may be reached at any time $0 \leq t \leq T_*$ along some game course. In any round t the number of forwarded (and received) units equals the minimum between the number of those who at t have a unit that at least one peer is still missing and the number of those who at t still miss some unit. Accordingly, streaming length is maximized when $|N_0^t|$ is kept to its minimum at each t . Any of the units minimally takes $m + 1$ rounds to be distributed over the whole population. While the source puts units in circulation, the number of those who have one unit that someone else misses minimally increases by 1 in each round, till either

all peers have a unit that someone else is missing, or else the content production ends, whatever first: $|N_0^t| \geq |N_0^{t-1}| + 1$ for $0 < t < \min\{2^m - 1, T\}$, with $N_0^0 = \{0\}$. Maximum duration occurs when this last inequality is an equality, requiring in turn that each unit, up to round $t = \min\{2^m - 1, T\}$ inclusive, is streamed along a same *chain*: each unit is received by peers in a fixed order, such as the natural one $1 < 2 < \dots < n$. If $T \leq 2^m - 1$, then such a linearly ordered streaming induces, at the end of round T , the game tree node $C^{T+1} = (C_1^{T+1}, \dots, C_n^{T+1})$ where $C_1^{T+1} = CON$ and $C_i^{T+1} = C_{i-1}^{T+1} \setminus c^{T-i+1}$ for $i = 2, \dots, n$. That is, maintaining the example given by the natural order for the sake of simplicity, at $T + 1$ peer 1 has the whole content CON , while each peer $i = 2, \dots, n$ misses the last $i - 1$ units $c^T, c^{T-1}, \dots, c^{T-i+1}$. Now consider the whole streaming tree: it must (eventually) consist of $(T+1)n$ (directed) edges, out of which $T + 1$ have the source as one end-vertex, while $(T + 1)(n - 1)$ are P2P links. At game tree node C^{T+1} the source has exhausted its role, and $\sum_{2 \leq i \leq n} (i - 1) = \binom{n}{2}$ P2P links are still missing. In particular, peer n , at time $T + 1$, still misses units $C^T, C^{T-1}, \dots, C^{T-n+1}$. Accordingly, this peer needs further rounds $T + 1, T + 2, \dots, T + n$ for getting the whole content, because in each round only one unit can be received. Then, given that the number of rounds has to include the very first round 0, the whole streaming length is $T + n + 1 = T + 2^m + 1$. The case $T > 2^m - 1$ is handled in the same fashion (see [17] for details). \bullet

Although worst-case equilibrium streaming length is linear in both the whole number $T + 1$ of produced units and the whole number $n = 2^m$ of peers, it can be rather greater than the socially optimal streaming length. This gap may be closed by means of a dynamic mechanism for constraining per-node strategies.

7. STRATEGY RESTRICTION

In the (traditional) static scenario, the issue of strategy restriction arises most interestingly when players can choose more facilities [7], and is sometimes approached by introducing *mediators* in the routing system [18]. Still, in our setting it remains worth being addressed even when strategies are time-sequences of singletons. In particular, the focus is on simple rules of the form: certain peers $i \in N$ at certain nodes $C \in \mathcal{N}$ cannot ask to receive from certain $j \in N_0$. Hence, restrictions apply to per-node strategies, which in any case (already) are singletons.

As explained above, the proposed model provides a symmetric monotone congestion game with no bad configuration, where equilibria are strong and Pareto-optimal. Adding per-node strategy restrictions has to deal with symmetry [7]. Here such restrictions result anyhow: peers' per-node payoffs depend only on valid requests, hence at any $C \in \mathcal{N}$ and for any $i, i' \in N, j, j' \in N_0$, it shall be $C_j \not\subseteq C_i \supseteq C_{j'}$ and $C_{j'} \not\subseteq C_{i'} \supseteq C_j$. That is, i can make a valid request to j but not to j' , and the converse for i' .

