Statistical and Simulation Analysis of Photonic Packet Switching Networks - The Emerging Functions Concept

Antonio de Campos Sachs, Ricardo Luis de Azevedo da Rocha, Fernando Frota Redígolo, and Tereza Cristina Melo de Brito Carvalho
Departamento de Engenharia de Computação e Sistemas Digitais (PCS)
Escola Politécnica da Universidade de São Paulo (EPUSP)
São Paulo, Brazil
antoniosachs@larc.usp.br; luis.rocha@poli.usp.br; fernando@larc.usp.br; carvalho@larc.usp.br

Abstract—**A transparent Optical Packet Switching network designed with Emerging Functions Concept is analyzed as a complex system showing desirable characteristics that emerges from the bottom-up organization. The objective is to describe the Emerging Functions Concept applied to different topologies. The bottom-up organization is obtained from simple rules executed by individual nodes. It is possible to create those simple rules, or fundamental individual functions executed by individual nodes, in order to potentiate scalability, robustness, and other desirable characteristics referred to as Emerging Functions. The scalability is enabled after the avoidance of all long distance signalizations. The robustness emerges from next neighbor signalizations and from the mesh topology with a large number of alternative paths. All emerging functions are better observed for networks with more nodes. The concept is described for the Manhattan Street Network to show the emergence of scalability and robustness. It is also applied to the National Science Foundation Network in order to generalize the procedure showing its applicability to nonsymmetrical and more real topologies. Failure effect segregation is shown for 256 nodes Manhattan Street Network.**

Keywords-complex network; emerging function; scalability; robustness; nondeterministic physical layer

I. INTRODUCTION

One important constraining factor for the scalability of the number of nodes in a network is the long time necessary for signalization between two distant nodes. The current approach, which treats the Optical Packet Switching (OPS) network as a complex system and the network nodes as autonomous entities, utilizes the Emerging Function Concept (EFC) described in the EMERGING-2012 international conference [1]. That approach avoids long distance signalization. A packet is sent from source to destination without any previous path determination. The routing procedure needs to use the shortest path table previously calculated at the moment of the network initialization. From that shortest path table, each node knows the address of the output port that corresponds to the shortest path connecting itself to any other node. Each packet carrying the destination address can find the path from source to destination from node to node in a multi hop schema using the output port corresponding to the shortest path or the alternative port in those cases in which the preferred one is not available.

The use of a simple switching device without optical buffers that forwards the arriving packet without delay to the preferential output port or to the alternative one is a procedure referred to as "hot potato routing" [2]. That operation can be performed by using an optical sample removed before a FDL (Fiber Delay Line). This sample can be converted into electrical media enabling the logical treatment that is performed by conventional electronic circuitry. The optical switch can easily be constructed based on SOA (Semiconductor Optical Amplifiers) devices [3]. The optical switch can be positioned to address the packet to the correct output port before the arrival of the packet that is traveling through the FDL. Such operations have been adopted since the precursor projects KEOPS [4] and DAVID [5].

There is an option to avoid the conversion from optical to electrical media that consists in the utilization of new photonic devices that can do all the jobs, including logical operations. With those photonic devices the switching operation can be performed in a fully optical process [6]. The network described herein works for any technology employed for reading the address and forwarding the packet to the output port. Whatever the technology used inside the node, the network can operate as a complex system, with a bottom-up organization. Each node has the autonomy to carry on the switching operation, performing its work exclusively with information locally obtained.

The use of a large number of nodes with a large number of alternative paths, provided by the mesh topology, is known to be important for the network survivability. Since the beginning of the digital telecommunication technology Baran [2] worked with mesh topology and got very strong robustness for a network with a large number of nodes. The survivability of a complex network is associated to the intrinsic robustness of complex systems. Carlson and Doyle [7] claim that all complex systems are intrinsically robust for the most frequent daily events; however they are very fragile due to rare and unexpected environment events. Barabási [8] claims “hubs make the network robust against accidental failures but vulnerable to coordinated attacks”. All agree that complexity is intrinsically related to robustness.

In addition to robustness, or acting together with the robustness, the self-organization is another property associated to a complex system [9]. With the self-organization ability the network gains autonomy and can reduces the external management effort. That characteristic is referred to as Autonomic Network Management (ANM). A survey into ANM is presented in [10]. The present work contributes to the ANM effort with the creation of a pure physical layer mechanism, that provides traffic self-distribution, scalability, self-protection and restoration.

Self-organization or bottom-up organization is a convenient and necessary approach to deal with large size networks [11]. Not restricted to the future, the complexity is already a reality at the optical metropolitan networks that is characterized by large number of nodes necessary for the capillarity, accessibility and high capacity. The scalability of those networks is limited by the utilization of a strictly deterministic approach. The present proposal brings the non-determinist behavior to the physical layer and calls attention to the convenience of using the complexity approach in all network layers.

Next, Section II describes aspects related to the EFC and how it can be applied to a network. Section III describes the network architecture and the operations responsible for the packet insertion without collision and for the protection and restoration properties. Section IV presents a generalization that shows the applicability of the EFC to virtually any topology. Section V describes the theory adopted for the statistical model and aspects of calculation performance. Section VI describes the simulation performed to validate the results achieved by the statistical model. Section VII presents discussions about the failure distribution effect for a 256-node study case. Section VIII presents the final conclusion and future work.

II. Emerging Function Concept Applied to a Network

The term Emerging Function is used in a number of different areas, such as physics, chemistry and biology. Although there is no single formal definition for the term, two main definitions can be inferred:

- A function that is not regularly present in a system and appears or is activated automatically in an emergency situation;
- A function that is always present in a system (it characterizes the system) and emerges from simple operations executed by its individual parts.

An emerging function is associated to the whole system and not to its individual parts although its emergence is the result of small changes in the normal operations (first definition) or of regular operations of individual parts of the system (second definition).

A system based on emerging functions can be classified as a bottom-up organized system or, equivalently, a self-organized system [9] and it is associated with a complex system composed by a large number of individual units following simple operation rules. It is difficult to deal with such complex systems, with a large number of elements, just employing a classical and reversible treatment that calculates all the possible events in all the system components. The models considering the probability of transition from one state to the next describing the system evolution seem to be a more feasible strategy. That is also the same strategy found in the chaos theory [12], where the final results cannot be derived from the initial conditions because there is high sensitivity to tiny fluctuations in the initial conditions.

The network routing layer (OSI network layer 3) does not control the routing function herein. It emerges from simple fundamental functions executed by each node individually. There is no high level entity responsible for the operation of the switches or for the path followed by each packet in the network. Instead, the node operation is based on the local situation and on the packet header information: each packet is sent to the preferred output port, or sent to the available port if the preferred one is busy. This operation rule, by itself, turns the network self-organized or bottom-up organized, and provides autonomic network operation. Therefore, it is possible to consider "routing" as a function emerging from individual node operations or, in other words, that routing is an Emerging Function.

Traffic distribution, which can be considered the set of all routes, is also an Emerging Function. As the shortest path is not always the one chosen, the traffic distribution obtained is better than the one obtained using only the shortest path.

The access to the network is made only if there is a time interval to accept the new packet without collision. This is possible because of a fiber delay line (FDL) positioned before any input port. Collision avoidance can also be interpreted as an Emerging Function, since it is not executed by any higher protocol layer, but it shows up after the careful local insertion procedure.

Protection is an important network function that can be enabled by means of the insertion of an extra individual node operation function based on a backward signalization sent to all the input ports. The output ports integrity can be checked through the signalization received from the next node. Protection can also be considered as an Emerging Function.

III. Network Architecture and Operation

The network architecture is based on the "Hot Potato Heuristic Routing Doctrine" [2] made up by network nodes executing simple well-defined rules. A set of Emerging Functions arise from those simple rules. The network complexity is related to its size and the number of nodes. Each node, in contrast, is idealized to be simple. The first simplification is the omission of optical buffers. Without optical buffers, it is necessary to use symmetrical nodes in order to avoid packet losses. In symmetrical nodes, with the same number of input and output ports, there is always a free output port for any arriving packet. The simplest possible symmetrical node to be connected in a mesh topology is a 2x2 switching node with two inputs and two outputs as shown in Fig. 1. Without optical buffers, the optical packet is immediately forwarded to the output port and the front part of the packet starts traveling through the next link while the back part of the packet is still arriving to the node input port. An eventual packet P2, arriving when the node is occupied, is immediately forwarded to the available output.

