World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:7, No:9, 2013

Comparison of Router Intelligent and Cooperative Host Intelligent Algorithms in a Continuous Model of Fixed Telecommunication Networks

Dávid Csercsik, Sándor Imre

Abstract—The performance of state of the art worldwide telecommunication networks strongly depends on the efficiency of the applied routing mechanism. Game theoretical approaches to this problem offer new solutions. In this paper a new continuous network routing model is defined to describe data transfer in fixed telecommunication networks of multiple hosts. The nodes of the network correspond to routers whose latency is assumed to be traffic dependent. We propose that the whole traffic of the network can be decomposed to a finite number of tasks, which belong to various hosts. To describe the different latency-sensitivity, utility functions are defined for each task. The model is used to compare router and host intelligent types of routing methods, corresponding to various data transfer protocols. We analyze host intelligent routing as a transferable utility cooperative game with externalities. The main aim of the paper is to provide a framework in which the efficiency of various routing algorithms can be compared and the transferable utility game arising in the cooperative case can be analyzed.

Keywords—Routing, Telecommunication networks, Performance evaluation, Cooperative game theory, Partition function form games

I. Introduction

Routing is the most important networking layer function in worldwide packed-switched telecommunication networks [1]. It guaranties that packets containing payload information are driven from a source node to a destination node. Although routing methods face various and sometimes different challenges in fixed and wireless networks there are several measures which allows evaluating them. These are: *reliability*, *stability (scalability)*, *fairness*, *optimality* and *complexity*. *Reliable* routing protocols are able to manage changes in the network topology (independently whether they emerge as a consequence of errors or they are intentional reconfigurations) while *scalable* protocols can handle changes in the number of customers or equivalently in the volume of traffic load. *Fairness* supports network management that users expecting different Quality of Service (QoS) parameters will be satisfied. For example, more delay-sensitive traffic can be directed over shorter routes. *Optimality* is responsible for using the network resources efficiently. This is often contradictory to complexity because typically more sophisticated algorithms are needed to achieve better efficiency.

Routing algorithms can be categorized according to several routing strategies. Non-adaptive (static) routing algorithms do not take into account the status of the network, the route is calculated in advance (e.g. PSTN). Adaptive (dynamic) solutions adjust the routes in compliance with the network topology and status (e.g. ATM). Alternate and multi-path routing methods use more than one route between the source and destination nodes. In the former case the best route is selected according to the decision rule while methods belonging to the latter family exploit more than one one route at the same time to deliver packets of a certain connection (e.g. IP).

The wired and wireless environment lay down different conditions to routing protocols. First of all wireless links are less reliable than fixed ones because of the interference and fading effects. Nowadays the amount of customers who access networks over wireless links is continuously increasing (WIFI, WIMAX, HSxPA, LTE, etc.). Moreover, smart metering and ad hoc networks in home automation or assisted living become more and more popular. These type of fully or almost fully wireless systems consist of nodes which are connected to one another via wireless links. Routing in such networks is either table-driven or source-driven. Table-driven networks periodically update the routing tables in the nodes introducing huge traffic overhead and in case of large networks certain inconsistency in exchange for fast payload packet delivery. Source-driven solutions first discover routes between the source and destination nodes and maintain them during the lifetime of the connection. Here the overhead appears in the form of delay which might be fruitful if the network is not stable. In practice hybrid solutions fitted to the application scenario are used. Nature inspired routing algorithms for fixed telecommunication networks are reviewed in [2].

While the literature of game theoretic approach of networking telecommunication problems is quite rich [3], cooperative approaches usually use the Nash Bargaining concept [4], and focus on bandwidth sharing [5], [6], [7], [8] or traffic control [9], [10] in this context. The paper [11] considers Stackelberg equilibria in the case of flow control as well. Network formation problems have been studied in [12], [13], [14], while coalitional approaches in telecommunication and wireless networks including cognitive radio network applications are addressed in the papers [15], [16], [17], [18], [19]

In this article it is assumed that the whole traffic of the analyzed network is originating from a finite number of hosts. In other word, any component of the traffic can be identified as a part of delivery (or routing) task of one of the distinguished hosts. This assumption can be considered as neglecting any other traffic in the network, which can be relevant in the case

D. Csercsik is with the Faculty of Information Technology Pázmány Péter Catholic University Budapest, Hungary H-1444 csercsik@itk.ppke.hu.

S. Imre is with the Department of Networked Systems and Services Budapest University of Technology and Economics Budapest, Hungary H-1521 imre@hit.bme.hu

of e.g. HD video streaming services, or high speed downloads requiring high bandwidth rates compared to which other traffic of the network can be considered insignificant. In addition this assumption can be relaxed and as we will see the proposed model is able to take into account traffic as well which can not be influenced by the distinguished hosts.

Furthermore, we assume that the data which has to be sent through the network can be divided to infintesimal parts, therefore we use a continuous model. We will use the term *packet* as a synonym for such an arbitrary small part of a flow corresponding to a certain routing task.

To distinguish between more and less delay critical delivery tasks according to quality of service (QoS) considerations, utility functions are assigned to each delivery task. The utility functions assign a utility value to the total latency cost of a certain routing task. The most simple interpretation of the utility functions is that they describe the willingness of the clients to pay for a certain QoS regarding the corresponding routing task to the hosts. Since cooperating players (hosts) may easily redistribute the wealth gained such way between themselves, the transferable utility assumption is straightforwardly validated in the cooperative case. This implies that in this case it makes sense to analyze the stability of the resulting cooperative game regarding the various payoffs.

We analyze two routing approaches. The *router intelligent* algorithm assumes an IP like routing, where packets hold information only about their destination, and the determination of flows is performed by the routers. In this case a simple adaptive congestion management algorithm is carried out by the routers to optimize the network loads.

The *host intelligent* algorithm assumes that hosts determine the explicit route for their data deliveries in order to maximize their overall utility. In the case of host intelligent routing, various predictive routing strategies and levels of cooperation among hosts are analyzed. While the router intelligent case can be analyzed as a distributed optimization problem, in which routers are about to distribute their inbound traffic in order to meet the latency constraints defined on their outbound links, the host intelligent method case can be described as a transferable utility cooperative game. Since in this game cooperation may also affect outsiders not taking part in a coalition, externalities may arise, thus the game can be described in partition function form.

