Modeling Flow and Congestion in Packet Switching Networks

by Anna Lawniczak[1,2], Peng Zhao[1], Alf Gerisch[1,2], Bruno Di Stefano[3]

[1]University of Guelph, Guelph, ON

[2]The Fields Institute for Research in Mathematical Sciences, Toronto, ON

[3] Nuptek Systems Ltd., Toronto, ON

Abstract

We investigate simplified models of packet switching networks and examine how an introduction of additional randomly generated links influences the performance of these networks. In general, the impact of additional randomly generated links on the performance of a network depends on a routing algorithm used in the network and on a "cost" assigned to each network link. With the shortest path full table routing, where the cost of each link is one, i.e. with the "minimum-hop full table routing", degradation of performance of a network is observed. However, if the cost of each link is equal to the queue size of the node from which the link originates or to the queue size plus one then an improvement in the network performance is observed.

Sommaire

Nous étudions les modèles simplifiés des réseaux de commutation par paquets et examinons comment l'introduction de liens générés aléatoirement influence la performance de ces réseaux. En général, l'impact de l'addition des liens générés aléatoirement sur la performance d'un réseau dépend de l'algorithme de « routing » utilisé dans le réseau et du « coût » assigné à chaque lien du réseau. Avec la table complète du « routing » du chemin le plus court, où le coût de chaque lien est égal à 1, c'est-à-dire avec le « minimum-hop full table routing », on observe une dégradation de la performance du réseau. Toutefois, si le coût de chaque lien est égal à la grandeur de la queue du noeud à partir duquel le lien origine (ou égal à la grandeur de la queue plus 1) alors une amélioration dans la performance du réseau est constatée.

1.0 Introduction

Understanding the dynamics of flow and congestion in data communication networks is of vital importance. Some of the aspects of this dynamics can be captured and investigated by studying simplified models of data communication networks. Often these models assume either regular topology of a network, in the form of a square or a hexagonal lattice, or a binary Cayley tree, or a random graph topology. The purpose of our work is to study how additional randomly generated links affect the dynamics of flow and congestion in packet switching networks. We investigate the performance of such networks for various routing algorithms and network topologies. In order to simulate simplified models of packet switching networks we developed a software tool called DataNetwork.exe. In this article we highlight only the developed methodology and present some selected results. This work is the continuation of the research commenced by one of the authors in [1,2], and it can be easily extended to simulate dynamics of more complex networks. The proposed methodology can be extended to model dynamics of flow of data in wireless networks.

2.0 Packet Switching Network Models

The purpose of a packet switching network is to transmit messages from points of origin to destination points. In our model, we assume that the entire message is contained in a single "capsule" of information, which, by analogy to packet-switching networks, is simply called a packet. In a real packet-switching network, a single packet carries the information "payload", and some additional information related to the internal structure of the network. Since our aim is to understand, for various routing algorithms, the effects of additional randomly generated links on network flow and congestion, we ignore the information "payload" entirely. Hence, in the considered models we assume that each packet carries only two pieces of information: time of its creation and its destination address.

Our simulated networks consist of a number of interconnected nodes. Each node can perform two functions: of a hosts, meaning that it can generate and receive packets, and of a router (message processor), meaning that it can store and forward packets. Packets are stored in queues and buffers which are maintained by the nodes. We assume that each queue can be of unlimited length and that each buffer stores temporarily only one packet which is ready for routing. Packets are created and moved according to a discrete time parallel algorithm. The structure of the considered networks and their routing algorithms are described in subsections which follow.

2.1 Connection Topologies

A packet-switching network topology can be viewed in an obvious way as a digraph, where each network node corresponds to a vertex, and each communication link between two nodes corresponds to a pair of parallel edges, each carrying data in one direction. With each direction of a link is associated a cost of transmission of a packet.

We consider the following network connection topologies: regular periodic (non-periodic) two-dimensional lattices L; and regular periodic (non-periodic) lattices L_l with "l" additional randomly generated links added to them. In particular, our software tool, DataNetwork.exe has been developed for square and hexagonal lattices.