The Manhattan-Street Network (MSN) [13] was chosen as the main topology for the scalability analysis, but any other mesh topology can be considered. This particular choice facilitates the calculations for increasing the number of nodes without changing the network symmetry. As an example of applicability to any topology the EFC was also applied to the National Science Foundation Network (NFSNet) [14].

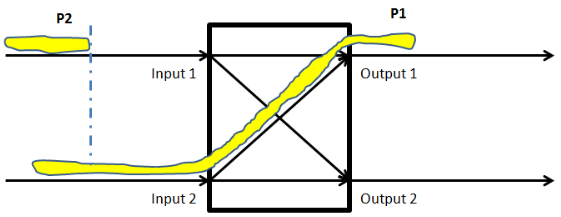


Figure 1. Simple optical switching node with two inputs and two output without optical buffer.

A. *Packet insertion without collision*

Each packet goes through a Delay Line Fiber (DLF) before arriving to the optical switch. An optical splitter takes a sample of the optical signal before the DLF. That sample is analyzed by a logical circuitry that accounts for reading the packet header, consulting a previously stored routing table and sending a signal to the optical switch control. The optical switch is prepared for dealing with that particular packet before its arrival. In addition to providing a secure time interval for the switching positioning the DLF has a second important function that consists in a sight of the near future that permits the insertion of new packets without collision. Fig. 2 illustrates the DLF functionalities. A small time interval, represented by t_1 in Fig. 2, is necessary for the header reading, logical procedure and switch positioning. That time interval is very small. One hundred nanoseconds should be enough for t_1. It is required a longer time interval for introducing a new packet in the network. The remaining time t_0-t_1, where t_0 represents the total time for the light to travel inside the DLF, defines the maximum packet size that can be inserted into the network without collision. For example, in a 10 Gb/s optical link, one single bit spends 0.1 ns in propagation and 12 thousand bits (1500 bytes) would require 1.2 µs. Taking t_0-t_1 equal to 1.2 µs and t_1=100 ns this results in t_0=1.3 µs. Thus the DLF, in this case, should be 260 meters long. A longer DLF can be adopted for jumbo packets or for a set of packets put together into a burst in an Optical Burst Switching (OBS) technology.

Concerning the example in Fig. 2, after time interval t_1, the node already knows which output port the packet is going to use. Given that P_3 arrived before P_1, it had priority and chose the port first. In that case, P_3 is going to use output 2 and P_1 is going to use output 1. In that situation, there is enough time for the insertion of a new packet P_2 with size t_2 < (t_0-t_1). P_2 can be inserted directly into the output 1 link and can be completely inserted before the arrival of P_1.

Occasionally, packets need to be stored for a long period of time before finding an appropriate time interval to be inserted. In this case, several packets can be lost at the full buffer condition. The buffer has a finite size and need to discard packets that arrives in that full condition. In order to minimize such losses, the suggestion is to use the FILO (First In Last Out) buffer strategy that will avoid small packets disposal while a long packet is waiting for a long and rare free time interval to be inserted.

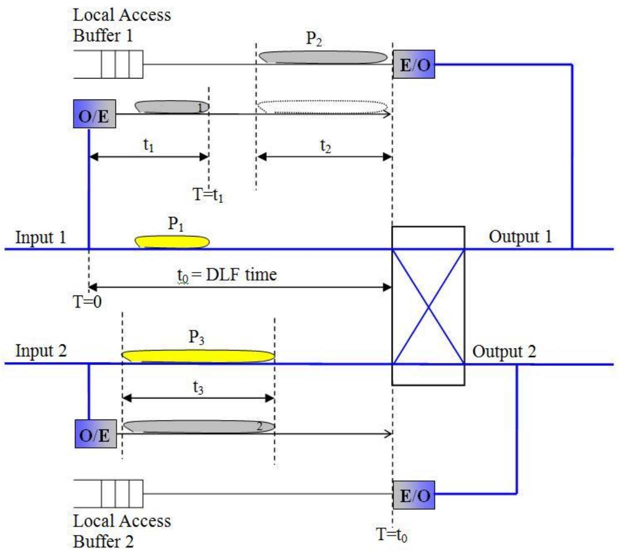


Figure 2. Delay Line Fiber (DLF) architecture.

B. *Protection emerging function*

The implementation of the protection emerging function requires the differentiation between the two output ports in order to define different link sub-domains. The idea is to deactivate only one sub-domain in the case of link failure. Fig. 3 is a MSN with 36 nodes showing clockwise and counterclockwise sub-domains as first described in [15]. Each node belongs to two sub-domains and each sub-domain contains four nodes. It is desirable to have a small number of nodes in each sub-domain because an entire sub-domain needs to be deactivated to deal with the failure.

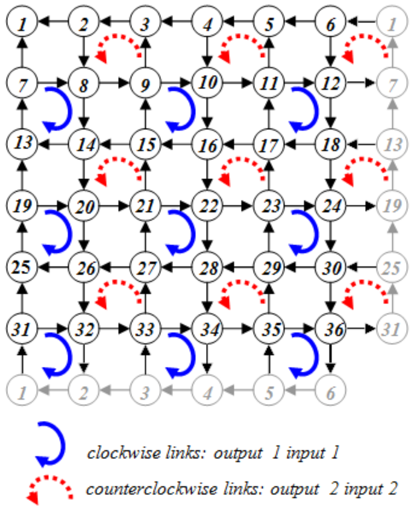
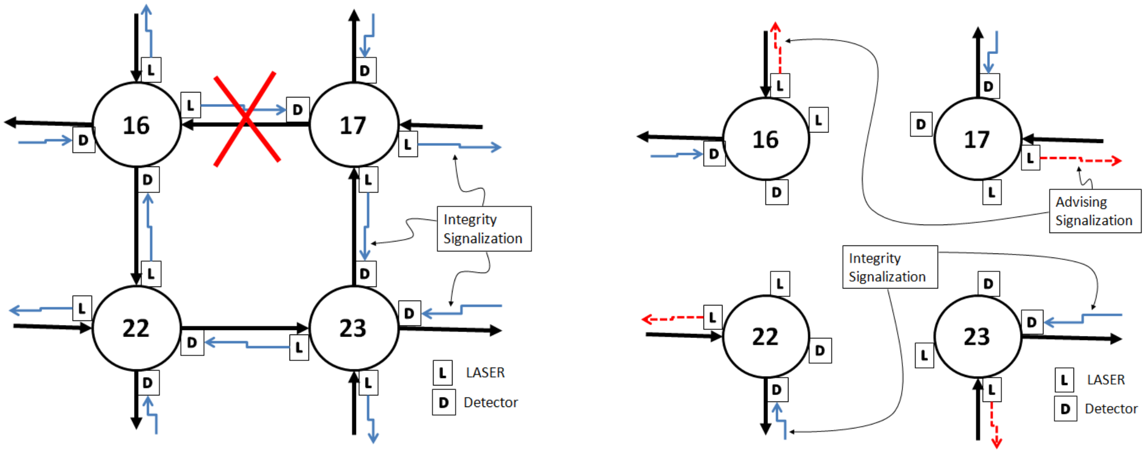

clockwise links: output 1 input 1
counterclockwise links: output 2 input 2

Figure 3. MSN organized with clockwise and counterclockwise link sub-domains [15].



(a) Before failure

(b) After failure

Figure 4. (a)Before failure, all nodes send a backward (opposed to the packet direction) Integrity Signalization. (b)After failure the Integrity Signalization is turned off for all links belonging to the failed sub-domain. The signalizations addressed to the four next neighbors that can access the failed sub-domain are changed to a second type of signalization called Advising Signalization represented by dashed red line arrow (see text for more details).

