J Internet Serv Appl (2011) 2:113–127
DOI 10.1007/s13174-011-0024-0

SI: FUTURE NET SERVICE MODELS & DESIGNS

A socially-driven topology improvement framework with applications in content distribution and trust management

Eleni Stai · Vasileios Karyotis · Symeon Papavassiliou

Received: 8 February 2011 / Accepted: 18 May 2011 / Published online: 22 June 2011

Abstract Contemporary networking infrastructures are required to be capable of adapting to the increasing trends of user demands, as well as the impairments of their operational environment. In this work, by exploiting the power varying capabilities of multihop wireless networks and inspired by social structures of the higher protocol layers, we develop a distributed and dynamic physical topology modification framework for weighted and directed multihop networks. The operational robustness and effectiveness of the proposed framework is demonstrated in two highly popular application areas, namely QoS-oriented content distribution and trust management in wireless multihop networks. We focus on the emerging trade-offs of topology modification, and through analysis and simulation, we demonstrate how social features can be used in improving the physical network topology and corresponding performance.

Keywords Wireless multihop networks · Evolutionary modification framework · Small-world phenomenon · Topology control · Trust management · Content distribution

1 Introduction and related work

Motivated by advances of Network Science [1], network engineering has followed a hierarchical approach in modeling and analyzing interactions and information flow in dynamic networks, identifying a three level design hierarchy. At the lowest level, physical networks such as the Internet and wireless networks, exhibit associations that correspond one-to-one in actual connectivity. The second level, denoted as logical networks (overlay and peer-to-peer), involves logical associations among peers. The third level, denoted as (online) social networks [2], involves complex interactions that take into account unpredictable/hidden social associations, apart from physical and logical connectivity. Bottom-up interactions and impact between the above design levels have been observed and studied extensively [3, 4]. Research and developments in the reverse direction, where social trends are considered for improving logical level mechanisms (e.g., improving the logical server network of Facebook or LinkedIn to accommodate more users/traffic), have emerged to a more restricted degree.

In this paper, we examine the potentials of extending the engineering impact flow to the top-to-bottom direction as well, leading potentially to an "evolutionary design loop" for wireless multihop networks, similar to several natural evolutionary learning processes [5]. Despite being cost-effective, wireless multihop networks, usually suffer from long average hop distances among randomly selected nodes, while multihop forwarding usually increases MAC overhead. They also present intermittent connectivity, especially when mobility is allowed, while the energy of each node is limited. Thus, they are currently unable to support demanding end-to-end applications and increasing social communication demands [2]. However the nodes of a wireless multihop network are interdependent due to their forwarding relay nature, similarly to the actors of a social network. Therefore, we introduce a framework for exploiting features of the higher modeling level (social) into the lower (physi-

E. Stai · V. Karyotis · S. Papavassiliou (✉)
School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechniou 9, 15780 Zografou, Athens, Greece
e-mail: papavass@mail.ntua.gr

E. Stai
e-mail: estai@netmode.ntua.gr

V. Karyotis
e-mail: vassilis@netmode.ntua.gr

cal). The proposed framework exploits the Topology Control capabilities of such networks [6], and by reversing such process to extending the transmission radii of selected nodes (inverse Topology Control—iTC), it obtains performance gains, improving several topology characteristic properties, e.g., average path length.

It is well established that among the most typical emerging social structures is that of the small-world effect [3, 4], which bears small average path length and high clustering. Such properties are desired in communication networks for improving end-to-end performance in various applications, e.g., content distribution, trust management, etc. Various evolutionary approaches suggest developing small-world characteristics and exploiting them for improving internal mechanisms [7–10]. Typical approaches add shortcuts to a regular grid, or randomly rewire its edges leading to a small-world structure [3, 4]. In the literature, the main focus has been so far on adding wired links aiming at reducing the average path length and increasing network capacity [11], or achieving energy efficiency in ad hoc or sensor networks [12, 13]. However, wired shortcuts are subject to a fixed deployment, high cost, and do not inherently suit distributed multihop networks.

The work in [14] first studied the possibility of small-world properties in purely multihop wireless networks. The spatial character of the networks was considered and the length of the rewired or added wireless links was restricted to a fraction of the network diameter, although a practical method to achieve this was not specified. In [15], the potential of inverse Topology Control (iTC) for multihop networks has been demonstrated, aiming to yield significant performance benefits, without significantly impacting energy consumption. However, the modification processes in [14, 15] were restricted on solely adding or rewiring communication links. Furthermore, these works did not offer a basis for topology optimization. Our work is mainly focused on providing such basis and eventually lead to automated and adaptive topology improvements.

A first attempt to infuse small-world properties in wireless multihop networks by exploiting all the processes characterizing network evolution was presented in [16]. In this work, we extend the model presented in [16] in weighted and directed RGGs and better adapt it to specific applications of high interest. Link weights express the heterogeneity in communicating node pairs, which should be taken into consideration along with connectivity. We mainly aim at reducing the mean distance between communicating nodes by taking into consideration the specific definitions of distances per different applications. The mathematical analysis of this paper is based on models proposed for the evolution of weighted random networks through edge and node addition. Indicatively, the model in [17] correlates the topology evolution of a social network (Internet traffic, network of airports, etc.) with dynamic variation and adaptation of weights. Similar models have been proposed for describing the dynamics of weighted communication networks [18] or the WWW topology [19]. Inspired by these models, we adapt the graph weights of a RGG during the topology modifications of our network churn model in a socially-aware fashion.

We present detailed demonstrations and adaptations of the proposed framework in two application areas of high research and practical importance for the considered decentralized and inherently stringent networking environments. The first one deals with QoS-oriented content distribution and the second with trust management in wireless multihop networks. The proposed methodology is properly adapted for each application and through analysis and simulation, we show how social features can be exploited in improving the physical topology supporting each application.

The rest of this paper is organized as follows. In Sect. 2, we describe the considered networking model and the employed system performance metrics, while in Sect. 3 the proposed topology modification framework is presented in detail. Sections 4 and 5 focus on detailed implementations of the proposed framework, based on the applications of trust management and content distribution respectively. Finally, Sect. 6 concludes the paper.

2 System model and metrics

We consider the general case of a directed and weighted wireless multihop network, represented by a corresponding weighted Random Geometric Graph (RGG) [20]. Links are established both at the physical and network layer. In the latter, based on the operation of the MAC protocol (e.g., DCF protocol) point-to-point links are established, thus avoiding excessive energy consumptions. RGGs express the spatial character of multihop networks, where neighboring relations are determined by the transmission radius (power) of nodes and they are characterized by long average path length. In this work, we acquire a heterogeneous RGG with small-world features by infusing shortcuts in the initial homogeneous (node range wise) RGG. We assume that the weights can be related to the traffic intensity of a link, i.e., the intention of a node to send content to a neighboring one. We consider asymmetric and nonnegative link weights. V is the set of vertices or nodes of the graph with cardinality $|V| = N$ and E the set of directed edges, where $e = (i, j)$ is an edge from node i to node j. We denote by $A = [a_{ij}]$ the adjacency matrix, where $a_{ij} = 1$ if there is a link from i to j, otherwise $a_{ij} = 0$. Similarly, we employ the weight matrix $W = [w_{ij}]$, where w_{ij} is the weight of the link (i, j).

In a directed graph, a node i is characterized by its out- and in-degrees, which are equal to $k_i^{\text{out}} = \sum_{j=1}^N a_{ij}$ and $k_i^{\text{in}} = \sum_{j=1}^N a_{ji}$, respectively. By analogy to node degrees, in weighted directed graphs, we may define the in- (s_i^{in}) and out- (s_i^{out}) node strengths, which express the total amount of weight that reaches or leaves node i correspondingly [17, 21]. Thus,

$$s_i^{\text{out}} = \sum_{j=1}^{N} w_{ij} \quad \text{and} \quad s_i^{\text{in}} = \sum_{j=1}^{N} w_{ji} \tag{1}$$

In this work, node selection takes place according to social features relevant to popularity or unpopularity. Such rules have been shown to be responsible for several emerging social structures [4]. According to the first one, denoted as in-strength preferential attachment, selection is performed proportionally to the total weight entering node i and is expressed by the probability:

$$\Pi_i^{\text{in}} = \frac{s_i^{\text{in}}}{\sum_{j=1}^{N} s_j^{\text{in}}} \tag{2}$$

while the second one is inversely proportional to the in-strength, namely:

$$\Pi_i^{\text{invs}} = \frac{\frac{1}{s_i^{\text{in}}+\theta}}{\sum_{j=1}^{N} (\frac{1}{s_j^{\text{in}}+\theta})} \tag{3}$$

Π_i^{invs} is obtained from (2), by replacing s_i^{in} with the inverse in-strength equal to $\frac{1}{s_i^{\text{in}}}$. As s_i^{in} decreases, $\frac{1}{s_i^{\text{in}}}$ increases, leading to higher selection probability for low-strength nodes. The positive parameter θ (taking very small values) ensures the rule works even for disconnected nodes with zero strength. Similarly, we can define the out-strength (inverse) preferential attachment by replacing in (2), (3) Π_i^{in}, Π_i^{invs} and s_i^{in} with Π_i^{out}, Π_i^{outvs}, and s_i^{out} respectively. Strength preferential attachment generalizes the degree-driven preferential attachment for weighted networks [22].