II. Materials and methods

Definition 1 *A network routing problem $\mathcal{D} = (N, \Gamma, \Delta, U)$ is a 4-tuple consisting of a player set N, a network Γ, a set of delivery tasks Δ, and a set of utility functions U. A network Γ is a two-tuple (G, l), represented by a directed graph $G(V, E)$, and a set of edge latency functions $l = \{l_e | e \in E\}$. A delivery task $\delta = (r, s, t) \in R_+ \times V \times V$ is described by a quantity and two nodes (a source and sink respectively). To each player $j \in N$ k^j delivery tasks and utility functions are assigned $\delta^j = \cup_{i=1}^{k^j} \delta^{j,i}$, $\delta^{j,i} = \{(r^{j,i}, s^{j,i}, t^{j,i})\}$, $u^j = \cup_{i=1}^{k^j} \{u^{j,i}\}$.*

We denote the flow of player j corresponding to task i on edge (or link) e with $f_e^{j,i}$, the total flow on edge e with f_e and the set of flows with $\mathcal{F}$. We say that $\mathcal{F}$ satisfies the network routing problem D if $\mathcal{F} \in R^{|E|}$ can be decomposed to N disjoint f^j player flows, each of which can be decomposed into k^j disjoint $f^{j,i}$ task flows corresponding to the delivery tasks of player j. The task flow $f^{j,i}$ is composed of a finite number of flows on partially disjoint directed paths connecting $s^{j,i}$ to $t^{j,i}$ and their overall value is $r^{j,i}$. An algorithm assigning a set of flows to Γ satisfying the network routing problem will be called a *routing method*. The routing method σ will describe whether host intelligent or router intelligent algorithm is applied, furthermore in the case of host intelligent algorithms, it will describe the level of cooperation and the routing strategy applied by the hosts.

It is commonly accepted to make some constrains on the latency functions, such as non-negativity, differentiability and non-decreasingness. The resulting latency cost corresponding to task i of player j can be calculated as

$$c^{j,i} = \sum_{e \in E} l_e(\mathcal{F}) \cdot f_e^{j,i} \tag{1}$$

where $f_e^{j,i}$ denotes the flow corresponding to task i of player j on edge e. The flow of a certain edge can be calculated as

$$f_e = \sum_{j=1}^{N} \sum_{i=1}^{k_j} f_e^{j,i}$$

The resulting utility value of player j denoted by $u(j)$ can be calculated as

$$u(j) = \sum_{i=1}^{k_j} u^{j,i}(c^{j,i}) \tag{2}$$

where $u^{j,i}$ is the non-increasing utility function of the task $\delta^{j,i}$.

If we would like to have some measure about the 'physical' efficiency of the network, independent of the utility functions we may want to use the total latency cost of the network, which is the sum of the total latency cost of the players, which in turn can be calculated as the sum of the task latency costs defined by equation 1.

$$C^{TN} = \sum_{j=1}^{N} \sum_{i=1}^{k_j} c^{j,i} \tag{3}$$

In reality, in the case of hard wired networks, packets suffer significant latency only at the routers. In the proposed model however, this latency is allocated on the edges. This assumption can be considered as inbound registers realizing the node-latency. The edge-latency in the proposed model, in contrast to traditional routing models [20], [21], [22], does not only depend on the traffic on the actual edge but also on the traffic on the other inward edges of the endnote of the actual edge (which represents the next router in the path). From this point of view the proposed model can be regarded as a generalization of the one presented in [20]. Furthermore, we will assume, that the edge latency is equal on each inward edge of a node, namely the latency function of the corresponding node. We will denote the latency function of node k by L_k where $i \in V$. The latency functions of the edges can be defined as: $l_e = L_k$ iff the edge e is an inbound edge of node k. The network model used in this article does not deal with explicit capacity constraints on links, however latency functions may be defined in a way which assigns extreme high values to traffic levels above a certain threshold.

A. Router intelligent method

In the case of the router intelligent method, we will assume, that every packet holds only information about its destination node. Furthermore we assume that each node (router) in the network has its own lookup table, which determines for every unit of inbound traffic (from a certain host corresponding to a certain delivery task) that how much of it will be forwarded on a certain outward edge. In other words, multi-path routing is assumed since the routers may distribute the incoming traffic of a delivery task on the outgoing paths. $w_e^{v,t}$ denotes the ratio of the traffic forwarded from node v on edge e if the destination is t.

$$\sum_{e \in E_{out}^{v,t}} w_e^{v,t} = 1$$

where $E_{out}^{v,t}$ denotes the set of outward edges of node v on which the destination t is reachable. Clearly, the progression of this algorithm on the nodes until each flow reaches its destination is a routing method.

A simple approach regarding the determination of the lookup tables of the nodes is the *shortest path* method. In this case every packet is forwarded on the shortest path to its destination. Formally, $w_e^{v,d} > 0$ only for the edges belonging to the shortest paths. In contrast to [12], where the shortest path algorithm routes only on one edge, defined by a lexicographic ordering, in our case if the shortest path is not unique the traffic will be divided equally amongst the shortest paths. Since shortest path routing does not take any information about the latency levels into account, it may easily result in congestion and high latency values.

Congestion management in the case of the router intelligent method will be carried out via the iterative modification of the lookup tables based on local congestion management approach. The initial routing configuration is determined by the shortest path approach, which is followed by the iterative application of the following method. We assume that every router detects the latency on its outward edges, and if this latency exceeds a certain value ($\bar{l}_i$) on edge f, it decreases the corresponding $w_f^{v,d}$ with a predefined η, and increases the $w_e^{v,d}$ parameters for all other $e \in E_{out}^{v,d}$ by $\eta/(|E_{out}^{v,d}|-1)$. It is possible that even decreasing the traffic on a certain edge to 0 is not enough to reach the predefined edge latency (since the edge latency does depend also on other traffic factors which can not be affect by the actual router). As a simplest approach in this paper we assume that edge latencies are defined in such a way that this does not happen.