After organizing the network links in small sub-domains, it is possible to create the protection function by including an operation rule for all network nodes. This rule is constituted by a continuous optical signal sent backward from all the nodes. That signal is denominated Integrity Signalization. It is received by the backward next neighbor indicating that the corresponding output is properly connected. The Integrity Signalization can be implemented by supplying each node with two lasers and two photo-detectors as shown in Fig. 4.

Nodes 16, 22, 23 and 17 are connected by four links in a counterclockwise sub-domain as shown in Fig. 4 as a zoom of the same nodes in Fig. 3. Each node has two lasers sending a continuous optical signal backwards (opposite the packet directions), and two detectors placed to receive the laser signals from the downstream neighbors. In the case of one link interruption, the detector that first stops receiving the signal turns off the laser corresponding to the same sub-domain. In Fig. 4, for example, a failure occurred in the link that connects node 17 to node 16. After the failure, node 17 does not receive the integrity signalization and stops sending packets through that link and also turns off the laser corresponding to node 23 that belongs to the same sub-domain as node 16. Node 23, in turn, after stopping receiving the integrity signalization from node 17, turns off its laser addressed to node 22. This one does the same and the final result is that all the four links in the ring are forced to be interrupted. The four lasers are turned off and, without the Integrity Signalization, no packet can be sent through those links.

The integrity signalization is enough for the protection schema operation, but a second type of signalization has been implemented aiming at better network performance. That second type of signalization consists in sending a different type of signal to the nodes outside the failed sub-domain to inform that the next node belongs to a sub-domain in failure, although the link still works properly. The implementation of that advising signalization can be performed by a square wave light signal replacing the continuous light signalization or, alternatively, the continuous laser can be used with half optical power to differentiate from the full optical power of the regular link integrity signalization. The implementation of both signalization types, indeed, can also be implemented by the utilization of smart photonic devices transmitting digital signalization and processing the signal in a fully optical process.

In the case of Fig. 4 (right side figure), the failure, at the same time, causes four links to stop the integrity signalization and also to start the advising signalization outside the failed sub-domain. From Fig. 3, it is possible to recognize that the nodes receiving the advising signalization are nodes 18, 10, 21 and 29.

The action of the node receiving the advising signalization is to deflect all packets to the other output port (to the port that is receiving the integrity signalization), with the exception of the packets addressed to the node that is sending the advising signalization. This is the only way a failed sub-domain node can receive a packet. That deflection corresponds to an adaption in the local preferential port table. All the nodes at a failed sub-domain, nevertheless, continue to work with only one input and one output port. In that situation, they continue to be symmetrical (one input and one output) and can transfer all the packets arriving from the input to the always available output port. All the other nodes, far from the failure, have no information about the failure and their procedure remains the same, including the use of the same preferential output port matrix.

Both modes of signalization were implemented in the calculations, and the results are shown in Fig. 5 for 16 nodes (N=16) and for 256 nodes (N=256). The caption ending in “F” refers to the first level protection schema, characterized by not using the advising signalization type. The caption ending in “F2” refers to the second level protection schema that includes the advising signalization type. For the 16-node case, the second level protection schema (N=16F2) shows an interference of the failure remarkably smaller than that observed at the first level protection schema (N=16F). The advising procedure reduces the mean number of hops.

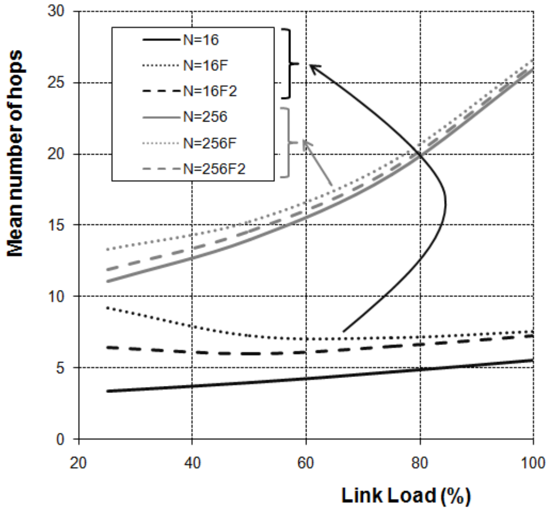


Figure 5. Mean number of hops degradation after failure for two types of signalizations.

One additional feature obtained with the second type of signalization is the correction of a strange behavior that occurs for low charge condition. In that region (Link Load < 50%) the failure causes a large enhancement in the number of hops and it is quite odd to see the number of hops decreasing for higher load condition (curve N=16F in Fig. 5). That behavior can be explained by the fact that in the low load condition, the packets take the preferential output port more often as compared to the large load condition. Consequently, the packets are more often forced to proceed through the failed region. At the large load condition, the packets are naturally dispersed and the failure does not cause too much degradation to the number of hops. The failure is more efficiently avoided with the second signalization, minimizing this effect. All the unnecessary trial through the failed region is avoided.

C. *Restoration emerging function*

After failure, the use of the same preferential output port matrix causes performance degradations. The network, however, can operate under acceptable condition, while waiting for physical reparation. After reparation, in order to get an automatic restoration, it is necessary to implement additional fundamental functions, to be executed by all nodes, only in the emergency case. First, the nodes involved in the failure need to change their states from “normal” to “emergency”, and, in that new state, it is prepared for the restoration. Second, a node in the “emergency” state must restart its backward integrity signal to reset the system after physical reparation of the failure.

Based on these fundamental functions, once the link is repaired, the integrity signal is propagated backwards, restoring the sub-domain. Again, the functions executed by individual nodes are responsible for the emergence of the global function restoration. That means that restoration is also considered an Emerging Function. It is important to stress that those functionalities can be easily implemented in practice.

For long emergency state time, it may be necessary to consider a network reset to get a new preferential output port matrix for the new topology (topology with failure and disabled sub-domain). In this case, it is convenient to store the old preferential output port matrix or to reset the network again after failure reparation. The restoration function should recover the original matrix as soon as it receives the integrity signal from another node and a new signalization should inform the entire network to reset or to recover the original topology matrix. That operation, including signalization for the entire network (twice), cannot be considered as Emerging Function because it is not carried out only by localized operations. It takes too much time and causes scalability limitations.

IV. Generalization

Optical network topologies, in real world, are not like the MSN but something more like the NFSNet (National Science Foundation Network) [14] that is shown in Fig. 6. The NFSNet has bidirectional links and each connection represents two optical fibers, one input and one output. Most of the nodes have three inputs and three outputs. Nodes 4 and 10 have four bidirectional connections and nodes 5 and 12 have only 2 inputs and 2 outputs. This network can be transformed into an equivalent representation that exhibits only 2x2 switches as described in sub-section A or it can be treated considering nodes 3x3 and 4x4 according to the discussions presented in sub-section B.

A. *Representation using only 2x2 nodes*

The same procedure employed for the MSN can be adopted for NFSNet by reconstructing all 3x3 and 4x4 nodes, present in the NSFNet, as a set of unidirectional 2x2 nodes. Each three bidirectional connection node can be represented, as shown in Fig. 7a, by three unidirectional 2x2 nodes. Generalizing this idea, each node with k bidirectional connections can be replaced by k unidirectional 2x2 nodes. After reconstruction 42 nodes containing only 2x2 optical switches as shown in Fig. 7b can represent the NFSNet .

The definition of link sub-domains, applied to MSN for developing the protection emerging function, can also be applied to NFSNet. The main idea is to choose two types of sub-domains such that each 2x2 node has one input and one output connected to the first type and the other pair input/output connected to the second type. Another idea to be reassigned is related to the size of each link sub-domain. Each sub-domain needs to be small because it is going to be deactivated in case of failure. The smaller the sub-domains, the better the network performance after failure. After those considerations, the protection can be applied to NFSNet using the same backward signalization adopted for the MSN, and the sub-domain types can be considered separating long-distances and short-distances links.