Table 1 summarizes the most important symbols used throughout this paper.

2.1 Path length and clustering coefficient

The average path length is defined as the mean number of hops of the shortest paths among all node pairs [21], i.e., usually: $L_{\text{avg}} = \frac{\sum_{i,j}(h_{i,j})}{N*(N-1)}$, where $(h_{i,j})$ is the shortest path (in hops) from node i to j, and may potentially differ from $(h_{j,i})$ in directed graphs.

In this work, we employ a generalized semiring based computation for shortest paths [23]. A semiring denoted as $(R, \oplus, \otimes, 0, 1)$ is a set R equipped with two binary operations $\oplus$ and $\otimes$, such that: (1) $(R, \oplus)$ is a commutative monoid with identity element 0; (2) $(R, \otimes)$ is a monoid with identity element 1 and absorbing element 0; and (3) $\otimes$ distributes over $\oplus$. In Mohri's algorithm [23], which solves the single source shortest path (SSSP) problem for all nodes, $\otimes$ is used to combine weights along paths and $\oplus$ across paths. The definition of the operation of these two operators and the set R, allows this shortest path algorithm to be adapted to a wide range of applications. For instance, if the weights express traffic cost, weights are added (+) along paths and their minimum value is taken across different paths. However, if the weights express trust values, then multiplication along paths and maximization across different paths is a more suitable computation [24, 25].

Table 1 Nomenclature

Symbol	Semantics
$W_{(ij)} \equiv w_{ij}$	Element in the i row, j column of the weight matrix W
$\mathcal{N}^{\text{in}}(i)$	The set $\{j \in V \mid w_{ji} > 0\}$ of in-neighbors of i
$\mathcal{N}^{\text{out}}(i)$	The set $\{j \in V \mid w_{ij} > 0\}$ of out-neighbors of i
s_i^{in}	In-strength of i
s_i^{out}	Out-strength of i
$\bar{x}$	Average value of x over all nodes or links of the network
p, q, r, v	Probabilities of edge addition/deletion, node addition/deletion
m_a, m_d, M_a, M_d	# of nodes participating in edge/node addition/deletion
$\mathbb{P}_i^a, \mathbb{P}_i^d, \mathbb{P}_i^{nd}$	Probabilities of node selection
$\delta^{1a}, \delta^{2a}, \delta^{La}, \delta^{1d}, \delta^{2d}, \delta^{\text{na}}$	Weight adaptation parameters (δ parameters)

In the case of undirected and unweighted graphs, the clustering coefficient C_i of a node i expresses the direct connections among its neighbors (forming triangles around i) [26], and the network clustering coefficient (CC) is the average $CC = \frac{\sum_{i=1}^N C_i}{N}$. In our work, we employed the approach of [26] for calculating the coefficient C_i according to which, C_i increases, not only with the increase in the number of triangles around node i, but also with the increase of the weights' values contained in these triangles. In [26], it is assumed that weights are normalized in $(0, 1)$, which is easily achieved if every weight is divided by the maximum value $w_{\max}$.

3 Topology modification framework and weight adaptation mechanisms

In this section, we describe and analyze the proposed topology modification framework. It consists of two basic mechanisms, inspired by dynamic network evolution, the Weighted

Edge Churn (WEC) and the Weighted Node Churn (WNC) mechanisms.

In both mechanisms, time is considered slotted. We consider a square deployment area of side L where N nodes are randomly and uniformly distributed. Initially, all nodes have the same transmission radius R_f. We assign an initial random weight in a specified interval to each directed link. Each node is capable of varying its transmission radius in the range R_{MIN}, R_{MAX} [6, 27]. $R_c(i)$ denotes the current transmission radius of node i. Each time step is characterized by two parameters, $R_{\min}(t)$, $R_{\max}(t)$, where $R_{\mathrm{MIN}} \le R_{\min}(t) \le R_{\max}(t) \le R_{\mathrm{MAX}}$. Initially, $R_{\mathrm{MIN}} = R_{\min}(0) = R_{\max}(0) = R_f$. At the beginning of each step t (starting at $t = 1$), $R_{\min}(t) = R_{\max}(t-1)$ and $R_{\max}$ is increased by a predefined step value a, i.e., $R_{\max}(t) = R_{\max}(t-1) + a$. The induced network must remain connected. However, deletion of nodes or edges may disconnect some network parts, thus higher addition rates of links and/or nodes ensure connectivity maintenance in the final network.

3.1 Weighted Edge Churn framework

The Weighted Edge Churn (WEC) method consists of a link addition and a link deletion mechanism, as explained in the following.

3.1.1 Edge addition with weight adaptation

Edge addition, takes place with probability p $(0 \le p \le 1)$ and m_a new connections are added to m_a selected nodes. Each one of the m_a nodes, selected with probability $\mathbb{P}_i^a$, increases its range and extends its neighborhood. Similarly to online social networks, whose members obtain new acquaintances, the m_a selected nodes become more "social" (popular) in the network. We choose a node i to perform link addition with probability $\mathbb{P}_i^a$, proportional to its popularity, thus proportional to the intensity that the other network nodes want to communicate with i, since popularity in terms of communication denotes traffic intensity.

Edge addition is realized at the Physical and Network protocol stack layers. More explicitly, when link addition is performed at step t, a node i is selected with probability $\mathbb{P}_i^a$ $(\sum_{i=1}^N \mathbb{P}_i^a = 1)$ and extends its range from $R_c(i)$ to $R_{\max}(t)$, as in Fig. 1. Although, i increases its transmission range to $R_{\max}(t)$ and thus, it can transmit in the Physical layer to all the nodes in this range, it forms a Network layer "long-range" connection, only with one node j, lying in the annulus from $R_{\min}(t)$ to $R_{\max}(t)$ centered at node i and denoted by $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$. Thus, i avoids depleting its energy by sending and forwarding to all the nodes in the newly added range. Node j is selected among all nodes in $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$ with probability $Q(j|i)$ $(\sum_{j \in \mathcal{A}_{R_{\min}}^{R_{\max}}(i)} Q(j|i) = 1)$ which denotes the selection of j given that i is selected. The form of

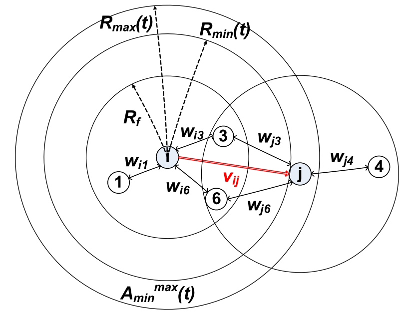


Fig. 1 Edge addition and weight adaptation

$Q(j|i)$ depends on the application and may be a function of the weights in the local neighborhoods of nodes i, j, i.e., $Q(j|i) = f_1(w_{kh}|k, h \in$ local neighborhoods of $i, j)$. Also, the weight v_{ij} of the new link is determined according to the application setting and $v_{ij} = f_2(w_{kh}|k, h \in$ local neighborhoods of $i, j)$. We will elaborate more on the selection of the above probabilities in Sects. 4 and 5. The addition of the directed link $i \to j$ with weight v_{ij} is depicted in Fig. 1 where initially j was two hops away from i and $R_c(i) = R_f$.

The above process is repeated for each one of the m_a nodes selected at a time step t and the links added are considered directional, as only the transmission radius of the node initiating the modification process increases. By controlling parameters p and m_a, we can either add a few links at a step but of different lengths, i.e., in many steps (with high p and low m_a) or add many links of the same length (low p, high m_a), or add many links at each step of different lengths (high p, high m_a), etc.

This addition of the long-range link will cause a variation in the weights of the local neighborhood of nodes i and j. It is intuitive that the links' weights will be influenced at least locally [22] due to the new connection. We assume that a node is able to adapt only its out-links, which is a reasonable assumption, as the traffic passing through the out-links of a node is controlled totally by this node, while the traffic passing through the in-links is controlled by neighboring nodes. Precisely, as in Fig. 1, node i, has added a long-range connection $i \to j$, and thus it can use more intensively this connection for traffic exchange toward this flow direction, while it can reduce the traffic sent through the rest of its out connections toward the specific direction. Similarly, node j has obtained a new in-connection which will result in more flow leaving node j especially when j is used as a relay node. The adaptation of weights followed in this work is motivated by the model in [17, 22] which considers that the establishment of a new edge with weight w to a vertex induces a total increase in the amount of traffic equal to δ which is distributed to the edges departing from this vertex proportionally to their previous weights.

Following a similar approach, we consider three constants, δ^{1a}, δ^{La}, and δ^{2a} which express the total change in the amount of traffic, for node i, for the link $i \to j$ and node j correspondingly. As a result, node i adapts the weights of its out-links as follows:

$$w'_{ih} = w_{ih} + \delta^{1a} \frac{w_{ih}}{s_i^{\text{out}}}, \quad h \in (1..N),\ h \neq j \tag{4}$$

The link $i \to j$ is adapted as

$$v'_{ij} = v_{ij} + \delta^{La} v_{ij} \tag{5}$$

Similarly for j,

$$w'_{jh} = w_{jh} + \delta^{2a} \frac{w_{jh}}{s_j^{\text{out}}}, \quad h \in (1..N) \tag{6}$$

We consider the permissible values of the above constants (δ^{1a}, δ^{La}, δ^{2a}) in the interval $[-1, 1]$. However, more exact definition fields of these constants can be defined with respect to different types of applications.