The simplest approach is to assume that $\bar{l}_i = \bar{l} \ \ \forall i$, in other words to assume a universal maximal edge delay.

In this article we assume that in the case of the given networks and delivery tasks this algorithm always stops (which means that the latency constraints are feasible).

B. Host intelligent method

In the case of the host intelligent method, we will assume that every packet holds the information describing its complete route. Furthermore it is assumed that every host is aware of the network structure and of the routers characteristics. Hosts may behave competitively or cooperate regarding the determination of their routing paths. We will call the cooperating disjoint subsets of players (hosts) coalitions. The key assumption is that as a coalition of hosts cooperate, they design the routing for their deliveries to maximize their *overall* utility. As the latencies of the network do depend on the traffic, as the resulting traffic changes, changes in latency also affect other players, not taking part in the cooperation implying *externalities*. This type of game can be described by the partition function form [23]. The partition function is explicitly defined via the utility functions. In the host intelligent case, and the arising transferable utility (TU) game is analyzed, while the utilities resulting from the router intelligent case will serve as reference values.

1) Routing strategies: Players and coalitions may route their delivery according to different possible strategies. These are shortly described below, and demonstrated in section III. The expression 'routing strategy' is interpreted in a wide sense, including information and beliefs about other players. The zero order strategy assumes that the players have no information about each other while in other cases the delivery tasks are common knowledge. The strategies presented here are pure in the sense that players may route their deliveries in several different paths in the same time but they do it with probability 1.

Let us introduce some notations.The load of edge e with respect to agent j is the traffic that goes through the edge not counting f_e^j. We denote this by λ_e^j, formally $\lambda_e^j = \sum_{k \neq j} f_e^k = f_e - f_e^j$. Similarly $\lambda_e^S = \sum_{k \notin S} f_e^k = f_e - f_e^S$. The expected load of edge e with respect to coalition S is the flow that goes through e not counting f_e^S according to the current knowledge of S (which depends on the coalition structure and σ). We denote this by $\widehat{\lambda}_e^S$.

Zero order strategy: This "dummy" strategy assumes that all coalitions neglect the activity of others, and route their deliveries in a way, which is optimal when no other traffic appears on the network. This strategy assumes that non-cooperating players/coalitions have no information of each others routing tasks. In other words $\widehat{\lambda}_e^S = 0$ for each edge $e \in E$ and for each coalition $S \subseteq N$.

First order predictive (FOPS) and n-th order predictive (nOPS) strategy: We define the first order predictive strategy as follows. Every coalition expects the remaining coalitions to route their deliveries according to the zero order strategy, and minimizes his routing costs according to this. This strategy assumes that the coalitions are aware of the other participants delivery contracts.

Let us denote the resulting flow of edge e in the zero order routing by $f_e(\sigma_0)$. In this case $\widehat{\lambda}_e^S = f_e(\sigma_0) - f_e^S(\sigma_0)$. In the second order predictive strategy all coalitions assume that the remaining ones will route their delivery according to the FOPS etc.

Routing under Nash-equilibrium: We will call a routing configuration a Nash-equilibrium (NE), if no player is able to improve his routing, assuming that other players do not change theirs. Formally, let A be an algorithm that computes a NE for a given routing problem $\mathcal{D}$. Furthermore let $\sigma(A)$ be the routing strategy that routes the delivery tasks as in the NE

World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:7, No:9, 2013

computed by A. Then $\mathcal{D}(N, \Gamma, \Delta, \sigma(A))$ is a delivery game. Note that the strategy of S is naturally equivalent to the set of flows of S, namely f^S.

2) Basic properties of predictive strategies: The predictive technique is an elemental way to strategically approach a game theoretical problem. The most difficult part is to guess the depth of reasoning of the other players. A fair assumption is that the players think that they go at least one step further than the others. Here we only analyze the case when the depth of reasoning is the same for all players and coalitions, and every actor thinks that the other players take one step less in the reasoning process. It is easy to see that if the n-th and the n+1-th order predictive strategies coincide, it means that the resulting routing is a Nash-equilibrium.

III. Results

A. The example routing problem

In this section we demonstrate the results of different routing methods on a simple example network (Network 1 is depicted in Fig. 1).

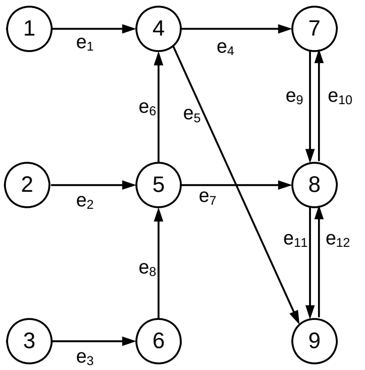

Fig. 1. Topology of network 1 and the indexing of nodes and edges.

Let us define the following routing tasks. $\delta^{11} = (3, 1, 8)$, $\delta^{21} = (2, 2, 9)$, $\delta^{22} = (3, 2, 9)$ and $\delta^{31} = (3, 3, 7)$. The latency functions of the nodes are described by equations 4.

$$\begin{aligned} L_4 &= 0.2 + 0.3(f_1 + f_6) \\ L_5 &= 1.5 + 0.2(f_2 + f_8) \\ L_6 &= 0.5 + 0.2 f_3 \\ L_7 &= 0.1 + 0.4(f_4 + f_{10}) \\ L_8 &= 0.5(f_7 + f_9 + f_{12}) \\ L_9 &= 0.4 + 0.4(f_5 + f_{11}) \end{aligned} \tag{4}$$

Let us note that the flows in the linear expressions of (4) are interchangeable in the case of this example. This refers to the assumption that the delay of a router depends only on the total amount of the incoming traffic, and not on its distribution on the incoming lines.

