Modeling Traffic of Information Packets on Graphs with Complex Topology

Bosiljka Tadić

Jožef Stefan Institute, Box 3000, 1001 Ljubljana, Slovenia
Bosiljka.Tadic ijs.si
http://phobos.ijs.si/ tadic/

Abstract. We present the algorithm introduced in [1] to grow the directed graph with self-organized in- and out-links and closed cycles and show that it has topological properties of the world-wide Web. We then implement the algorithm of simultaneous transport of information packets that are created with a given rate and navigated on that graph by local up-to-next-neighbour search algorithm. We determine the statistics of transit times of packets from posting until their arrival to given destinations on the graph when creation rate is varied.

1 Introduction

Importance of technological, social, and biological networks, was recently emphasized in sciences. Most of these networks are not static but evolve in time, based on a set of microscopic rules individual for each network structure. In the coarse of time these rules lead to *emergent* structures of links via the dynamic constraints and self-organization processes [2,3]. Ubiquity of the scale-free networks, having no a typical scale in the degree distributions, can be related to socially motivated *preferential linking* in real networks, and to the functional stability of such topologies to error correction [4] in naturally evolved networks.

Understanding functional properties of networks, both autonomous and man made, makes the necessity to study the dynamic processes (such as gene coregulation, search and access on the Web, packet trafficking on the Internet) on graphs with complex topology. The motivation is both theoretical—we encounter complex graph geometries, the full impact of which is not yet clear even in the simple random walk processes [5], and practical—improving potential applications, costs planning, adapting the efficiency of processes with respect to the underlying network structure.

Here we study numerically traffic of information packets on graphs with scale-free organization of links and closed cycles, with the statistical properties similar to those in the world-wide Web and the Internet. Apart from the graph's topology, several other parameters are relevant for packet trafficking. These are external parameters, such as *packet creation rate*, and internal parameters such as hardware (*link capacity* and *buffer sizes*) and software properties (*search algorithm* and *queuing discipline*) of the network.

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, pp. 136–143, 2003.
© Springer-Verlag Berlin Heidelberg 2003

We first implement an algorithm for growth of the graphs with given scale-free structure using the microscopic rules originally proposed in Ref. ([1]) to model growth of the world-wide Web. The emergent structure statistically resembles the one in the real Web, showing the occurrence of the hub and authority nodes, and closed cycles [1,6,7].

We then implement *simultaneous transport* of packets created with a given rate and navigated by the local *nnn*-search algorithm [8] towards their respective destinations on the graph. We present several results quantifying the collective dynamic properties of moving packets, that can be compared with measured quantities in the real packet traffic [9,10]. Some other recent attempts of modeling packet transport on graphs are: for scale-free tree graphs [11] (see also references therein), and for strictly hierarchical [12] and optimized [13] graph structures.

2 Growth Rules & Topology of the Web Graph

We present an algorithm originally proposed in [1] to grow a graph with scale-free structure and flexible wiring diagram in the class of the world-wide Web. Objectives are to grow a graph that has statistically the same properties as measured in the real Web [14]: scale-free degree distributions both for in- and out-links (exponents $\tau_{in} \approx 2.2$ and $\tau_{out} \approx 2.6$); clustering properties; and occurrence of a giant component and the distribution of clusters with the exponent $\tau_s \approx 2.5$. As demonstrated in Ref. [1] a minimal set of microscopic rules necessary to reproduce such graphs include *growth, attachment*, and *rewiring.* Time is measured by addition of a node, which attempts to link with probability $\tilde{\alpha}$ to a node k. Else, with probability $1-\tilde{\alpha}$ a preexisting node n rewires or adds a new out-link directed to k. Nodes k and n are selected with probabilities $p_{in} \equiv p_{in}(k,t)$, $p_{out} \equiv p_{out}(n,t)$

$$p_{in} = (M\alpha + q_{in}(k,t))/(1+\alpha)Mt \; ; p_{out} = (M\alpha + q_{out}(n,t))/(1+\alpha)Mt \; , \quad (1)$$

which depend on current number of respective links $q_{in}(k,t)$ and $q_{out}(n,t)$. M is average number of links per time step (see [1,6] for more details). The graph flexibility, which is measured by the degree of rewiring $(1-\tilde{\alpha})/\tilde{\alpha}$, is essential both for the appearance of the scale-free structure of *out-links* and for occurrence of closed cycles, which affect the dynamic processes on the graph. An example of the emergent graph structure is shown in Fig. 1.