It is possible that at an equilibrium strategy profile A each content unit $c^t, 0 \leq t \leq T$ reaches 2^{k-1} *new* peers in each round $t + k, k = 1, \dots, m$, and thus the whole population in $m + 1$ rounds, which is optimal in terms of streaming length. Yet, such a profile A cannot be deterministic, as any equilibrium profile A must result in a congestion $\sigma_C^j(A) > 1$ for some (j, C) -entries of the associated matrix $\sigma(A)$. In fact, each (greedy) peer $i \in N$ makes a valid request to receive from each node $\{C_0, C_1, \dots, C_n\} = C \in \mathcal{N}$ such

that $C_i \neq CON$. That is, equilibrium profiles A surely yield some congestion, and provide a streaming length which ranges from the optimal (i.e. minimum) one $T + m + 1$ to the worst-case one $T + 2^m + 1$. Then, for a social planner there are two priorities when designing restrictions: *i*) at equilibrium (with restrictions) streaming length should be $T + m + 1$, the same as with fastest streaming profiles; *ii*) congestion should be minimized.

A key fact is that those restrictions yielding minimal streaming length, i.e. *i*), also minimize congestion, i.e. *ii*). Indeed, in fastest streaming profiles there is a whole central phase Ph2 where streaming occurs through one-to-one matching involving all peers both as senders and as receivers (apart from one peer, who never forwards, and the source, who never receives). Therefore, any equilibrium satisfying *i*) also fulfills this latter requirement, which quantitatively translates into null congestion over phase Ph2.

Consider a strategy restriction mechanism which specifies from what $j \in N_0$ each peer $i \in N$ can ask for content at each node $C = C^t = (C_1^t, \dots, C_n^t)$ the game may reach up to any time t . In other terms, the mechanism specifies for any node $C = (C_0, C_1, \dots, C_n)$ and for any $i \in N, j \in N_0$ such that $C_j \not\subseteq C_i$, whether it may be $j = A^i(C)$ or not. In view of the above recurrence Re1-3, consider the following per-node restriction mechanism: for all $i \in N$ and $C^t \in \mathcal{N}$
Rm1: if $C_i^t \supseteq c^{t-k}$ for some $k \leq m - 1$, then $A^i(C^t) \neq j$ for all $j \in N_0$ such that $C_j^t \supseteq c^{t-k'}$ for some $k' < m$;
Rm2: if $C_i^t \not\supseteq c^{t-m}$, then $A^i(C^t) = j$ for some $j \in N_0$ such that $C_j^t \supseteq c^{t-m}$.

In this way, for any reachable game tree node C^t , strategies for round t are constrained precisely in the manner established by above constraint Re1-3 applying to streaming tree evolution. If, given previous history, a peer in t has some content unit c^{t-k} that must be forwarded in round $t + 1$ (i.e. such that $t - k + m \geq t + 1$), then in this round t the peer cannot ask to receive from those $j \in N_0$ who in t have units $c^{t-k'}$ to be also forwarded in round $t + 1$ (i.e. such that $t - k' + m > t$). Still, note that while Re1-3 are stated from the perspective of an overall coordinator (i.e. specifying forwarders and receivers), this Rm1-2 is more broadly stated in terms of contents. Then, the demand for new units gets synchronized (i.e. timely coordinated), and can thereby be driven by selfish behavior.

Any profile $A \in N_0^{\kappa n}$ yields a probability distribution $p_{C^t}^A$ over \mathcal{N}^{t+1} for each game tree node C^t that may be reached at any time $0 \leq t \leq T_*$. That is, $p_{C^t}^A(C^{t+1})$ is the probability of reaching node C^{t+1} from node C^t when chosen strategies are $A^i(C^t), i \in N$. Thus $\sum_{C^{t+1} \in \mathcal{N}^{t+1}} p_{C^t}^A(C^{t+1}) = 1$ for all $t \geq 0, C^t \in \mathcal{N}^t$ and $A \in N_0^{\kappa n}$. For given underlying probabilistic model (anyhow handling multiple valid requests whenever there are, see above), any strategy profile $A \in N_0^{\kappa n}$ puts a probability p^A on each game course $\{C^0, C^1, \dots, C^{T_*}\}$ (or sequence of content distributions) obtained as the following product of conditional probabilities

$$p^A(\{C^0, C^1, \dots, C^{T_*}\}) = \prod_{t=0}^{T_*-1} p_{C^t}^A(C^{t+1}), \quad (4)$$

where any t -th round³ starts at t and ends at $t + 1$. Let \mathcal{C}^A denote the set of all game courses that may prevail with strictly

³If T_* is the maximum conceivable streaming length, in rounds and starting with 0, then a game course ends at time T_* , when round $T_* - 1$ ends, although the whole content shall generically

positive probability through profile A . In other terms, \mathcal{C}^A contains all $T_* + 1$ -sequences $\{C^0, \dots, C^{T_*}\}$ or game courses on which A puts strictly positive probability, that is, such that $p^A(\{C^0, \dots, C^{T_*}\}) > 0$ as defined by (4).