In this article we consider piecewise linear utility functions, which are monotone decreasing with the latency cost of the actual delivery task.

$$\begin{aligned} u^{1,1} &= m_1^{1,1} c^{1,1} + a_1^{1,1} \\ u^{2,1} &= m_1^{2,1} c^{2,1} + a_1^{2,1} \\ u^{2,2} &= m_1^{2,2} c^{2,2} + a_1^{2,2} \quad if \;\; c^{2,2} < t^{2,2} \\ u^{2,2} &= m_2^{2,2} c^{2,2} + a_2^{2,2} \quad if \;\; c^{2,2} \geq t^{2,2} \\ u^{3,1} &= m_1^{3,1} c^{3,1} + a_1^{3,1} \end{aligned} \tag{5}$$

where $u^{j,i}$ denotes the utility of player j corresponding to task i as the function of $c^{j,i}$ ($m_k^{j,i} < 0 \;\; \forall i, j, k$). As defined in 2, the total utility of the players is simply the sum of the utility values regarding the various tasks. The parameters are as follows.

$$\begin{aligned} &m_1^{1,1} = -0.2 \quad a_1^{1,1} = 6 \quad m_1^{2,1} = -0.25 \quad a_1^{2,1} = 5 \\ &m_1^{2,2} = -0.25 \quad a_1^{2,2} = 8.375 \quad m_2^{2,2} = -0.75 \\ &a_2^{2,2} = 21.125 \quad m_1^{3,1} = -0.1 \quad a_1^{3,1} = 1 \quad t^{2,2} = 25.5 \end{aligned}$$

In reality, the QoS requirements often result in threshold type utility functions (eg. the client only pays for a certain service, if the delay is below a predefined limit). These step functions can also be approximated by piecewise linear functions, which are more beneficial regarding optimization procedures.

B. Router intelligent method

The initial flows and edge delays defined by the shortest path approach in the case of network 1 are depicted in Fig. 2.

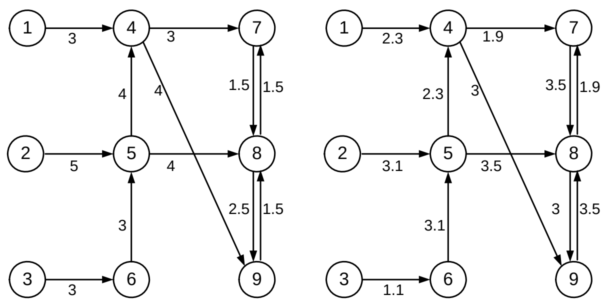

Fig. 2. Initial flows and edge delays of network 1 assuming router intelligent method.

First, let us recall that since the routers have no information about the utility functions, the router intelligent algorithm is modifying the network flows only in order to decrease the load on the edges on which the latency exceeds the values predefined in $\bar{l}_e$. However, as we will see, the appropriate choice of $\bar{l}_e$ may significantly reduce the players overall latency cost.

We will examine three cases in the router intelligent method. In the first two cases, a universal $\bar{l}$ is assumed for all edges, while in the third case, $\bar{l}_e$ is different for each $e \in E$. If we use an universal $\bar{l}$, the lowest value at which the local adaptive congestion management algorithm described in section III-B assuming $\eta = 0.01$ converges is $\bar{l} = 3.1$.

- In the first case we assume a slightly conservative guess of the optimal value, and use $\bar{l} = 1.05 \bar{l}_{opt}^u = 3.35$. To demonstrate the local adaptive congestion management algorithm, Table I and II shows the evolution of flows, edge delays and the resulting latency costs of the players respectively. The overall social cost (C^{TN}) is 96 at the end of the iteration. If we calculate the utility values for the players, we get $[U(1) \quad U(2) \quad U(3)] = [1.09 \quad 1.57 \quad 1.32]$

- If we consider $\bar{l} = 3.1$, we get the results $C^{TN} = 95.84$ and the utility values $[1.1 \quad 1.61 \quad 1.33]$.
- We may consider the optimization problem in which we are looking for the vector $\bar{l_e}$ which minimizes C^{TN}, under the constraint that the local congestion management algorithm should converge. The result for this C^{TN}-optimal $\bar{l_e}$ is

$$[4.46 \ 3.58 \ 2.06 \ 2.21 \ 4.59 \ 3.22 \ 2.72 \ 3.40 \ 3.2 \ 3.51 \ 4.75 \ 3.88]$$

which results in $C^{TN} = 94.64$. The resulting utilities of the players in this case are 1.24, 1.77 and 1.34 respectively.

Iteration	f_1	f_2	f_3	f_4	f_5	f_6
0	3	5	3	3	4	4
1	3	5	3	3.03	4.05	4.08
2	3	5	3	3.06	4.1	4.16
3	3	5	3	3.09	4.15	4.24
4	3	5	3	3.12	4.2	4.32

Iteration	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
0	4	3	1.5	1.5	2.5	1.5
1	3.92	3	1.5	1.47	2.45	1.5
2	3.84	3	1.5	1.44	2.4	1.5
3	3.76	3	1.5	1.41	2.35	1.5
4	3.68	3	1.5	1.38	2.3	1.5

Iteration	l_1	l_2	l_3	l_4	l_5	l_6
0	2.3	3.1	1.1	1.9	3	2.3
1	2.32	3.1	1.1	1.9	3	2.32
2	2.35	3.1	1.1	1.9	3	2.35
3	2.37	3.1	1.1	1.9	3	2.37
4	2.4	3.1	1.1	1.9	3	2.4

Iteration	l_7	l_8	l_9	l_{10}	l_{11}	l_{12}
0	3.5	3.1	3.5	1.9	3	3.5
1	3.46	3.1	3.46	1.9	3	3.46
2	3.42	3.1	3.42	1.9	3	3.42
3	3.38	3.1	3.38	1.9	3	3.38
4	3.34	3.1	3.34	1.9	3	3.34

TABLE I
EVOLUTION OF FLOWS AND EDGE LATENCIES IN NETWORK 1 DURING THE ITERATIONS OF THE LOCAL ADAPTIVE CONGESTION MANAGEMENT ALGORITHM IN THE CASE OF ROUTER INTELLIGENT ALGORITHM AND $\bar{l} = 3.35$.

