Static and Dynamic Properties of Small-world Connection Topologies Based on Transit-stub Networks

Carlos Aguirre[*]
Fernando Corbacho
Ramón Huerta[†]

Computer Engineering Department,
Universidad Autónoma de Madrid,
28049 Madrid, Spain

Many real complex networks are believed to belong to a class called small-world (SW) networks. SW networks are graphs with high local clustering and small distances between nodes. A standard approach to constructing SW networks consists of varying the probability of rewiring each edge on a regular graph. As the initial substrate for the regular graph some specific topologies are usually selected such as ring-lattices or grids. However, these regular graphs are not suitable for modeling certain hierarchical topologies. A new regular substrate is proposed in this paper. The proposed substrate resembles topologies with certain hierarchical properties more accurately. Then, different dynamics inspired by networking protocols are used to characterize dynamical properties of a network. Measuring transmission times and error rates lead us to consider networks with SW features as the most reliable and fastest, regardless of the routing policies.

1. Introduction

In the past few years, growth in the field of communication networks [2, 3, 4], multi-agent systems [5], architectures [7], and complex systems have generated many studies focused on network topologies. In this framework graphs represent the most adequate abstract representation of a network, where each node represents a router or host and each edge represents a connection between routers or hosts. There are many investigations that deal with graphs that model networks [8, 9, 10]. A common feature in all these studies is the set of graphs subjected to study such as star, ring, grid, regular, and random networks. Recently the use of transit-stub networks [11] has been proposed as an improvement for

[*] Electronic mail address: Carlos.Aguirre@ii.uam.es.

[†] Also Institute for Nonlinear Science, University of California, San Diego, La Jolla, CA, 92093-0402.

modeling real communication networks. Transit-stub networks capture most of the relevant topological characteristics in terms of the following metrics: diameter, average degree, average path length, number of connected components, and number of biconnected components.

Many interesting properties arise when the network topology itself is an intrinsic parameter that can be modified using some specific rules. These methods allow slowly varying the metrics of the graphs in such a way that the dynamical behavior of any given problem can be carefully analyzed. In [13] a method to study the dynamic behavior of networks when the network is shifted from a regular, ordered network to a random one is proposed. The method is based on random rewiring with a fixed probability p for every edge in the graph. We obtain the original regular graph for $p = 0$, and a random graph for $p = 1$. This method shows that the characteristic path length (i.e., the average distance between nodes measured as the minimal path length between them) decreases with increasing values of p much more rapidly than does the clustering coefficient (i.e., the average number of neighbors of each node that are neighbors between them). It was found that there is a range of values for p where paths are short but the graph is highly clustered. The topologies in this range are known as small-world (SW) topologies. SW topologies present some very interesting features that make them suitable for efficient transmission of commodities [13, 14, 15]. Moreover, they appear in many real life networks [16, 17], the World Wide Web [18], as a result of natural evolution [19], or as result of a learning process [20]. In [6], for example, it is shown that on SW networks coherent oscillations and temporal coding can coexist in synergy in a fast time scale on a set of coupled neurons.

As the initial substrate for the generation of SW, the use of a ring-lattice or a grid is usually proposed [14, 21]. They are used because these graphs are connected, present a good transition from regular to random, and there are no specific nodes on them. The substrate selected in [13] presents a single biconnected component, independent of the number of nodes in the graph and does not present a hierarchical structure. In many networks there are special nodes that connect backbones with subnetworks. Furthermore, in [11], the graphs generated are organized following a clear hierarchical structure and present a high number of biconnected components. Formally, a biconnected component is a maximal set of edges such that any two edges in the set lie on a common single cycle. Intuitively, a biconnected component is a subnetwork connected by a set of outputs to the network. The number of biconnected components in the stub-domain graphs grows with the number of nodes in the graph. Realistic models of network topologies are necessary in order to obtain correct behavior for most kinds of algorithms and policies over these hierarchical networks [12]. These models and methods should also scale to increasingly larger networks because communication networks continue to grow in size and importance. Our objective is to construct regular graphs that can resemble the topology of hierarchical networks but are able to shift from regular to random in the same way ring-lattices do. We also want to study some well known routing policies over this new kind of regular graph.

In this work we present a theorem of existence along with a method for building regular graphs with a high number of biconnected components. In these graphs the number of biconnected components is a function of the number of nodes in the graph and the number of neighbors of each node. We provide an analytic expression for the number of biconnected components, the characteristic path length, and the cluster coefficient. Next, we investigate the behavior of some networking inspired dynamics using this transit-stub based network. In particular, we analyze by means of intensive computer simulations the dynamical behavior of information broadcasting, multicasting, and unicasting on a set of graphs ranging from transit-stub regular to random.

The structure of the article is as follows. First, we show the existence of regular graphs with several biconnected components and present an algorithm for building this type of graph. Additionally we provide an analytic expression for the characteristic path length and cluster coefficient for this type of graph. Second, we study several static parameters for this kind of graph and investigate topological metrics when the graph is shifted from a regular to a random situation. Then, we compare these metrics with the results obtained by the regular-lattice substrate. And finally, we analyze by means of computer simulations the behavior of information broadcast, multicast, and unicast over both types of substrates to determine those graphs that achieve minimum transmission error and delivery delay.

2. Biconnected small-worlds

This section describes an algorithm to generate a SW graph with a higher number of biconnected components than do lattices. In the literature about SW, the ring-lattice is used as the initial graph substrate due to the following advantages.

- Ring substrates are connected.
- They have a good transition from "large" to "small."
- They have no special nodes such as trees or stars do.

The ring-lattice substrate has some characteristics that make it unsuitable for modeling some specific types of networks such as hierarchical multi-agent networks or the Internet. These networks have special nodes, the ones that connect stubs with backbones, and present a number of biconnected components greater than one. However, the number of biconnected components for any graph obtained from the ring-lattice substrate by the Watts–Strogatz method is close to one.

In this section we develop a constructive method for building a finite regular graph with multiple biconnected components that we call a "transit-stub regular graph." This graph is the initial substrate for our SW model of hierarchical networks. It is not clear *a priori* that such graphs exist and, if they exist, it needs to be proved that they can be built algorithmically for given values of n and k. For a given graph G, we use n, $|G|$ as the number of nodes in the graph, k_i as the number of neighbors that node i has, and we use k to denote $\sum_{i=1}^{n} k_i/n$, that is, the average number of neighbors for each node. Remember that these graphs must be connected and have a good transition from large to small in the same way that ring-lattice substrates have.

Here we present a theorem that establishes the existence of regular finite graphs with several biconnected components. The theorem is constructive, that is, it proves existence by building such a graph. The algorithm builds a regular ring-lattice and then attaches an almost-regular stub to every node in the central ring. This algorithm produces a k-regular graph with n nodes and a number of biconnected components higher than one. We show that for fixed k the number of biconnected components grows linearly with n. The pseudocode of the algorithm is given in appendix A. We now state our theorem.

Theorem 1. Provided that k is odd and $(k + 3)$ divides n, there is a regular graph with n nodes, k neighbors per node, and $2n/(k + 3) + 1$ biconnected components.

Proof. The proof is constructive. First we build a central ring-lattice with $n/k + 3$ nodes and $k - 1$ neighbors per node. Then we build a subgraph for each node in the ring-lattice with $k + 2$ nodes where each node connects to k neighbors, except for one of them, which connects only to $k - 1$ neighbors (this is possible since k is odd, and therefore $k + 2$ is also odd). This last node connects to its corresponding node in the ring-lattice. We call this special node (i.e., the node in the subgraph) the *stub node*. ■

For convenience, we also call the subgraph joint to the node in the ring-lattice the *stub graph*, such that every node in the ring-lattice belongs to one single stub graph. The previous algorithm generates a regular graph with n nodes and k neighbors per node. The number of biconnected components can be calculated by observing that the central ring is one biconnected component. Every stub generates a new biconnected component since it is only connected by a single node to the central ring. The edges that connect the stubs to the central ring generate another biconnected component. Finally, $2n/(k + 3) + 1$ biconnected components exist in the graph.