Claim: if $A \in N_0^{\kappa n}$ is an equilibrium under Rm, then for any $\{C^0, \dots, C^{T_*}\} \in \mathcal{C}^A$ both the following hold:

(a) $|\{i \in N : c^t \in C_i^{t+k}\}| = 2^{k-1}$ for $k = 1, \dots, m + 1$ and all $0 \leq t \leq T$,

(b) $\sum_{j \in N_0} \max\{0, \sigma_{C^t}^j(A) - 1\} = \begin{cases} 2^m - 2^{t+1} + 1 & \text{for } 0 \leq t < m \\ 0 & \text{for } m \leq t \leq T, t = T + m \\ 2^{t-T-1} & \text{for } T < t < T + m \end{cases}$

Proof: without restrictions, at any equilibrium, peers make a valid request to receive content in each round as long as they miss at least one unit. With restrictions, until the whole CON is not received, peers make a valid request whenever restrictions allow them to. Therefore, the demand for content units (to be further forwarded) is always provided by peers' utility maximization at any node $C^t \in \{C^0, \dots, C^{T_*}\} \in \mathcal{C}^A$. Under constraint Rm1-2, content demand at each node is conveyed towards valid requests which, whenever satisfied, allow for minimal streaming length. On the supply side, peers (and the source) always satisfy precisely one (random) valid request among those received, and thus equilibrium conditions under Rm1-2 yields that any resulting game course distributes each content unit $c^t, 0 \leq t \leq T$ as desired.

Concerning (b), $\sigma_{C^t}^j(A) - 1$ is the number of *excess* valid requests for any pair (j, C) , that is, the number of non-satisfied valid requests that $j \in N_0$ receives at node C . Accordingly, $\sum_{j \in N_0} \max\{0, \sigma_{C^t}^j(A) - 1\}$ measures the whole (i.e. aggregate) number of excess valid requests at any reached node C^t . Given (a), any game course in \mathcal{C}^A provides a streaming with minimal duration, and thus evolves through the same phases Ph1-3 as fastest streaming, but non-deterministically. In other words, there is congestion, but only in the initial and final phases (Ph1,3), and in order to measure it, we have to check how many times the above reasoning on fastest streaming profiles uses one-to-one matchings between sets of different cardinalities. More specifically, if cardinality is the same, then the problem of finding some one-to-one matching is solved by equilibrium condition (3) above⁴. Conversely, if cardinality is different, then some of those in excess shall be left out. The number of these latter, at node C^t , is precisely $\sum_{j \in N_0} \max\{0, \sigma_{C^t}^j(A) - 1\}$. In fact, in any initial round $t = 0, 1, \dots, m - 1$ if a peer receives a unit then this peer will make no valid requests until round $t = m$ (i.e. the initial of Ph2)⁵. Summing up, in initial rounds $t = 0, 1, \dots, m - 1$, the numbers of (restricted) excess valid requests respectively are $2^m - 2^0, 2^m - 2^0 - 2^1, \dots, 1$, that is, $2^m - 2^{t+1} + 1$. Similarly, in rounds $t = T + 1, T + 2, \dots, T + m - 1$ the numbers of (restricted) excess valid requests respectively are $2^0, 2^1, \dots, 2^{m-2}$, that is, 2^{t-T-1} . Finally, in round $T + m$ clearly there are no excess valid requests. •

Restriction mechanism Rm1-2 above is useful for exploiting selfish behavior toward socially desirable outcomes. In particular, the mechanism is simple and, most importantly,

be completely distributed much in advance.

⁴When a set of senders and a set of receivers, both of same cardinality, have to match, at equilibrium receivers make their valid request each to a different sender.

⁵Otherwise, if a valid valid request was made and satisfied, then this peer would have two units to forward.

specifies conditions only in terms of the generic node C^t that may be reached at some time t during game course. The pattern through which the system reaches this node is irrelevant; all that matters for strategy restriction is content distribution over peers, which is precisely captured by the node itself. Any outcome or game course constrained through the mechanism provides minimal streaming length and (consequently, given Ph2) also minimizes congestion.

