A Coalitional Game Based Approach for Multi-Metric Optimal Routing in Wireless Networks

Eleni Stai[1], Symeon Papavassiliou[1], John S. Baras[2]
[1] School of Electrical & Computer Engineering, National Technical University of Athens (NTUA), Athens, Zografou, 15780, Greece.
[2]Institute for Systems Research, Department of Electrical & Computer Engineering, University of Maryland, College Park, MD 20742, USA.
Emails: estai@netmode.ntua.gr, papavass@mail.ntua.gr, baras@umd.edu

***Abstract*—Achieving high Quality of Service (QoS) over wireless multihop networks calls for enhanced routing/scheduling algorithms. Towards this direction it has been shown in the literature that the Greedy Backpressure algorithm which combines routing based on greedy hyperbolic embedding with backpressure scheduling, achieves to improve delay while remains throughput optimal. However, the performance of such an approach is significantly affected by the selection of the corresponding spanning tree used for greedily embedding the network into the hyperbolic space. Our work aims exactly at addressing this issue, that is the construction of an appropriate spanning tree that improves the cost of the paths used by the Greedy Backpressure approach, when considering a more generic weighted network graph modeling. The latter allows us to take into consideration the link costs in the routing process, which in turn may result in the simultaneous improvement of multiple performance metrics. To address the problem under consideration, we propose a coalition formation game framework among the network nodes, so that they can decide cooperatively for the spanning tree, via trading their value functions designed to depend on the link weights. We prove that the stable outcome of the coalitional game is a spanning tree of the network, and study through simulations the induced improvement in the network performance. Furthermore, we extend the framework for a scenario with multiple costs on each link through multi-tree hyperbolic embedding.**

I. Introduction

Modern communication systems support interactive applications that require high Quality of Service (QoS) (e.g. high throughput, low delay, high trustworthiness etc.). In this paper the emphasis is placed on wireless multihop networks and particularly on the improvement of the QoS of the routing process. In our previous work [1], [2], we have developed and enhanced the Greedy Backpressure (GBP) routing/scheduling algorithm which combines the original Backpressure algorithm (BP) [3] with greedy routing in hyperbolic space. It has been shown that every graph has a greedy embedding in hyperbolic space which ensures the successfulness of greedy routing [4].

For performing greedy routing, the network is greedily embedded in hyperbolic space [5] via a randomly chosen spanning tree. It is shown through modeling and simulations that GBP improves the throughput-delay trade-off [1] while at the same time remains throughput optimal similar to the original BP [3]. It is noted that the selection of the corresponding spanning tree is of paramount importance with respect to the performance of the routing process. Let us denote as "greedy" paths, the paths consisting of nodes with strictly decreasing hyperbolic distances to their destinations (may consist also of non-spanning tree links). More explicitly, a different spanning tree of the network defines different greedy paths between the source-destination pairs, affecting the performance of GBP. For non-weighted graphs, the greedy paths corresponding to different spanning trees may differ on their number of hops, thus differentiating the gains in time delay. Under the assumption of link costs, this dependence becomes stronger, since the costs of the greedy paths may differ significantly among different spanning trees, which in turn may negatively affect the cost of routing. This paper aims exactly at addressing this issue and studies the problem of the selection of the spanning tree for the network's greedy embedding in hyperbolic space, so that to improve the cost of the paths used by GBP.

Towards this direction we use cooperative game theory [6] and build a coalition formation game framework for constructing the spanning tree. Cooperative game theory provides the mathematical tools to study the cooperation among the rational, selfish nodes of a wireless multihop network and their organization into coalition structures. In this paper, we focus on the construction of the spanning tree, for a weighted network graph, where each link weight $w(i, j)$ represents the cost of the corresponding link (i, j). In this case, simply reducing the time delay is not enough as, additionally, the incurred cost of the selected paths needs to be taken into consideration leading to a multi-metric problem. Since GBP does not consider the link weights in the routing process, the greedy paths followed may consist of high cost links. The optimal greedy paths for each node may correspond to different spanning trees and as a result it may not be possible to define a unique spanning tree ensuring minimum cost greedy paths for all node pairs.