By solving the corresponding rate equations we find that the local connectivities $< q_{in}(s,t) >$ and $< q_{out}(s,t) >$ at a node added at time s increase with time t as

$$q_\kappa(s,t) = A_\kappa \left[(t/s)^{\gamma_\kappa} - B_\kappa\right] \; . \quad (2)$$

with $\kappa = in$ and out, and $\gamma_{in} = 1/(1+\alpha)$ and $\gamma_{out} = (1-\tilde{\alpha})/(1+\alpha)$. We use the original one-parameter model introduced in [1] with $\tilde{\alpha} = \alpha = 0.25$ and $M = 1$. When $\tilde{\alpha} = 1$ the emergent structure is tree like with one out-link per node.

In Fig. 2 (left) we show simulated local connectivities for $t = N = 10^4$ nodes in agreement with Eq. (2). This implies the power-law behavior of the emergent degree distributions $P(q_\kappa) \sim q_\kappa^{-\tau_\kappa}$, where the respective exponents are given by the exact scaling relation $\tau_\kappa = 1/\gamma_\kappa + 1$, in agreement with simulations in [1,6]. Measurements in the Internet maps [15] suggest that a similar structure of in-links occurs as in the above graph, with more symmetry between in- and out-linking and less flexibility in the wiring diagram.

In addition, the Web graph grown from the rules in our model shows a nontrivial correlation between local in- and out-connectivity, which is related to clustering property of the graph (cf. Fig. 1). In Fig. 2 (right) we show average out-degree $< q_{out} >_{nn}$ of nodes which are near neighbours to a node of given in-degree q_{in}. The power-law decay of the curve (slope is 0.42 within error bars) indicates correlations similar to the ones measured in the Internet maps [15].

3 Packet Traffic Algorithm

The information or data traffic on the Internet, e.g., a Web application, occurs in few steps. First, the information enters TCP/IP protocol at server node, where it is divided into a set of smaller data packets, and each packet is given a unique address on the network. Then the packets are transfered from node to

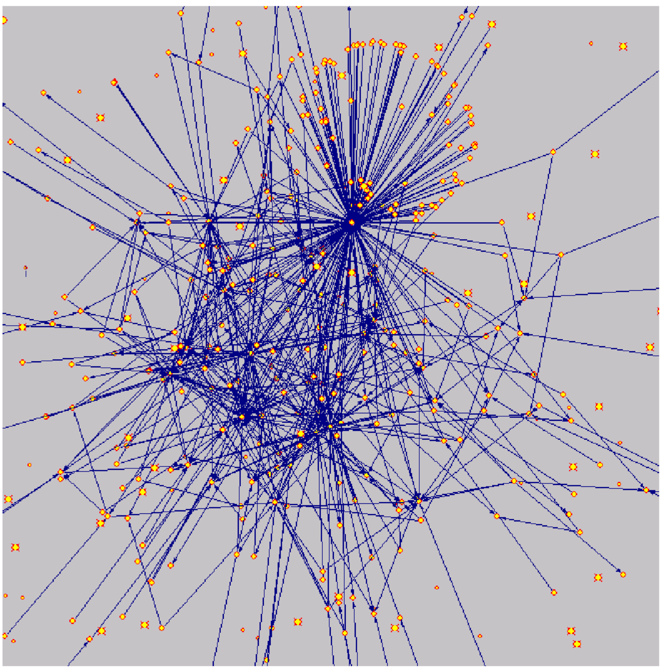

Fig. 1. Emergent structure of directed links with hub and authority nodes and closed cycles in the graph grown from the linking rules in Eq. (1), shown after $t = N = 10^3$ added nodes.

node towards their destination address along generally different paths which are dynamically available that connect the client and the server address on the network. Upon arrival they are eventually re-assembled by TCP at the destination (client) node. We are modeling the kinetics of packets on the network.

We implement traffic of the information packets on the network as a set of *simultaneously moving intentional random walks*, each of which has a specified pair of nodes representing client/server addresses on the graph. The packets are initiated by given rate R at a random node and by the creation each packet is given the destination address, which is selected from other nodes in the network. In each time step the whole network is updated and each node that has a packet tries to transfer it to one of its neighbours, according to a *search algorithm* that is specified below. If more than one packet is on that node, the priority is given according to a specified *queuing discipline*. We adapt LIFO (last-in-first-out queue), which is familiar in the queue theory. When more packets are moving towards same node they form a queue at that node since only one packet can leave the node at one time step. For simplicity, we assume that each link has the capacity one and that all nodes have a finite buffer size H. If the queue at a target node is already full, packet can not be delivered to that node and remains at current node until further occasion that it can be moved. When a packet reaches its destination node it is considered as delivered and it is removed from the traffic.