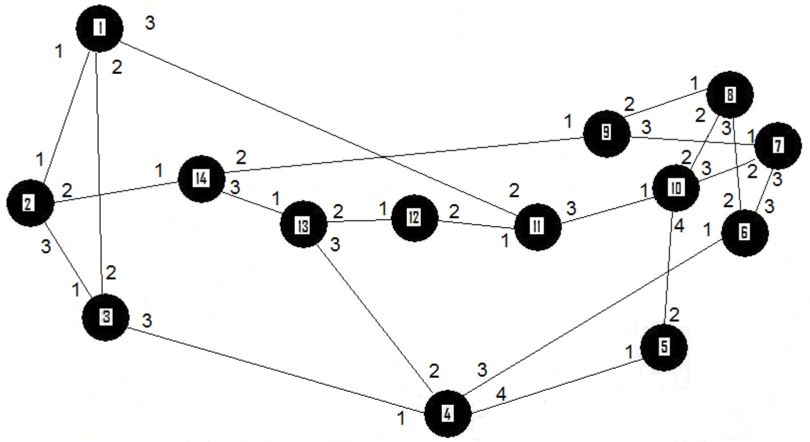
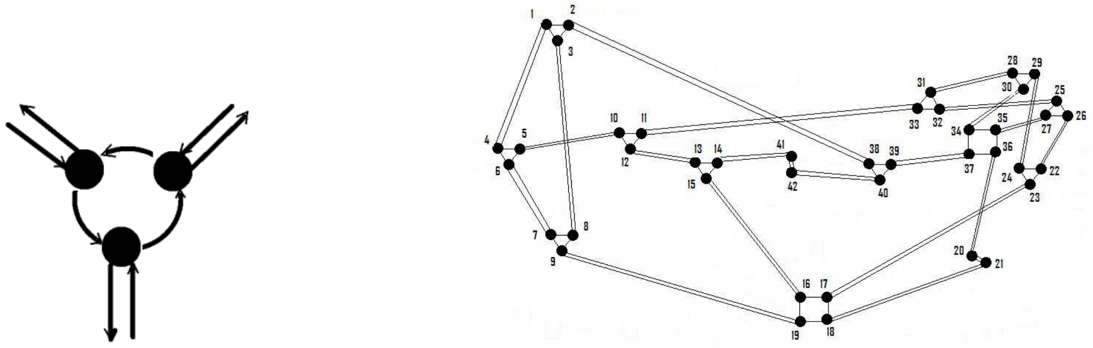


Figure 6. NFSNet with 14 nodes. All links are bidirectional.



(a)

(b)

Figure 7. Three 2x2 nodes replacing one node with three bidirectional connections (a) and the NFSNet designed with 42 nodes (b). A simple optical switch with two input ports and two output ports represents each node.

As illustrated in Fig. 7a, all nodes have a pair input/output for the long distance links and a second pair for the short distance links. The separation of groups of sub-domains to be disconnected in case of failure considers that all long distances links belong to one type of sub-domain and all short distance links constitute the second type of sub-domain. Each node belongs to two types of sub-domain and all the long distance sub-domains have only two links, while the short distance sub-domain can have 2, 3 or 4 links depending on the case.

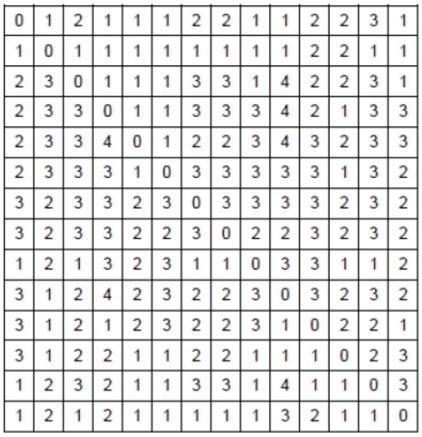
After the transformation of the original NFSNet (Fig. 6), by using the alternative shown in Fig. 7a, it results in a network with 42 nodes as shown in Fig. 7b.

B. *Generalization for NxN node type*

It is also possible to consider that all 2x2 nodes physically located in the same place can, almost instantly, have the same information. With the knowledge of the arriving packet and its destination, the network performance seemed to be better, but the calculations may be harder to be implemented because it is necessary to find out at least two better output port options to send each packet. Considering the original NFSNet topology (Fig.6), with 2x2, 3x3 and 4x4 nodes, it is necessary to find out a second preferential output port for all 3x3 and 4x4 nodes. It may not be important to deal with the third preferential port because in most cases it is the last available port and in the 4x4 nodes, the third option can be considered to be the path going back to the same place it came from. If the smallest possible path (first option) has one hop, the second option can have two hops and considering the third option to be returning to try again, it has only three hops. In some cases the way back (return to try again) is the second option.

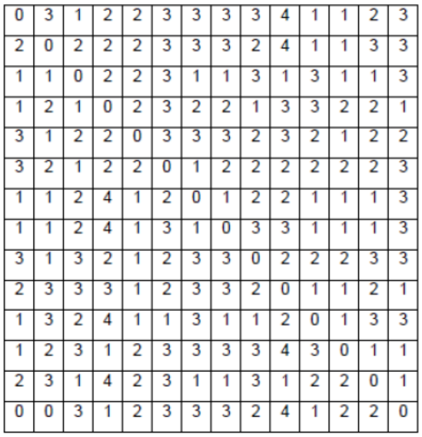
After those considerations, it is possible to construct a table of preferred output port just for the first and second preferential output ports. Table I shows an example obtained from the NFSNet represented in Fig. 6. Columns represent the actual position of a packet, lines represent the final destination and the numbers represent the first or the second options output port. The number of each output port can be confronted with the numbers presented in Fig. 6. For example, from node number 1 (first column) to node number 3 (third line), the first option is port number 2 (Table Ia) and the second option is port number 1 (Table Ib).

TABLE I
First option (a) and second option (b) for sending packets from any of the 14 nodes in the NFSNet shown in Fig. 6. Columns 1 to 14 represent the actual position of a packet, lines 1 to 14 represent the final destination and the numbers inside the table are the first or second options output port numbers.



0	1	2	1	1	1	2	2	1	1	2	2	3	1
1	0	1	1	1	1	1	1	1	1	2	2	1	1
2	3	0	1	1	1	3	3	1	4	2	2	3	1
2	3	3	0	1	1	3	3	3	4	2	1	3	3
2	3	3	4	0	1	2	2	3	4	3	2	3	3
2	3	3	3	1	0	3	3	3	3	3	1	3	2
3	2	3	3	2	3	0	3	3	3	3	2	3	2
3	2	3	3	2	2	3	0	2	2	3	2	3	2
1	2	1	3	2	3	1	1	0	3	3	1	1	2
3	1	2	4	2	3	2	2	3	0	3	2	3	2
3	1	2	1	2	3	2	2	3	1	0	2	2	1
3	1	2	2	1	1	2	2	1	1	1	0	2	3
1	2	3	2	1	1	3	3	1	4	1	1	0	3
1	2	1	2	1	1	1	1	1	3	2	1	1	0

(a) First option



0	3	1	2	2	3	3	3	3	4	1	1	2	3
2	0	2	2	2	3	3	3	2	4	1	1	3	3
1	1	0	2	2	3	1	1	3	1	3	1	1	3
1	2	1	0	2	3	2	2	1	3	3	2	2	1
3	1	2	2	0	3	3	3	2	3	2	1	2	2
3	2	1	2	2	0	1	2	2	2	2	2	2	3
1	1	2	4	1	2	0	1	2	2	1	1	1	3
1	1	2	4	1	3	1	0	3	3	1	1	1	3
3	1	3	2	1	2	3	3	0	2	2	2	3	3
2	3	3	3	1	2	3	3	2	0	1	1	2	1
1	3	2	4	1	1	3	1	1	2	0	1	3	3
1	2	3	1	2	3	3	3	3	4	3	0	1	1
2	3	1	4	2	3	1	1	3	1	2	2	0	1
0	0	3	1	2	3	3	3	2	4	1	2	2	0

(b) Second option

Results for the mean number of hops as a function of the network load are shown in Fig. 8. The case of NFSNet with 14 nodes and 42 links as compared to a MSN with 16 nodes and 32 links is shoen. The NFSNet is better (uses less number of hops) because of the larger number of links available.