There are several examples in literature of the use of coalitional games (see [6], [7] and the references therein) in routing, cognitive radio networks, MIMO systems, rate allocation etc. The value of the coalition, depends both on the participants of the coalition and on the interconnections among them. In our work, we aim at designing value functions for the coalitions so that all nodes participate in a unique coalition with spanning tree structure. Furthermore, we propose an

978-1-4577-1348-4/13/$31.00 ©2013 IEEE

extension of the spanning tree construction framework in the case of multiple weight costs on each link. The simulation results, show both the possible gains by a suitable spanning tree design and the effect of different choices of spanning trees on the QoS performance.

The rest of the paper is organized as follows. Section II describes the system model. In Section III, we present the necessary theory of coalition formation games which motivates our proposed framework. Section IV, is devoted to the development and analysis of the spanning tree construction framework, while Section V contains a numerical example providing a proof of concept of the effectiveness of the proposed approach, along with some indicative simulation results demonstrating the induced performance gains. Finally, Section VI extends GBP for a scenario with multiple costs on each link through multi-tree hyperbolic embedding.

II. System Model

Let us consider a static wireless multihop network over a set of $\mathbb{N}$ nodes, with cardinality N. The time t is considered slotted. Each node i stores a queue $q_i^d(t)$ for each destination d. We denote with $\mu_{ij}(t)$, $\mu_{ij}^d(t)$ the total number of packets and the number of packets for destination d respectively, served by the link (i,j) at time t. We represent the network by two types of graphs: the physical layer undirected graph represented by the adjacency/weight matrix $W=[w(i,j)]$, where $w(i,j)=w(j,i)>0$ if i is a direct neighbor of j at the physical layer and the social layer directed graph represented by the matrix S_c, consisting of network flows where $S_c(i,j)=1$ if i is source of packets with destination the node j. The term $\mathbf{I_S}$ refers to the set of all independent sets of the physical graph, i.e. maximal sets of links that do not interfere with each other and their corresponding service rates.

As aforementioned, in our previous work [1], [2], we proposed the GBP routing/scheduling algorithm for wireless multihop networks based on the BP algorithm [3] and the greedy embedding of the network in hyperbolic space. A greedy embedding in hyperbolic space is a correspondence between nodes and hyperbolic coordinates such that the greedy routing algorithm, employed in hyperbolic space, does not have local minima, i.e. every node can find at least one neighbor closer than itself to all possible destinations [4], [5]. The greedy embedding is constructed by choosing a spanning tree of the graph of the initial network and then embedding the spanning tree into the hyperbolic space according to [5], by assigning hyperbolic coordinates to every node. If a spanning tree of the graph is greedily embedded in hyperbolic space then the whole graph is also greedily embedded [4]. By definition, the greedy embedding ensures the existence of at least one greedy path between each source-destination pair in the case of static networks. Let us denote as $dist_H(i,d)$ the hyperbolic distance between nodes i,d.

GBP performs routing and scheduling as follows. At each time slot t, for each link (i,j), GBP computes the differences $P_{ij}^d(t)=(q_i^d(t)-q_j^d(t))$, for all d for which j is a greedy neighbor of i towards d, i.e. $dist_H(i,d)>dist_H(j,d)$, and then finds $d^*(i,j)$ that maximizes the difference $P_{ij}^d(t)$ and $P_{ij}(t)=\max(\max_d P_{ij}^d(t),0)$. Afterwards, similarly to the original BP, an independent set, $I_S(t)$, is chosen for transmission by solving the maximum weight matching $[\mu_{ij}(t)]=arg\max_{\mu'\in\mathbf{I_S}}\sum_{(i,j)}\mu'_{ij}P_{ij}(t)$, where $[\mu_{ij}(t)]$ is a $N\times N$ matrix, denoting the communication traffic for all links at time t. Each link (i,j) included in $I_S(t)$, serves the maximum possible number of packets of the destination $d^*(i,j)$ for slot t. GBP, similarly to BP, has the important advantage of throughput optimality, i.e. it can support every arrival rate which can be supported by any other scheduling algorithm [1]. Beyond throughput optimality, GBP was shown in [1], [2], to improve the delay performance of the original BP algorithm by restricting the number of paths used by BP to only greedy paths, especially in light traffic conditions [1]. For higher traffic rates, the combination of the greedy routing constraints with appropriate weighting of the link queue differences can lead to significant delay reductions compared to BP [2].

