View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by VBN

Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Topological Routing in Large-Scale Networks

Pedersen, Jens Myrup; Knudsen, Thomas Phillip; Madsen, Ole Brun

Publication date:
2003

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Pedersen, J. M., Knudsen, T. P., & Madsen, O. B. (2003). Topological Routing in Large-Scale Networks. Department of Control Engineering: Aalborg Universitetsforlag.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
? You may not further distribute the material or use it for any profit-making activity or commercial gain
? You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: April 25, 2017

Topological Routing in Large-Scale Networks

Jens Myrup Pedersen, Thomas Phillip Knudsen and Ole Brun Madsen
Aalborg University, Institute of Electronic Systems
jens@control.auc.dk, thomas@control.auc.dk, obm@control.auc.dk

***Abstract* — A new routing scheme, Topological Routing, for large-scale networks is proposed. It allows for efficient routing without large routing tables as known from traditional routing schemes. It presupposes a certain level of order in the networks, known from Structural QoS. The main issues in applying Topological Routing to large-scale networks are discussed. Hierarchical extensions are presented along with schemes for shortest path routing, fault handling and path restoration. Further research in the area is discussed and perspectives on the prerequisites for practical deployment of Topological Routing in large-scale networks are given.**

***Keywords* — Communication system planning, communication system routing, network, routing, topology, WAN.**

1. Introduction

The growth of large-scale networks, particularly the global networks comprising the Internet, has put pressure on traditional routing schemes for such networks; this includes size of routing tables and ability to support the increasing demands for QoS. From the field of multiprocessor systems table-free routing schemes have been known for years. These schemes are not directly applicable to large-scale networks; they rely on the structure having highly regular properties and operating on a limited scale, conditions which are not practicable in large-scale networks. Recent work, however, in the field of large-scale networks has proposed the design of networks with global structural properties for the improved support of QoS, termed Structural QoS or SQoS [1],[2]. Such network design offers the opportunity for applying concepts from multiprocessor systems to large-scale networks, taking advantage of the global properties to introduce table-free routing. This class of routing schemes, taking advantage of defined structural properties, is labelled Topological Routing. In this paper, characteristics of Topological Routing will be described in relation to large-scale networks.

2. Methods

The concept of Topological Routing is introduced and related to already known schemes from multiprocessor systems. Two structures known from multiprocessor systems are dealt with in details, and it is analysed how the structural demands of large-scale networks differ from these. This leads to a discussion on how the structures and schemes can be revised in order to satisfy the demands of large-scale networks. A number of problems are identified, and some solutions suggested.

The 4-regular grid structure and the honeycomb structure [3] form the basis of the further study. Both structures support Topological Routing. The structures can be studied as meshes or toruses. Throughout this paper, the meshes are considered. However, the algorithms are provided without considering the problems that occur due to nodes on the edge of a structure having smaller degree than the nodes of the structure in general. The algorithms are easily extended to the toruses.

3. Notation

Basic notation is given here. Throughout the paper more notation is introduced as it is used. A structure $S=N\bigcup L$ consists of a set of nodes N and a set of undirected lines L, such that each line is interconnecting two different nodes. Every node has a name/address associated to it. A path of length n between a node u and another node u', is a sequence of nodes $(u=u_0),u_1,u_2,\ldots,u_{n-1},(u_n=u')$ such that all pairs of nodes (u_i,u_{i+1}) for $0\le i\le n-1$ are connected by a line e_{i+1}. Thus it can also be written $u,e_1,u_1,e_2,\ldots,e_{n-1},u_{n-1},e_n,u'$. The hop distance between two nodes, u and v, corresponds to the length of the shortest path between these two nodes, and is written $d_h(u,v)$. $N(u)$ denotes the set of neighbours of a node u and consists of all nodes w such that $d_h(u,w)=1$.

In this paper it is assumed that all structures without failures are finite and connected: For every pair of nodes $u,v\in N$, there exists a path between u and v. In case of failures, a structure S' can contain a number of components. A component of S is a set of nodes $N'\subseteq N$ and a set of lines $L'\subseteq L$ so that every such line is incident only with nodes in N'. Furthermore, from a node $u\in N'$ a path exists to another node v if and only if $v\in N'$.