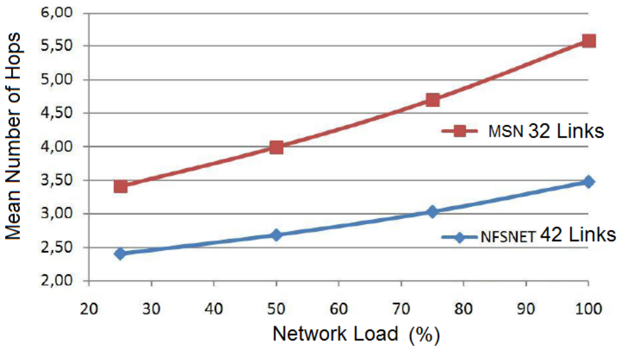


Figure 8. Mean number of hops of NFSNet 14 nodes and 42 links compared to MSN with 16 nodes and 32 links.

V. CALCULATIONS

To deal with scalability, the number of nodes can be higher than practical calculations can support. It is impractical to implement calculations for an arbitrarily large number of nodes. In order to minimize the time and memory used, the connection matrix "c" and the preferential output port matrix "pp", were calculated separately. Data were saved in files that could be interpreted by the main program. The algorithm employed to obtain the MSN connection matrix "c" is shown in sub-section A. The shortest path calculation is presented in sub-section B. The algorithm description for the mean number of hops calculation is presented in sub-section C. The model validation carried out by comparison with the simulation model is presented in sub-section C. One important result, the segregation of the failure effect, is presented in sub-section D.

A. Algorithm for MSN Connection Matrix

The connection matrix "c" for a MSN with N nodes is a matrix NxN where the columns represent all the N network nodes. Each column has only two non-zero element in the position corresponding to the two nodes that can be reached directly. Instead of using number "one" to represent an adjacent element as in an ordinary adjacency matrix, here two different values are used to represent two different sub-domains.

It is not necessary to find out a general formula for all MSN types. The goal is just to obtain a sequence of MSN types, each one with an increased number of nodes. That calculation was obtained by constraining the number of nodes "N" to those that are the square of an even number "n". With that restriction ($N=n^2$, n even), an algorithm to obtain the connection matrix for an arbitrarily large number of nodes was implemented. Following Fig. 3 for link sub-domain identification, the algorithm constructs a matrix that attributes the value "one" to the output port connected to the counterclockwise link sub-domain (red in Fig. 3) and attributes the value "two" to the output port connected to the clockwise link sub-domain (blue in Fig. 3). The code lines for the algorithm are presented below in a syntax used for the scilab simulation platform [16].

```
//OUTPUT PORT 1:COUNTERCLOCKWISE (RED)
  for i=1:2:n-1; //ODD LINES
    for j=2:2:n-1; //EVEN COLUMNS
    c((i-1)*n+j,(i-1)*n+j+1)=1; c(i*n+j,(i-1)*n+j)=1;
    c(i*n+j+1,i*n+j)=1; c((i-1)*n+j+1,i*n+j+1)=1;
    end;
    c(i*n,i*n-n+1)=1; c(i*n+n,i*n)=1;
    c(i*n+1,i*n+n)=1; c(i*n-n+1,i*n+1)=1;
  end;
```

```
//OUTPUT PORT 2:CLOCKWISE (BLUE)
 for i=2:2:n-1; //EVEN LINES
  for j=1:2:n-1; //ODD COLUMNS
  c((i-1)*n+j+1,(i-1)*n+j)=2; c(i*n+j+1,(i-1)*n+j+1)=2;
  c(i*n+j,i*n+j+1)=2; c((i-1)*n+j,i*n+j)=2;
  end;
 end;
 for j=1:2:n-1;
  c(j,j+1)=2; c(N-n+j,j)=2;
  c(N-n+j+1,N-n+j)=2; c(j+1,N-n+j+1)=2;
 end;
```

B. *Shortest Path Calculation*

The shortest path to reach the destination is calculated once for a non-failed MSN topology. As the packet can be deflected to any output port, it must be able to find out the destination shortest path from any place in the network and not only from the origin. The packet is informed about the shortest path through a preferential port matrix "*pp*" with dimensions *NxN*, where *N* is the total number of nodes. Each column of the "*pp*" matrix represents the actual position of a packet. The lines represent the final destination and the matrix elements are numbers indicating the best option: number 1 for output 1 or number 2 for output 2 or number 3 to indicate the "don't care" condition, in which there is a shortest path starting from both outputs.

The preferential port matrix "*pp*" is constructed from connection matrix "*c*". This is done column by column, in a nondeterministic procedure [17] just at the beginning of the algorithm. The procedure is nondeterministic because it is necessary to calculate the smallest path starting from all possible packet positions. From column number 1, which represents node number 1, a tree is established and the starting level is called Level Zero. Based on the Manhattan-Street network with *N* = 36, presented in Fig. 3, the calculation procedure considers the evolution of the tree presented in Fig. 9. Level 1 is formed by the two branches going down from node 1 to node 6 and from node 1 to node 31. Those are the two nodes directly reachable from node 1. The "*pp*" matrix lines 6 and 31 can now be filled with the preferential output port, respectively number 2 and number 1. Those numbers can be obtained from the already known "*c*" matrix for connections 1 to 6 and 1 to 31. The first connection (output port number 2) will be used by all nodes found on the left side and the second (output port number 1) will be the preferential port for all nodes found on the right side of the tree.

Following this procedure, the number of elements in each level would increase exponentially. That is a strong limitation for the number of nodes scalability. To fix that scalability problem, the adaptive tree mechanism was deployed [17]. In that mechanism, the tree is adapted at each level and all the nodes that were already used at earlier levels are removed. This is exemplified in Fig. 9, where node 1 was removed twice in Level 4 because it was already used in Level zero at the beginning. Several branches are removed in Level 5. For example, node 3 in Level 4 is connected to nodes number 2 and 33 (the connections can be identified in Fig. 3). Those nodes were already reached at Level 3. That adaptive procedure reduces the computational complexity from exponential to polynomial growth as a function of the total number of nodes.

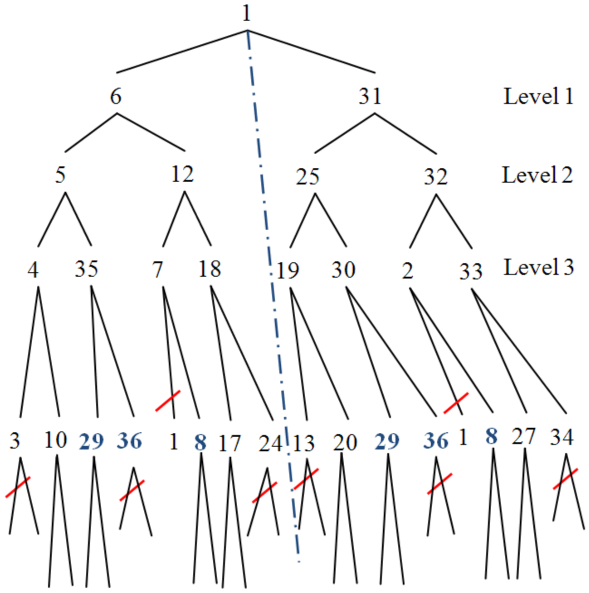


Figure 9. Adaptive tree for shortest path calculation.

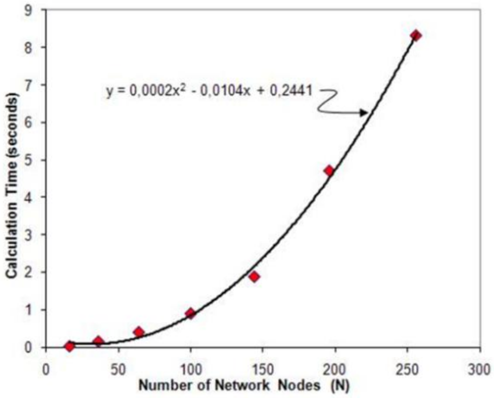


Figure 10. Calculation time to determine preferential output port.