The network hosts and routers are located at the nodes of the lattice L. The communication links between nodes are represented by the lattice L edges. The extra links are constructed using the following procedure. First, we select randomly a node n_1 on a lattice L. Next, we select randomly another node n_2 on the lattice L, different from the node n_1. We connect these two nodes with a direct communication link. By repeating this procedure independently l times we obtain the lattice L_l having l additional randomly generated links. In the described model it can happen that the nodes n_1 and n_2 can be selected again to form a new link. Hence, in the network the same nodes can be connected directly by several links. However, this procedure can be easily modified. We want to emphasise that all the connections in our model are static, they do not change during the simulation period. Randomly generated links are added before the simulation starts and remain unchanged. This property can be easily modified.

2.2 Routing Decisions

In the network models under discussion, each packet is transmitted from its source node through various links and packet switches to a destination node according to some routing decision. This is equivalent to finding a path (route) through the graph. Depending on the costs assigned to network links we consider routing decisions based on the following least-cost criterion: minimum path distance and minimum path length.

We consider three types of link cost functions called "One", "QueueSize" and "QueueSizePlusOne". In the case of the link cost function "One" all links in the network are assigned a cost equal to one. Using the minimum path distance criterion for this link cost function, the number of hops from source to destination is minimized for each packet. The routing based on this criterion is called the minimum-hop routing. In the case of the link cost function "QueueSize" all links in the network are assigned a cost equal to the number of packets awaiting transmission in

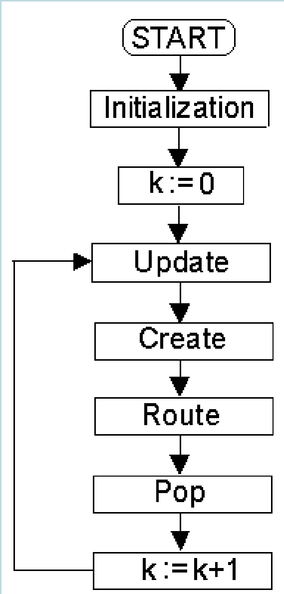

Figure 1: Simplified flow chart of the discrete time network algorithm.

the node from which the link originates. In the case of the link cost function "QueueSizePlusOne" all links in the network are assigned a cost equal to the number of packets awaiting transmission in the node from which the link originates plus the cost of a single hop (which is equal to one). For these two types of link cost functions the minimum path length criterion selects for a packet in a buffer the next node on its path to its destination. This node is on a path with a minimum sum of queue sizes or with the smallest sum of queue sizes plus the number of hops, respectively, from the packet's node to its destination, depending on the current state of the network.

2.3 Discrete Time Network Algorithm

For all three link cost functions discussed in the previous section the routing decisions can be expressed in terms of routing tables. Here we consider only full routing tables, i.e. each node has a routing table with entries for all nodes in the network. Based on these tables we employ two types of full table routing algorithms, the centralized full table routing algorithm and the distributed one.

In the centralized full table routing algorithm at any discrete time each node knows what are the costs of the least-cost paths to all other nodes in the network. This case corresponds to the situation where a central

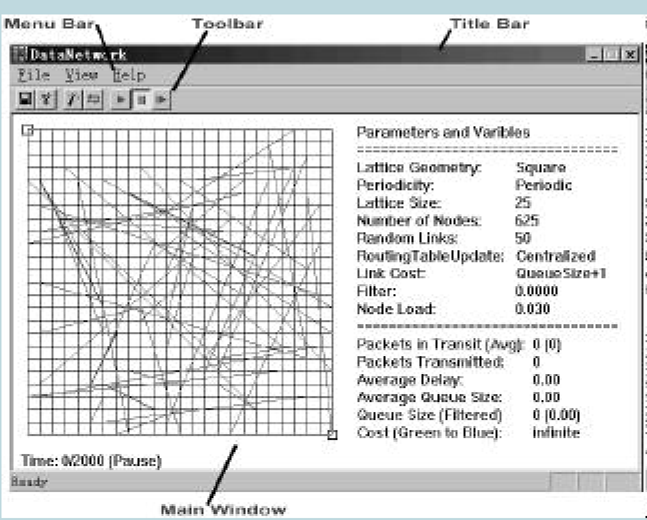

Figure 2: The main window of the DataNetwork.exe tool.