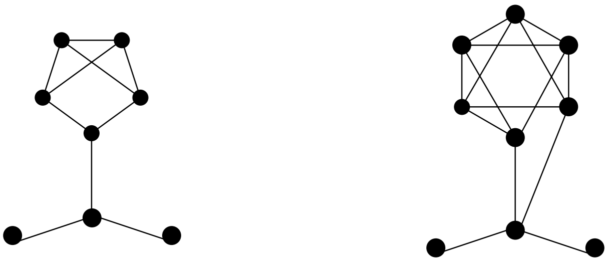

Figure 1. Nodes of stub graphs for $k = 3$ left, and $k = 4$ right. Note that, by our definition, the two nodes at the extremes of each base do not belong to the stub graph but are depicted here for clarity.

A similar theorem can be established when k is even.

Theorem 2. Provided that k is even and $(k + 3)$ divides n, there is a regular graph with n nodes, k neighbors per node, and $n/(k + 3) + 1$ biconnected components.

Proof. The proof is again constructive. First we build a central ring-lattice with $n/k + 3$ nodes and $k - 2$ neighbors per node. Then we build a subgraph for every node in the ring-lattice with $k + 2$ nodes where each node connects to k neighbors, except for two of them, which connect only to $k - 1$ neighbors (this is possible since k is even, and therefore $k + 2$ is also even). These nodes connect to their corresponding node in the ring-lattice. ■

Figure 1 shows how the stub graphs are built for k even and odd.

3. Characteristic path length and cluster coefficient scaling

In this section we give analytical expressions for both the cluster coefficient and the characteristic path length for regular transit-stub graphs.

3.1 Cluster coefficient

Intuitively the cluster coefficient is the average number of neighbors of each node that are neighbors between them. More precisely, for a vertex v let us define $\Gamma(v)$ as the subgraph composed by the neighbors of v (without v itself). Let us define the cluster coefficient for a given node v as:

$$C_v = \frac{|E(\Gamma(v))|}{\frac{1}{2}k_v(k_v - 1)}, \tag{1}$$

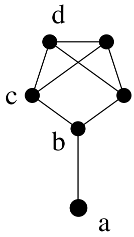

Figure 2. Distribution of nodes in a stub for $k = 3$.

where $|E(\Gamma(v))|$ is the number of edges in the neighborhood of v. The cluster coefficient for a given graph G can now be defined as

$$C = \frac{\sum_{i=1}^{n} C_i}{n}. \tag{2}$$

In order to obtain an expression for the cluster coefficient for regular odd transit-stub graphs, let us observe that there are four types of nodes in each stub. These are distributed in the following way (as shown in Figure 2).

- One node in the central ring, node "a."
- One stub node in each stub, node "b."
- $k-1$ nodes in each stub connected with the stub node, node "c."
- Two nodes in each stub not connected with the stub node, node "d."

We count the nodes by type to obtain

$$\begin{aligned} C_a &= \frac{3}{4}\frac{k-3}{k}, \quad C_b = \frac{k-3}{k}, \\ C_c &= \frac{(k-3)(k+1)+2}{(k)(k-1)}, \quad C_d = \frac{k-1}{k}. \end{aligned} \tag{3}$$

Then we weight them by averaging over all vertices in the stub:

$$\begin{aligned} C = \frac{1}{k+3}\left(C_a + C_b + C_c + C_d\right) = \frac{1}{k+3} \\ \cdot\left(\frac{3}{4}\frac{k-3}{k} + \frac{k-3}{k} + (k-1)\frac{(k-3)(k+1)+2}{(k)(k-1)} + 2\frac{k-1}{k}\right), \end{aligned} \tag{4}$$

which can be simplified to

$$C = 1 - \frac{1}{4}\left(\frac{5\,k+33}{k^2+3k}\right). \tag{5}$$

Note that the transit-stub graphs differ from the most clustered ones only by an amount $O(1/k)$. Furthermore, when $n \to \infty$, k can be made arbitrarily large without violating the sparseness condition ($n \gg k$). This makes a clear difference with respect to ring-lattices, where

$$C = \frac{3}{4}\frac{(k-2)}{(k-1)}.$$

In this case, we have that $C \to 3/4$ as $k \to \infty$, which will never reach the optimal value of the cluster coefficient.

3.2 Characteristic path length

The characteristic path length of a graph indicates how far the nodes are from each other. For a vertex v let us define its characteristic length as

$$L(v) = \frac{\sum_{i=1}^{n} d(v,i)}{n}, \tag{6}$$

where $d(v,i)$ indicates the length of the shortest path connecting v and i. Using $L(v)$ we define the characteristic length over a graph as

$$L = \frac{\sum_{i=1}^{n} L(i)}{n}. \tag{7}$$

In order to estimate the characteristic path length of a transit-stub regular graph we calculate the average distance between the nodes in the same stub d_l, the average distance between nodes in different stubs d_g, and two length scales: L_l as the characteristic path length for nodes in the same stub, and L_g as the characteristic path length between stubs. This method for estimating the characteristic path length of a regular graph is similar to the method followed in [21].

In each stub we have $(k+3)(k+2)/2$ possible connections organized as follows.

- Two of them with a distance of 3 (these correspond to connections between a and d nodes).
- There are $k+1$ pairs of nodes with a distance of 2 (a, c and d, b pairs).
- There are $k(k+3)/2$ pairs of nodes with a distance of 1.

Hence:

$$\begin{aligned} d_l = L_l &= \frac{2}{(k+2)(k+3)}\left[2 \cdot 3 + 2\,(k+1) + \frac{k(k+3)}{2}\right] \\ &= 1 + \frac{2(k+5)}{(k+2)(k+3)}. \end{aligned} \tag{8}$$

We can see that $d_l \to 1$ when $k \gg 1$. If we think of each stub graph as a super-vertex of the central ring, the average distance between the nodes of different stubs d_g is determined by L_g and L_l. In [21] it is shown that a regular ring-lattice with n nodes and k neighbors has the following value:

$$L = \frac{n(n+k-2)}{2k(n-1)}. \tag{9}$$

In the transit-stub graph the central ring that connects the stubs is a regular ring-lattice with $n_r = n/(k+3)$ and $k_r = k-1$ where n_r and k_r represent the values of n and k for the central ring. Hence, we have

$$L_g = \frac{\frac{n}{k+3}\left(\frac{n}{k+3} + (k-1) - 2\right)}{2(k-1)\left(\frac{n}{k+3} - 1\right)} = \frac{n\left(\frac{n}{k+3} + k - 3\right)}{2(k-1)(n-k-3)}. \tag{10}$$

A path from a node v in one stub to a node u in another stub consists of three components: the edges contained in the stub with v, the edges in the central ring, and the edges in the stub with u. We add up these terms to obtain

$$d_g = 2L_l + L_g = 2 + \frac{4(k+5)}{(k+2)(k+3)} + \frac{\frac{n}{k+3}\left(\frac{n}{k+3} + (k-1) - 2\right)}{2(k-1)\left(\frac{n}{k+3} - 1\right)} \tag{11}$$

that, for $n \gg k \gg 1$, we have

$$d_g \approx \frac{1}{2k}\left(\frac{n}{k+3} + k\right).$$

There are $N_l = n(k+2)/2$ pairs of nodes that are in the same stub and $N_g = n(n-k-3)/2$ pairs of nodes that are in different stubs. We average them to calculate

$$\begin{aligned} L &= \frac{2}{n(n+1)}(N_l \cdot d_l + N_g \cdot d_g) \\ &\approx \frac{k+2}{n+1} + \frac{(n-k-3)\left(\frac{n}{k+3} + k\right)}{2k(n+1)} \approx \frac{1}{2k}\left(\frac{n}{k+3} + k\right) \end{aligned} \tag{12}$$

where we assume that $n \gg k \gg 1$, that is, sparseness. Note that this transit-stub network has a shorter characteristic path length than the characteristic path length of ring-lattices.