After the addition of the directed link $i \to j$ (Fig. 1) the strengths of nodes i and j are altered as follows:

$$\begin{aligned} s_j^{\text{in}} &\leftarrow s_j^{\text{in}} + v_{ij}\left(1 + \delta^{La}\right) \\ s_j^{\text{out}} &\leftarrow s_j^{\text{out}} + \delta^{2a} \\ s_i^{\text{out}} &\leftarrow s_i^{\text{out}} + v_{ij}\left(1 + \delta^{La}\right) + \delta^{1a} \end{aligned} \tag{7}$$

It is important to mention that we consider the adaptation of the weights preceding the change of strengths, meaning that the adaptation in (4), (5), and (6) takes place according to the old values of strengths of i and j. In the following, we construct differential equations expressing the rate of change of the in-strength and out-strength of each node. The out-strength of node i changes when:

1. The node is the one initiating the process, i.e., the node from where the long-range link starts (node i in Fig. 1). In this case, i is selected with probability $\mathbb{P}_i^a$ and s_i^{out} changes by $v_{ik}(1 + \delta^{La}) + \delta^{1a}$ with k chosen with probability $Q(k|i)$.
2. The node i is the end of the long-range link (i.e., node j in Fig. 1). This occurs if one of i's neighbors, i.e., $k \in \mathcal{A}_{R_{\min}}^{R_{\max}}(i)$ is chosen to initiate the process and chooses i with $Q(i|k)$. In this case, s_i^{out} changes by δ^{2a} and it takes place with probability

$$P_G(i) = \sum_{k \in \mathcal{A}_{R_{\min}}^{R_{\max}}(i)} \mathbb{P}_k^a Q(i|k) \tag{8}$$

However, v_{ik} does not depend only on i and this creates a difficulty in constructing the differential equation for the rate of change of s_i^{out}. Thus, instead of using v_{ik}, we use the average of the possible v_{ik} for all neighbors k of i in the $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$, expressed by the equation:

$$\hat{v}_i = \frac{\sum_{k \in \mathcal{A}_{R_{\min}}^{R_{\max}}(i)} v_{ik}}{\frac{N|\mathcal{A}_{R_{\min}}^{R_{\max}}(i)|}{L^2}} \tag{9}$$

From all the above, the rate of change for s_i^{out} is

$$\frac{ds_i^{\text{out}}}{dt} = m_a p\left[\mathbb{P}_i^a\left(\hat{v}_i\left(1 + \delta^{La}\right) + \delta^{1a}\right) + P_G(i)\delta^{2a}\right] \tag{10}$$

Similarly, the in-strength of node i changes when:

1. The node is an out-neighbor of the one initiating the process (nodes 1, 3, 6 in Fig. 1). This happens with probability $\sum_{k \in \mathcal{N}^{\text{in}}(i)} \mathbb{P}_k^a$.
2. The node is the end of the long-range link (node j in Fig. 1) with probability $P_G(i)$.
3. The node is an out-neighbor of the end of the long-range link (nodes 3, 4, 6 in Fig. 1). This happens with probability $\sum_{k \in \mathcal{N}^{\text{in}}(i)} P_G(k)$.

Thus, the evolution equation for the s_i^{in} is:

$$\begin{aligned} \frac{ds_i^{\text{in}}}{dt} = m_a p\Bigg[& \sum_{k \in \mathcal{N}^{\text{in}}(i)} \mathbb{P}_k^a \delta^{1a} \frac{w_{ki}}{s_k^{\text{out}}} \\ & + \sum_{k \in \mathcal{A}_{R_{\min}}^{R_{\max}}(i)} \mathbb{P}_k^a Q(i|k)(\delta^{La} + 1)\hat{v}_k \\ & + \sum_{k \in \mathcal{N}^{\text{in}}(i)} P_G(k)\delta^{2a} \frac{w_{ki}}{s_k^{\text{out}}}\Bigg] \end{aligned} \tag{11}$$

3.1.2 Edge deletion with weight adaptation

Edge deletion takes place with probability q $(0 \leq q \leq 1)$ and m_d links, one from each one of m_d selected nodes, are deleted. We propose selecting a node i to initiate edge deletion with probability $\mathbb{P}_i^d$ ($\sum_{i=1}^N \mathbb{P}_i^d = 1$). Our choice for $\mathbb{P}_i^d$ depends on the popularity of a node and more precisely, nodes with low popularity are chosen w.h.p. to perform deletion. This choice is intuitive as deleting one of the out-links of an unpopular node would not influence significantly the network flow. Only the forward direction of a link is deleted $i \to j$, in case both of them exist, where such process is achieved at the logical layer with proper routing modification. The node j from where the edge is deleted is chosen with probability $Q^d(j|i)$ ($\sum_{j \in \mathcal{N}^{\text{out}}(i)} Q^d(j|i) = 1$) among the out-neighbors of i where $Q^d(j|i) = f_3(w_{ij})$. As an example, in Fig. 2 node i is selected to delete its link with its one-hop neighbor j.

Similarly to edge addition, we consider an adaptation of weights in the local neighborhood of nodes i and j, due to

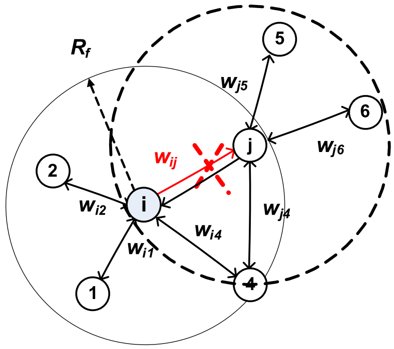


Fig. 2 Edge deletion and weight adaptation

the deletion of the link $i \to j$. As mentioned above, nodes can influence and control only their out-strengths. Thus, the locally induced variation of weights concerns only the out-links of nodes i, j. We define two constants δ^{1d}, δ^{2d}, expressing the total amount of change of flow in the remaining links of i and j, respectively. Therefore, we consider the following weight adaptations:

$$w'_{ih} = w_{ih} + \delta^{1d} \frac{w_{ih}}{s_i^{\text{out}} - w_{ij}}, \quad h \in (1..N),\ h \neq j \tag{12}$$

Similarly for j,

$$w'_{jh} = w_{jh} + \delta^{2d} \frac{w_{jh}}{s_j^{\text{out}}}, \quad h \in (1..N) \tag{13}$$

We consider that the weights' variation takes place after the link is deleted, and for this reason in the denominator in (12) instead of using the old strength s_i^{out} of node i, we subtract the weight of the deleted link, i.e., $s_i^{\text{out}} - w_{ij}$. The definition fields of the constants δ^{1d}, δ^{2d} follow the same rules as in the edge addition mechanism. After the deletion of the directed link $i \to j$ (Fig. 2) the strengths of nodes i and j are altered as follows:

$$\begin{aligned} s_j^{\text{in}} &\leftarrow s_j^{\text{in}} - w_{ij} \\ s_j^{\text{out}} &\leftarrow s_j^{\text{out}} + \delta^{2d} \\ s_i^{\text{out}} &\leftarrow s_i^{\text{out}} - w_{ij} + \delta^{1d} \end{aligned} \tag{14}$$

Sequentially, following the same approach as in the edge addition mechanism, we describe strengths' rates of change through differential equations due to edge deletion [17]. Specifically, the out-strength of node i changes when:

1. The node is the one initiating the process, i.e., the node deleting an out-link e.g., $i \to j$ (node i in Fig. 2). In this case, node i is selected with probability $\mathbb{P}_i^d$ and s_i^{out} changes by $\delta^{1d} - w_{ij}$.
2. The node i is losing an in-link (i.e., node j in Fig. 2). This happens if one of its neighbors k in $\mathcal{N}^{\text{in}}(i)$ is chosen to initiate the process and chooses i with $Q^d(i|k)$. Thus, the probability is

$$P_G^d(i) = \sum_{k \in \mathcal{N}^{\text{in}}(i)} \mathbb{P}_k^d Q^d(i|k) \tag{15}$$

In this case, s_i^{out} changes by δ^{2d}.

However, w_{ij} does not depend only on i and this creates a difficulty in constructing the differential equation for the rate of change of s_i^{out}. Thus, instead of using w_{ij}, we use the average weight, $\bar{w}$, of the links of the whole network.

Thus, the evolution equation for the s_i^{out} is

$$\frac{ds_i^{\text{out}}}{dt} = m_d q \left[\mathbb{P}_i^d \left(\delta^{1d} - \bar{w}\right) + P_G^d(i)\delta^{2d}\right] \tag{16}$$

Similarly, the in-strength of a node i changes when:

1. The node is an out-neighbor of the one initiating the process (nodes $1, 2, 4$ in Fig. 2). This happens with probability $\sum_{k \in \mathcal{N}^{\text{in}}(i)} \mathbb{P}_k^d$.
2. The node i is the node from where the in-link is deleted (node j in Fig. 2) with probability $P_G^d(i)$.
3. The node is an out-neighbor of the node that loses its in-link (nodes $4, 5, 6$ in Fig. 2), which happens with probability $\sum_{k \in \mathcal{N}^{\text{in}}(i)} P_G^d(k)$.