8. STRATEGY RESTRICTIONS AT WORK

We now detail an algorithm, hereinafter referred as ConGaS (*Congestion Games for Streaming*) for achieving minimal streaming length, when the number of peers is a power of two. If peers must be able to coordinate themselves towards fastest streaming, then they must be constantly endowed with perfect information. That is, they must be capable to “see”, during each round, who sends on-going content units to who. Technically, peers share a seed to randomly generate same sequences of pseudo-random numbers. Such shared seed serves as the needed coordination mean among nodes (i.e. it is employed to randomly select those peers that receive any given content unit).

ConGaS is described in the Algorithm that follows. During the initialization, the broadcaster (i.e. node 0) sends to all peers a generated seed value. The distribution loop consists of an iterative behavior: each iteration t corresponds to the production, at the broadcaster, of a novel content unit c^t to be distributed. Meanwhile, on-going content units $c^{(t-k)}$, $k = 1, \dots, m$ not yet been delivered to all peers are disseminated according to our method (lines 3-7 of the distribution loop). The `MANAGEDISTRIBUTION()` procedure, executed by all nodes, defines who sends what content unit to who. Differences in peers’ *actions* are simply determined accordingly to their *ids*.

At each iteration t , each peer is selected to receive a (single, new) content unit. This is achieved by picking nodes from an auxiliary list *NextFree*, which is initialized to N (line 2), and then progressively emptied through different calls of `MANAGEDISTRIBUTION()` (line 5). Specifically, given a content unit c^k being distributed ($t - m \leq k \leq t$), a bijection between those who have the unit (senders S^k in the code) and (some of) those who do not (receivers R^k in the code) is provided. Thus, for any content unit, at each h -th step of distribution, 2^h nodes have the content unit and 2^h are selected to be the receivers. Once the distribution for the on-going content units is specified, a new content unit is produced at the broadcaster (lines 9-11) and `MANAGEDISTRIBUTION()` is called for the novel content unit.

As for the *functions*, in `MANAGEDISTRIBUTION()` a new receiver *recv* is identified through `NEXTRECV()`. Next, `MUSTSEND()` schedules the delivery of the content unit from sender p to *recv*. Moreover, *recv* is added into S^k (the list of c^k senders; line 4)⁶. Accordingly, *recv* is removed from the list of receivers R^k (as well as from the list of those who may be selected as receivers of ongoing content units; lines 5-6). `NEXTRECV()` randomly selects receivers. In simple words, a (novel) node is picked until someone is found in the intersection of *NextFree* and *AvailRecvs* (i.e. the set of those who have not already been selected as receivers of the considered unit). Finally, `MUSTSEND()` makes thus far identified senders actually send the on-going units, based on the nodes’ *id*.

⁶In this way, this peer is marked as the c^k owner in the future.

Algorithm 1 ConGaS

Initialization

```

1:  $p = \text{IDNODE}()$ 
2: if ( $p == 0$ ) then
3:    $s \leftarrow \text{GENERATESEED}()$ 
4:   BROADCAST( $s$ )
5: end if

```

Distribution Loop

```

1: for all  $t \in [0, T]$  do
2:    $\text{NextFree} \leftarrow N = N_0 \setminus \{0\}$ 
3:   for  $k = \max(t - m, 0)$  to  $t$  do
4:     for all  $p \in S^k$  do
5:       MANAGEDISTRIBUTION( $p, k$ )
6:     end for
7:   end for
8:    $p = \text{IDNODE}()$ 
9:   if ( $p == 0$ ) then
10:     $c^t = \text{NEWCHUNK}()$ 
11:   end if
12:   MANAGEDISTRIBUTION( $0, t$ )
13: end for

```

```

1: function MANAGEDISTRIBUTION( $p, t$ )
2:  $\text{recv} \leftarrow \text{NEXTRECV}(R^t, s)$ 
3: MUSTSEND( $p, \text{recv}, c^t$ )
4: ADD( $\text{recv}, S^t$ )
5: REMOVE( $\text{recv}, R^t$ )
6: REMOVE( $\text{recv}, \text{NextFree}$ )

```

```

1: function NEXTRECV( $\text{AvailRecvs}, \text{seed}$ )
2: repeat
3:    $n \leftarrow \text{RANDOMPEER}(\text{seed})$ 
4: until  $n \in \text{NextFree} \cap \text{AvailRecvs}$ 
5: return  $n$ 