In a random graph with fixed k it can be shown that L scales as $\log(n)$ and C scales to 0 as n tends to infinity [27]. This means that the transit-stub graphs have a different scaling regime than the random graph model. This makes us expect that at some point, when we shift from these regular models to a random graph, there must be a phase change both in the value of L and C. If this phase change is produced at different values of p for L and C we can build SW graphs using this regular substrate, and therefore, SW graphs exist for these models. Furthermore, if the value of p that changes the phase of L is small, the SW graphs generated from regular transit-stub graphs will have a high number of biconnected components. This is because each rewired edge will decrease the number of biconnected components at most by one. This means that we need to rewire a minimum of $2n/(k+3)+1$ edges for k odd and $n/(k+3)+1$ edges for k even in order to reduce the number of biconnected components to one.

4. Static metric behavior

In order to study the static metric behavior of our graphs, we follow a modified version of the standard procedure described in [13]. Our method consists of rewiring iteratively every edge with probability p. Once an edge is evaluated, it is not reconsidered. Other procedures for building SW graphs have been described in [22, 23, 24]. However these methods increase the number of edges and/or nodes in the graph or can only be applied to ring-lattice substrates. Our method maintains a constant number of nodes and edges and can be applied to any kind of substrate.

Figure 3 displays the process of shifting a regular transit-stub graph to random. When $p = 0$ no changes are made over the substrate and the graph remains unchanged. As p increases shortcuts appear in the graph. In the limit of $p = 1$ a random graph is fully developed.

The metrics used to classify the graphs are the characteristic path length L, the characteristic cluster C [21], the number of biconnected components B [25], and the average edge euclidean length D. These parameters are the most extensively studied in the existing literature. The diameter of the graph is tightly related to L (being in fact an upper bound) [27]. Therefore we consider L because it provides more specific information about the distance of the nodes in the graph. In the same way, the length diameter [11] is an upper bound of D.

In Figure 4 we show the behavior of L and C in a ring-lattice and in a regular transit-stub graph for both k even and odd when they are shifted

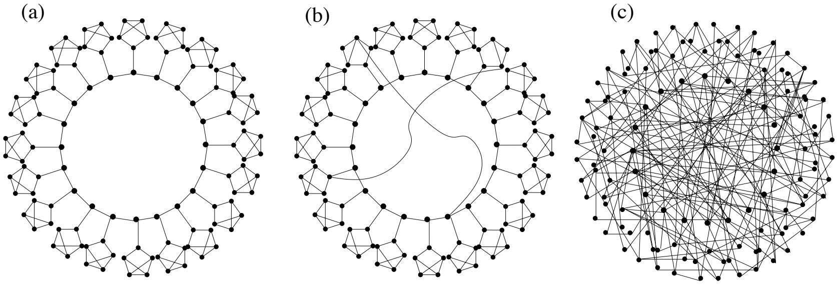

Figure 3. Transit-stub graphs for $n = 120$ and $k = 3$. (a) Regular transit-stub graph for $p = 0$. (b) SW transit-stub graph for $p = 0.02$. (c) Random graph for $p = 1$.

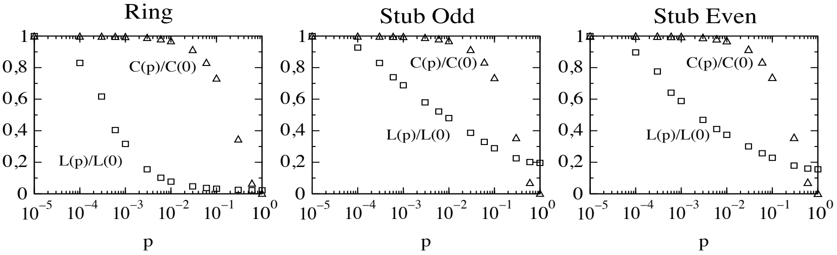

Figure 4. L (squares) and C (triangles) for ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$.

from a regular to a random situation. Note that all of them present a clearly visible SW area (i.e., low L and high C). However the decrease of L in the transit-stub models is smoother because the characteristic paths are shorter for regular transit-stub substrates than for ring-lattice ones. In the ring-lattice substrate the value of the characteristic path L is very close to the maximum for any regular graph and has a very fast descent. On the other hand, when we calculate the characteristic cluster C we do not find significant differences between these types of substrates.

The number of biconnected components in a graph is a useful measure of "connectedness" or "edge redundancy." Formally speaking a biconnected component is a set of edges such that any two edges in the set lie on a common single cycle. A biconnected component cannot be disconnected by removing a single node or a single edge. Furthermore, each pair of nodes in a biconnected component is connected by at least two paths. Intuitively a biconnected component is a subgraph connected to the rest of the graph by articulation nodes (see [25] for more information about biconnectivity). Figure 5 displays the number of biconnected components in the graph as a function of p. Observe that the number of biconnected components is reduced to a very low number as p increases in an initially regular transit-stub graph substrate. However it remains almost constant when a ring-lattice graph is employed. Therefore, we can see that the main difference between the regular and the transit-stub graph is found in the number of biconnected components. Notice also that in both models, there is a small increase in the number of biconnected components in the random graph area.

Finally, for a given embedding Φ of a graph G into an euclidean space $\mathcal{L}$ we can assign a length to each edge in the graph. This length is the euclidean distance between the two vertices connected by the edge. As the graph becomes more random, the edges get longer in the euclidean space. In Figure 6 we plot the average length D of the edges for both

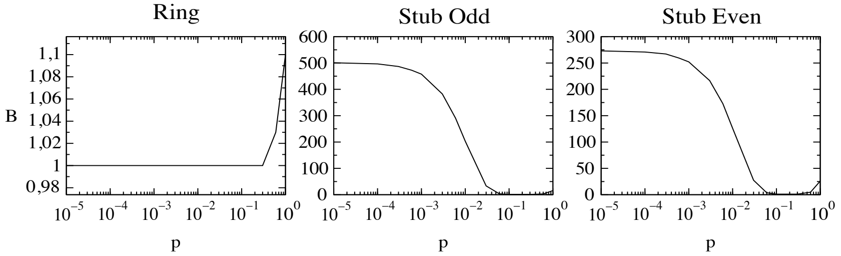

Figure 5. Number of biconnected components for ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$ (the average of 100 experiments is plotted).

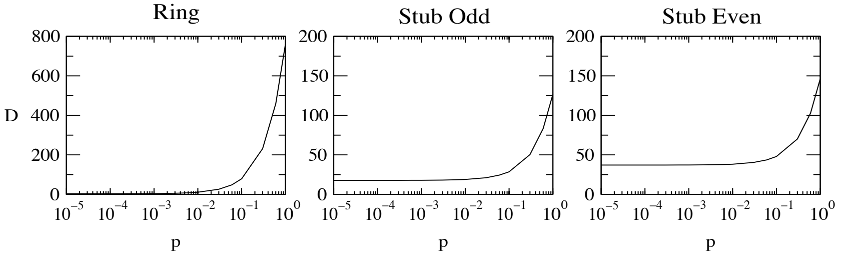

Figure 6. Average euclidean edge length for ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$. The nodes of the ring-lattice lie over a circumference with 600 units of radium. In the case of the transit-stub graphs the backbone nodes are over a circumference of 578 units of radium, each stub has 1 unit of radium and is 20 units away from its corresponding node in the backbone (the average of 100 experiments is plotted).

kinds of substrates when they are shifted from regular to random, that is,

$$D_\Phi = \frac{\sum_{i=1}^{|E|} d_\Phi(i)}{|E|}, \tag{13}$$

where $d_\Phi(i)$ is the euclidean length of edge i. The embeddings selected are such that the nodes of the ring-lattice lie over a circumference with 600 units of radium. In the case of the transit-stub graphs the backbone nodes are over a circumference of 578 units of radium, each stub has 1 unit of radium and is a distance of 20 units from its corresponding node in the backbone. Notice that both substrates behave in a similar way.