III. Background on Coalition Formation Network Games

The "players" of the network game making decisions are the nodes. A coalition is a connected subset of nodes, i.e. for any two nodes belonging in the same coalition there is either a direct link or a multihop path connecting them. The matrix of the coalition S with graph structure G_s, i.e. the pair (S,G_s), denoted as W_s, can be constructed by erasing from W the entries corresponding to nodes and links, not belonging to the coalition. We denote by $|S|$ the cardinality of the coalition S.

In this paper we are studying coalition formation games [8] in graph form (or called simply graph games) where the value function depends on both the number of nodes participating in the coalition and the structure of the coalition (links), i.e. the same set of nodes with different connections among them can lead to different results for the value function. Each node i computes its "value" function, $v_i(G_s)$, for belonging in the coalition S with graph structure G_s as the difference between its gain and cost. We study Non-Transferable Utility Games (NTU), i.e. the payoff of each player coincides with its value, in the sense that the players cannot exchange payoffs. The value function of the coalition S with graph structure G_s, is equal to $v(G_s)=\sum_{\forall i\in S}v_i(G_s)$ [7]. We assume that a node $i\in S$ with payoff $v_i(G_s)$ can also gain every payoff less than $v_i(G_s)$. Assuming this, we can define the set of payoff vectors on G_s, $V(G_s)=\{y\in\mathbb{R}^N|y_i\leq v_i(G_s)\ \forall\ i\in S\}$. If S consists of all the nodes, it is the grand coalition symbolized by $\mathcal{N}$. Let us symbolize with G^s, all the possible connected graphs on the coalition S and $\mathbb{G}^N=\cup_{S\subseteq\mathbb{N}}G^s$. Finally, $V(\mathcal{N})$ denotes the set of all possible payoff vectors over all graphs on $\mathcal{N}$.

The described network game is iterative, i.e. at each time slot t each node decides about its coalition and evaluates its value function according to both its decision and the other nodes' decisions. More specifically, at slot t, node i decides its action, independently from the rest of the nodes, and the action is expressed as a $1\times N$ real valued vector, $x_i^t=[x_{i1}^t\ x_{i2}^t...x_{iN}^t]$, where $x_{ij}^t=1$ if i has decided to form a direct connection with node j and $x_{ij}^t=0$ otherwise. All possible actions of node i form the set X_i (time independent). It should be noted that the decisions of the wireless nodes about their possible one-hop connections, are restricted by the power limitations characterizing the wireless terminals. A link (i, j) is formed at slot t only if $x^t_{ij} = x^t_{ji} = 1$. The symbol x^t_{-i} is used to denote the decisions of all other payers except from i, at slot t. We denote with $x^t = (x^t_1, x^t_2, ..., x^t_N)$, the action set at time t of all network nodes. All the possible action sets of the network form the set $X = \Pi_{i=1}^N X_i$. The probability of having an action set $x \in X$, at time slot t, is $p^t(x) = \Pi_{i=1}^N p^t_i(x_i)$, where $p^t_i(x_i)$ is the probability that node i performs $x_i \in X_i$. In the sequel, we present two stability notions applying to coalitional network games and we briefly describe a learning algorithm presented in [7] for leading the coalitional game to equilibrium.