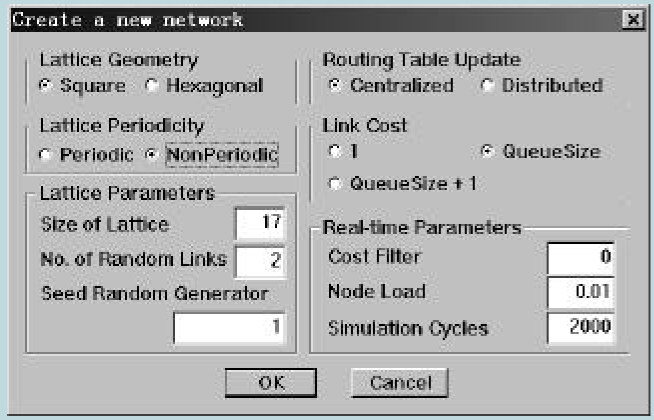

Figure 3: The dialog box of DataNetwork.exe tool.

site is in charge of updating routing tables. Nodes provide their link costs to the central site, which in turn provides new routing tables to all the nodes. At each discrete time new routing tables are calculated based on the criterion of least-cost routing for link cost functions "One", "Queue-Size", or "QueueSizePlusOne", respectively. However, for the link cost function "One" the routing table can be computed *a priori* any simulation. We use the shortest path backward tree algorithm to calculate the new routing tables [3].

In the case of the distributed full table routing algorithms nodes exchange information about their link costs in time in order to determine the least-cost paths from one node to another one. The routing tables are not built up in one cycle as in the case of the centralized full table routing algorithm. They are build up gradually in time. To calculate the routing tables we use the distributed version of the shortest path backward tree algorithm [3].

For the networks with centralized full table routing algorithms or distributed full table routing algorithms the dynamics are governed by the discrete time algorithms as described by the simplified flow chart in Figure 1. The implementation of the modules in the main loop is in parallel over the network nodes.

For each type of the considered routing algorithm we start the network simulation with all queues empty and with discrete time clock "k" set to zero. Then the following operations are repeated in sequence:

1. **Update:** at each node the routing table for centralized or distributed full table routing algorithm is calculated.
2. **Create:** at each node independently of the other nodes, a packet is created with probability p, called the presented node load. Its destination address is randomly selected among all other nodes in the network with uniform probability distribution. The newly created packet is placed at the end of the queue.
3. **Route:** at each node the packet from the buffer is forwarded, if there is one, to the next node along the least-cost path to its destina-

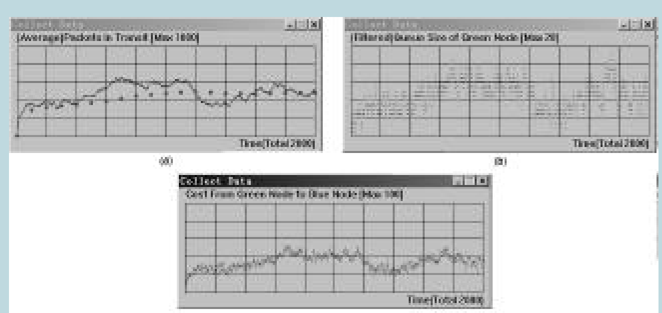

Figure 4: Real-time output data windows of DataNetwork.exe

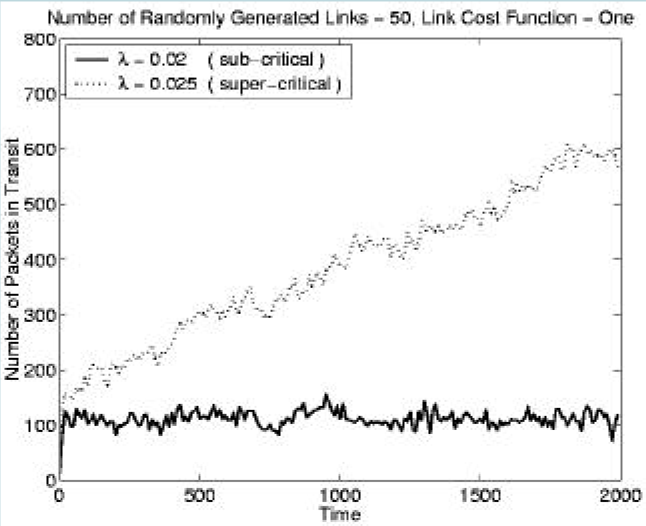

Figure 5: Time evolution of the Number of Packets in Transit for a sub- and a super-critical value of presented node load.