We can now define the euclidean path length $H_\Phi(v)$ for a vertex v as

$$H_\Phi(v) = \frac{\sum_{i=1}^{n} H_\Phi(v,i)}{n}, \tag{14}$$

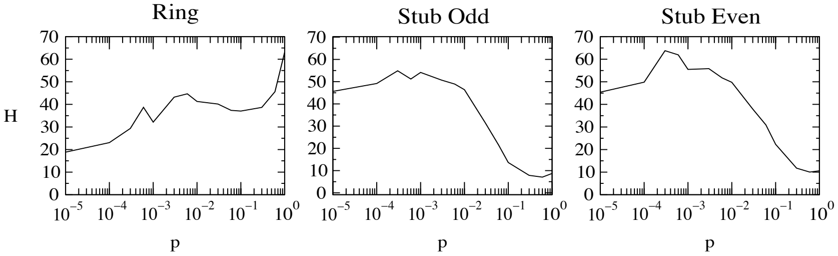

Figure 7. Normalized characteristic euclidean path length for ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$. The circumferences and distances are the same as Figure 6 (the average of 100 experiments is plotted).

where $H_\Phi(v,i)$ is the total euclidean length of the edges that lie in the shortest path between vertices v and i with the embedding Φ. Now, the characteristic euclidean path length for a given graph G and a euclidean embedding Φ can be defined as

$$H_\Phi(G) = \frac{\sum_{i=1}^{n} H_\Phi(i)}{n}. \tag{15}$$

The average euclidean path length of a graph is a measure of how far the nodes are in terms of the euclidean distance, instead of the number of hops.

In contrast with the average edge euclidean length, observe in Figure 7 the different qualitative behavior in the characteristic euclidean path length for both kinds of substrate. The ring-lattices display an average euclidean path length with a "plateau" in the SW area. The highest value for the average euclidean path length is reached in the regular area, and the lowest value is reached in the random area. For transit-stub graphs, the minimum value is reached near the random area ($p \approx 1$ but $p < 1$), meanwhile the highest euclidean path length is observed in the SW area. The embeddings considered are the same utilized in the calculus of H.

This measure is important when the cost of the network is considered as, for example [14, 15], or when the error rate in the transmission of information packages depends on the distance from source to destination. If each edge has a probability which reflects how prone it is to transmission error, longer links tend to produce more transmission errors. In our model, the probability of error in an edge is linearly proportional to its length. This is due to the fact that if p is the probability of error by unit of length, the probability p_m of error in an edge of length m units of length is given by:

$$p_m = 1 - (1-p)^m \approx mp \text{ if } p \ll 1. \tag{16}$$

From the definition of H it is clear that the average number of corrupted packages due to transmission errors should be proportional to H when the routing protocol transports packages from a source node to a destination node using the shortest path between both nodes.

5. Dynamic metrics

We consider two metrics for evaluating the efficiency of information transmission over the network. One of the them is the error rate, which may determine the degree of redundancy to include in messages. The second one is the transmission time. Forecasting transmission times of the Internet has been studied using neural networks and ergodic techniques [31]. As pointed out in [13, 14], transmission times should decrease with the graph characteristic path length.

We define the transfer time as the average time a packet takes to reach the target starting from a given source node in several experiments. We compute it by the number of time steps it takes to deliver a set of packets to the given set of target agents in the network averaged by the number of nodes in the target set and the number of packets sent to each agent, that is,

$$T = \frac{t}{l(n-1)}, \tag{17}$$

where t is the number of time steps until the system stops, l is the number of packets, and n is the number of nodes in the target set. The error rate increments with the length of the edges and with the characteristic path length of the graph.

The other metric we compute is the average error rate that we define as the number of packets that become corrupted on their way through the network. We use the Hamming distance between the original packages and the packages at each node:

$$E = \frac{\sum_{i=2}^{n} \sum_{v=1}^{l} f(H(p_{0v}, p_{iv}))}{l(n-1)}, \tag{18}$$

where p_{iv} is the package v at node i, H is the usual Hamming distance, and $f(x)$ is the step function

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{otherwise.} \end{cases} \tag{19}$$

The node 0 is assumed to be the source. The error rate would depend essentially on the characteristic path length and on the number of long-range connections present in the graph.

To better understand the balance between time delay and error rate we arbitrarily propose analyzing an average of both normalized measures:

$$P = \frac{1}{2}\left(\frac{T - T_{\min}}{T_{\max} - T_{\min}} + \frac{E - E_{\min}}{E_{\max} - E_{\min}}\right), \tag{20}$$

where $T_{\min}$, $T_{\max}$ are the minimum and maximum values of T and $E_{\min}$, $E_{\max}$ are the minimum and maximum values of E.

6. Dynamical simulations over the networks

Comparisons based on topological metrics provide differences between graph types. However, it does not mean much unless a link to a function is established. Therefore, we study how different static topologies affect performance in some networking inspired dynamics, such as broadcast, multicast, and unicast of information packages.

As pointed out in [1], performance parameters associated with a telecommunication network warrants a meaningful model for the complexity of the communication system. A relevant modeling of such complexity should include stochasticity of interacting resources and in the flow of information between the nodes of the network. Furthermore, some of these parameters are fuzzy in nature [26]. An accurate modeling of the Internet, or even smaller communications systems, is not an easy task and is out of the scope of this work due to the high heterogeneity of the elements present in the network and the high number of variables that affect the performance of a modern communication network.

In our model, a set of qualitative properties found in communication networks has been added to each of the agents in the network. In fact, all sorts of additional information about the network can be added to the topological structure by associating information with the nodes and edges. For instance, nodes may be assigned numbers representing their buffer capacity, that is, the number of packages that they can hold in the queue (including a stop symbol). The queue has a saturation limit beyond which new incoming packages are discarded. An edge may also have values of several types; including costs, such as the propagation delay on the link; and constraints, such as the bandwidth of the link. In our experiments most of the parameters were constant between the different networking problems (queue sizes, number of packages per file, lifetime of each package, etc.). This means that some of the parameters have no influence on the result of the experiments for a particular problem but does allow us to maintain some consistency between experiments.

For our simulations we assume that the substrates are embedded in the usual two-dimensional euclidean space. The nodes of the ring-lattice substrate lie over a circumference with 600 units of radium. In the case of the transit-stub graphs we assume that the backbone nodes are over a circumference of 578 units of radium. Each stub has 1 unit of radium and is 20 units away from its corresponding node in the backbone. These distances (but not the topology) correspond with the measures of the Spanish Research network RedIris [33]. We also assume that the probability of corrupting a package per unit length is 10^{-5}.

6.1 Simulations based on broadcasting

This section presents a dynamics inspired by simplified broadcasting using the selective diffusion algorithm described in [32] to understand the dynamic advantages of different substrate-based network topologies. Broadcast is the process of sending a package or a set of packages to all possible destinations in the network. Broadcast is a costly process in terms of the amount of packages that are introduced in the network. Moreover, broadcast is frequently used both as stand-alone processes (e.g., military networks, live TV and radio broadcast, etc.) and as a part of more complex routing algorithms (e.g., OSPF [35]).

In our simulations each node maintains a finite queue of packages. In the l first time instants l packages of a file containing routing information are sent by one fixed agent using selective diffusion to the whole set of nodes in the network. Each package has an unique identification number and a time-to-live (*ttl*) counter that decreases by one for each hop the package crosses. The package also maintains a list of visited nodes in order to trace the route the package has followed and to effectively implement the selective diffusion. When the *ttl* counter becomes 0, the package is considered obsolete and removed from the network. In addition, the package can become corrupted at any bit, including *ttl*, route, or identification number. For convenience, the packets are of fixed size.

The following happens at each time instant for each node.

- The first valid package is obtained from its queue and the package is sent by selective diffusion. A package becomes obsolete if there is another copy of the package in the queue with a higher *ttl* or its *ttl* takes the value 0.

- If the destination node is congested (i.e., its queue is full) the package is removed from the network reflecting the congestion properties of the network. At every time instant all obsolete packages are removed from the queues. This allows queues to maintain the most recent version of the package.

- At each crossed hop, the package decrements its *ttl* and updates its route register. The package has a maximum number of hops that corresponds to its *ttl*, if the maximum is reached (i.e., *ttl* = 0), the package is removed from the network.

- At each hop the package has a probability of becoming corrupted as it crosses links with a certain probability of corrupting packages.