Following the definition of [7], the Nash Equilibrium of the coalition formation game is a probability distribution p on all possible action sets X, if no player can deviate from p and improve its pay-off, i.e. $\forall\ i, x'_i \in X_i, \sum_{x_{-i} \in X_{-i}} v_i(x'_i, x_{-i}) \Pi_{j \neq i} p_j(x_j) \leq \sum_{x \in X} v_i(x) \Pi_j p_j(x_j)$, where v_i is the value/payoff function of i expressed here via the nodes' decisions. As analytically proven in [7], an algorithm that can lead to the set of Nash Equilibria when the number of iterations tends to infinity, is the following. At each time slot t, each node i either continues its previous action or chooses another action with probability proportional to how much accumulated gain in payoff would have experienced if it had always selected this action in the past. The average payoff for i up to time t is equal to $\bar{v}^t_i = \frac{\sum_{1 \leq s \leq t} v_i(x^s_i, x^s_{-i})}{t}$. For every action $x'_i \in X_i$, the regret up to time t is defined as: $\bar{r}^t_i(x'_i) = \max\{\frac{\sum_{1 \leq s \leq t} v_i(x'_i, x^s_{-i})}{t} - \bar{v}^t_i, 0\}$. According to the learning strategy, at time $(t+1)$, the node i chooses action x_i with probability $p^{(t+1)}_i(x_i) = \frac{\bar{r}^t_i(x_i)}{\sum_{x'_i \in X_i} \bar{r}^t_i(x'_i)}$.

A stronger notion of equilibrium in a coalition formation game is the core, C, of the graph game [9]. The core of a NTU graph game is the set of payoff vectors $v = [v_1\ v_2 ... v_N] \in \mathbb{R}^N$ satisfying that $v \in V(\mathcal{N})$ and there does not exist a combination of a coalition S and graph structure, $G_s \in G^s$, and a payoff vector corresponding to this combination $y \in V(G_s)$, such that $y_i(G_s) > v_i,\ \forall\ i \in S$. In order to determine if the core is non empty, we need the notion of the balanced graph game and the balanced core. Let $S \subseteq \mathbb{N}$ be a coalition of players with graph G_s. Then, we define a power measure vector $m(G_s) \in \mathbb{R}^N$, such that $m_i(G_s) = 0$ if $i \notin S$, and $\sum_{i \in S} m_i(G_s) = 1$. For a given power measure m over $\mathbb{G}^N$, a family of graphs in $\mathbb{G}^N$, $\mathbb{F} = \{\mathbb{F}_1, \mathbb{F}_2, ..., \mathbb{F}_k\}$ is graph balanced if there exist positive numbers $\lambda_1, \lambda_2, ..., \lambda_k$, such that $\sum_{j=1}^k \lambda_j m(\mathbb{F}_j) = \frac{\mathbf{1}}{N}$ where $\mathbf{1}$ is the unity vector. If for every graph balanced family $\mathbb{F}$, it holds that $\cap_j V(\mathbb{F}_j) \subset V(\mathcal{N})$ then the NTU game with power measure m is a balanced graph game. The balanced core, is a special case of the core with the extra requirement that there should exist a graph balanced family $\mathbb{F}$, such that $v \in \cap_j V(\mathbb{F}_j)$.

Next we quote the conditions for the nonemptiness of the balanced core and since the latter is subset of the core, it is implied that the core is non-empty. From Theorem 4.3 of [9], the balanced core is non empty if: (*i*) Any isolated node (graph $G_{\{i\}}$) can achieve an upper bounded payoff, i.e. $v_i(G_{\{i\}}) \leq a_i$ with $a_i \in \mathbb{R}$. (*ii*) For every combination $S \subseteq \mathbb{N}$, G_s, the set $\{(l_i)_{i \in S} \in \mathbb{R}^{|S|} | l \in V(G_s)$ and $l_i \geq a_i\ \forall\ i \in S\}$ is bounded. (*iii*) Every set $V(G)$ is closed and comprehensive, i.e. if $x \in V(G)$, then $y \in V(G)$ for all $y \leq x$. (*iv*) The game is graph balanced.