Iteration	$c^{1,1}$	$c^{2,1}$	$c^{2,2}$	$c^{3,1}$	C^{TN}
1	24.7	17.96	26.94	26.94	96.55
2	24.65	17.93	26.89	26.89	96.35
3	24.61	17.89	26.84	26.84	96.17
4	24.56	17.86	26.79	26.79	96

TABLE II
EVOLUTION OF TASK LATENCY COSTS IN THE CASE OF NETWORK 1 DURING THE ITERATIONS OF THE LOCAL ADAPTIVE CONGESTION MANAGEMENT ALGORITHM IN THE CASE OF ROUTER INTELLIGENT ALGORITHM AND $\bar{l} = 3.35$.

If we compare the reference cases of the router intelligent method, we can see that although the local adaptive congestion management algorithm does not takes the utility functions explicitly into account, as an indirect effect, through the decrease of edge latencies and latency costs, the values of the utility functions are increased. Although this may be considered as a typical case, a counterexample may be easily constructed. For example, since the optimization is carried out to minimize C^{TN}, it is possible that the total latency cost of a certain player increases as C^{TN} is decreased.

C. Host intelligent method

In this case, the hosts are about to maximize their utility assuming different cooperation structures and strategies. Cooperating players maximize the sum of their utilities. To describe the data routing of the hosts, we define the so called *routing variables*. In the case of Network 1, player one, who has only one delivery task, to transfer 3 units from node 1 to node 8, has two alternative paths, namely $P_1^{1,1} = 1-4-9-8$ and $P_2^{1,1} = 1-4-7-8$. $x_1^{1,1}$ will describe the amount of data routed on $P_1^{1,1}$ (the remaining will be routed via P_2^{11}).

Player 2 has two delivery task, in the case of this simple example with the same source and destination. $x_1^{2,1}$ describes the amount of his first task routed on $P_1^{2,1} = 2-5-8-9$, $x_2^{2,1}$ corresponds to the path $P_2^{2,1} = 2-5-4-9$ and the rest is routed via $P_3^{2,1} = 2-5-4-7-8-9$. Similarly, regarding his second task, $x_1^{2,2}$ describes the amount of his first task routed on $P_1^{2,2} = P_1^{2,1}$, $x_2^{2,2}$ corresponds to the path $P_2^{2,2} = P_2^{2,1}$ and the rest is routed via $P_3^{2,2} = P_3^{2,1}$.

Regarding player 3, $x_1^{3,1}$ describes the amount of his first task routed on $P_1^{3,1} = 3-6-5-4-7$, $x_2^{3,1}$ corresponds to the path $P_2^{3,1} = 3-6-5-4-9-8-7$ and the rest is routed via $P_3^{3,1} = 3-6-5-8-7$.

To demonstrate how the routing variables, task latency costs and utilities change while the predictive strategies of increasing converge to a NE, we provide the detailed results in the case of the all singleton partitions.

The evolution of routing variables, task latency costs and the players utility values are summarized in Tables III, IV and V respectively.

	$\{1\},\{2\},\{3\}$						
σ	x_1^{11}	x_1^{21}	x_2^{21}	x_1^{22}	x_2^{22}	x_1^{31}	x_2^{31}
0	1.31	0.75	1.25	1.25	1.75	1.75	0
1	0.81	2	0	0	3	2.19	0
2	0.81	1.94	0.06	0	3	2.19	0
3	0.81	1.94	0.06	0	3	2.16	0
4	0.81	1.91	0.09	0	3	2.16	0
5	0.81	1.91	0.09	0	3	2.14	0
6	0.81	1.89	0.11	0	3	2.14	0
7	0.81	1.89	0.11	0	3	2.13	0
8	0.81	1.88	0.12	0	3	2.13	0
9	0.81	1.88	0.12	0	3	2.13	0
10	0.81	1.88	0.12	0	3	2.13	0

TABLE III
THE EVOLUTION OF ROUTING VARIABLES OF NETWORK 1 TOWARDS NEs IN THE CASE OF THE HOST INTELLIGENT METHOD AS THE ORDER OF STRATEGIES σ IS INCREASED IN THE CASE OF PARTITION $\{1\},\{2\},\{3\}$

If we compare Table V to the utility results of the router intelligent case, we can see that the application of predictive strategies implies higher utilities compared even to the best router intelligent case. This coincides with our expectations, since in the router intelligent case the delivery routes are

	$\{1\},\{2\},\{3\}$				
σ	$c^{1,1}$	$c^{2,1}$	$c^{2,2}$	$c^{3,1}$	C^{NT}
0	24.12	17.55	26.4	26.85	94.92
1	23.66	17.46	25.44	27.3	93.86
2	23.62	17.39	25.5	27.31	93.82
3	23.64	17.42	25.47	27.31	93.84
4	23.62	17.38	25.5	27.32	93.82
5	23.63	17.4	25.49	27.32	93.83
6	23.62	17.38	25.5	27.32	93.82
7	23.63	17.39	25.49	27.32	93.83
8	23.62	17.38	25.5	27.32	93.82
9	23.62	17.38	25.5	27.32	93.82
10	23.62	17.38	25.5	27.32	93.82

TABLE IV
THE EVOLUTION OF TASK LATENCY COSTS OF NETWORK 1 TOWARDS NES IN THE CASE OF THE HOST INTELLIGENT METHOD AS THE ORDER OF STRATEGIES σ IS INCREASED IN THE CASE OF PARTITION $\{1\},\{2\},\{3\}$.