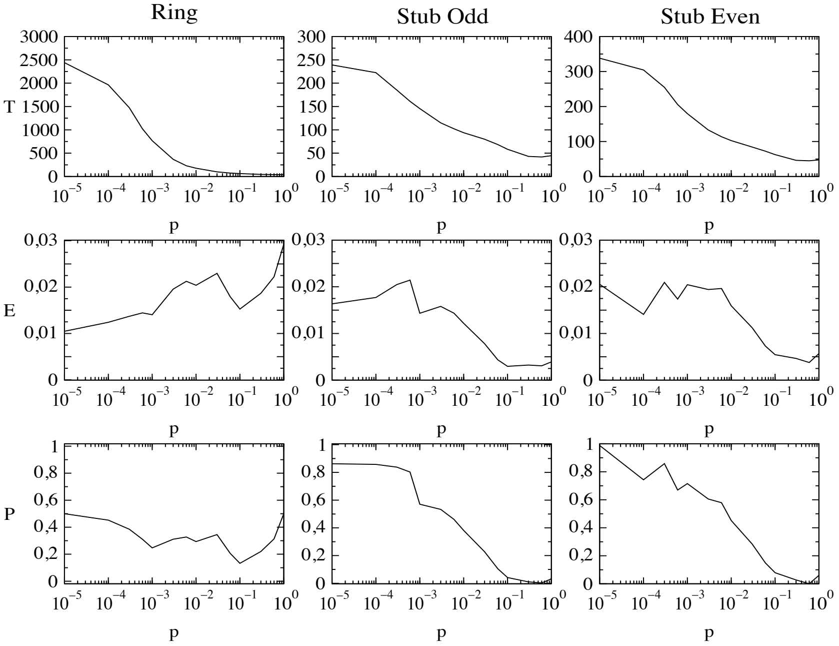

Figure 8. T, E, and P for broadcast in ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$. $l = 10$, queue size = 100, and $ttl = 380$. Results are averaged over 100 experiments.

The system stops when the whole set of nodes owns a copy of the complete file or there are no more packages to deliver in the agent's queue. When the system stops the transfer time T and the average error rate E are computed.

We have carried out our simulations for each of the different topologies obtained by applying the modified Watts–Strogatz method in ring-lattice and transit-stub substrates. All the metrics are averaged over 100 different experiments. In Figure 8 we display the transfer time T, the error rate E, and the time/error average P for the three substrates. The left column shows the dynamic metrics values for the ring-lattice, the central and right columns show the dynamic metrics for the transit-stub graphs.

We can observe that, in both substrates, the average time T decreases when p grows (i.e., when the graph becomes more random). The reduction in the average time rate is due to shorter paths that are typical of random graphs (see Figure 4). On the other hand, the error rate E behaves differently in the case of ring-lattices than for transit-stub graphs. In the case of ring-lattices, the error increases with the value of p; meanwhile, the error decreases with p in the case of transit-stub substrates. In both substrates the error rate is related with the behavior of the characteristic euclidean path length.

If an average measure of the two normalized parameters P is used as in [14, 15], an optimal point for building networks with an optimal error-time delay ratio would be those corresponding to the random area $p \approx 1$ for transit-stub graphs. Meanwhile the optimal ratio for ring-lattices is when the graph is in the SW area. Note that we do not intend to establish a comparison between ring-lattice and transit-stub networks, but, given a specific substrate and an embedding, we want to determine the best performance.

6.2 Simulations based on multicast

In the past few years, especially with the emergence of multimedia over the Internet and distributed database systems, there has been an increase in the number of applications that need to establish communications between groups of hosts. Specific algorithms have been developed for multicast, and even some specific networks with their own topologies as, for example, MBONE [34], have been developed for multicast services.

Multicast requires the existence and management of groups of nodes. Each node of the network can join one or several groups or can be separate from any group to serve as a mere transport node. The creation and management of groups are independent tasks of the package delivery process over the network and have their own algorithms and protocols. For this reason, as a step prior to each multicast simulation, two groups composed of 1% of the nodes in the network are created. The nodes that form a group are selected at random with uniform probability over the whole set of nodes in the network. A node can belong to none, one, or several groups. The nodes that do not belong to any group work as transport nodes. We assume that the members of each group do not change once they are established. In our simulations we have implemented a multicast routing algorithm based on spanning trees [32]. A prerequisite to carrying out the simulations is the calculation of the spanning trees that originate in the source node.

As in section 6.1, each node maintains a queue of packages. In the l first time steps, $2l$ packages of two different files are sent by a source agent. Each file consists of l packages and is addressed to one of the two groups in the network. Each package has a file number, an identification number inside the file, and a *ttl* counter. The package also maintains a list of visited nodes and when the *ttl* counter becomes 0, the package is considered obsolete. As in the previous experiment, the package can become corrupted at any bit, including *ttl*, route, file, or identification number. Fixed-size packages are again used.

The following happens at each time instant for each node.

- The first valid package is obtained from its queue and the group to which the package is addressed is checked.
- The node sends the package only to neighbor nodes that belong to the group to which the package is addressed, or are on the way to nodes of that group, avoiding the neighbor node that sent the package.
- When the destination node is congested, the package is lost on its way to that node.
- At every time instant all obsolete packages are removed from the queues.
- At each crossed hop, the package decrements its *ttl* and updates its route register.
- The packages have some given probability of becoming corrupted as they cross links.

The system stops when the whole set of nodes expecting files (i.e., the nodes belonging to the two groups) owns a copy of the corresponding file or files or there are no more packages to deliver. When the system stops, the average error and transfer time are computed.

As in section 6.1 the Watts–Strogatz method has been applied in ring-lattice and transit-stub substrates. All the metrics are averaged over 100 different experiments.

Multicast results for T, E, and P are plotted in Figures 9 and 10. As an illustration of the consistency of the results we can observe in Figure 10 the same qualitative results when the number of packages and *ttl* are increased. Multicast presents its worst results both for transfer time and error rate in the regular area ($p = 0$). Optimal points are obtained for the SW area in the case of ring-lattices and in the random area for transfer-stub graphs. This is consistent with the broadcast results due to the fact that both are minimal path based algorithms.

6.3 Point-to-point transport (unicast)

In unicast, or point-to-point transmission, a package has to cross a number of intermediate machines (routers) in order to go from the source to the destination. Multiple routes or paths of different length are possible. In this framework, routing algorithms play an important role in the efficient transmission of packages. In the Internet, for example, each router obtains routing information (link state, topology, congestions, etc.) from other routers. Routing information can be provided by routers in the same subnetwork or, for border network routers, from the neighbor subnetwork [35]. The information provided by the routers allows maintaining the best possible route at each moment. In

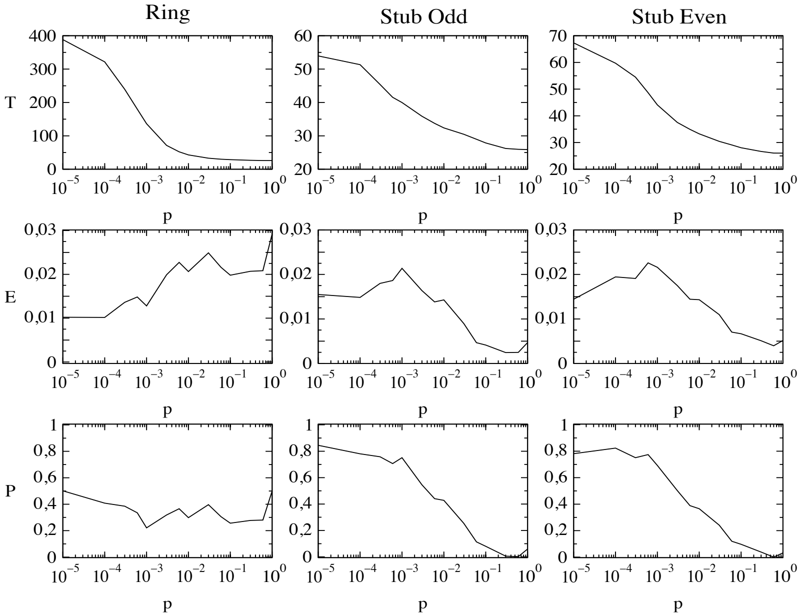

Figure 9. T, E, and P for multicast with spanning trees routing in ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$. $l = 10$, queue size = 100, and $ttl = 380$. Results are averaged over 100 experiments.