IV. Spanning Tree Construction: Framework and Analysis

In this section we focus on the construction of a stable spanning tree of the network to be used for the hyperbolic embedding. The source-destination pairs indicated by the social layer matrix S_c, demand low delays and low cost end-to-end communication paths. GBP can significantly improve delay without however taking into consideration the link costs. Since there are specific source-destination pairs in the network, we are interested in selecting greedy paths between each communicating pair consisting of low cost links. As already mentioned in Section I, we will develop a coalitional iterative graph game for the construction of the spanning tree for the hyperbolic embedding, aiming at reducing the cost of the greedy paths as this is defined by the weights of the network graph. Towards this direction, we observe that the nodes' gain from cooperation is twofold. First, they gain by the formation of a spanning tree that will allow them to apply the GBP algorithm and improve the throughput-delay trade-off. Second, gain is also induced by selecting as greedy paths mostly low-cost paths. Also, multiple greedy paths are preferable to enable multi-path routing and avoid congestion. We assume that all nodes have the incentive to cooperate, even if not acting as sources, due to the cooperative nature of wireless multihop networks, for getting reimbursed by being able to deliver or receive their own packets through the cooperation of the rest of the nodes, when they need to.

The aim is to design the value function $v_i(G_s)$, depending on the graph structure G_s of the coalition S, so that the following three objectives are achieved: (*i*) The grand coalition $\mathcal{N}$ is formed. (*ii*) The structure of the grand coalition is a spanning tree. (*iii*) Each node tries to select its preferred subset of paths towards its destinations as greedy paths.

Let us denote as $N^c_i(G_s)$ the number of nodes with which i is connected either directly or via multiple hops, when participating in coalition S with graph structure G_s. Similarly, $N^d_i(G_s)$ is the number of single-hop neighbors of i. We propose the following value function:

$$v_i(G_s) = B_1 N^c_i(G_s) - B_2 N^d_i(G_s) + f_i(W_s),\ \forall\ i, \qquad (1)$$

where B_1, B_2 are constant values appropriately defined as it will be explained below and $f_i(W_s)$ is a positive function of the link weights and the coalition structure. We assume that the addition of links and nodes in a particular graph structure can only increase or leave unchanged the value of $f_i(W_s)$. The first summand of $v_i(G_s)$ corresponds to the connectivity gain and the third summand to the gain of i by satisfying its routing preferences. The second summand expresses the cost induced by the direct connectivity as this is desirable to be kept low for achieving a tree structure. The following two propositions concern the tree structure and the stability of the coalition if each node is using the value function of Eq. (1).

Proposition 1. *If $B_2 > max_{\{i,W_s\}} f_i(W_s)$, all coalitions formed will have a tree structure.*

Proof. Let us suppose that there exists a coalition structure G_s which is not a tree. In this case there is a node i in G_s which can delete the connection with one of its direct neighbors j, maintaining simultaneously the connectivity inside the coalition. Let us define $G'_s = G_s - \{(i,j)\}$ with matrix W'_s. We observe that, $N_i^c(G_s) = N_i^c(G'_s)$, while $N_i^d(G'_s) = N_i^d(G_s) - 1$. We compute the change in the value function, $v_i(G'_s) - v_i(G_s) = B_2 + f_i(W'_s) - f_i(W_s)$, and thus by considering the assumptions of the proposition it holds that $v_i(G'_s) > v_i(G_s)$. As a result, a non-tree structured coalition cannot be formed. □

Proposition 2. *If $B_1 > B_2$, the core is non-empty consisting only of the grand coalition.*

Proof. Let us suppose that the grand coalition is not formed. Then there is a node i belonging in coalition S, which can add one more direct neighbor belonging in coalition R, creating the coalition $S' = S \cup R$. Then for i, $N_i^d(G_s)$ will be increased by one and $N_i^c(G_s)$ by at least one and $v_i(G_{s'}) - v_i(G_s) \leq B_1 - B_2 + f_i(W_{s'}) - f_i(W_s)$, i.e. $v_i(G_{s'}) > v_i(G_s)$, leading to the formation of the grand coalition. Note that $f_i(W_{s'}) - f_i(W_s) \geq 0$, due to our initial assumptions for the function f and the fact that the addition of a new direct neighbor from a node only adds links and nodes in its current coalition. In a similar way it can be shown that the values of the other nodes participating in S' increase compared to the corresponding values in S, R. As a result only a coalition containing all the nodes can be stable.