	$\{1\},\{2\},\{3\}$		
σ	$u(1)$	$u(2)$	$u(3)$
0	1.18	1.95	1.32
1	1.27	2.65	1.27
2	1.28	2.65	1.27
3	1.27	2.65	1.27
4	1.28	2.65	1.27
5	1.27	2.65	1.27
6	1.28	2.66	1.27
7	1.27	2.66	1.27
8	1.28	2.66	1.27
9	1.28	2.66	1.27
10	1.28	2.66	1.27

TABLE V
THE EVOLUTION OF THE PLAYERS UTILITY VALUES IN THE CASE OF NETWORK 1, ALL-SINGLETON COALITIONS AND HOST INTELLIGENT METHOD, AS THE ORDER OF PREDICTIVE STRATEGIES (σ) IS INCREASED.

carried out based on information of the global structure and characteristics of the network explicitly optimizing the utility functions, while the adaptive congestion management method of the router intelligent case uses only local information and does not consider the form of the utility functions. The resulting utilities in the case of other coalition structures can be found in Appendix A. In the case of total cooperation (the grand coalition in the terminology of game theory) the utilities of the players will be as follows $[u(1) \;\; u(2) \;\; u(3)] = [1.14 \;\; 3.25 \;\; 1.09 \;\;]$. The total latency cost is $C^{TN} = 93.6$ in this case. In the following we consider the Nash routing in the case of the various coalitional structures, and analyze the stability properties of the arising transferable utility PFF game.

D. Stability

Before analyzing the stability properties of the arising game, let us recall some basic notions of cooperative games [23]. A *cooperative game with transferable utility* or simply a *TU-game* is an ordered pair (N, v) consisting of the player set $N = \{1, 2, \ldots, n\}$ and a characteristic function $v : 2^N \to \mathbb{R}$ with $v(\emptyset) = 0$. The value $v(S)$ is regarded as the worth of coalition S. The members of S can achieve this value by cooperating regardless of how players outside the coalition react. In a partition function form (PFF) game $v(S)$ depends also on the partition to where S belongs [23]. Formally a partition function form game is a pair (N, V) where $V : \pi \to (2^N \to \mathbb{R})$ is the partition function which assigns characteristic functions (v) to each partition $\pi \in \Pi(N)$ (where $\Pi(N)$ denotes the set of partitions of N). For $S \in \pi$, the worth of $V(S, \pi)$ denotes the amount that the players in S can guarantee themselves by cooperating, when the coalition S is embedded in the partition π. We call the pair $\omega = (y, \pi)$ an outcome, where $\pi \in \Pi(N)$ is a partition and $y = (y^1, \ldots, y^n) \in \mathbb{R}^N$ is a payoff vector satisfying feasibility; $\sum_{i \in (S \in \pi)} y^i \leq V(S, \pi)$ for all $S \in \pi$. Let us denote the the set of outcomes in (N, V) by $\Omega(N, V)$.

Analyzing the stability properties of a PFF TU game means the determination of the set of stable outcomes. The basic assumption regarding stability analysis is that the conditions of individual and coalitional rationality have to hold after the redistribution of coalitional vales in the case of a certain coalition structure (outcome). If any player or coalition is unsatisfied with his payoff, and thinks he can reach a higher value by acting alone, he may deviate from the actual coalition structure.

To analyze stability we use the concept of the *recursive core* [24], [25], that allows the remaining, residual players in a PFF game to freely react and form a core-stable partition before the payoff of the deviating coalition is evaluated.

First we define the *residual game* over the set $R \subsetneq N$. $\Pi(N)$ denotes the set of partitions of N. Assume $\overline{R} = N \setminus R$ have formed $\pi_{\overline{R}} \in \Pi(\overline{R})$. Then the residual game $(R, V_{\pi_{\overline{R}}})$ is the PFF game over the player set R with the partition function given by $V_{\pi_{\overline{R}}}(S, \pi_R) = V(S, \pi_R \cup \pi_{\overline{R}})$.

Definition 2 (Recursive core [24]) *For a single-player game the recursive core is trivially defined. Now assume that the core $RC(N, V)$ has been defined for all games with $|N| < k$ players. For an $|N|$-player game an outcome (y, π) is dominated if there exists a coalition Q forming partition π' and an outcome $(y, \pi' \cup \pi_{\overline{Q}}) \in \Omega(N, V)$, such that $y_Q > y_Q$ and if $RC(\overline{Q}, V_{\pi'}) \neq \varnothing$ then $(y_{\overline{Q}}, \pi_{\overline{Q}}) \in RC(\overline{Q}, V_{\pi'})$. The (recursive)* core *$RC(N, V)$ of (N, V) is the set of undominated outcomes.*

Based on the concept of the Recursive Core, a *minimal claim* function can be defined, which describes the minimal claim of each coalition in the corresponding PFF game reduced to that coalition. This function, termed v^{mc} in the following, may be applied in the same spirit as a characteristic function, since it assigns a unique value to each coalition, which they can secure for themselves if they deviated, assuming that the players of the residual game will form a stable partition. The formal definition of v^{mc} is as follows.

Definition 3 *Let us consider the residual game $(\bar{S}, V_{\pi_S})$ over the player set $\overline{S}$ defined by the partition function $V_S(R, \pi_{\overline{S}}) = V(R, \pi_{\overline{S}} \cup S)$ where $R \in \pi_{\overline{S}} \in \Pi(\overline{S})$. Let us denote the Recursive Core of the residual game by* $\mathrm{RC}(\bar{S}, V_S)$. *The* *(pessimistic) minimal claim function* v^{mc} *can be defined as*

$$v^{mc}(S) = \begin{cases} \min_{\sum_{i\in S} y^i} \{\Omega(N,V) | (y, P^{\overline{S}}) \in RC(\bar{S}, V_S)\} \\ \quad if \;\; RC(\bar{S}, V_S) \neq \emptyset \\ \min_{\sum_{i\in S} y^i} \{\Omega(N,V)\} \\ \quad if \;\; RC(\bar{S}, V_S) = \emptyset \end{cases}$$

where $v^{mc}(S)$ *is the minimal claim of coalition* S.

With the help of the minimal claim function, a characterization of the Recursive Core can be given as follows.

Lemma III.1 *The Recursive Core* $RC(N,V)$ *of the game* (N,V) *is a collection of Pareto efficient outcomes* $(y,\pi) \in \Omega(N,V)$*, such that there is no coalition* S *with* $v^{mc}(S) > \sum_{i\in S} y^i$.