Thus, the evolution equation for the s_i^{in} is

$$\begin{aligned} \frac{ds_i^{\text{in}}}{dt} = m_d q \Bigg[&\sum_{k \in \mathcal{N}^{\text{in}}(i)} \mathbb{P}_k^d \delta^{1d} \frac{w_{ki}}{s_k^{\text{out}} - \bar{w}} - P_G^d(i)\bar{w} \\ &+ \sum_{k \in \mathcal{N}^{\text{in}}(i)} P_G^d(k) \delta^{2d} \frac{w_{ki}}{s_k^{\text{out}}} \Bigg] \end{aligned} \tag{17}$$

3.1.3 Weighted Edge Churn (addition and deletion)

Combining edge addition and deletion is important, since the first infuses small-world features in the multihop topology, while the second ensures that the induced network does not become overcrowded, and thus suffers from excessive interference. It is obvious that the mechanisms proposed to infuse social properties in wireless multihop networks take into consideration their spatial character and do not impact their character as Random Geometric Graphs. In WEC mechanism, edge addition happens with probability p and m_a links are added to m_a nodes while edge deletion happens with probability q and m_d links are deleted. We assume that $p + q \leq 1$ and if $p + q < 1$ then with probability $1 - p - q$ the network topology does not change. In the combined mechanism, the rate of change $\frac{ds_i^{\text{in}}}{dt}$ is obtained by the superposition of the rates in (11) and (17), while the rate $\frac{ds_i^{\text{out}}}{dt}$ is obtained by the superposition of the rates in (10) and (16). However, in the general case, the solution of these equations is cumbersome and requires that all quantities included (weights, probabilities) are expressed as time functions.

Finally, the change of the number of links at each time step due to edge churn can be computed by the equation $\ell(t) = pm_a - qm_d$. Thus, up to step t, we have a total change of $(pm_a - qm_d)t$ of the number of links in the network.

3.2 Weighted Node Churn framework

Weighted node churn is based on node variation which is a natural process in wireless multihop networks. For instance, some nodes may deplete their energy and become inactive, while others recharge and become operational again. Node churn is also evident in online social networks where new users enter the network, possibly invited by their friends, or existing nodes choose to leave the network especially when their intensity of participation in the network processes is low. Therefore, inspired by social networks, we formulate the node churn mechanism in such a way that it permits the infusion of social properties in the initial RGG of the wireless multihop network.

3.2.1 Node addition with weight adaptation

At a time slot, node addition takes place with probability r and M_a new nodes are added in the network. The new nodes have increased radius compared to the original ones (R_f) in the network. This serves the purpose of reducing the average path length, as in the other case, the result would be simply a larger network bearing the same characteristics, i.e., the path length as the initial one.

Each new node, added at step t, has a radius $R_{\max} = R_f + at$. The new connections are considered bidirectional in an area of πR_f^2 and directional in the annulus extending from radius R_f to radius $R_{\max}$ with center the newly added node (Fig. 3). The weights of the new links are chosen according to a probability distribution depending on the application. For instance, the beta $\beta(m,n)$ distribution can be used as it takes different forms depending on m, n, and thus can be adapted to a wide range of applications.

The addition of the new node with increased radius is considered to cause a variation in links' weights as follows. Inside the disk with radius R_f (Fig. 3), we assume that there is no adaptation of the weights as the links are bidirectional and, therefore, both the out- and the in-strength of a node are influenced leading to a relative balance in flow. However, in the annulus $\mathcal{A}_{R_f}^{R_{\max}}(i)$, the links added are considered directional increasing the in-flow of the corresponding nodes, i.e., in Fig. 3 the incoming flow of nodes 1, 6, 7 has increased. Therefore, their out-strength has to be adapted so as they can drain the additional flow. Each one of the nodes

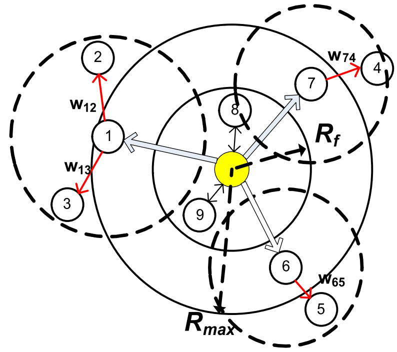


Fig. 3 Node addition and weight adaptation

in $\mathcal{A}_{R_f}^{R_{\max}}(i)$, where i is the newly added node, changes the weights of its out links as follows:

$$w'_{jh} = w_{jh} + \delta^{na}\frac{w_{jh}}{s_j^{\text{out}}}, \quad j \in \mathcal{A}_{R_f}^{R_{\max}}(i),\ h \in (1..N) \tag{18}$$

where δ^{na} is a constant parameter chosen as the corresponding ones in WEC case and depends on the application. Also, the weight adaptation, as in WEC is considered local and concerns only the out-links of the corresponding nodes. As an example, in Fig. 3, node 7 changes the weight w_{74} following (18).

If considering each new link having a weight equal to the average of the probability distribution used for assigning weights to the new links and denoted as $E(P)$, then the change of the total weight of the network due to node addition is approximately equal to:

$$\begin{aligned} dw(t) &= 2\pi M_a N(t) r \frac{R_f^2}{L^2} E(P) \\ &\quad + \pi r M_a N(t) \frac{(R_f + at)^2 - R_f^2}{L^2} \\ &\quad \times \left(E(P) + \delta^{na}\right) \end{aligned} \tag{19}$$

In (19), the first term corresponds to the bidirectional links in the range R_f (factor of 2), while the second term corresponds to the directional links. Also, $N(t)$ is the number of nodes in the network at step t.

3.2.2 Node deletion

At time slot t, node deletion takes place with probability v and M_d nodes are chosen with probability $\mathbb{P}_i^{nd}$, to be deleted. The probability $\mathbb{P}_i^{nd}$ ($\sum_{i=1}^{N} \mathbb{P}_i^{nd} = 1$) is chosen inversely proportional to the participation of a node in the network and as a result, a node with no strong connections with its neighbors is w.h.p. deleted. As expected, all the in- and out-connections of the deleted nodes are also deleted. We note that node deletion is used for replacing some unpopular nodes (according to the specific definition of unpopularity employed in each application), with others more popular and set more intelligently in the topology. In node deletion, we assume that there is not weight adaptation, as once a node is deleted all of its connections are deleted bidirectionally. Considering that each deleted link has, at $t-1$ step, an average weight $\bar{w}(t-1)$ and the deleted node has average (in- or out-) degree $\bar{k}(t-1)$, then the total change of weight induced by node deletion is calculated as $-2vM_d\bar{k}(t-1)\bar{w}(t-1)$.

3.2.3 *Weighted Node Churn (addition and deletion)*

Node addition and deletion are complementary mechanisms and combined they form the WNC mechanism. Both can be part of the natural evolution of the network, where nodes leave and enter the network. Essentially, the first cannot be considered without the second and vice versa, since the first improves the structure (average distance), while the second ensures a balanced operation without excessive interference.

In the node churn case, at each time step t, node addition happens with probability r where M_a nodes are added and node deletion happens with probability v where M_d nodes are deleted. Similarly, with edge churn, $r+v\leq 1$. The average number of nodes at each time step t is theoretically equal to: $N(t)=N+M_art-M_dvt$ with N, the initial number of nodes in the network.

The average out-strength $\bar{s}^{\text{out}}$ (the same equation holds for $\bar{s}^{\text{in}}$), over all nodes, changes from step $t-1$ to the next step t as in the following equation:

$$\begin{aligned}\bar{s}^{\text{out}}(t) &= \bar{s}^{\text{out}}(t-1)\frac{N(t-1)}{N(t)} + 2\pi M_a r\frac{R_f^2}{L^2}E(P)\\ &\quad + \pi r M_a\frac{(R_f+at)^2-R_f^2}{L^2}\big(E(P)+\delta^{na}\big)\\ &\quad - 2vM_d\frac{\bar{k}(t-1)\bar{w}(t-1)}{N(t)} \qquad (20)\end{aligned}$$

The total change in the number of links at each time step t may be approximated as

$$\begin{aligned}L_n(t) &= \left(2M_a\pi r\frac{R_f^2}{L^2} + \pi r M_a\frac{(R_f+at)^2-R_f^2}{L^2}\right)N(t)\\ &\quad - 2vM_d\bar{k}(t-1) \qquad (21)\end{aligned}$$

The approximate recursive equation of the average node degree is

$$\bar{k}^{\text{in}}(t)=\bar{k}^{\text{out}}(t)=\bar{k}(t)=\frac{E+\sum_{k=1}^{t}L_n(k)}{N(t)} \qquad (22)$$

where E is the number of directed links in the initial RGG.