We prove now the non-emptiness of the core by examining the four conditions highlighted at the end of Section III. We define the power measure function $\forall\ i\ \in (S, G_s)$ as $m_i(G_s) = m_i(S) = \frac{1}{|S|}$. According to this measure, each graph of the grand coalition ($G^{\mathcal{N}}$) is self-graph balanced ($\lambda_1 = 1$). In addition k different graphs on the grand coalition are graph balanced with coefficients $\lambda_1 = ... = \lambda_k = \frac{1}{k}$. For every graph balanced function (not only those defined for the grand coalition) $\mathbb{F}$, it holds that $\cap_j V(\mathbb{F}_j) \subset V(\mathcal{N})$, since each node achieves always higher payoff if belonging in a graph of the grand coalition. Therefore the graph game is balanced and the condition (*iv*) is satisfied. The conditions (*i*), (*ii*) are obviously satisfied since the payoffs are bounded (finite network). Finally, regarding condition (*iii*), it holds from the initial assumption for the payoff vectors, i.e. the node i belonging in (S, G_s) can receive any payoff $y_i \leq v_i(G_s)$. As a result, the balanced core is not empty and since it contains only stable coalitions it will include only one or more spanning trees (by taking also into account the Proposition 1). Indeed each spanning tree belongs to a graph balanced family consisting of itself. □

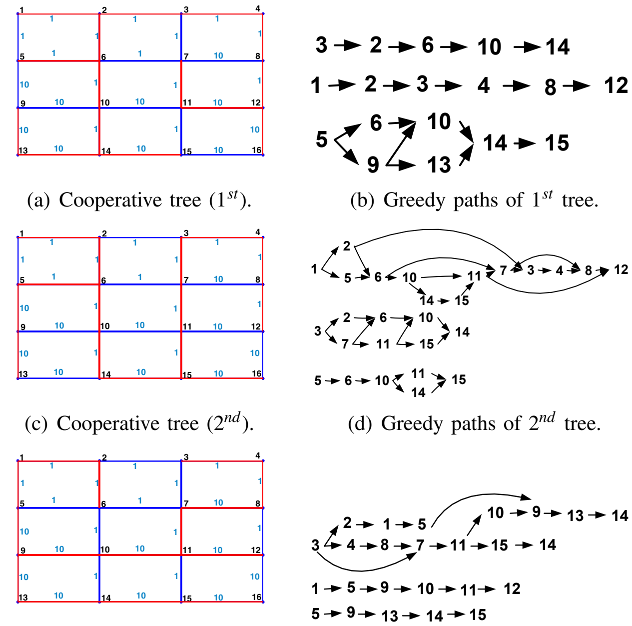
The first two summands of Eq. (1) are responsible for the construction of the spanning tree while the last summand affects the selection of the spanning tree according to the routing preferences of the nodes. In our work we propose a specific type for the function $f_i(W)$, that takes into consideration the greedy embedding of the current coalition formed, and the routing preferences of the sources. As a result, if i is not a source of packets, $f_i(W) = 0$. Differently, let us denote the set of greedy paths from i to destination d as $P_g(i,d)$ and then $f_i(W) = \sum_{\{d|S_c(i,d)=1\}} \sum_{\{(j,k)\in P_g(i,d)\wedge I_{nc}(jk)\}} \frac{1}{w(j,k)}$, where $I_{nc}(jk)$ is an index that the weight $w(j,k)$ is not counted again in the sum. For ensuring uniqueness of the greedy embedding of the spanning tree for a tree coalition S, we can assign distinct weights to all the nodes and choose the node with the highest weight as root of the tree. The proposed value function can be computed easily by short message exchanges among the nodes, so that each node discovers its one hop and multihop neighbors and the weights lying on the greedy paths towards its destinations. Furthermore, v_i has a higher value when more links with low cost are included in the greedy paths. This approach gives rise to both multi-path and low cost routing.