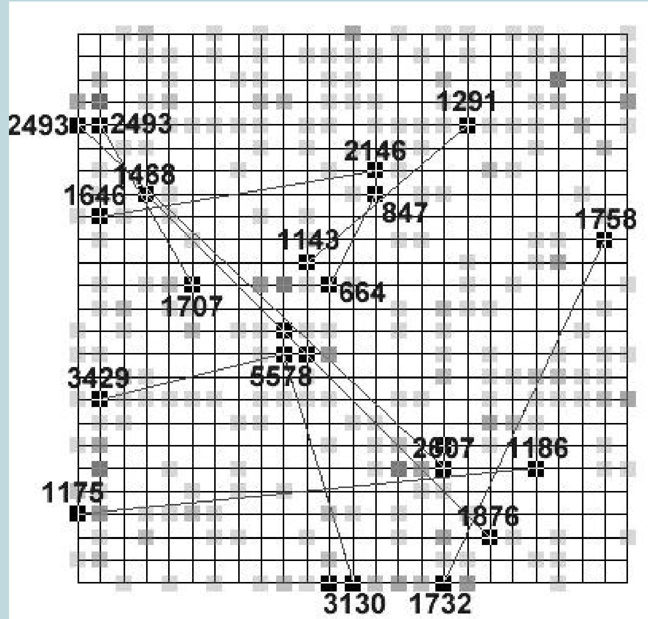

Figure 7: Colour coded queue sizes (from black corresponding to a large queue size to white corresponding to a small queue size) at time 2000 at the nodes a 25 by 25 periodic square lattice network with 10 additional randomly generated links under a presented node load of 0.065. The link cost function is “One”. There are approximately 98% of all undelivered packets concentrated in nodes from which extra links originate (queue sizes given there).

tion. If there are several least-cost paths, then one of them is selected randomly with uniform probability. If the next node is the packet destination, then the packet is destroyed immediately. Otherwise, it is placed at the end of the queue of the next node.

4. **Pop:** at each node, if the queue is not empty then the first packet from the queue is picked up and stored in the node's buffer.
5. Time k is incremented by 1.

The above described sequence of events constitutes a single time step update. It can be repeated an arbitrary number of times. The state of the network is observed after sub-step 5 (clock increase).

3.0 DataNetwork.exe

In order to simulate our models of networks we developed a program called DataNetwork.exe. The program has been developed using the Microsoft Visual C++ 5.0 compiler and is capable of running under Microsoft Windows 95 and 98. Figure 2 depicts the main window of the tool with its various components. The dialog box used to create a new network configuration is given in Figure 3.

DataNetwork.exe allows for real-time output of several network characteristics. These can be displayed during the simulation in special data windows, see Figure 4.

The source code of DataNetwork.exe is currently being ported to UNIX/Linux operating systems. We also plan to incorporate some new features and improvements.

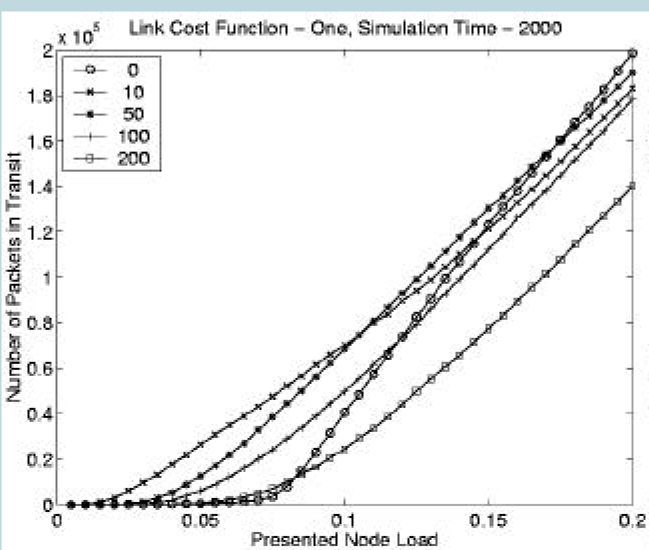

Figure 6: Number of Packets in Transit as a function of presented node load for different numbers of added randomly generated links.