Assuming that W_s is the sum of the weights in the initial RGG, the change in total weight at time step t is:

$$\begin{aligned}dw(t) &= 2\pi M_a N(t) r\frac{R_f^2}{L^2}E(P)\\ &\quad + \pi r M_a N(t)\frac{(R_f+at)^2-R_f^2}{L^2}\big(E(P)+\delta^{na}\big)\\ &\quad - 2vM_d\bar{k}(t-1)\bar{w}(t-1) \qquad (23)\end{aligned}$$

Therefore, the average weight in the network at time t is given by (24):

$$\bar{w}(t)=\frac{W_s+\sum_{k=1}^{t}dw(k)}{E+\sum_{k=1}^{t}L_n(k)} \qquad (24)$$

Thus, if the initial average connectivity ($\bar{k}(0)$), the initial average weight of the network ($\bar{w}(0)$) and the initial average strength of the network ($\bar{s}^{\text{out}}(0)$) are known, we can compute the average in- or out-strength (20) with the aid of (22) and (24).

3.3 Combined mechanism (WEC and WNC)

Finally, node churn and edge churn can be combined together and in this case the topology modification mechanism consists of all the processes of the network evolution. In this case, the mechanisms and their probabilities are depicted in Table 2 (where $p+r+q+v\leq 1$).

In the following Sects. 4 and 5, we describe and evaluate the adaptation and application of the proposed mechanisms in two different application scenarios. With respect to the operation of the proposed mechanisms, the two applications are differentiated by the semantics of the weights in each case. It is important to mention that the proposed mechanisms apply to a wide range of applications by properly specifying the probabilities $\mathbb{P}_i^a$, $\mathbb{P}_i^d$, $\mathbb{P}_i^{nd}$, $Q(k|i)$, $Q^d(k|i)$ and the values of the parameters δ^{1a}, δ^{2a}, δ^{La}, δ^{1d}, δ^{2d}, v_{ik} so as to fit in the framework of the specific application. A possible implementation of the proposed combined mechanism should take into consideration the distributed nature of the wireless multihop network and the restricted energy resources by using optimal parameter values for each process (Table 2), so as to lead to the highest possible gain with the smallest energy consumption.

Table 2 Combined mechanism

Processes	Probability	# of nodes or links participating
Edge addition	p	m_a links added
Edge deletion	q	m_d links deleted
Node addition	r	M_a nodes added
Node deletion	v	M_d nodes deleted

4 Application 1: trust management in wireless multihop networks

Commonly used trust models and mechanisms are mainly based on weighted and directed graph model, in order to denote trust relations between node pairs (personal beliefs, statistical recommendations, etc.). Two interacting nodes i, j, may assign local trust values w_{ij}, w_{ji} expressing their biased trust perception. These two values may differ, as the opinion of i for the trustworthiness of j may be different from the opinion of j for i. The obtained weighted directed graph describing such relations is called trust graph [24, 25]. In the static wireless multihop networks considered in our work, the trust graph coincides with the physical topology due to the locality of wireless communications and the weight matrix W expresses the trust values for all node pairs. In this work, a trust value equal to 0 means that node i has no direct communication with node j.

4.1 Framework specification and operation

In trust establishment/management, where nodes are more interested in using the most trusted paths, rather than the shortest (hop-wise) paths, social properties like small-world and high clustering are important. As a result, in this section, we adapt the proposed topology modification framework by determining the probabilities $Q(k|i)$, $Q^d(k|i)$, $\mathbb{P}_i^a$, $\mathbb{P}_i^d$, $\mathbb{P}_i^{nd}$ presented above, aiming at jointly reducing the mean hop count of the most trusted paths and increasing the corresponding path trust value. The definition of these probabilities can also express other policies and lead to different topology modifications than the one described in this work. We use a simple definition for the total trust value of a path, as the multiplication of link weights of a path. Thus, the most trusted path separating two nodes, is the one possessing the highest trust value. We denote this value as distance in the trust graph. The weights are normalized in the interval $[0, 1]$ for simplicity. Based on the above and following the analysis in this paper we essentially employ the following semiring definition $(R, \oplus, \otimes, 0, 1) \equiv ([0, 1], \max, \times, 0, 1)$ for computing trusted path quantities.

In the deletion process, a node i should choose with high probability a distrusted out-neighbor k to remove from its neighbor list, using probability $Q^d(k|i)$, which expresses inverse preferential attachment with respect to the weight w_{ik}:

$$Q^d(k|i) = \frac{w_{\max} - w_{ik}}{\sum_{j \in \mathcal{N}^{\text{out}}(i)} (w_{\max} - w_{ij})} \tag{25}$$

A node k with low value of w_{ik} is more probable to be selected to loose its in-link from i. The value of $w_{\max}$ is equal to 1 in this case.

In addition, we identify $\mathbb{P}_i^a$ with Π_i^{in}, based on the observation that a node with high in-strength is trusted by its one-hop neighbors, and thus more preferable to participate in the creation of shortcuts in the trust graph. Similarly, for edge deletion, $\mathbb{P}_i^d$ is chosen as Π_i^{invs}, so as to avoid distrusted nodes included in paths. On the contrary, the rule $\mathbb{P}_i^{nd}$ will delete nodes having low trusted connections, i.e., with probability inversely proportional to their out-strength, thus $\mathbb{P}_i^{nd} = \Pi_i^{\text{outvs}}$.

It can be observed that when $\mathbb{P}_i^a = \Pi_i^{\text{in}}$, the creation of chains of long-range connections may take place. This is possible, if for instance the link $i \to j$ is formed, and thus s_j^{in} increases, leading to a higher probability that j will be chosen at a next step.

In the sequel, we propose two variations of the probability that the shortcut $i \to k$ is formed $Q(k|i)$. In the first one, denoted by local, nodes require local information only in order to perform edge addition, while in the second, denoted by global, nodes need a network wide view. We define $Q(k|i)$ in two possible ways:

Local algorithm: In this case, node i requests information from its local neighbors (1 or 2 hops away), in order to learn the trust values of its potential neighbors in $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$, so that eventually the long link added is limited to 2 or 3 hops. Let h_2 be the nodes in $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$ that are 2-hops away from i and h_3 the nodes 3-hops away from i.

- 2 hops: If node k is 2-hops away from i, its probability of being selected to form the link $i \to k$ is

$$\begin{aligned} Q(k|i) &= \frac{\sum_h w_{ih} w_{hk}}{\sum_{h,(m \in h_2)} w_{ih} w_{hm} + \sum_{h,g,(m \in h_3)} w_{ih} w_{hg} w_{gm}} \\ &= \frac{W_{(ik)}^2}{\sum_{m \in h_2} W_{(im)}^2 + \sum_{m \in h_3} W_{(im)}^3} \end{aligned} \tag{26}$$

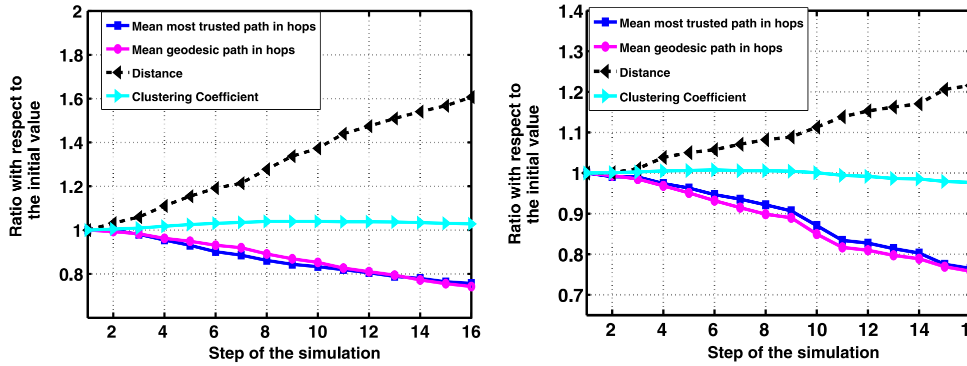
The weight assigned to the long-link is:

$$v_{ik} = \max_h (w_{ih} w_{hk}) \tag{27}$$

- 3 hops: If k is 3-hops away from i, its probability of being selected is

$$\begin{aligned} Q(k|i) &= \frac{\sum_{h,g} w_{ih} w_{hg} w_{gk}}{\sum_{h,(m \in h_2)} w_{ih} w_{hm} + \sum_{h,g,(m \in h_3)} w_{ih} w_{hg} w_{gm}} \\ &= \frac{W_{(ik)}^3}{\sum_{m \in h_2} W_{(im)}^2 + \sum_{m \in h_3} W_{(im)}^3} \end{aligned} \tag{28}$$

$$v_{ik} = \max_{h,g} (w_{ih} w_{hg} w_{gk}) \tag{29}$$



(a) Newly added trusted nodes (b) Newly added neutral nodes

Fig. 4 Node Churn mechanism performance for trusted and neutral newly added nodes

Global algorithm: In this case, there is no limitation on the hops covered by the added long link, while node i is assumed to obtain information from the whole network aiming to select w.h.p. the most trusted node in $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$. In this case, the node i should compute the path semirings for all nodes in $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$ and the probability that $k \in \mathcal{A}_{R_{\min}}^{R_{\max}}(i)$ is selected by node i is given as follows:

$$Q(k|i) = \frac{D_{i\to k}^t}{\sum_{j\in\mathcal{A}_{R_{\min}}^{R_{\max}}(i)} D_{i\to j}^t} \tag{30}$$

$$v_{ik} = D_{i\to k}^t \tag{31}$$

where $D_{i\to k}^t$ is the distance between i and k in the trust graph as noted above. In both algorithms, node i chooses w.h.p. a node k for which the trust path $i \to k$ has a high value.