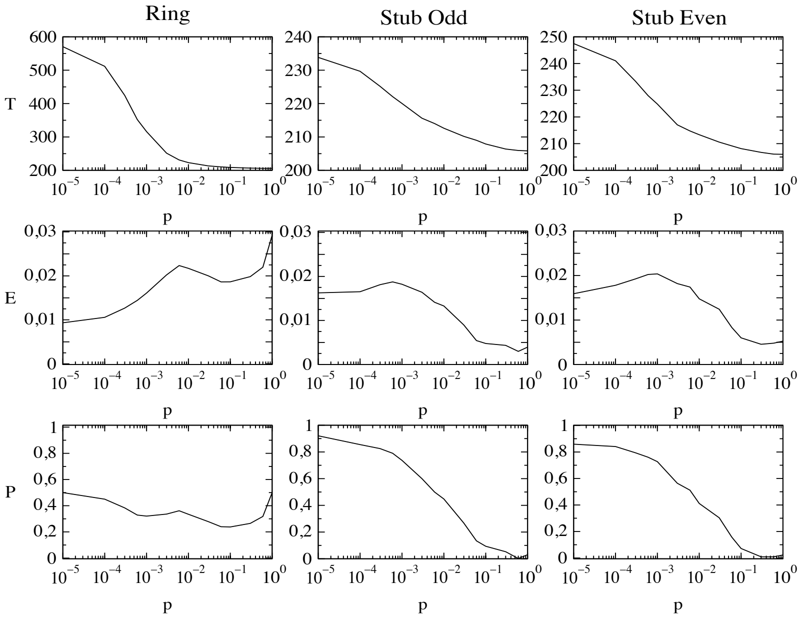

Figure 10. T, E, and P, for multicast with spanning trees routing in ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$. $l = 100$, queue size = 110, and $ttl = 750$. Results are averaged over 100 experiments.

our model, each agent maintains a routing table based on minimal distance to the other agents in the network. Note that in reality the routing process does not follow minimal paths [36]. However, minimal distance routing can be considered as a lower bound of routing algorithms.

In our unicast experiment, each node maintains a queue of packages. In the l first time steps, one node selected at random sends a file of l packages to each of two different randomly selected destination nodes. Each package has an origin and destination number, a file number, an identification number inside the file, and a *ttl* counter. The package also maintains a list of visited nodes and when the *ttl* counter becomes 0, the package is considered obsolete. As in the previous experiment, the package can become corrupted at any bit, including destination, origin, *ttl*, route, file, or identification number. Fixed-size packages are again used.

The following happens at each time instant for each node.

- The first valid package is obatined from its queue and the node to which the package is addressed is checked.
- The node sends the package to a neighbor node that is closer to the destination address.
- When the destination node is congested, the package is lost on its way to that node.
- At every time instant all obsolete packages are removed from the queues.
- At each crossed hop, the package decrements its *ttl* and updates its route register.
- Packages have a probability of becoming corrupted as they cross links.

Unicast results for T, E, and P are plotted in Figures 11 and 12. Both substrates present a very similar qualitative behavior with respect to the transfer time. The transfer time in unicast is clearly related to the characteristic path of the substrate. Both substrates now present different characteristics in the error rate. The error rate increases with p in the case of ring-lattices, whereas in transit-stub graphs the error rate has a minimum in the random region, reaching its worst values in the regular extreme of p ($p = 0$). The optimal time/error average is in the SW area for ring-lattices and in the random area for transit-stub graphs.

6.4 Multihop networks simulations

Multihop networks have a small number of neighbors per node and have been mainly used for computer clusters and lightwave networks. Some very specific topologies have been studied for multihop networks, such as the toroidal and diagonal mesh [37] or the Manhattan street [38].

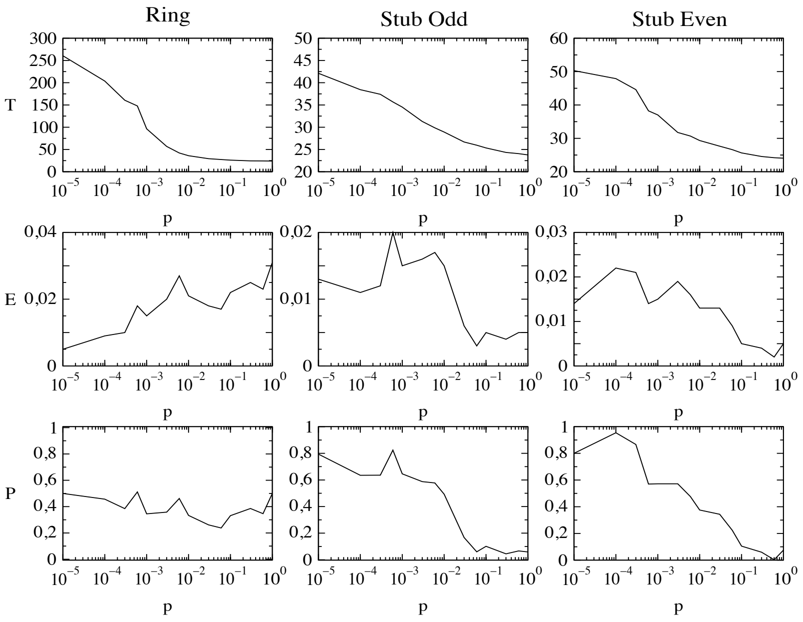

Figure 11. T, E, and P for unicast with minimal distance routing in ring-lattices with n = 2992 and k = 8, transit-stub graphs (k odd) with n = 3000 and k = 9, and transit-stub graphs (k even) for n = 2992 and k = 8. l = 10, queue size = 100, and ttl = 380. Results are averaged over 100 experiments.

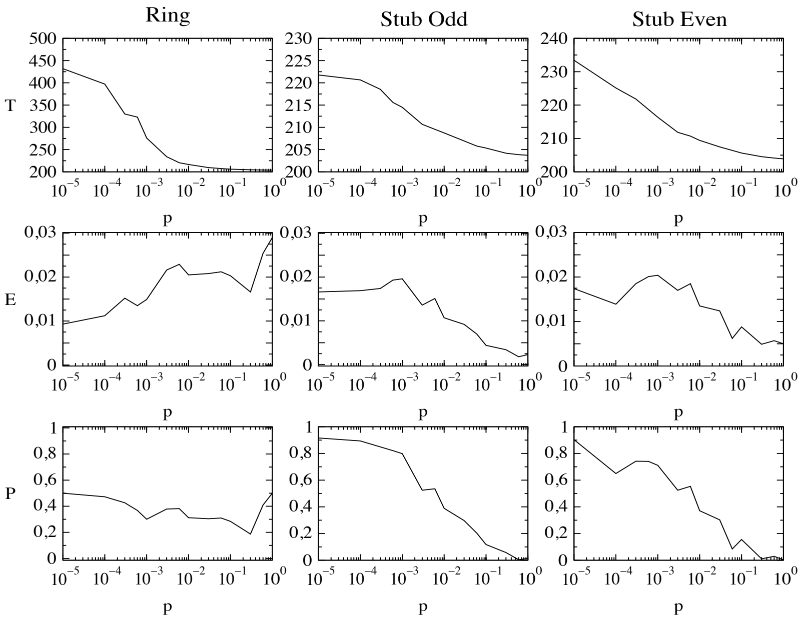

Figure 12. T, E, and P for unicast with minimal distance routing in ring-lattices with n = 2992 and k = 8, transit-stub graphs (k odd) with n = 3000 and k = 9, and transit-stub graphs (k even) for n = 2992 and k = 8. l = 100, queue size = 110, and ttl = 750. Results are averaged over 100 experiments.

Deflection routing or "hot potato" routing is a popular unicast routing strategy, specially used over multihop networks. Deflection routing is a bufferless routing algorithm. Packages are sorted by a deflection criterion as, for example, the time the package is in the network, or the distance to its destination. Packages with higher priority are routed to the optimal links. Store and forward routing is another algorithm used over multihop networks. Store and forward is a buffered algorithm and all packages are routed over the shortest path.