```

9. CONGAS EVALUATION

In order to evaluate our streaming method, ConGaS, a main issue is what benchmark should it be compared with. In the literature, such an issue is known as the *price of anarchy* [3]. In simple terms, given that ConGaS actually selects a subset of (Nash) equilibria, the aim is to compare the average outcome of such equilibria with respect to the average over *all* equilibria. Technically, an outcome is a (expected) utility value (i.e. a real quantity) for each player. The approach adopted here is more general: the focus is not on a specific congestion game, but rather on a congestion game form. Accordingly, results are valid for the whole class of monotone games derived from the game form [7]. Thus, the benefits of ConGaS over a generic equilibrium outcome quantitatively vary as the utility specification varies, but in any case the sooner peer receives content units, the better. Basically, anarchy is implemented simply by ignoring restriction mechanism Rm1-2. More specifically, at any stage $t \geq 0$, each peer $i \in N$ who still misses some unit randomly selects some valid forwarder $j \in N_0$, if any, and then receives the oldest unit in $C_j^t \setminus C_i^t$. Whenever the number of receivers exceeds that of forwarders, the probability of being among those who receive (and thus also that of being among those who do not) is the same across all potential receivers.

In order to compare two equilibrium strategy profiles without utility specification, we need to unify any such a profile into an index measuring its goodness. We already paid much attention to one such an index, that is, streaming length.

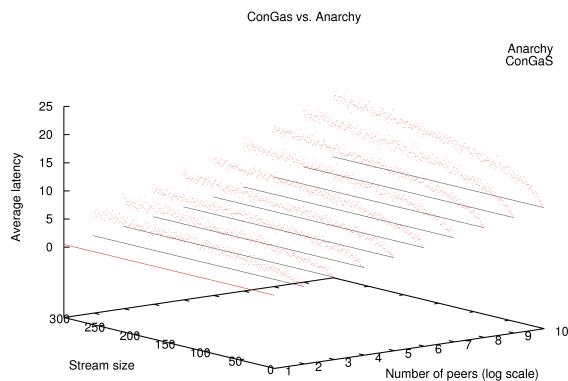


Figure 1: Average latency: varying number of both peers and stream size.

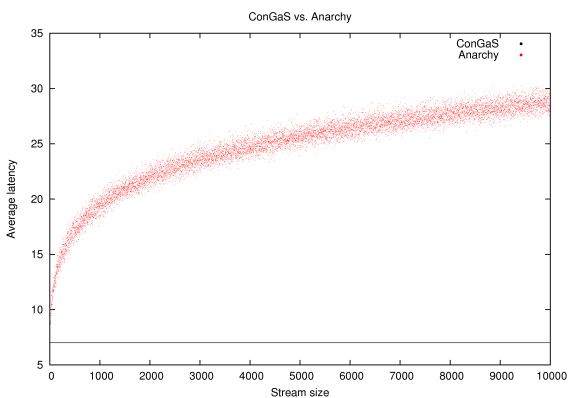


Figure 2: Average latency: number of peers= 2^8 , varying stream size.

Now, we turn to *latency*: the real quantity Λ such that, by picking at random both a peer and a unit c^t , the former receives the latter (on average) in round $t + \Lambda$. Note that, in this reasoning the “bad” outcomes are nevertheless equilibrium ones, and therefore anarchy is rather rational. Put it differently, although ConGaS performs substantially better than such a term of comparison (for any number of content units and peers, see below), still one may like to consider that such a term of comparison already performs rather good as it corresponds to an equilibrium. In particular, such an “*equilibrium anarchy*” performs even much better than typical gossiping protocols and other unstructured mechanisms.

Figure 1 shows the average latency, measured in rounds, experienced by peers in receiving any content unit. In this experiment, we vary both the number of peers (reported here in \log_2 scale) and the number of content units or *size* of the stream $T + 1$. As predicted by our analytical results, ConGaS obtains very stable results in terms of latency when such a size varies (Figure 1, black lines). In other words, the length of the stream has no evident impact on performance. Conversely, in the anarchical environment (Figure 1, light