4.2 Evaluation and discussion

In the sequel, we present simulation results concerning the average path length (geodesic) in hops, the trust value of the most trusted paths (distance), the number of hops of the most trusted paths and the clustering coefficient. The simulations were performed using Matlab. We consider an initial weighted directed RGG of $N = 250$ nodes, $R_f = 100$ m, square deployment area of side $L = 800$ m, $R_{\text{MAX}} = 250$ m, $a = 10$ m and the results were averaged over 5 different network topologies. Initially, we assign positive weights randomly on the directed links. Weight adaptation does not have a special meaning in trust graphs, as the direct opinion of a node for its neighbors does not change with edge variations. Therefore, we consider $\delta^{1a} = \delta^{2a} = \delta^{1d} = \delta^{2d} = \delta^{na} = 0$.

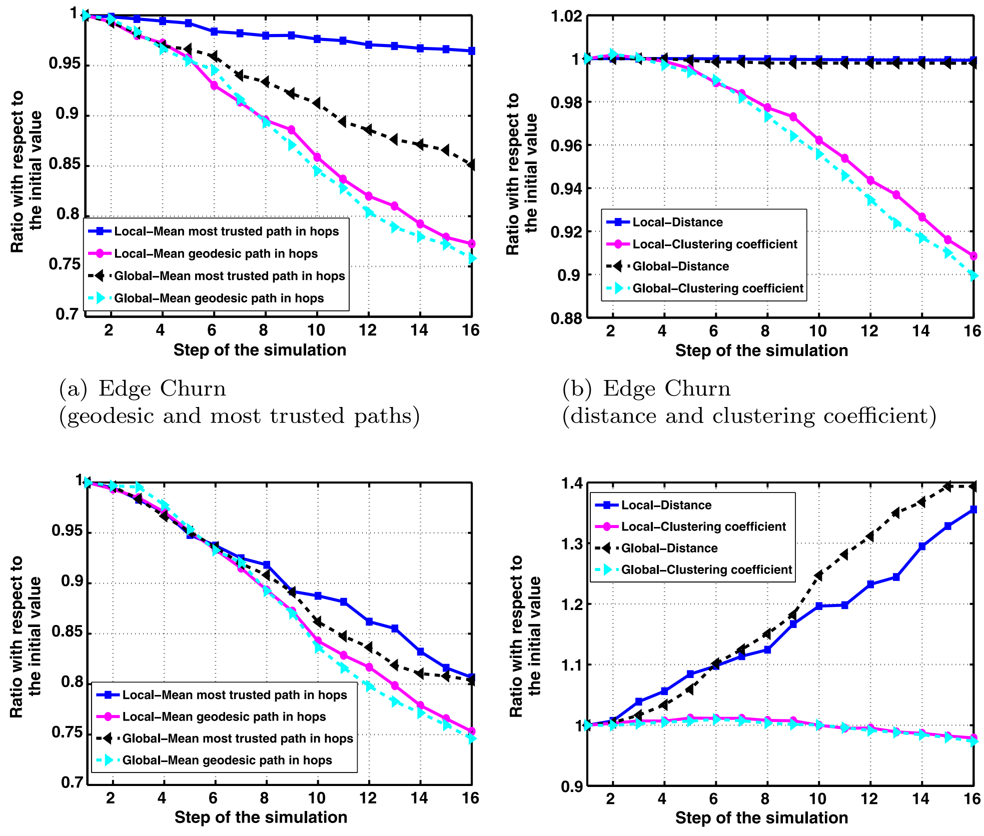
If only edge churn is performed, the parameters used are: $p = 0.7$, $q = 0.3$, $m_a = 25$, $m_d = 8$. In the case of node churn, the parameters used are $r = 0.7$, $v = 0.3$, $M_a = 3$, $M_d = 1$. For the combined mechanism, the parameters used for each process are $p = 0.35$, $m_a = 25$, $q = 0.15$, $m_d = 8$, $r = 0.35$, $M_a = 3$, $v = 0.15$, $M_d = 1$.

In Node Churn, shown in Fig. 4(a), we consider the newly added nodes as trusted and assign weights to their in-links according to the beta $\beta(5, 1)$ distribution which gives w.h.p. values close to 1, and to their out-links, according to $\beta(2, 2)$ which gives w.h.p. values close to 0.5. In Fig. 4(b), we assign weights from $\beta(2, 2)$ to both the in- and out-connections of the newly added nodes, so as to consider their trustworthiness as neutral.

It can be observed in Fig. 4 that the Node Churn mechanism achieves both the reduction of the mean geodesic path (in hops) and of the mean most trusted path (in hops) at approximately the same percentage. In addition, it maintains a stable clustering coefficient. The links of the new nodes with higher range serve as shortcuts, while nodes with distrusted connections are deleted leading to higher values of most trusted paths (distances). When the newly added nodes are highly trusted, the hop number of the most trusted paths reduces with higher rate and the average distance increases more (Fig. 4(a)).

We examine two cases of parameter δ^{La}, i.e., $\delta^{La} = 0$, $\delta^{La} = 0.1$. Initially, Fig. 5 presents the results when weight adaptation is not considered ($\delta^{La} = 0$). In Edge Churn, the most trusted paths and the geodesic paths decrease for both local and global algorithms. However, the global algorithm leads to higher reduction as far as the number of hops of the most trusted paths is concerned, due to the more complete network knowledge it has available. In the case of the global algorithm the new link $i \to k$ attains the value v_{ik} of the most trusted path between i, k, while in the local algorithm $v_{ik} = \max_h(w_{ih}w_{hk})$ or $v_{ik} = \max_{h,g}(w_{ih}w_{hg}w_{gk})$ which is not surely the highest value of trust between i and k. For the global algorithm, the semiring path algorithm will choose either the multihop most trusted path or the new shortcut in computing the path between two nodes. Both of them have the same trust value, and thus the path algorithm does not ensure the selection of the one hop shortcut, so that eventually it cannot achieve the largest reduction in the mean most trusted path in hops. For the local algorithm, the value v_{ik} may not be equal to $D_{i\to k}^t$ and thus not selected

Fig. 5 Performance of the proposed framework—comparison of local and global algorithms for the Edge Churn and combined mechanism (without weight adaptation $\delta^{La} = 0$)



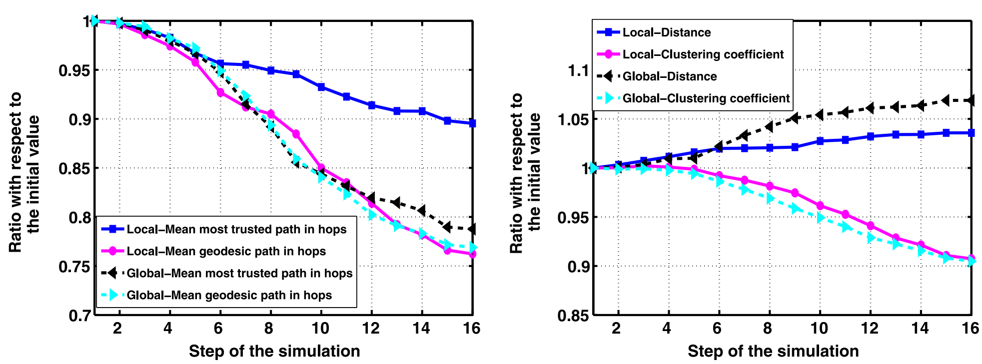
(a) Edge Churn (geodesic and most trusted paths)

(b) Edge Churn (distance and clustering coefficient)

(c) Combined Mechanism (geodesic and most trusted paths)

(d) Combined Mechanism (distance and clustering coefficient)

Fig. 6 Performance of the proposed framework—comparison of local and global algorithms for the Edge churn (with weight adaptation $\delta^{La} = 0.1$)



(a) Edge Churn (geodesic and most trusted paths)

(b) Edge Churn (distance and clustering coefficient)

by the semiring based algorithm. The clustering coefficient is reduced due to the addition of long links which reduce the proportion of triangles to triplets (if one considers the corresponding definition of clustering coefficient). Also, the mean distance remains stable, as we do not perform weight adaptation.

The Combined mechanism exploits the characteristics of both the Edge Churn and Node Churn mechanisms. The new nodes are considered as trusted. The clustering coefficient is approximately stable, the distance is increased, and the influence of the locality in the local algorithm is reduced due to the Node Churn mechanism. Thus, the geodesic paths are reduced, and the number of hops of the most trusted paths is also reduced with a percentage close to this of the geodesic paths. As it can be observed, for this choice of parameters, the results of the local algorithm are close to those of the global algorithm leading to the choice of the local algorithm for less overhead.

Figure 6 shows that a low value of $\delta^{La} = 0.1$ makes the improvements more evident, especially for Edge Churn.