In addition, the repetition inside the same level needs an extra verification. It is verified if they belong to a different half of the tree, which means that the destination can be reached with the same number of hops from any output port from node number 1. This is known as the "don't care" situation and the "*pp*" matrix element is set to be equal to 3.

The procedure continues until column one in the "*pp*" matrix is completed. And then the procedure is repeated for all the other columns, which represent all the possible origin nodes. The resulting time for "*pp*" calculation shows a quadratic polynomial growth with respect to the number of nodes as can be seen in Fig. 10, which was obtained by performing all calculations in a Pentium 4 personal computer with 3GHz CPU and 2GB memory.

There are other algorithms that could be used to find out the shortest path. Dijkstra's algorithm, most commonly used, would need to be modified in order to consider the "don't care" cases and not only the traditional shortest path first (spf) found. The following algorithm based on the adaptive tree (Fig. 9) is comparable to a modified Dijkstra's algorithm and can meet all needs for this case, including the registration of the don't care situations. The lines of the code presented here allow calculating preferential port matrix "*pp*" from any number of nodes *N* and for a MSN connection matrix "*c*" previously calculated as described in sub-section A. The code syntax is also adapted to the scilab platform [16].

```
    for      j=1:N;      fff=find(c(:,j));      ppp(fff(1),j)=0;
ppp(fff(2),j)=0; end;
     for ii=1:N;
    [w]=find(c(:,ii),2); per=[w(1) w(2)]; nivel=1;
    while sum(ppp(:,ii))>0; nivel=nivel+1;
     perpro=[];
     k1=sum(size(per))-1;
     for i=1:k1;    [w]=find(c(:,per(i)),2);
      for j=1:2;
        ori=ii; des=w(j); pai=per(i);
        if ppp(des,ori)==1;
         if pp(pai,ori)<>0;
          pp(des,ori)=pp(pai,ori);
          ppp(des,ori)=0;
          pniv(des,ori)=nivel;
          perpro=[perpro w(j)];
         end;
        elseif
pp(des,ori)<>pp(pai,ori)&pniv(des,ori)==nivel&pp(pai,ori)<
>0;
         pp(des,ori)=3; end;
      end;
     end;
     per=perpro;
    end; end;
```

C. Mean number of hops calculation

The network performance is measured by the mean number of hops $\overline{H}$ a packet completes traveling from origin to destination. The main program, used to calculate the main number of hops is based on the evolution of a vector *P(x)* with *N* dimensions. The *x* variable is the discrete position for the packet *(x = 1, 2, ..., N)*. Each vector represents the probability of finding a hypothetical packet in each node. That is called probability distribution vector. The mathematical treatment for the evolution of a probability distribution through time corresponds to the application of an operator *"U"* to probability vector $P_t(x)$ at any instant of time "*t*" to obtain probability vector $P_{t+1}(x)$ at the instant of time "t+1" after a discrete time interval. The unitary increment of time corresponds to one hop from one node to the following in the packet traveling from source to destination.

$$P_{t+1}(x)=UP_t(x) \tag{1}$$

Operator *U* is analogous to the "Perron-Frobenius operator" employed in the chaos theory for calculating the time evolution of a probability distribution [12]. An analogy can be constructed with the chaos theory, in which the idea of trajectory is abandoned and replaced by the evolution of a probability distribution. In this work, the idea of a path that a packet should follow from its origin to its destination is replaced by the probability distribution vector with a time evolution described by (1). Acampora and Shah [18] considered a similar statistical procedure to describe the behavior of a store-and-forward routing as a comparison with hot-potato routing. Due to the fact that the probability to go directly from one node to the other is zero for almost all nodes, except for those two directly connected, most of the elements of operator *U* are zero. Each column has only two non-zero elements. The preferential output port has probability *Ppp* and the alternative port, corresponding to the deflection port, has probability *Pd* given by:

$$Pd = 1 - Ppp \tag{2}$$

A packet is sent to the preferential port in three cases:

a) There is no other packet in the competitor link that could arrive before it.

b) There is another packet that could arrive before it, but that has a local final address and is going to be removed before competition.

c) There is another packet arriving before it that is not a local packet, but it has a different preferential output port.

Link occupation probability *Poc* defines the probability of the first case to be 1− *Poc*. Given that case, a) is not true, the local packet probability *Plp* defines the second case probability term as *Poc*∗*Plp*. Finally, given that case a) and case b) do not apply, considering *Pop* as the probability of the competitor packet to have a different preferential port (another port), the third term is defined as *Poc*∗(1-*Plp*)∗*Pop*. Hence, the final probability of a packet to go through preferential port *Ppp* is given by:

$$Ppp = 1 - Poc + Poc{*}Plp + Poc{*}(1-Plp){*}Pop \tag{3}$$

In (3), *Poc* is the occupation probability associated to the link load. A fully loaded link (not considering the FDL length) is assumed to corresponds to *Poc*=1. The probability of a packet preference pointing to another port *Pop* is assumed to be 50% and *Pop*=0.5 in all cases. Local packet probability *Plp* is evaluated to be $1/\overline{H}$, with $\overline{H}$ calculated as a preliminary mean number of hops obtained with a first guess value *Plp*=1/(*N*–1).

The $Plp = 1/\overline{H}$ hypothesis is originated by the fact that every packet, at any time, is positioned in some place in its own path that starts on the origin and finish on destination. The mean number of hops in all possible paths is $\overline{H}$ and the packet is considered to be a local packet only in the last of those $\overline{H}$ hops. That means that $1/\overline{H}$ is the probability of a packet to be positioned at the last hop of its path from origin to destination.

Without failure, the MSN architecture belongs to a symmetry group called automorphism [13]. In this group, it is impossible to differentiate any node from the other concerning its position in the network. The mean number of hops is the same regardless of the position of the final address node. Yet, introducing a failure, the symmetry is broken and the mean number of hops may assume different values for different final destinations. In this case, it is necessary to calculate the mean number of hops for all the possible destinations and to adopt the arithmetic mean of those values as the final network mean number of hops.

One more consideration should be made about the "don't care" nodes. They are already identified and signalized by number 3 in the preferential port matrix "*pp*". In that case, the model considers that the packet plays no role in the decision of the preferable output port. The position of the switch may be adjusted to the preferred output port of the packet eventually arriving in the competitor link. This case corresponds to considering *Ppp=Pd=50%* in all "don't care" situations.

The number of hops is obtained recursively by (1) starting with $P_1(x)$, which represents the probability to reach the destination with one hop, to calculate $P_2(x)$, which represents the probability to reach the destination with two hops. That procedure is repeated *k* times while the total probability is less than 100%, with an arbitrary criteria chosen to be $\Delta P=10^{-6}$. Further reduction of that criterion interferes only with the calculation time and no change is observed in the results for $\Delta P=10^{-5}$.

The mean number of hops for each destination *x* is calculated by the equation:

$$\overline{H} = \sum_{1}^{k} tP_t(x) \tag{5}$$

With the condition:

$$1-\Delta P < \sum_{1}^{k} P_t(x) \leq 1 \tag{6}$$

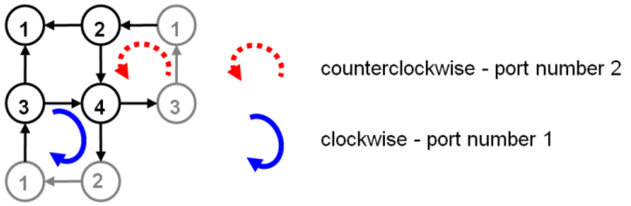


Figure 11. MSN with four nodes (N=4) showing link sub-domains.

D. *Trivial case calculation example*

As an example, consider the network with four nodes shown in Fig.11. Suppose that the link load is very near zero (without any charge). Then, *Poc=0*, *Ppp=1* and *Pd=0* are the values used. In order to find out the mean number of hops it suffices to calculate the mean number of hops to get to node number 1. Calculations for all the other destinations will yield the same value because of the MSN automorphism.