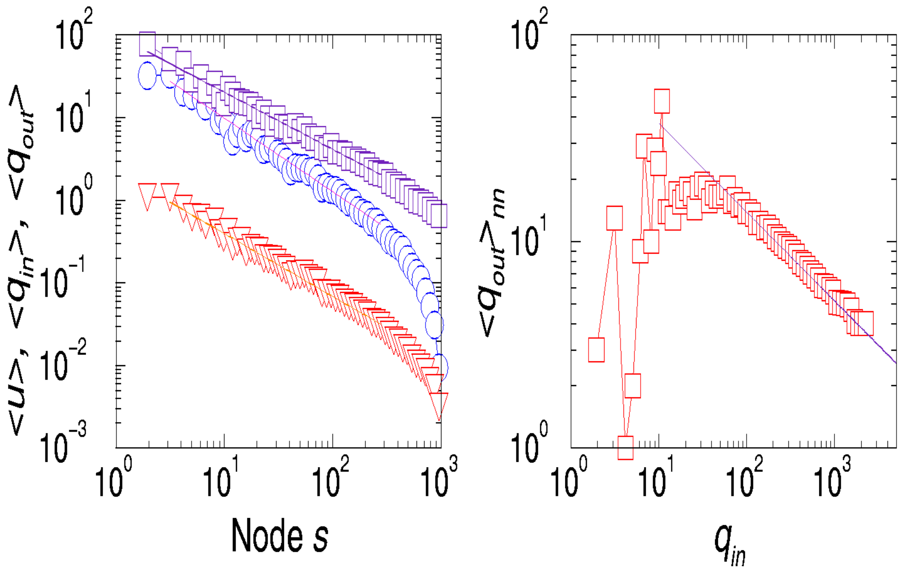

Fig. 2. Left panel: Average connectivity of node s for (top to bottom) out-links, in-links and number of visits $< u >$ for non-interacting (sequential) random walks in the Web graph (Figure 1 from Ref. [8]). Right panel: Near neighbour connectivity correlations between in- and out-links in the Web graph with 6×10^3 nodes. All data log-binned.

The main part of traffic is the algorithm that each node uses in trying to find the way how to forward the packet. We adapt *nnn*-search, that consists of *local* up to next-neighbour search for the packets address: If the packet's address is one of the near neighbours of the current node, packet is delivered to that node. Else, it is delivered to a node whose neighbour is the packet's address, if it is the case, otherwise it is delivered to a randomly selected neighbour. The local *nnn*-search was shown [8] to be quite effective on scale-free graphs, and in particular on cycled scale-free graphs, where it makes use of both hub and authority nodes.

In order to implement this problem numerically, we first grow the graph according to the rules described in Section 2, and store its adjacency matrix. Here we use graph with $N = 10^3$ nodes. We initiate packets on this graph and keep track of destination address, current position, and position in the queue at current node for each packet. Among packets we mark a given number (2000) of them with additional time labels: posting time, and elapsed time that they spend on the network. The implementation of transport includes the following steps:

- start cumulative time loop; with probability R create a packet at a random node and give it a destination address;
- mark given number of packets with time labels—initialize the objects array;
- update in parallel all nodes; node with packets identifies the top packet label address and searches for that destination node;
- move the packet to selected neighbour node (if its buffer is not full);
- when packet is delivered at its destination node delete its data;
- update until all marked packets reach their destinations;

4 Transit Time Statistics

In low traffic intensity, i.e., at zero driving (posting) rate $R = 0$, individual packets walk without waiting in queues. The statistics of survival time depends on graph topology and search algorithm [8]. For a finite creation rate $R > 0$ packets start interacting that leads to formation of queues mostly at hub nodes, and at large intensity the congestion slowly spreads to nodes linked to hubs etc. In general, the interaction and queuing leads to waiting times of packets and thus total elapsed time (transit time) of a packet before it reaches its destination is larger than number of steps that the packet has to perform along the path. (In other implementations of the algorithm the path may also change.)