Specifically, for the global algorithm it distinguishes the new link from the corresponding multihop path with the same trust value, while for the local algorithm the increase in v_{ik} will probably render the weight of the long link greater than $D^t_{i\to k}$, helping the path algorithm to select the new shortcut. The distance is increased as expected, due to the increment of the weights of the long-range links and both the local and the global algorithms lead to higher reductions in the number of hops of the most trusted paths. The mean geodesic path does not change so much with regard to the case of $\delta^{La}=0$, as expected. The same observations hold for the combined mechanism as well if a small value of $\delta^{La}=0.1$ is used (the simulation results are omitted due to space limitations), but in a more restricted degree, meaning that the combined mechanism works equivalently for both the local and global algorithms and with or not weight adaptation. For the Edge Churn mechanism, the global algorithm or the local one with weight adaptation give good results. However, the Combined mechanism is highly influenced by the added trusted nodes. When the newly added nodes are not trusted or Node Churn is performed with lower probability than Edge Churn, the behavior of combined mechanism is more influenced from the Edge Churn mechanism, i.e., the local and global algorithms behave differently as it happens in Edge Churn.

5 Application 2: content distribution in wireless multihop networks

Content distribution is the main reason for building robust and efficient networking infrastructures. The traffic distribution can be accurately expressed through link weights in directed graphs, the latter representing the cost of communication for a node i to communicate with node j. Consequently, in this section, we adapt the topology modification framework for improving network structure in order to better serve content distribution in wireless multihop networks. The adaptation of the framework is different than the case of trust management, however, it remains simple and efficient as it will be shown in the following.

5.1 Framework specification and operation

Assuming the same network model as the one described earlier, we aim at infusing social properties in the traffic graph and mainly at reducing the mean hop distance among nodes. We consider the initial weights in the interval $[0, w_{\max}]$. Contrary to the interpretation of weights in Sect. 4, lower weight values are now preferable for communication. We define as traffic cost of a path, the addition of all weights along the path and as distance between two nodes the minimum traffic cost for all paths between them. We essentially employ the semiring (Sect. 2.1) $(R,\oplus,\otimes,0,1)=(R_+\cup\infty,\min,+,\infty,0)$.

A local and a global algorithm may be defined again, depending on the choice of $Q(k|i)$. Probability $Q^d(k|i)$ expresses preferential attachment with respect to w_{ik}:

$$Q^d(k|i)=\frac{w_{ik}}{\sum_{j\in\mathcal{N}^{\text{out}}(i)} w_{ij}} \tag{32}$$

A link with high communication cost (thus a less desirable link), is more probable to be deleted. In addition, we identify $\mathbb{P}^a_i$ with Π^{invs}_i, based on the observation that a node with low in-strength has low-cost incoming connections and is preferable to participate in the creation of shortcuts. Similarly, $\mathbb{P}^d_i$ is chosen as Π^{in}_i, so as to avoid nodes with costly in-links. A node with high probability Π^{in}_i, is not desired by its local neighbors, as the links leading to it are costly. Therefore, deletion impacts w.h.p. nonpopular nodes, leading to a smaller negative impact on the network flow. Also, $\mathbb{P}^{nd}_i=\Pi^{\text{out}}_i$, meaning that we delete nodes with probability proportional to their out-strength, i.e., nodes with costly connections.

In edge addition, a node i initiating the process, chooses w.h.p a node k in $\mathcal{A}^{R_{\max}}_{R_{\min}}(i)$ having high cost of communication, in order to replace such a costly connection with a less costly one. We define $Q(k|i)$ in two possible ways:

Local algorithm: Similarly to the corresponding algorithm regarding trust applications, node i selects node k with probability proportional to the sum of its 2 or 3-hop distances from i. The weights (costs) of the local links needed for the computation of this probability are obtained via requests of i to its local neighbors. Let h_2 be the nodes in $\mathcal{A}^{R_{\max}}_{R_{\min}}(i)$ that are 2-hops away from i and h_3 be the nodes 3-hops away from i.

- 2 hops: If k is 2-hops away from i, its probability of being selected to form the long link $i\to k$ is:

$$\begin{aligned}Q(k|i)=\sum_h(w_{ih}+w_{hk})&\\ \times\Bigg[\sum_{h,(m\in h_2)}(w_{ih}+w_{hm})&\\ +\sum_{h,g,(m\in h_3)}(w_{ih}+w_{hg}+w_{gm})\Bigg]^{-1}&\end{aligned} \tag{33}$$

$$v_{ik}=\min_h(w_{ih}+w_{hk}) \tag{34}$$

- 3 hops: In the case that k is 3-hops away from i, the probability becomes:

$$Q(k|i) = \sum_{h,g}(w_{ih} + w_{hg} + w_{gk}) \times \left[\sum_{h,(m\in h_2)} (w_{ih} + w_{hm}) + \sum_{h,g,(m\in h_3)} (w_{ih} + w_{hg} + w_{gm}) \right]^{-1} \quad (35)$$

$$v_{ik} = \min_{h,g}(w_{ih} + w_{hg} + w_{gk}) \quad (36)$$

Global algorithm: In this case, node i should compute the traffic semirings for all nodes in $\mathcal{A}_{R_{\min}}^{R_{\max}}(i)$ and form a network wide view of the least-cost path to k. The probability that $k \in \mathcal{A}_{R_{\min}}^{R_{\max}}(i)$ is selected from i is

$$Q(k|i) = \frac{D_{i\to k}^T}{\sum_{j\in\mathcal{A}_{R_{\min}}^{R_{\max}}(i)} D_{i\to j}^T} \quad (37)$$

$$v_{ik} = D_{i\to k}^T \quad (38)$$

where $D_{i\to k}^T$ is the distance in the traffic graph as explained above.

In both algorithms, node i chooses w.h.p. a node k for which $D_{i\to k}^T$ is of higher value, so as to replace it with another less costly one.

5.2 Evaluation and discussion

In the sequel, we present simulation results concerning the infusion of small-world properties in network traffic graphs. The parameters are the same as those used in Sect. 4, with the exception that in this case the δ parameters take different values. In all cases, $\delta^{La} < 0$ as the new link is desired to cost less than the replaced multihop path. For reasons noted in Sect. 4, $\delta^{La} \neq 0$. We consider two possibilities of determining the rest of parameters δ^{1a}, δ^{2a}, δ^{1d}, δ^{2d}, δ^{na}. According to the first, the cost of the link depends on the traffic passing through it and, therefore, $\delta^{1a} < 0$, $\delta^{2a} > 0$, $\delta^{1d} > 0$, $\delta^{2d} < 0$, $\delta^{na} > 0$. According to the second, where, $\delta^{1a} > 0$, $\delta^{2a} < 0$, $\delta^{1d} < 0$, $\delta^{2d} > 0$, $\delta^{na} < 0$, we adapt the cost of the links depending on which links we want the traffic to pass through. For example, when the link $i \to j$ is added, we increase the cost of the other out-links of i so as to let more traffic pass through the shortcut.

Figure 7 depicts simulation results for the Node Churn mechanism. We consider the new nodes as introducing lower cost in the network, and thus their out-links are assigned weights according to the $\beta(2,5)$ distribution. The hop count of the least cost paths and of the geodesic paths is reduced, while the distance is reduced, i.e., the corresponding cost is less.

Figure 8 presents the Edge Churn mechanism and Combined mechanism regarding the first consideration of the δ parameters. The Edge Churn mechanism behaves efficiently when the global algorithm is performed as both the number of hops of the mean least cost path and the distance are reduced more. The reason is similar to the case of Sect. 4, as the global algorithm finds with certainty the least cost path for a pair of nodes. The combined mechanism has a similar behavior for both algorithms leading to the choice of the local algorithm for less overhead in the network. However, it is possible for the Edge Churn mechanism to have a similar behavior with the Combined one, regarding local and global algorithms, i.e., by using higher absolute values for the δ parameters.