The initial hypothesis is that a generically chosen packet is not in node number 1 but is addressed to node number 1. This condition is represented by initial probability vector $P_0(x)$ given in (7).

$$P_0(x) = \begin{bmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \tag{7}$$

Connection matrix "*c*" is given by (8):

$$c = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 \end{bmatrix} \tag{8}$$

Preferential port matrix "*pp*" is given by (9).

$$pp = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 3 & 1 \\ 1 & 3 & 0 & 2 \\ 3 & 2 & 1 & 0 \end{bmatrix} \tag{9}$$

After one hop, probability vector $P_1(x)$ is obtained by (1) as shown in (10).

$$P_1(x) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \begin{bmatrix} 2/3 \\ 1/6 \\ 1/6 \\ 0 \end{bmatrix} \tag{10}$$

Operator "*U*" in (10) has a null first column because the packet in node 1 is going to be removed (has arrived to destination) and cannot go anywhere. After two hops, the probability vector $P_2(x)$ yields:

$$P_2(x) = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2/3 \\ 1/6 \\ 1/6 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

Condition (6) is achieved because $P_1(1)+P_2(1)=1$ and the mean number of hops given by (5) results in $1.P_1(1)+2.P_2(1)=2/3+2/3=4/3$. This value can be confirmed by analytical formulae deduced in [13] for MSN and shown in (12) valid for the cases in which *N* is an even number and the shortest path is always used. The trivial example presented here uses *N*=4 and *n*=2. In this case, *n*/2=1, so (*n*/2) odd number case formula applies. The (*n*/2) even number case formula (12) can be used for the empty link (*Link Load*=0%) and both *N* and *n*/2 must be even numbers.

$$\begin{cases} \overline{H} = \dfrac{(N/2)(n+2)-4}{N-1} \rightarrow (n/2)\ even \\ \overline{H} = \dfrac{(N/2)(n+2)-2n-2}{N-1} \rightarrow (n/2)\ odd \end{cases} \qquad (12)$$

E. Results and discussions

The results for the mean numbers of hops are shown in Fig. 12 for the MSN with *N* nodes (*N=64, N=144 and N=256*). All the results were calculated for both cases: with and without failure.

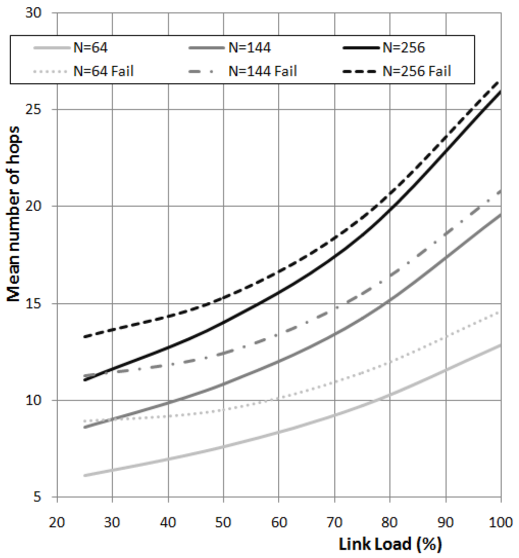


Figure 12. Mean number of hops versus Link Load condition.

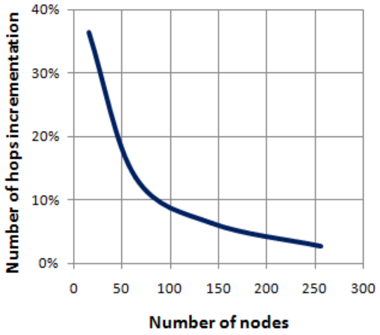


Figure 13. Relative Mean Number of hops degradation after failure as a function of the number of nodes.

Figure 12 shows that the degradation of the mean number of hops due to the failure is smaller for 100% link load condition when compared to the degradation for non-full load condition. Also, that degradation is smaller for higher number of nodes than for small networks. Fig. 13 shows a plot of the relative mean number of hops degradation at that full load condition (*Link Load* = 100%) as a function of the number of nodes. It confirms that larger number of nodes results in better robustness of the complex system. That conclusion was previously discussed for calculations up to 144 nodes [15].

VI. Simulation Model

The time domain simulation model (TDSM) was developed over the OMNeT++ platform [19]. The simulation model considers all the nodes sending packets to all the others and following the same rules used in the analytical model. Every packet arriving to one 2x2 node is addressed to the better output port, unless the node is already occupied with a competitor packet. In this case, the packet is sent to the available output port. The destination and the exact instant of packet generation are randomly chosen. Each link load condition is governed by the packet size. A packet with half the link size is used to simulate the 50% link load condition. Each packet that reaches the destination stimulates the insertion of a new one, addressed to a new randomly chosen destination. That procedure ensures the maintenance of the link load condition all along the simulation time. The simulation considers a 40Gbps bit rate and one kilometer link length. The delay line fiber length is considered to be equal to the link length, the same hypothesis employed in the analytical model.

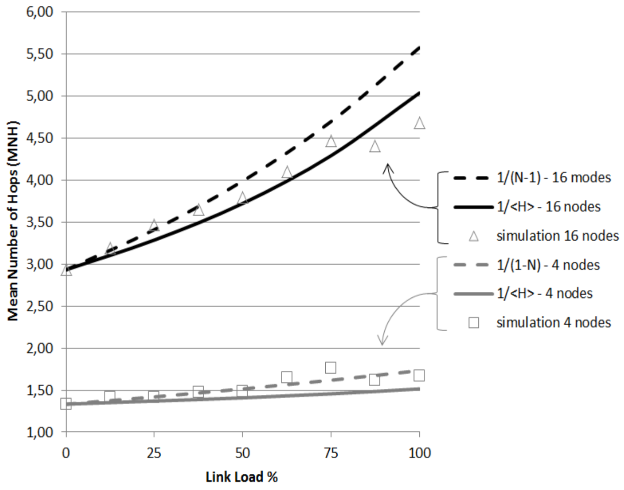


Figure 14. Analytical and simulation models.

Figure 14 shows the simulation results compared to the analytical model for two hypotheses used for the evaluation of the local packet probability *Plp*. The agreement between models is better for hypothesis $Plp = 1/\overline{H}$ as compared to the hypothesis of the first guess *Plp* = 1/(*N*-1). In fact, that first guess hypothesis is very close to simulation results for small number of nodes but tends to decrease faster than $Plp = 1/\overline{H}$, producing wrong results for a higher number of nodes. The simulation time is far higher than the analytical calculation time, limiting its utilization for scalability issues. The simulation model was important to validate the statistical analytical model and to determine that the first guess used for the *Plp* probability was not correct for large number of nodes.

VII. Failure Effect Distribution Map

The last calculation performed was the failure distribution effect. In case of failure, the symmetry is broken and the mean number of hops is no longer the same for any destination. Then, it is necessary to calculate the mean number of hops executed by an arbitrary packet addressed to all the 256 nodes. The overall mean value was considered to be the arithmetic mean of those previously calculated values. Considering the full load traffic condition (100% link load), the map in Fig. 15 shows an important distribution feature. The map shows nodes 1 to 16 in the first line and 16 nodes per line up to node number 256. The failure occurs in a link belonging to the clockwise sub-domain connecting nodes 89, 90, 106 and 105. Most of the destination nodes are not perturbed by the failure and remain with the same average number of hops (ANH) they had before failure (ANH<26). The ANH increases only for the destinations near the failure. Outside the contour lines, the ANH is less than 26. Crossing one contour line, the ANH is less than 27. Increased by one unit after crossing each contour line, the ANH will be less than 34, near failure, after crossing 8 contour lines.