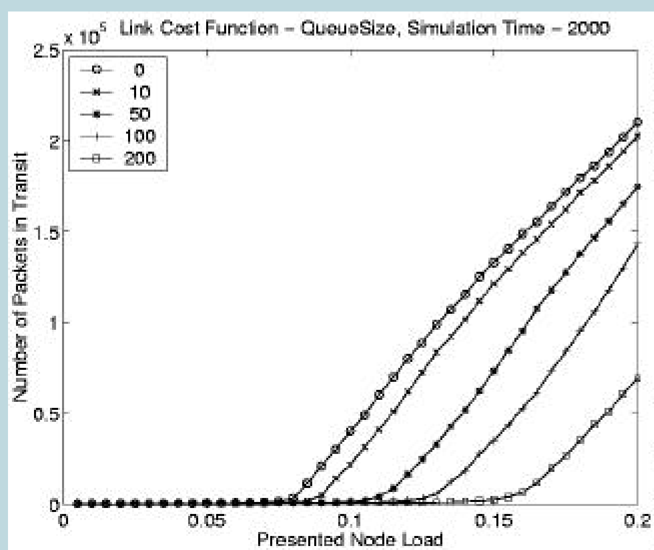

Figure 8: Number of Packets in Transit as a function of presented node load for different numbers of added randomly generated links.

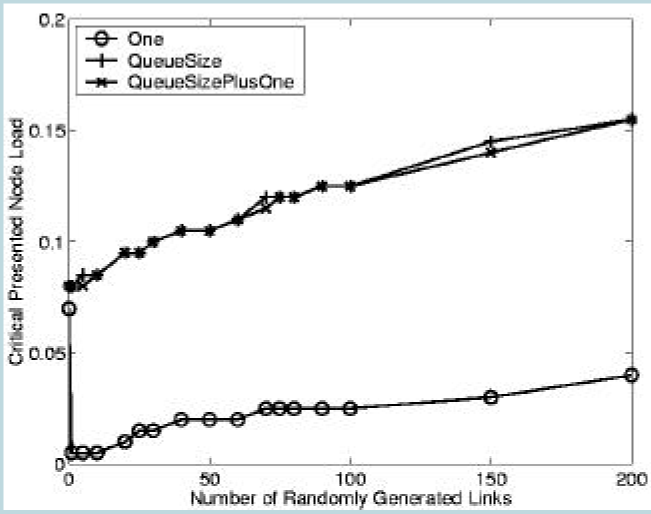

Figure 9: Estimated values of critical presented node load as a function of the number of randomly generated links for the three link cost functions considered.

4.0 Selected Simulation Results

We conducted a large variety of data network simulation experiments with the DataNetwork.exe tool. We first want to stress that the dynamics of a simulated network using the centralized full table routing algorithm turned out to be very similar to the behavior of the network with the same network topology which instead employs the distributed full table routing algorithm.

However, in terms of simulation costs (CPU time) the distributed algorithm is much more cost effective. In this presentation we describe only some of the results obtained in the simulations with the centralized full table routing algorithm. All simulations are performed for 2000 discrete time steps on a periodic square lattice with 25 nodes in each spatial direction. We add randomly generated links to this regular network in a manner such that the set of links of a network with l_1 randomly generated links is a subset of the set of links of the network with l_2 randomly generated links for all $l_2 \geq l_1$. We study the influence of the addition of randomly generated links on the performance the network. To this end, we monitor some characteristic network quantities during the time of simulation. These are, beside some others, the Number of Packets in Transit (i.e., the number of all packets in the network which have not yet arrived at their respective destination node) and the Average Delay Time (i.e., for all packets, which have reached their destination since the simulation was started, we form the sum of their delay times and divide by the number of those packets; the delay time of a packet is the simulation time elapsed between creation of the packet at its source node and arrival at its destination node).

Our main interest is to determine the maximum admissible presented node load (the critical presented node load) for a network configuration such that the flow in the network does not become congested. We decide on this critical node load value by considering the time evolution of the Number of Packets in Transit in the network. A value of the presented node load is called sub-critical if the Number of Packets in Transit is constant (beside some fluctuations caused by the randomness in the network) after a transient phase at the beginning of the simulation. If the Number of Packets in Transit increases with time then the presented node load value is called super-critical. The typical behavior of the Number of Packets in Transit as a function of simulation time for a sub- and a super-critical presented node load value is shown in Figure 5.