In Fig. 3 we show a sequence of elapsed time between two consecutive moves for first 800 packets at rate $R = 0.08$. In addition, we show the transit time distributions at zero posting rate (non-interacting walks) and at a large posting rate $R = 0.08$. It shows that at a finite rate $R > 0$ two types of processes are taking part on the cyclic graph: first some packets move quickly, reaching their destination within a short time, while some other remain buried in long queues (at hub nodes), resulting in large transit times. As the Fig. 3 shows, the distribution of transit times exhibits different behavior for short and for long transit times. On the other hand, the overall behavior of the time distribution is governed by the graph's topology, which is reflected in the time distribution at zero rate $R = 0$, where the transit time equals the path length between initial and destination node. In this case, the distributions have a power-law behavior on structured graphs with the exponent *decreasing* with decreased efficiency of the search algorithm at the underlying graph topology. In the cyclic Web graph the exponent is close to 1.6 within numerical error bars, which is comparable with the distribution of round-trip times measured in the Internet (see Ref. [8] for simulations in some other graph topologies).

Within this numerical algorithm we measure more temporal properties that characterize packet traffic, such as density of packets arriving on a hub node, number of active nodes, network's output rate etc. These quantities show cor-

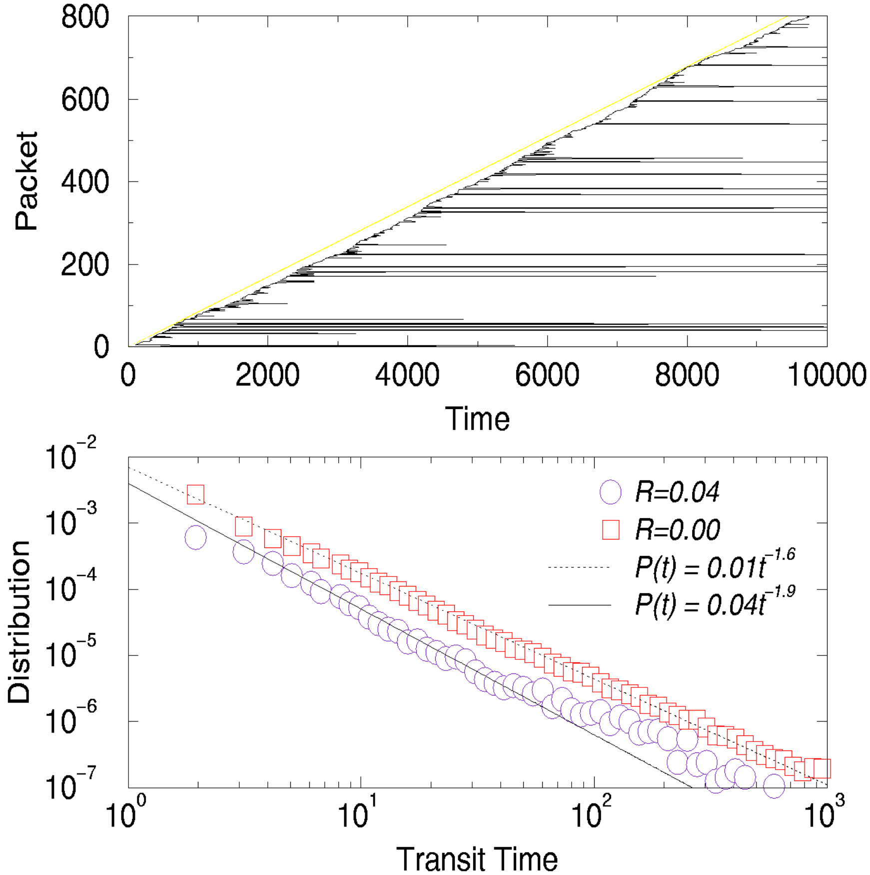

Fig. 3. Top panel: Sequence of elapsed times between two latest moves of packets at driving rate $R = 0.08$. Lower panel: Transit time distributions for zero driving rate (path length) and for finite driving rate (elapsed time) in the Web graph with $N = 10^3$ nodes. Data log-binned.

related temporal behavior that is characteristic for the network's topology and driving conditions. In addition we study waiting times statistics and the role of buffer sizes to the onset of jamming. More detailed study will be published elsewhere. See also Ref. [11] for the case of scale-free tree graph.

5 Conclusions

For numerical modeling of the traffic of information or data packets on the Internet we need two types of algorithms: First an algorithm to grow a graph of given structure, and then an algorithm to implement packet traffic on that graph. Here we have demonstrated how this can be done using an algorithm for growth of a scale-free cyclic graph [1] and implementing simultaneous traffic of many *intentional* random walks as packets on that graph. We applied local *nnn*-search algorithm to navigate walkers through that graph.