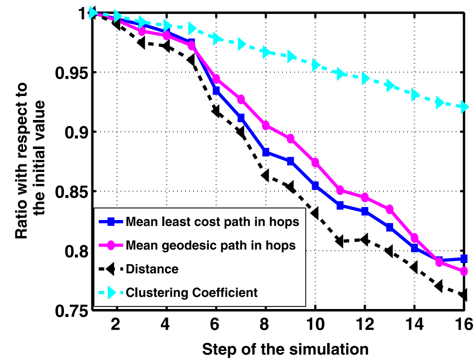


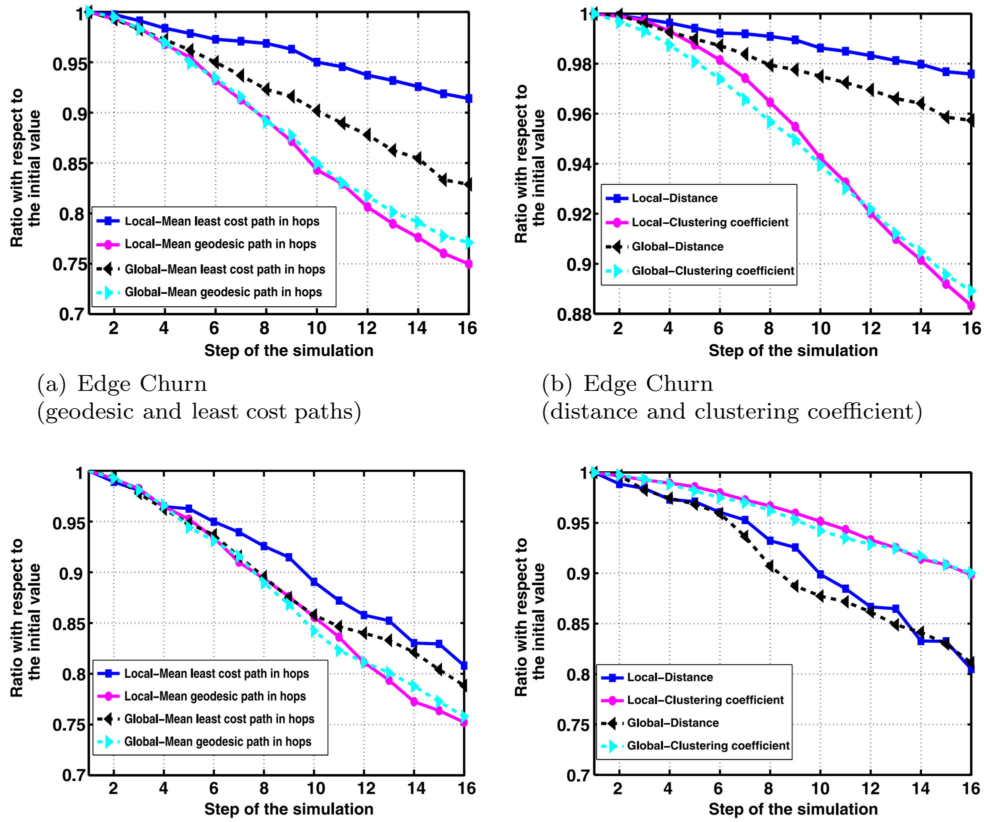
Fig. 7 Node Churn mechanism performance with $\beta(2,2)$ in in-links and $\beta(2,5)$ in out-links of the newly added nodes and $\delta_{na} = 0.08$

Finally, it should be noted that, as mentioned in Sect. 3.1.3, for WEC, the in-strength s_i^{in} can be derived from the superposition of (11) and (17), and the out-strength s_i^{out} from the superposition of (10) and (16). Since these equations are cumbersome to be solved analytically, we used numerical methods to obtain the corresponding solutions (i.e., Finite Differences). In Fig. 9, we verify that the results obtained through the numerical methods are very close to those obtained via simulations.

6 Conclusions

In this work, we presented a generic framework for reengineering the topology of wireless multihop networks following a top-down modification approach. The proposed approach exploits features of the higher protocol layers and mainly social structures for improving the physical topology, based on nominal power variations and inverse Topology Control capabilities of modern radio devices. We demonstrated the operation of the proposed methodology for two key application areas in wireless networks, namely content distribution and trust management and showed through analysis and simulations the obtained benefits and performance improvements. The framework describes the inherent trade-offs for each application area and allows exploiting the emerging control parameters for further improving

Fig. 8 Performance of the proposed framework—comparison of the local and global algorithms for the Edge Churn and combined mechanism ($\delta^{1a} = -0.08$, $\delta^{2a} = 0.08$, $\delta^{1d} = 0.08$, $\delta^{2d} = -0.08$, $\delta^{na} = 0.08$, $\delta^{La} = -0.2$)



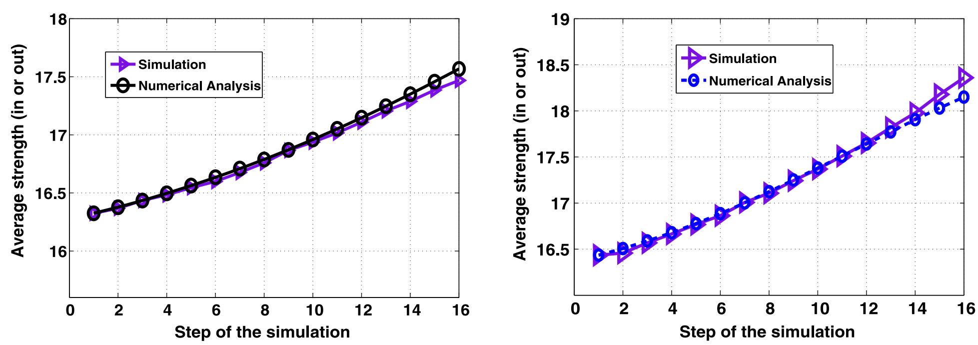
(a) Edge Churn (geodesic and least cost paths)

(b) Edge Churn (distance and clustering coefficient)

(c) Combined Mechanism (geodesic and most trusted paths)

(d) Combined Mechanism (distance and clustering coefficient)

Fig. 9 Comparison of numerical vs. simulations results for the WEC mechanism ($\delta^{1a} = -0.08$, $\delta^{2a} = 0.08$, $\delta^{1d} = 0.08$, $\delta^{2d} = -0.08$, $\delta^{na} = 0.08$, $\delta^{La} = -0.2$)



(a) Global Algorithm: Average Strength

(b) Local Algorithm: Average Strength

the topology and operation of networks of the future. In that sense, optimizing the control parameters of the proposed framework is a promising direction for automating topology adaptation and making it more efficient.

References

1. Lewis TG (2009) Network science: theory and practice. Wiley, Hoboken
2. Churchill EF, Halverson CA (2005) Social networks and social networking. IEEE Internet Comput Mag, 14–19
3. Albert R, Barabasi A-L (2002) Statistical mechanics of complex networks. Rev Mod Phys 74(1):47–97
4. Newman MEJ (2003) The structure and function of complex networks. SIAM rev 45(2):167–256
5. Honneth A, Joans H (1988) Social action and human nature. Cambridge University Press, Cambridge
6. Santi P (2005) Topology control in wireless ad hoc and sensor networks. ACM Comput Surv 37:164–194
7. Olfati-Saber R (2005) Ultrafast consensus in small-world networks. In: Proc of American control conference, vol 4, pp 2371–2378
8. Ye X, Xu L, Lin L (2008) Small-world model based topology optimization in wireless sensor networks. In: Proc of international symposium on information science and engineering, pp 102–106
9. Afifi N, Chung K-S (2008) Small world wireless mesh networks. In: Proc of international conference on innovations in information technology (IIT), pp 500–504

10. Luo X, Yu H (2010) Constructing wireless sensor network model based on small world concept. In: Proc of international conference on advanced computer theory and engineering (ICACTE), pp 501–505
11. Reznik A, Kulkarni SR, Verdu S (2004) A 'small-world' approach to heterogeneous networks. Commun Inf Syst 4(4):325–348
12. Chitradurga R, Helmy A (2004) Analysis of wired short-cuts in wireless sensor networks. In: Proc of IEEE/ACS int'l conference on pervasive services, pp 167–177
13. Sharma G, Mazumdar RR (2008) A case for hybrid sensor networks. IEEE/ACM Trans Netw 16(5):1121–1132
14. Helmy A (2003) Small worlds in wireless networks. IEEE Commun Lett 7(10):490–492
15. Stai E, Karyotis V, Papavassiliou S (2010) Socially-inspired topology improvements in wireless multi-hop networks. In: Proc of IEEE international conference on communications (ICC) workshop on social networks (SOCNETS)
16. Stai E, Karyotis V, Papavassiliou S (2010) Enhanced service provisioning in wireless multi-hop networks via socially-driven inverse topology control. In: Proc of IEEE Globecom 2010 workshop on enabling the future service-oriented Internet (EFSOI)
17. Barrat A, Barthelemy M, Vespignani A (2004) Weighted evolving networks: coupling topology and weight dynamics. Phys Rev Lett 92(22)
18. Wang W, Wang B, Hu B, Yan G, Ou Q (2005) General dynamics of topology and traffic on weighted technological networks. Physical Review Letters 94(18)
19. Barrat A, Barthelemy M, Vespignani A (2004) Traffic-driven model of the World Wide Web graph. Lecture Notes in Computer Science, vol 3243, pp 56–67
20. Penrose M (2003) Random geometric graphs. Oxford University Press, New York
21. Opsahl T, Agneessens F, Skvoretz J (2010) Node centrality in weighted networks: generalizing degree and shortest paths. Soc Netw 32(3):245–251
22. Barrat A, Barthelemy M, Vespignani A (2008) Modeling the evolution of weighted networks. Phys Rev Lett 70(6)
23. Mohri M (2002) Semiring frameworks and algorithms for shortest-distance problems. J Autom Lang Comb 7(3):321–350
24. Kamvar SD, Schlosser MT, Garcia-Molina H (2003) The EigenTrust algorithm for reputation management in P2P networks. In: Proc of the 12th int'l WWW conference
25. Theodorakopoulos G, Baras JS (2006) On trust models and trust evaluation metrics for ad hoc networks. IEEE J Sel Areas Commun 24(2):318–328
26. Fagiolo G (2007) Clustering in complex directed networks. Phys Rev Lett 76(2)
27. Chlamtac I, Conti M, Liu JJ-N (2003) Mobile ad hoc networking: imperatives and challenges. Ad Hoc Netw 1(1):13–64