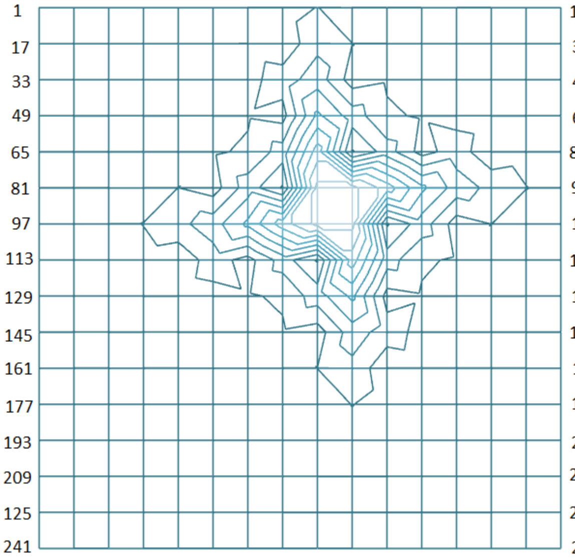


Figure 15. Failure segregation. Each contour line corresponds to one more hop from source to destination. Outside the contour lines, the Average Number of Hops (ANH) is less than 26. Inside all the lines, the ANH is less than 34.

VIII. Conclusion and Future Work

The approach used in this work allowed treating a large number of nodes network as a complex system, working as a bottom-up organization system. The approach was fully analyzed with both a statistical analytical model and a simulation model. It was possible to investigate the scalability, the protection and the restoration as physical layer emerging functions. Protection and restoration are achieved by local signalizations that modify only the node operations around the failure. No signalization needs to be transmitted over a long distance regardless of the size of the network and that behavior is responsible for the scalability. A map with the number of hops after failure illustrates that the network performance degradation occurs only around the failure. The segregation of the failure effects represents a new feature that could be observed due to the new approach. Generalization was performed for the National Science Foundation Network (NFSnet) working as a complex system in a bottom-up type of organization. The approach used herein can be considered as an Autonomic Network Architecture (ANA) [20] extension to the physical layer. Future work will provide more details about the complex behavior through the use of the same statistical analysis for larger networks (more than 256 nodes). With more nodes, new Emerging Functions can be revealed because they can be emphasized for larger number of nodes. Several new features can be proposed or investigated. Burst switching (packets larger than the link length) traffic distribution behavior, delay and delay variation are the same candidates to be analyzed as Emerging Functions. Considering the case of long emergency state time and the EFC, one possible future work is to apply a machine learning technique to automatically rebuild the routing tables. This can be done from the initial failure point recursively until reaching all nodes; the signalization is the trigger of this procedure. In this case, the adaptive tree is changed without being commanded to restart. The bottom-up organization and the complex system treatment at the physical layer allow this good performance and robustness for network.

Acknowledgment

The authors thank FAPESP – The State of São Paulo Research Foundation – sponsor of the KyaTera Project. The authors also thank Lucas Pauli Simões for the contributions to the simulation results. The second author also wants to thank FAPESP for the support through grant 2011/17096-5.

References

[1] A. Sachs, R. Rocha, F. Redígulo, and T. Carvalho, "Emerging function concept applied to photonic packet switching network", EMERGING 2012: The Fourth International Conference on Emerging Network Intelligence, IARIA, Barcelona, Spain, September 23-28, 2012, pp.22-26. ISBN: 978-1-61208-239-4.

[2] P. Baran, "On distributed communications netwoks", IEEE Transactions on Commnications Systems, CS-l2, 1964, pp.19.

[3] A. C. Sachs, "Self-organized network architecture deployed by the utilization of optical packet switching technology" (in Portuguese). 2011. Doctoral Theses (Digital Systems) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2011. Available at: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-05082011-152444/>. Retrieved: July, 2012.

[4] C. Guillemot, M. Renaud, P. Gambini, C. Janz, I. Andonovic, R. Bauknecht, B. Bostica, M. Burzio, F. Callegati, M. Casoni, D. Chiaroni, F. Clerot, S. L. Danielsen, F. Dorgeuille, A. Dupas, A. Franzen, P. B. Hansen, D. K. Hunter, A. Kloch, R. Krahenbuhl, B. Lavigne, A. Le Corre, C. Raffaelli, M. Schilling, J. C. Simon, and L. Zucchelli, "Transparent optical packet switching: the european ACTS KEOPS Project

approach", J of Lightwave Technology, vol. 16, pp. 2117-2134, 1998.

[5] L. Dittmann, C. Develder, D. Chiaroni, F. Neri, F. Callegati, W. Koerber, A. Stavdas, M. Renaud,, A. Rafel, J. Solé-Pareta, W. Cerroni, N. Leligou, Lars Dembeck, B. Mortensen, M. Pickavet, N. Le Sauze, M. Mahony, B. Berde, and G. Eilenberger, "The european IST Project DAVID: a viable approach towards optical packet switching", JSAC Special Issue on High-Performance Optical/Electronic Switches/routers for High-Speed Internet II. IEEE Journal on Selected Areas in Communications, vol. 21, pp. 1026 – 1040, 2003.

[6] C. Stamatiadis, M. Bougioukos, A. Maziotis, P. Bakopoulos, L. Stampoulidis and H. Avramopoulos, "All-optical contention resolution using a single optical flipflop and two stage all-optical wavelength conversion", paper OThN5 Proceedings of OSA / OFC/NFOEC 2010. Available at: <http://www.photonics.ntua.gr/PCRL_web_site/OFC_10_OThN5.pdf>. Retrieved: July, 2012.

[7] J. M. Carlson and J. Doyle, "Complexity and robustness", Proceedings of the National Academy of Sciences - PNAS, February 19, vol. 99, suppl. 1, 2002, pp. 2538–2545.

[8] A. L. Barabási, "The architecture of complexity", IEEE Control Systems Magazine, vol. 27, 2007, pp. 33-42.

[9] D. L. Turcotte and J. B. Rundle, "Self-organized complexity in the physical, biological, and social sciences", in Proceedings of the National Academy of Sciences – PNAS, February 19, vol. 99, suppl. 1, 2002, pp. 2463–2465.

[10] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle, "A survey of autonomic network architectures and evaluation criteria", IEEE Comm. Surveys & Tutorials, vol. 14, n. 2, 2012, pp.464-490.

[11] A. L. Barabási and R. Albert, "Emergence of scaling in random networks", Science, vol. 286, Oct. 1999, pp. 509–512.

[12] I. Prigogine, Le leggi del caos, Roma-Bari, Editori Laterza, 1993.

[13] A.G.Greenberg and J.Goodman, "Sharp approximate models of adaptative routing in mesh networks", Telegraffic Analysis Computer Performance Evaluation. Elsevier Science -North Holland, 1986, pp. 255-269.

[14] NSFNET: A Partnership for High-Speed Networking Final Report 1987-1995. Available at: http://www.merit.edu/networkresearch/projecthistory/nsfnet/pdf/nsfnet_report.pdf. Retrieved: set. 2008.

[15] A. Sachs, C. M. B. Lopes, and T. C. M. B. Carvalho, "Protection schema for optical packet switching network with large number of nodes", Microwave and Optoelectronics Conference (IMOC) 2009 SBMO/IEEE MTTS-International, 3-6 Nov 2009, pp.47-50.

[16] Scilab, free and open source software for numerical computation. Available att: http://www.scilab.org. Retrieved: Feb, 2013.

[17] H. Pistori, J. J. Neto, and M. C. Pereira, "Adaptive non-deterministic decision trees: general formulation and case study". INFOCOMP Journal of Computer Science, Lavras, MG, 2006. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.1885&rep=rep1&type=pdf>. Retrieved: July, 2012.

[18] A. S. Acampora and S. I. A. Shah, "Multihop lightwave network: a comparison of store-and forward and hot potato routing", IEEE Transactions on Communications, vol. 40, 1992, pp. 1082-1090.

[19] OMNeT++, discrete event simulation environment free for academic and non-profit use. Available at http://www.omnetpp.org. Retrieved: July, 2012.

[20] M. Sifalakis, A. Louca, A. Mauthe, L. Peluso, and T. Zseby, “A functional composition framework for autonomic network architectures”, Autonomic Network Architecture (ANA) Project, which is sponsored by the EU-IST initiative on Future and Emerging Technologies, 2006.