V. Numerical Example and Results

In this section, in order to demonstrate the improvement achieved by the proposed approach, we describe a numerical example for comparing the cost of the greedy paths of cooperatively built trees with a randomly built tree. For the solution of the iterative coalitional graph game, we use the value function specified in Section IV and the learning algorithm presented in Section III. We stop the learning algorithm at $t = 3000$ slots and observe the current coalition structure. Regarding the network topology, we consider a 4x4 grid ($N = 16$) with link weights as shown in Fig. 1. The social layer matrix S_c consists of 3 unity entries, $S_c(1,12) = 1$, $S_c(5,15) = 1$, $S_c(3,14) = 1$. We define $B_1 = 100$, $B_2 = 99$, since $max_i f_i(W) \leq 24$, due to the fact that the grid has 24 links. Fig. 1(a), 1(c) show two possible trees derived from the coalitional graph game, while Fig. 1(e) depicts a possible random tree. Figures 1(b), 1(d), 1(f), show the greedy paths between the source-destination $(s-d)$ pairs in each case. The arrows represent possible transitions from a node to one of its neighbors, while the numbers are the IDs of the nodes as specified in Fig. 1(a). Note that the greedy paths may contain some non-tree edges with the constraint of diminishing hyperbolic distances towards the destination. For all trees we consider as root of the spanning tree the node 1.

By simple computations with the hyperbolic coordinates, the costs and the number of the greedy paths derived by each spanning tree are presented in Table I. The third column shows for every path cost, the number of paths with this cost (inside the parenthesis). The multiple greedy paths between a source-destination pair are not distinct, but differ in one or more links. Obviously, the spanning trees constructed by the coalitional graph game will achieve lower cost routing than the random tree, since GBP uses only greedy paths for routing. This is obvious in the fourth column of Table I, where the cooperatively built trees have lower average costs. Compared to the random tree, the 1^{st} tree constructed via the coalition formation game leads to low cost paths for pairs $1-2$ and $3-14$ while it increases the path diversity for the last pair. Similarly, the 2^{nd} tree decreases significantly the cost of the greedy paths and simultaneously increases the path diversity.

The improvement of the path costs by a suitable choice of the spanning tree used in the hyperbolic embedding, il-



(a) Cooperative tree (1^{st}).

(b) Greedy paths of 1^{st} tree.

(c) Cooperative tree (2^{nd}).

(d) Greedy paths of 2^{nd} tree.

(e) Randomly constructed tree.

(f) Greedy paths of random tree.

Fig. 1. Spanning tree selection for the greedy hyperbolic embedding.

TABLE I

RESULTS AFTER 3000 SLOTS RUNNING TIME OF THE LEARNING ALGORITHM.

(s,d) pair	Spanning Tree	Cost (Number) of Greedy paths	Average cost (Total #)
$(1,12)$	1^{st} Tree	5(1)	5(1)
	2^{nd} Tree	5(1), 7(2), 14(2), 18(2), 25(4), 23(2), 20(2), 27(2)	19(17)
	Random	41(1)	41(1)
$(3,14)$	1^{st} Tree	4(1)	4(1)
	2^{nd} Tree	4(2), 13(2)	8.5(4)
	Random	33(1), 24(1), 53(1), 13(1), 42(1)	33(5)
$(5,15)$	1^{st} Tree	13(1), 31(1), 40(1)	28(3)
	2^{nd} Tree	13(2)	13(2)
	Random	40(1)	40(1)

lustrates the importance of the selection of the spanning tree in weighted graphs for reducing the cost of communication. Regarding the time delay, GBP was simulated in light traffic conditions (throughput achieved is very close to 1) under the three trees. For traffic rate equal to 0.075 packets per slot for all sources each one tree of Fig. 1 led to an average time delay 4.66, 5.8, 5 slots respectively. It can be observed that the 2^{nd} tree leads to increased time delay due to its longer (in hops) greedy paths for the pair $1-12$.