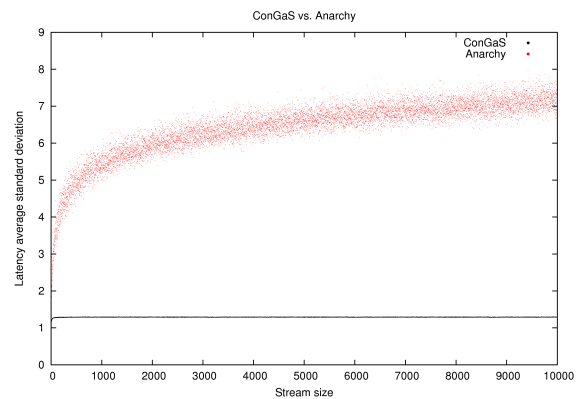


Figure 3: Latency average standard deviation: number of peers= 2^8 , varying stream size.

gray points), the higher the number of content units, the higher the experienced average latency. In addition, such gap also grows with the number of peers.

Figure 2 shows the behavior of the two compared approaches with a fixed number 2^8 of peers and increasing the size (of the streaming). Again, the average latency (over all peers and content units) obtained by ConGaS is independent of the size and optimal given the assumptions that only one unit can be sent/received by each peer in every round. Conversely, anarchy results deteriorate substantially as the size increases.

Another index we finally consider has to deal with fairness, in that we compute the variance across peers of the average latency (over all units) they experience during the whole streaming. In fact, Figure 3 displays the (average) standard deviation of latency, which is much more limited with ConGaS rather than in the anarchy equilibrium outcomes. We claim this is an important result for the implementation of viable live streaming systems, as it corresponds to a jitter reduction, which is a main requirement in most multimedia systems.

10. CONCLUDING REMARKS

A main novelty in this paper is the modeling of P2P real-time streaming by means of multistage congestion game forms: in each stage peers aim at receiving some additional content unit, until all units got received. As each peer receives only one unit, at most, per stage, and also forwards precisely one (at most and randomly selected) unit, our setting is crucially constrained in terms of streaming length (or duration), intended as the number of stages needed to spread the whole content over the whole population. Content distribution over peers changes in time, and for any such a distribution (or game tree node) prevailing at any stage, the associated peers’ per stage payoffs depend on how many requests to forward are addressed to each peer. The idea is that an underlying and unknown probabilistic model handles all those cases where two or more peers ask to receive from a common other peer or from the source: however these cases are handled, the greater the number of peers who all ask to receive from a common other peer or from the source (i.e. the higher congestion), the less likely it becomes, for each of them, to be precisely the one who actually receives. For a

simple (monotone) payoff specification, an exact potential is provided, any of whose maximizers is a pure-strategy equilibrium. In particular, strategies become sequences of singletons: for a peer a strategy specifies, for any (per stage) content distribution over the whole population, who to send a request for additional content. With these singleton strategies, the potential is strong: any of its maximizers is a strong (and Pareto-optimal) equilibrium. Note, once again, that assuming peers and the source to satisfy, at each stage, precisely one of the received requests to forward entails that in our model free-riding is not a feasible strategy. As for each content unit our model builds a new substreaming tree specifying how the unit gets routed through the whole population via P2P exchanges, any reputation-based mechanism that makes free-riding a dominated strategy in tree-based multicast systems is applicable.

Upon these premises, our analysis develops from a fundamental result: if the number $n = 2^m$ of peers is a power of 2, then P2P content exchange can be scheduled so that each unit reaches everybody in $m + 1$ (consecutive) stages, with corresponding whole duration $T + m + 1$. Thus, for any (monotone) payoff specification, this scheduling is socially optimal, allowing to spread each unit in the minimum conceivable number of stages. Given this finding, we first investigate worst-case equilibrium outcomes, and next provide a strategy restriction mechanism which, for any (per stage) content distribution over peers, restricts the set of feasible forwarders. Although simple, this mechanism is very useful: it impedes the satisfaction of all those requests submitted by selfish peers at equilibrium which must be ignored to achieve minimal duration. Hence, the streaming length at any (constrained) equilibrium is actually at its minimum $T + m + 1$, as well as congestion is minimized.

Finally, equilibrium constrained outcomes are compared, through simulations, with non-constrained ones. This quantifies the price of anarchy, although this latter traditionally refers (more strictly) to games rather than (more generally) to game forms. Instead of plotting streaming length, our figures report on average latency and standard deviation, which seem suitable to evaluate both overall performance and fairness in comparison with other P2P protocols.