For the graph that we use in this simulations we have shown that it belongs to the class of Web graphs, having scale-free structure for both in- and out-links, closed cycles, and occurrence of the hub and authority nodes. In addition to known structure of this class of graphs [1,6,7], here we have shown that these graphs exhibit a nontrivial correlations between local in- and out-connectivity (cf. Fig. 2), resembling the ones in the real Internet. In implementing packet traffic we use walks along in- and out-links with equal probability.

Occurrence of closed cycles on the graph improves its searchability, since the applied up-to-next-neighbour search algorithm can make use of both hub and authority nodes [8], thus affecting the traffic of packets. In comparison to scale-free tree graphs [11], here we find different statistics for fast and slow processes. The overall transport for low traffic intensity is determined by the graph topology. The power-law behavior of the transit time distribution (cf. Fig. 3) agree with measured distribution of the round-trip-time [9]. Our results suggest that observed distributions depend on the posting rate R. Here we used large buffer sizes in order to concentrate on the effects that topology and posting rate have on the traffic.

Our main conclusions that may have a practical impact are the following. The local low cost search algorithm that uses up-to-next-neighbour search appears to be quite effective on scale-free cyclic graphs such as the Web and Internet. The observed short transit times are compatible with the topology with dominant hubs and authority nodes. On the other hand, frequent use of the hubs by the search algorithm involves inevitable queuing at hubs when posting rate increases. This leads to large waiting times in queues and, consequently, to large transit times for queuing packets. At a critical rate $R^\star$ diverging transit times (congestion of the network) occur, which spreads from hubs through the connected cluster (giant component) of the graph. According to our model, the ways to shift the occurrence of congestion towards higher posting rates, which is of practical importance in real networks, could be achieved by adjusting the output rate at hubs (i.e., by increasing out-link capacities) and by introducing stronger clustering of the graph. The present algorithm can be easily extended for more realistic modeling, including different buffer sizes and link capacities, and allows for computing various other properties (see also [11]) of the traffic of packets.

References

1. Tadić, B.: Dynamics of directed graphs: the world-wide Web. Physica A **293** (2001) 273-284; e-print:cond-mat/0011442.
2. Albert, R., Barabasi,A.-L.: Statistical Mechanics of Complex Networks. Rev. Mod. Phys. **74** (2002) 47-101
3. Dorogovtsev, S., Mendes, J.F.F.: Evolution of networks. Adv. Phys. **51**, (2002) 1079–1146
4. Jeong, H., Tombor, B., Albert, R., Oltavi, Z.N., Barabasi, A.-L.: The large-scale organization of metabolic networks. Nature **407**, (2000) 651
5. Tadić, B.: Adaptive random walks on the class of Web graphs. European Physical Journal B, **23** (2001) 221–228
6. Tadić, B.: Growth and structure of the World-Wide Web: Towards realistic modeling. Computer Physics Communications **147** (2002) 586–590.
7. Tadić, B.: Temporal fractal structures: origin of power laws in the world-wide Web. Physica A **314** (2002) 278–283.
8. Tadić, B.: Exploring Complex Graphs by Random Walks. Proceedings of The Granada Seminar on Computational Physics "Modeling Complex Systems", P.L. Garrido and J. Marro (Eds.), AIP Conference Proceedings (in press).
9. Takayashu, M., Takayashu, H., Sato, T.: Critical behavior and 1/f noise in computer networks. Physica A **233** (1996) 824–834.
10. Chong, K. B., Y. Choo, Y.: Fractal analysis of internet traffic time series. arXiv/physics/0206012.
11. Tadić, B., Rodgers, G.J.: Packet transport on scale-free networks. Advances in Complex Systems **5** (2002) 445-456
12. Arenas, A., Diaz-Guilera, A., Guimera, R.: Communication in networks with hierarchical branching. Phys. Rev. Lett. **86** (2001) 3196-3199
13. Guimera, R., Diaz-Guilera, A., Vega-Redondo, F., Cabrales, A., Arenas, A.: Optimal network topologies for local search with congestion. Phys. Rev. Lett. (in press).
14. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the Web. Comput. Networks **33** (2000) 309.
15. Pastor-Satorras, R., Vazquez, A., Vespignani, A.: Dynamical and correlation properties of the Internet. cond-mat/0105161.