In our case the partition function is defined by the utility functions in a straightforward way. Let us take the NE routing, which is practically equal to $\sigma = 10$. In this case the partition function V will be given by Table VI.

partition (π)	values of coalitions ($v(S)$)
{1},{2},{3}	1.28, 2.66, 1.27
{1,2},{3}	4.26, 1.17
{1,3},{2}	2.36, 0.91
{1},{2,3}	1.27, 3.73
{1,2,3}	5.48

TABLE VI
PARTITON FUNCTION IMPLIED BY THE UTILITY FUNCTIONS IN THE CASE OF NASH ROUTING.

From the partition function given by given by Table VI, we can derive the minimal claim function, described in Table VII

Coalition (§)	($v^{MC}(S)$)
{1}	1.28
{2}	2.66
{3}	1.17
{1,2}	4.26
{1,3}	2.36
{2,3}	3.73
{1,2,3}	5.48

TABLE VII
MINIMAL CLAIM FUNCTION IMPLIED BY THE UTILITY FUNCTIONS IN THE CASE OF NASH ROUTING.

It is trivial that the only partition which may be stable is the grand coalition in our case, since the overall payoff here is the highest. In general this will be true for all games arising from this routing problem formulation, since any routing which appears in the case of any other partition is also feasible in the grand coalition. This shows that the total utility of the grand coalition has to be at least as high as the total utility in any other coalition structure.

The recursive core will be a convex polytope in the payoff space depicted in Fig. 3.

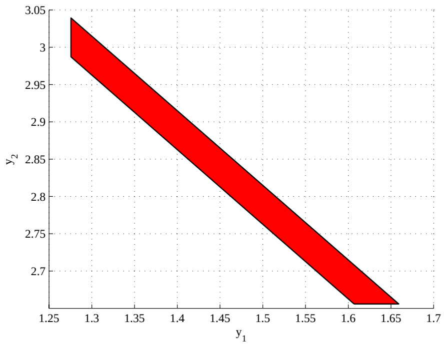

Fig. 3. The recursive core in the case of Nash routing. The payoff of player 3 is $y_3 = 5.48 - y_1 - y_2$

Since the recursive core depends on the network, on the routing tasks and on the utility functions as well, the existence of stable partitions (and so the non-emptiness of the recursive core) is not a necessary property of the arising game. For example, if we change only the utility function of the routing task belonging to player 3 to parameters $m_1^{3,1} = -0.4$; and $a^{3,1} = 16$ there will be no stable partitions in the arising game, considering Nash routing. This phenomenon shows that the transferable utility PFF games may serve as a useful approach when one tries to determine which cooperations can be mutually beneficial for the hosts.

IV. CONCLUSIONS

We have introduced a continuous model for network routing which is capable to compare the efficiency of router intelligent and host intelligent methods in terms of task latencies, total latency cost and utility values.

It was presented that for the simple example network 1 predictive host intelligent algorithms may reduce the total routing cost by 1-2%. However this small improvement in the total latency cost may correspond to a significant increase in the players' utilities (eg. we can experience a 183% increase in the utility of player 2). The utility functions of the various tasks result in the reconfiguration of network flows, which has no serious effect on the total latency costs, but guarantees some kind of higher priority and minimal congestion for the deliveries with high utility.

We have shown how to analyze the stability of the arising transferable utility game in the case of the host intelligent method, and demonstrated that scenarios can be found in which no global mutual satisfaction is possible.

The above results were derived for a specific network structure, where the possibilities of the hosts for the explicit route design were limited. We suppose that the proposed approach will give even more significant results if the network is more complex, and problems arise which are avoidable with intelligent routing design (e.g. bottleneck type shortest path congestions).

APPENDIX A

A. Coalition structure $\{1,2\},\{3\}$

Only that many iterations are included, for which the resulting utility values do change in the used precision.

	$\{1,2\},\{3\}$		
σ	$u(1)$	$u(2)$	$u(3)$
0	1.19	3.07	1.16
1	1.21	3.05	1.17
2	1.20	3.06	1.17
3	1.21	3.06	1.17
4	1.20	3.06	1.17
5	1.20	3.06	1.17

TABLE VIII
THE EVOLUTION OF THE PLAYERS UTILITY VALUES IN THE CASE OF NETWORK 1, PARTITION $\{1,2\},\{3\}$ AND HOST INTELLIGENT METHOD, AS THE ORDER OF PREDICTIVE STRATEGIES (σ) IS INCREASED.

B. *Coalition structure* $\{1,3\},\{2\}$

	$\{1,3\},\{2\}$		
σ	$u(1)$	$u(2)$	$u(3)$
0	0.93	0.78	1.43
1	0.94	0.91	1.42
2	0.94	0.91	1.42

TABLE IX
THE EVOLUTION OF THE PLAYERS UTILITY VALUES IN THE CASE OF NETWORK 1, PARTITION $\{1,3\},\{2\}$ AND HOST INTELLIGENT METHOD, AS THE ORDER OF PREDICTIVE STRATEGIES (σ) IS INCREASED.

C. *Coalition structure* $\{1\},\{2,3\}$

	$\{1\},\{2,3\}$		
σ	$u(1)$	$u(2)$	$u(3)$
0	1.18	1.94	1.3
1	1.27	2.45	1.29
2	1.27	2.45	1.29
3			

TABLE X
THE EVOLUTION OF THE PLAYERS UTILITY VALUES IN THE CASE OF NETWORK 1, PARTITION $\{1\},\{2,3\}$ AND HOST INTELLIGENT METHOD, AS THE ORDER OF PREDICTIVE STRATEGIES (σ) IS INCREASED.

ACKNOWLEDGMENT

This work was supported by the Hungarian National Fund OTKA NF104706.