We conclude by outlining our planned future developments: the model shall be extended so to allow for *i*) a generic number of peers, and *ii*) a varying number of peers. As for *i*), if the number n of peers is not a power of 2, then one may distribute the content by partitioning the population into blocks (each of which has cardinality equal to some power of 2), with one distinct source serving each block separately. In other terms, blocks' cardinalities would be $n_1 = 2^{\lfloor \log(n) \rfloor}$, $n_2 = 2^{\lfloor \log(n-n_1) \rfloor}$, \dots , $n_k = 2^{\lfloor \log(n-n_1-\dots-n_{k-1}) \rfloor}$. In the worst case, the number n of peers is $n = 2^0 + 2^1 + \dots + 2^h$ for some natural h , that is, $n = 2^{h+1} - 1$, which would require the source to multiply by $h + 1$ its activity for serving $h + 1$ different blocks. Streaming length would be $T + \lfloor \log(n) \rfloor + 1$. If this was too demanding, another option is for the source to serve only the block with cardinality n_1 . In fact, for the functioning of this n_1 -cardinal block there is a peer who is never required to forward, but receives each unit with latency $\lfloor \log(n) \rfloor$. This peer may well act as the source for the (next) block with cardinality n_2 , and so on, so that latency adds through blocks of (decreasing) cardinalities $n_1 \leq n_2 \leq \dots \leq n_k$. Turning to *ii*), allowing peers to leave the system as well as new ones to join it, at any time,

clearly has great impact on ConGaS design. From a system design perspective, this could be dealt with by keeping the peer population constantly partitioned into groups. Then, ConGaS would act on such groups, while an additional dissemination protocol would govern intra-group exchanges.

11. REFERENCES

- [1] M. Afegan and R. Sami. Using repeated games to design incentive-based routing systems. In *Proc. of INFOCOM06*, 2006.
- [2] M. Charikar, H. Karloff, C. Mathieu, J. S. Naor, and M. Saks. Online multicast with egalitarian cost sharing. In *Proc. of 20th symposium on Parallelism in algorithms and architectures*, pages 70–76, 2008.
- [3] C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. S. Naor, and A. Orda. Non-cooperative multicast and facility location games. *IEEE Journal on Selected Areas in Communications*, 25(6):1193–1206, 2007.
- [4] Y. R. Choe, D. L. Schuff, J. M. Dyaberi, and V. S. Pai. Improving VoD server efficiency with bittorrent. In *Proc. of 15th Conf. on Multimedia*, 2007.
- [5] Y. Chu, S. G. Rao, and H. Zhang. A case for end system multicast. In *Proc. of SIGMETRICS'00*.
- [6] A. Epstein, M. Feldman, and Y. Mansour. Strong equilibrium in cost sharing connection games. *Games and Economic Behavior*, 2008.
- [7] R. Holzman and N. Law-Yone. Strong equilibrium in congestion games. *Games and Economic Behavior*, pages 85–101, 1997.
- [8] I. Keidar, R. Melamed, and A. Orda. Equicast: scalable multicast with selfish users. In *Proc. of PODC '06*, pages 63–71. ACM, 2006.
- [9] H. Konishi, M. Le Breton, and S. Weber. Equilibrium in a model with partial rivalry. *Journal of Economic Theory*, 72:225–237, 1997.
- [10] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. Bar gossip. In *Proc. of OSDI '06*, pages 191–204, 2006.
- [11] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [12] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.
- [13] T. Moscibroda, S. Schmid, and R. Wattenhofer. On the topologies formed by selfish peers. In *Proc. of PODC '06*, pages 133–142, 2006.
- [14] T. Ngan, D. S. Wallach, and P. Druschel. Incentives-compatible peer-to-peer multicast. In *Proc. of 2nd Workshop on Peer-to-Peer Systems*, 2004.
- [15] M. Quant, P. Borm, and H. Reijnierse. Congestion network problems and related games. *European Journal of Operational Research*, 172:919–930, 2006.
- [16] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, pages 65–67, 1973.
- [17] G. Rossi, S. Ferretti, and G. D'Angelo. Equilibrium selection via strategy restriction in multi-stage congestion games for real-time streaming. Technical Report UBLCS-2009-11, University of Bologna, 2009. www.cs.unibo.it/pub/TR/UBLCS/2009/2009-11.pdf.
- [18] O. Rozenfeld and M. Tennenholtz. Routing mediators. In *Proc. of IJCAI'07*, 2007.