In this experiment, l packages of a file are sent by a source node to a random destination node. Each package has an identification number, a destination address, and a *ttl* counter. As in the previous sections, the package maintains a list of visited nodes and when the *ttl* counter becomes 0, the package is considered obsolete. Package corruption is also possible.

For deflection routing the highest priority is given to packages that have low *ttl*. This is the optimal criteria as shown in [37, 39]. At each time instant, the algorithm works as follows.

- Each node gets every valid package from its queue.
- If the package has a high *ttl*, it is sent to a random neighbor. If the *ttl* of the package is below a given threshold (in our experiments we take this threshold to be the diameter of the ring-lattice of size n), then it is sent to the following node by using the shortest path to the destination node.

Deflection routing results for T, E, and P are plotted in Figure 13. As expected, the ring-lattice presents the worst transfer time, but there are no big differences in respect to the transfer time of the transit-stub graphs. This is due to the fact that transfer time is mainly determined in both substrates for the "random walk" part in the route of each package. Some differences are found in the error rate. In the case of ring-lattices, the error decreases until the minimum value is situated in the SW region, then the error rate increases to a similar value observed in the regular ring substrate. For transit-stub graphs, the error rate remains almost constant during the SW area, increasing only when p is close to 1. Note also that in both substrates the optimal time/error average is obtained in the SW area. Deflection routing is not a minimal distance algorithm, so we cannot expect the error rate to follow the behavior of the characteristic euclidean path length. In this algorithm, the error rate is determined mainly by the random walk part of each route.

We now analyze store and forward routing. This routing algorithm performs the following at each time instant.

- Each node gets the first valid package from its queue and sends it to the following node using the shortest path to the destination node of the package.

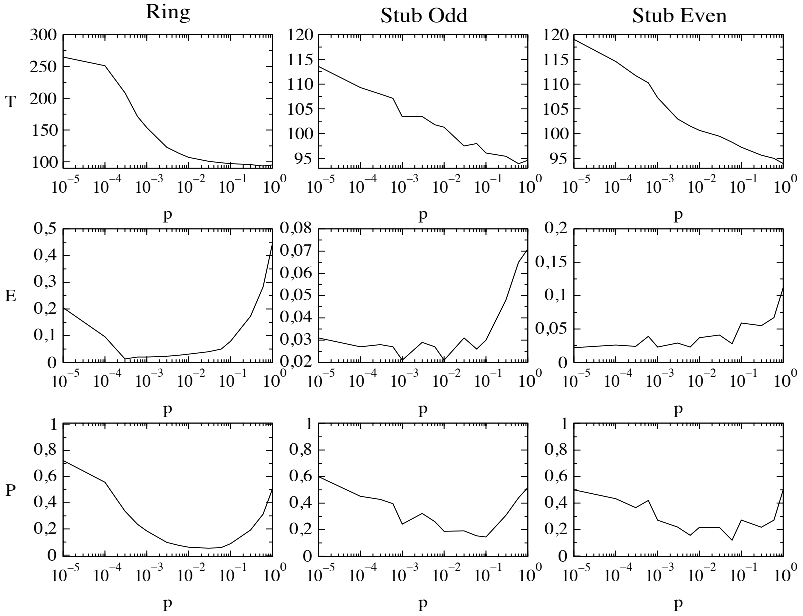

Figure 13. T, E, and P for the deflection routing policy in ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$. $l = 10$, and $ttl = 380$. Results are averaged over 100 experiments.

As in the previous experiments, node congestion, obsolete packages, corruption of packages, and *ttl* countdown have been implemented.

Store and forward results for T, E, and P are plotted in Figure 14. When very few packages $l = 10$ are present in the network the algorithm shows a high velocity, with a very small (close to the optimal) transfer time. The transfer time results respond to the expected values (decrease in p and higher T values for the ring-lattice substrate). The error rate increases with p in the ring-lattice substrate and decreases for transit-stub graphs, resembling the plot of the characteristic euclidean path length. Note that the optimal time/error average is obtained in the SW area for the ring-lattice substrate, while the transit-stub graphs reach the optimal value in the random area.

7. Conclusions and discussion

We have developed a method for building initial regular substrates by considering topological metrics different from the usual ones, as for example, the number of biconnected components. We studied a set of networking inspired dynamics over these new topologies by comparing them with the same dynamics over the usual small-world (SW) substrate.

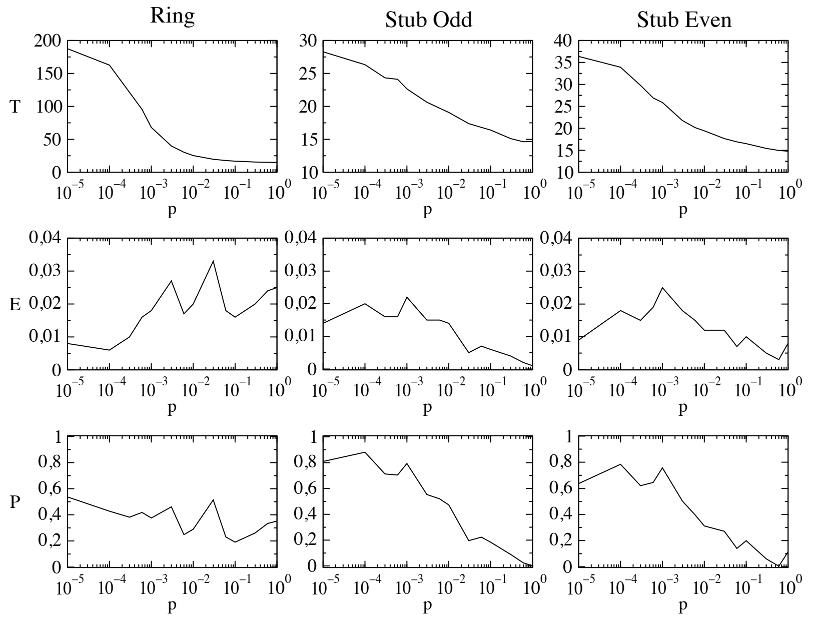

Figure 14. T, E, and P for the store and forward routing policy in ring-lattices with $n = 2992$ and $k = 8$, transit-stub graphs (k odd) with $n = 3000$ and $k = 9$, and transit-stub graphs (k even) for $n = 2992$ and $k = 8$. $l = 10$, queue size = 100, and $ttl = 380$. Results are averaged over 100 experiments.

In our simulations we study a very simplified deterministic model based on a set of homogeneous agents that interact following policies inspired by some of the dynamics found in intercommunication networks. The results obtained could be very far from the performance parameters given in real communication systems such as the Internet. However, we think that these results can possibly be appliyed to small, homogeneous systems and can show that, in these situations, the underlying network topology heavily determines the dynamic behavior of the network.

We conclude with the following remarks.

- Regular graphs exist that present a high number of biconnected components. These networks better resemble hierarchical networks than structures like stars, rings, or grids.
- These new transit-stub networks present a similar SW area when they are shifted from a regular to a random situation in a way similar to ring-lattice substrates. However, transit-stub networks present a slower descent in their characteristic path.
- The selected substrate changes the values of the dynamical metrics of a set of well known internetworking inspired dynamics over a network of agents.

- The SW area is an optimal region for time/error averages for ring-lattice graphs over several internetworking inspired dynamics. There is one exception, the deflection routing algorithm, because this parameter is controlled mainly by a "random walk" part of the route.

Acknowledgments

We thank the MCyT (BFI 2000-015). (RH) was also funded by DE-FG03-96ER14092 and (CA) was supported by ARO-MURI grant DAA655-98-1-0249 during a four month stay at UCSD. We also thank Lev Trimsing for useful discussion.