VI. EXTENSION TO MULTI-METRIC ROUTING

In this section, we propose multi-metric routing through multi-tree embedding. Let us suppose that each link is characterized by a vector of weights, $w(i,j) = [w_1(i,j)\ w_2(i,j)...w_K(i,j)]$ where K is the number of the metrics of interest expressed in the form of cost values. In addition, assume that for each packet and/or source a different metric is critical for the quality of the packet delivery process. In this case we propose to construct and embed in the hyperbolic space a different spanning tree for each metric, T_l, $l = 1,...,K$. Furthermore, each node i stores a separate queue for each destination d and for each spanning tree T_l, i.e. $q_i^{d,T_l}(t)$. When a new packet arrives at the source, the source decides which metric is more important for its delivery and stores the packet to the corresponding queue. We denote the communication traffic on link (i,j) for destination d and tree T_l at time t as $\mu_{ij}^{d,T_l}(t)$. The GBP algorithm can be modified as follows: each link (i,j) maximizes the queue difference $q_i^{d,T_l}(t) - q_j^{d,T_l}(t)$, $\forall\ d, T_l$, while the maximum weight matching for choosing the transmitting schedule at time t is performed as in the GBP. If the link (i,j) is included in the transmitting schedule, it serves the packets of the selected node d over the selected tree T_l (those maximizing the queue difference). By using Lyapunov drift techniques as in [1], [3], it is straightforward to prove that the modified GBP algorithm is throughput optimal.

In addition, one can borrow the notion of Pareto Optimality in order to solve the multi-metric problem [10]. In this case node i can compute its value for tree T_j as a vector $v_i^{T_j} = [v_i^{T_j}(1)...v_i^{T_j}(K)]$, where $v_i^{T_j}(l)$ is the value of node i for the l metric (i.e. considering only the weights $w_l(i,j)$, $\forall\ (i,j)$) over the tree T_j. We say that the spanning tree T_1 Pareto dominates the spanning tree T_2 for node i if $v_i^{T_1}(h) \geq v_i^{T_2}(h)$, $\forall\ h = 1...K$ and one at least strict inequality holds. As a result a node will prefer for routing the set of its Pareto optimal spanning trees (trees that are not Pareto dominated by others), and choose among them given also other criteria regarding its performance metrics of interest.

Acknowledgment This research is co-financed by the European Union (European Social Fund) and Hellenic national funds through the Operational Program 'Education and Lifelong Learning' (NSRF $2007-2013$). John S. Baras was supported by US NSF grant CNS-1018346.

REFERENCES

[1] E. Stai, J. S. Baras, S. Papavassiliou, "Social Networks over Wireless Networks", *inv. to 51st IEEE Conf. on Dec. and Cont. (CDC)*, Dec. 2012.
[2] E. Stai, J. S. Baras, S. Papavassiliou, "A Class of Backpressure Algorithms for Networks Embedded in Hyperbolic Space With Controllable Delay-Throughput Trade-off", *in Proc. of ACM MSWiM Int'l Conf.*, Oct. 2012.
[3] L. Tassiulas, A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks", *IEEE Trans. on Autom. Control*, Vol. 37, No. 12, pp. 1936-1948, December 1992.
[4] R. Kleinberg, "Geographic Routing Using Hyperbolic Space", *Proc. of IEEE INFOCOM*, pp. 1902-1909, May 2007.
[5] A. Cvetkovski, M. Crovella, "Hyperbolic Embedding and Routing for Dynamic Graphs", *Proc. of IEEE INFOCOM*, pp. 1647-1655, April 2009.
[6] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, T. Basar, "Coalitional Game Theory For Communication Networks", *IEEE Signal Processing Management*, Vol. 26, No. 5, pp. 77-97, September 2009.
[7] T. Jiang, J. S. Baras, "Coalition Formation through Learning in Autonomic Networks", *Int'l Conf. on Game Theory for Networks*, May 2009.
[8] F. Forgo, J. Szep, F. Szidarovszky, "Introduction to the Theory of Games: Concepts, Methods, Applications", *Kluwer Academic Publishers*, 1999.
[9] P. J.-J. Herings, G. van der Laan, D. Talman, "Cooperative Games in Graph Structure", *Res. Memoranda, Maastricht Res. Sch. Econ., Technol., Org., Memo. Maastricht, The Netherlands*, No. 11, August 2002.
[10] F. R. Fernández, M. A. Hinojosa, J. Puerto, "Multi-criteria Minimum Cost Spanning Tree Games", *European Journal of Operational Research*, Vol. 158, No. 2, pp. 399-408, Oct. 2004.