REFERENCES

[1] S. Ryu, C. Rump, and C. Qiao, “Advances in internet congestion control,” *Communications Surveys Tutorials, IEEE*, vol. 5, no. 1, pp. 28 –39, quarter 2003.
[2] H. Wedde and F. Muddassar, “A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks,” *Journal of Systems Architecture*, vol. 52, no. 8, pp. 461–484, Aug. 2006. [Online]. Available: http://dx.doi.org/10.1016/j.sysarc.2006.02.005
[3] E. Altman, T. Boulognea, R. El-Azouzi, T. Jimenez, and L.Wynter, “A survey on networking games in telecommunications,” *Computers & Operations Research*, vol. 33, pp. 286 – 311, 2006.
[4] J. Nash, “The bargaining problem,” *Econometrica*, vol. 18, pp. 155 – 62, 1950.
[5] H. Park and M. van der Schaar, “Bargaining strategies for networked multimedia resource management,” *Signal Processing, IEEE Transactions on*, vol. 55, no. 7, pp. 3496 –3511, july 2007.
[6] C. Touati, E. Altman, and J. Galtier, “Utility based fair bandwidth allocation,” in *Proceedings of the IASTED International Conference on Networks, Parallel and Distributed Processing and Applications (NPDPA 2002)*, 2002, pp. 126–131.
[7] ——, “Generalized Nash bargaining solution for bandwidth allocation,” *Computer Networks*, vol. 50, no. 17, pp. 3242 – 3263, 2006.
[8] H. Yaiche, R. Mazumdar, and C. Rosenberg, “A game theoretic framework for bandwidth allocation and pricing in broadband networks,” *Networking, IEEE/ACM Transactions on*, vol. 8, no. 5, pp. 667 –678, oct 2000.
[9] R. Mazumdar, L. Mason, and C. Douligeris, “Fairness in network optimal flow control: optimality of product forms,” *Communications, IEEE Transactions on*, vol. 39, no. 5, pp. 775 –782, may 1991.
[10] F. Kelly, “Charging and rate control for elastic traffic,” *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997. [Online]. Available: http://dx.doi.org/10.1002/ett.4460080106
[11] C. Douligeris and R. Mazumdar, “A game theoretic perspective to flow control in telecommunication networks,” *Journal of the Franklin Institute*, vol. 329, no. 2, pp. 383 – 402, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/001600329290041E
[12] R. Johari, S. Mannor, and J. Tsitsiklis, “A contract-based model for directed network formation,” *Games and Economic Behavior*, vol. 56, no. 2, pp. 201 – 224, 2006.
[13] E. Arcaute, R. Johari, and S. Mannor, “Network formation: Bilateral contracting and myopic dynamics,” in *Internet and Network Economics*, ser. Lecture Notes in Computer Science, X. Deng and F. Graham, Eds. Springer Berlin Heidelberg, 2007, vol. 4858, pp. 191–207.
[14] W. Saad, Z. Han, M. Debbah, and A. Hjorungnes, “Network formation games for distributed uplink tree construction in ieee 802.16j networks,” in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, 30 2008-dec. 4 2008, pp. 1 –5.
[15] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes, “A selfish approach to coalition formation among unmanned air vehicles in wireless networks,” in *Game Theory for Networks, 2009. GameNets ’09. International Conference on*, may 2009, pp. 259 –267.
[16] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, “Coalitional game theory for communication networks,” *Signal Processing Magazine, IEEE*, vol. 26, no. 5, pp. 77 –97, september 2009.
[17] W. Saad, Z. Han, M. Debbah, and A. Hjorungnes, “A distributed merge and split algorithm for fair cooperation in wireless networks,” in *Communications Workshops, 2008. ICC Workshops ’08. IEEE International Conference on*, may 2008, pp. 311 –315.
[18] W. Saad, “Coalitional game theory for distributed cooperation in next generation wireless networks,” Ph.D. dissertation, University of Oslo, 2010.
[19] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, and T. Basar, “Coalitional games for distributed collaborative spectrum sensing in cognitive radio networks,” in *INFOCOM 2009, IEEE*, april 2009, pp. 2114 –2122.
[20] J. Wardrop, “Some theoretical aspects of road traffic research,” *Proc. Inst. Civ. Eng.*, vol. 1, pp. 325 – 378, 1952.
[21] T. Roughgarden, *Selfish Routing and the Price of Anarchy*. 55 Hayward Street Cambridge, MA 02142-1493 USA: MIT Press, 2005.
[22] ——, “Selfish routing and the price of anarchy,” Department of Computer Science, Stanford University, Tech. Rep., Jan. 2006. [Online]. Available: http://theory.stanford.edu/ tim/papers/optima.pdf
[23] R. Thrall and W. Lucas, “n-person games in partition function form,” *Naval Research Logistics Quarterly*, vol. 10, no. 4, pp. 281–298, Dec. 1963.
[24] L. Á. Kóczy, “A recursive core for partition function form games,” *Theory and Decision*, vol. 63, no. 1, pp. 41–51, August 2007.
[25] ——, “Sequential coalition formation and the core in the presence of externalities,” *Games and Economic Behavior*, vol. 66, no. 1, pp. 559–565, May 2009.

Dávid Csercsik was born in Budapest, Hungary, in 1981. He received his M.Sc. degrees in electrical engineering and biomedical engineering from the Budapest University of Technology and Economics, and his Ph.D. degree in informatics from the Interdisciplinary Doctoral School at the Pázmány Péter Catholic University in 2005, 2007, and 2010, respectively. His research interests include cooperative game theory, networks and systems theory, systems biology and computational neuroscience.

Sándor Imre (Member, IEEE) was born in Budapest, Hungary, in 1969. He received the M.Sc. degree in electrical engineering, the dr.univ. degree in probability theory and mathematical statistics, and the Ph.D. degree in telecommunications from the Budapest University of Technology (BME), Budapest, Hungary, in 1993, 1996, and 1999, respectively, and the D.Sc. degree from the Hungarian Academy of Sciences, Budapest, Hungary, in 2007. Currently he is Head of Department of Networked Systems and Services at BME. He is also Chairman of the Telecommunication Scientific Committee of the Hungarian Academy of Sciences. Since 2005, he has been the R&D Director of the Mobile Innovation Centre. His research interests include mobile and wireless systems, quantum computing, and communications. He has made wide-ranging contributions to different wireless access technologies, mobility protocols, security and privacy, reconfigurable systems, quantum-computing-based algorithms, and protocols. Prof. Imre is on the Editorial Board of two journals: Infocommunications Journal and Hungarian Telecommunications.