Appendix

Pseudocode to generate regular multi-biconnected graphs for *k* odd.

```
Procedure CreateRegularStubsOdd(n,k)

  //n number of nodes in the graph
  //k neighbors per node

  // Build the central ring

  For j = 1 to n/(k+3)
    For l = 1 to (k-1)/2
      // bidirectional edge
      AddEdge (j, (j+l) mod (n/k+3))

  // Attach a stub to each
  // node of the central ring

  For j = 1 to n/k+3
    AddStubOdd(j)

End Procedure CreateRegularStubs

Procedure AddStubOdd(x)
// x = node to attach the stub

  // for each node in the stub

  For j = n/(k+3) + x*(k+2)
     to  n/(k+3) + x*(k+2) + k + 1

    // number of neighbors
    For v=1 to (k-1)/2

      // check cyclic connections in the stubs
       if  j+v < n/(k+3) + x*(k+2) + k + 2
          ind = j+v
       else
          ind = (j+v)-(k+2)
      AddEdge(j,ind)

   // join pairs of nodes in the
   // stub except the connection
   // with the central ring

   For j = n/(k+3) + x*(k+2)+1 to
     n/(k+3) + x*(k+2) + (k + 1)/2
     AddEdge(j,j+((k+1)/2))

  // and last connect the stub
  // with the central ring
  AddEdge(n/(k+3) + x*(k+2),x)

End Procedure AddStubOdd
```

References

[1] P. S. Neelakanta and W. Deecharoenkul, “A Complex Systems Characterization of Modern Telecommunication Services,” *Complex Systems*, **12** (2000) 31–69.

[2] B. Cheswick and H. Burch,“Mapping the Internet,” *Computer*, **32**(4) (1999) 97.

[3] K. Claffy, “CAIDA: Visualizing the Internet,” *Internet Computing Online*, http://computer.org/internet/v5n1/caida.html (2001).

[4] K. Claffy, “Measuring the Internet,” *IEEE Internet Computing*, **4**(1) (2000) 73–75.

[5] F. Corbacho and M. A. Arbib, Schema-based Learning: “Towards a Theory of Organization for Adaptive Autonomous Agents,” *Proceedings of the First International Conference on Autonomous Agents*, Santa Monica, CA, 1997 (ACM, New York, 1997).

[6] L. F. Lago, R. Huerta, F. Corbacho, and J. A. Siguenza, “Fast Response and Temporal Coding on Coherent Oscillations in Small-world Networks,” *Physical Review Letters*, **84**(12) (2000) 2758–2761.

[7] M. Fletcher, E. Garcia-Herreros, J. H. Chirstensen, S. M. Deen, and R. Mittman, “An Open Architecture for Holonic Cooperation and Autonomy,” *Proceedings of the Workshop on Database and Expert Systems Applications* DEXA 2000, Greenwich, London (IEEE Computer Society, Los Alamitos, California, 2000).

[8] M. Doar and I. Leslie, “How Bad Is Naive Multicast Routing?” *Proceedings of IEEE INFOCOM’93* (IEEE Computer Society, Los Alamitos, California, 1993).

[9] L. Wei and D. Estrin, “The Trade-offs of Multicast Trees and Algorithms,” *Proceedings of the International Conference on Computer Communications and Networks 1994* (IEEE Computer Society, Los Alamitos, California, 1994).

[10] R. Kumar, P. Raghavan, S. Rajagopalan, and D. Sivakumar, "The Web as a Graph," *Proceedings of the 19th ACM Symposium on Principles of Database Systems 2000* (ACM Press, New York, 2000).

[11] E. W. Zegura, K. L. Calvert, and M. J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topology," *IEEE/ACM Transactions on Networking*, **5**(6) (1997) 770–783.

[12] C. Aguirre, F. Corbacho, and R. Huerta, "A Realistic Substrate for Small-world Networks Modeling," *Proceedings of the 12th International Workshop on Database and Expert Systems Applications*, Munich, Germany (IEEE Computer Society, Los Alamitos, California, 2001).

[13] D. J. Watts and S. H. Strogatz, "Collective Dynamics of Small-world Networks," *Nature*, **393** (1998) 440.

[14] C. Aguirre, J. Martinez-Munoz, F. Corbacho, and R. Huerta, "Small-World Topology for Multi-Agent Collaboration," *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, Greenwich London (IEEE Computer Society, Los Alamitos, California, 2000).

[15] C. Aguirre, J. Martinez-Munoz, F. Corbacho, and R. Huerta, "A New Topology for Multi-Agent Collaboration," *Proceedings Autonomous Agents 2000 Workshop Agents in Industry*, Barcelona, España (available from carlos.aguirre@ii.uam.es).

[16] A. G. Phadke and J. S. Thorp, *Computer Relaying for Power Systems* (Wiley, New York, 1988).

[17] S. Milgram, "The Small World Problem," *Psychology Today*, **2** (1967) 60–67.

[18] L. A. Adamic, "The Small World Web," *Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries, ECDL'99* (Springer-Verlag, Heidelberg, Germany, 1999).

[19] T. B. Achacoso and W. S. Yamamoto, *AY's Neuroanatomy of C.elegans for Computation* (CRC Press, Boca Raton, Florida, 1992).

[20] T. Araújo and R. Vilela Mendes, "Function and Form in Networks of Interacting Agents," *Complex Systems*, **12** (2000) 357–378.

[21] D. J. Watts, *Small Worlds: The Dynamic of Networks between Order and Randomness* (Princeton University Press, Princeton, New Jersey, 1999).

[22] M. E. J. Newman and D. J. Watts, "Renormalization Group Analysis of the Small-World Network Model," *Physical Review E*, **60** (2000) 7332–7342.

[23] F. Comellas, J. Ozón, and J. G. Peters, "Deterministic Small-world Communication Networks," *Information Processing Letters*, **76** (1-2) (2000) 83–90.

[24] S. N. Dorogovtsev and J. F. F. Mendes, "Exactly Solvable Analogy of Small-world Networks," *Europhysics Letters*, **50**(1) (2000) 1–7.

[25] R. Sedgewick, *Algorithms in C++* (Addison-Wesley Publishing Company, Reading, Massachusetts, 1990).

[26] R. Cheng, "Design of Fuzzy Traffic Controller for ATM Networks," *IEEE/ACM Transactions on Networting*, **4** (1996) 460–469.

[27] B. Bollobas, *Random Graphs* (Harcourt Brace Jovanovich, Orlando, Florida, 1985).

[28] M. Pěchouček, V. Mařík, and O. Štepánková, "Coalition Information in Manufacturing Multi-agent Systems," *Proceedings of the Workshop on Database and Expert Systems Applications DEXA 2000*, Greenwich, London (IEEE Computer Society, Los Alamitos, California, 2000).

[29] J. McGhee, M. J. Grimble, and P. Mowforth, *Knowledge-based Systems for Industrial Control* (IEE Books, Stevenage, UK, 1990).

[30] A. Chowdhury, E. Burger, and D. Grossman, "DRS: A Fault Tolerant Network Routing System for Mission Critical Distributed Applications," *Proceedings of the IEEE-IC3N Sixth International Conference On Computer Communication and Networks* (IEEE Computer Society, Los Alamitos, California, 1997).

[31] J. C. Park, "Chaos and Predictability of Internet Transmission Times," *Complex Systems*, **12** (2000) 297–316.

[32] A. S. Tanenbaum, *Computer Networks* (Prentice Hall, Upper Saddle River, New Jersey, 1996).

[33] "Rediris Network Services: The National Backbone,"
http://www.rediris.es/red/index.en.html#red troncal (2001).

[34] M. R. Macedonia and D. P. Brutzman, "MBone Provides Audio and Video Across the Internet," *IEEE Computer*, **27**(4) (1994) 30–36.

[35] C. Huitema, *Routing on the Internet* (Prentice Hall, London, 2000).

[36] A. Broido, E. Nemeth, and K.C. Klaffy, "Internet Expansion, Refinement and Churn," *European Transactions on Telecommunications*, January 2002.

[37] K. W. Tang and S. A. Pabudidri, "Diagonal and Toroidal Mesh Networks," *IEEE Transactions on Computation*, **43** (1994) 815–826.

[38] N. F. Maxemchuck, "The Manhattan Street Network," *Proceedings of IEEE Globecom'85*, New Orleans, LA, 1985 (IEEE Computer Society, Los Alamitos, California, 1985).

[39] H. Y. Huang, T. Robertazzi, and A. A. Lazar, "A Comparison of Information Based Deflection Strategies," *Computer Networks ISDN Systems*, **29** (1995) 1388–1407.