We now consider the Number of Packets in Transit at final simulation time as a function of the presented node load. Figure 6 gives the resulting graphs for the simulations with link cost function "One" for 0, 10, 50, 100, and 200 added randomly generated links. We clearly observe degradation of the networks performance if less than 200 additional randomly generated links are added. The reason for this behavior is that the additional randomly generated links provide short-cuts which lead to reduced path costs if packets utilize these links. Hence, many packets are attracted to nodes from which extra links originate and subsequently these nodes become congested (only one packet can be dealt with at any time step), leading to the increase of the number of undelivered packets in the network. It is important to note that not the whole network is congested but only areas around the origins of additional randomly generated links.

This can be deduced by looking at the Average Delay Time instead of the Number of Packets in Transit (not presented here) and more clearly seen by looking at the actual queue sizes at the nodes, see Figure 7. A consequence of the described behavior is that the critical presented node load for networks with cost function "One" drops sharply as soon as the first extra link is added and grows only slowly as we add more and more randomly generated links, see Figure 9.

The situation is improved if we change the link cost function to "QueueSize". Now the queue size is taken into account when making the routing decision and packets can circumvent possible congested areas of the network. The result is that the additional links are now utilized by the packets but they do not attract more packets than can be transmitted. This results in a network behavior which is "monotone" in the number of additional randomly generated links, see Figure 8 for plots of the Number of Packets in Transit as a function of the presented node load. Also, we see from Figure 9 that we can present a much higher node load to a network with link cost function "QueueSize" (compared to link cost function "One") without causing congestion.

However, there is one drawback with this kind of link cost function: the packets have no "incentive" at all to reach their destination; they just try to avoid congestion. This results in increased delay times for the packets. This shortcoming, which is especially notable for small values of presented node load, is overcome by the third link cost function "QueueSizePlusOne".

If the link costs of a network are assigned according to the link cost function "QueueSizePlusOne" then both, the number of hops of a path in the network and the queue sizes of the path nodes, contribute to the cost of this path. Therefore, for low presented node load path costs are dominated by the number of hops, and consequently packets travel along paths with a minimum number of hops to their destination. This leads to small delay times of the packets. Otherwise, as the presented node load increases, the sizes of the queues are growing and therefore the cost of a path is dominated by these queue sizes. The result is that packets avoid congested areas (as it is the case with link cost function "QueueSize") and travel around them. This avoids over utilization of the randomly generated links (short cuts). The graphs of the Number of Packets in Transit as a function of the presented node load for networks with link cost function "QueueSizePlusOne" are very similar to those in Figure 8.

Therefore we do not give these plots here. Further, from Figure 9, we see that the critical presented node load for networks with link cost functions "QueueSizePlusOne" and "QueueSize" are almost identical. The advantage of "QueueSizePlusOne" is the improved performance of the network for low values of presented node load.

5.0 Final Remarks

This paper has presented a methodology to model flow and congestion in packet switching networks. DataNetwork.exe is the software tool developed to conduct this work and is the most recent of a series of software tools developed by Dr. Lawniczak's research team. Further research is being conducted to study various aspects of packet switching networks such as data traffic type (short message text (SMS), video, etc.) and network behavior under various levels of degradation. Extensions to wireless networks are being planned. The software tool is being improved to handle more complex models.

Also, we remark that significant performance gains can be observed for the shortest path reduced table routing (where each node holds routing information for a part of the network's nodes only), even when the cost of each link is one, or for the routing based on "geometrical distance" [1,2].

Interested readers wishing to know more and to be kept informed about future work should email to alawnicz@uoguelph.ca.

6.0 Acknowledgement

A.T. Lawniczak, Peng Zhao and Alf Gerisch acknowledge partial support from the University of Guelph. A.T. Lawniczak acknowledges additionally partial support from the Natural Science and Engineering Research Council (NSERC) of Canada and The Fields Institute for Research in Mathematical Sciences. Alf Gerisch acknowledges additionally partial support from The Fields Institute for Research in Mathematical Sciences. Bruno Di Stefano acknowledges total financial support from Nuptek Systems Ltd. The authors acknowledge the use of the SHARCNET computational resources at the University of Guelph. The authors thank Dr. H. Fuks for helpful discussions.

7.0 References

[1]. H. Fuks and A.T. Lawniczak, "Performance of data networks with random links", Mathematics and Computers in Simulation, 51, 1999, pp. 101-117.

[2]. H. Fuks, A.T. Lawniczak, and S. Volkov, "Packet Delay in Models of Data Networks", ACM-Transactions on Modeling and Computer Simulation, Vol. 11, No. 3, July 2001, pp. 1-18.

[3]. W. Stallings, "High-speed networks: TCP/IP and ATM design principles", Prentice Hall, New Jersey, 1998.

About the authors

Prof. Anna T. Lawniczak holds a PhD in Mathematics from Southern Illinois University (USA) and a M.Sc.Eng. from Wroclaw Technical University, Poland.

Since 1989 she is an associate professor at the University of Guelph. Dr. Lawniczak has held a number of visiting positions at various institutions (i.e. Los Alamos National Laboratory, and The Fields Institute for Research in Mathematical Sciences, University of N. Carolina, Bartol Research Institute at the University of Delaware, the University of Roma, etc).

She conducts research in the areas of Mathematical Modelling and Simulation of Dynamics & Complex Systems.

Peng Zhao holds a M.Eng. from the Xi'an University of Technology (PRC), specializing in Automatic Control. He is currently completing a M.Sc. in mathematics at the University of Guelph and will soon defend a thesis on flow and congestion in data communication networks. His advisor is Prof. Lawniczak. Peng intends to continue his studies with a Ph.D.

Alf Gerisch holds a Ph.D. in mathematics from the University of Halle-Wittenberg (Germany), a M.Sc. (Numerical Analysis & Programming) from the University of Dundee (UK). He holds a post-doctoral fellowship, jointly at the University of Guelph and at The Fields Institute. He collaborates with Prof. Lawniczak, conducting research to model flow and congestion of data communication networks and the dynamics of semiconductor nanostructures.

Alf has several papers on numerical methods and their applications to the simulation of tumor angiogenesis and taxis-diffusion-reaction systems.

Bruno Di Stefano, P.Eng., is president of Nuptek Systems Ltd., a consulting company specialising in real-time embedded systems, fuzzy logic, OOA/OOD, C/C++. Previously, he was a senior design engineer for AES Data Ltd. and Delphax Systems.

His professional activities include teaching within the Professional Development Program of Engineering at the University of Toronto since 1986 and previously with Ryerson Polytechnic University. Bruno is also active with the IEEE and with PEO, holding various positions since 1979 and 1992 respectively.

IEEE Canada

Guidelines for submitting an article to the IEEE Canadian Review

Contact any one of the Associate Editors (address information is available on page 2 of this journal). A short abstract should be sent in and will be reviewed before approvals are given for a full paper submission.

Follow these broad-based guidelines for paper preparation:

- Papers (in English or French) must be on topical subjects of interest to the vast majority of our members across Canada,
- Papers selected for the CR are not the type that would normally be submitted to a Conference or Transactions/Journal type of publication. Any mathematical material must be kept to a minimum,
- Papers are normally about 4 pages long (about 2000 words), but articles can be as short as 1 page or as long as 6 pages,
- Submit text in ascii (or WORD) format, as well as a typeset version in pdf format,
- Graphics and images must only be in two colors (Black and Reflex Blue) or Grayscale; Graphics should be scanned at 300 dpi, in 8 bit grayscale, and submitted electronically in either jpeg or gif formats,
- A short Abstract (about 200 words) in English and French (we can provide assistance in translation) is required,
- A short biographical note (about 150 words) about the author(s) is also required, together with a passport size photo,
- For samples of previously published articles, see our website at www.ieee.ca, and follow the links to the *IEEE Canadian Review.*
- Items are normally submitted on-line directly to the Editor.
- We particularly encourage articles from small businesses on topics in electrical, electronic, computer or telecommunications related industries etc.
- We encourage academic institutions to present their new programs so that the community is aware of recent program changes,
- Newsworthy items such as senior appointments in academia and industry are also welcome.

For further information, contact Vijay Sood, Managing Editor, at:

v.sood@ieee.org

Phone: 450-652-8089