4. Topological Routing

The basic idea of Topological Routing is the routing schemes known from multiprocessor systems as described in e.g. [3], [4], [5]: In all cases an addressing scheme is provided, and from any node any packet can be routed from only knowledge of the address of the current node as well as the destination node. Therefore, the name Topological Routing has been chosen.

The principle is illustrated in the 4-regular grid structure shown in fig. 1. Clearly, this scheme will route packets from source to destination in a number of hops corresponding to the sum of the differences of the coordinates in the two directions. This routing principle is generalised in the following.

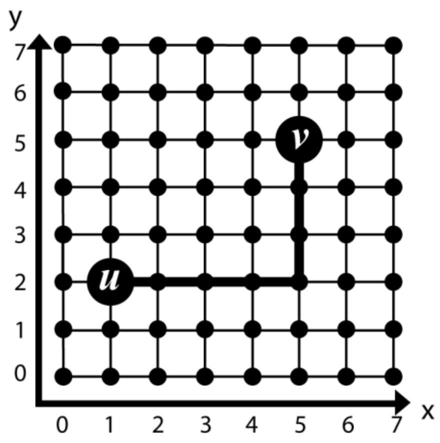

Figure 1. An example of a 4-regular grid structure. The nodes are addressed according to the coordinate system. Routing a packet from *u* to *v* is done hop-by-hop: In every node, the address of *v* is compared to the current address. Based on the differences in x and y coordinates a next hop is chosen, which reduces the difference in one of the directions. The scheme is not deterministic since in some nodes two next hops can be chosen between. In this example, the packet is always send in the x direction if such a choice is to be made.

To every node, an address is associated. This address can be formulated by words, coordinates or numbers, forming ordered sequences. Every node knows its own address as well as the addresses of its neighbours. Topological Routing works on a hop-by-hop basis: Given a packet p and its destination node v, in every node receiving p, say u, the address of v, is compared to the addresses of all neighbours of u, and p is sent to an address closest to v. The notation $d_a(u,v)$ is used for the address distance between two nodes. How this distance is calculated depends on the chosen addressing scheme. In this paper $d_a((x_1,y_1),(x_2,y_2))=|x_2-x_1|+|y_2-y_1|$ is used as the addressing distance measure for the 4-regular grid. For any node u in the 4-regular grid structure we write $u=(u_x,u_y)$, where u_x and u_y corresponds to x and y coordinates of the address of u.

Clearly, the performance of such a scheme depends on both network structure and addressing scheme. In the following, sufficient conditions for an addressing scheme to support Topological Routing are stated. Two levels of support are introduced: Schemes supporting Topological Routing, and schemes strongly supporting Topological Routing. In schemes supporting Topological Routing, any packet will finally reach its destination, while in schemes strongly supporting Topological Routing, an additional condition is that a chosen path always has to be a shortest path.

Let S be a structure consisting of a set of nodes N, and a set of lines L. It is assumed that $d_a(u,v)=d_a(v,u)\forall u,v\in N$, that $d_a(u,v)$ is always finite and that $d_a(u,v)=0\Leftrightarrow u=v$. If for every pair of nodes, $u,v\in N, u\neq v$, there exists a node $w\in N(u)$ such that $d_a(u,w)<d_a(u,v)$, then S supports Topological Routing: Clearly, when routing, the addressing distance is reduced by every hop, and thus a packet routed will always reach its destination, assuming S without errors.

The following further conditions ensure strongly support of Topological Routing: Assume $u\neq v$. Let $w\in N(u)$ such that $d_a(v,w)$ minimum. Then $z\in N(u), z\neq v$ must imply that $d_h(z,w)\geq d_h(v,w)$. As a consequence Topological Routing in such a structure ensures that a shortest path is chosen. In the 4-regular grid $d_h(u,v)=d_a(u,v)=|u_x-v_x|+|u_y-v_y|\forall u,v\in S$. This is sufficient to ensure strong support of Topological Routing. The same is true for the honeycomb structure. The addressing is similar to the scheme above, but extended to three coordinates [3]. This is illustrated in fig. 2.

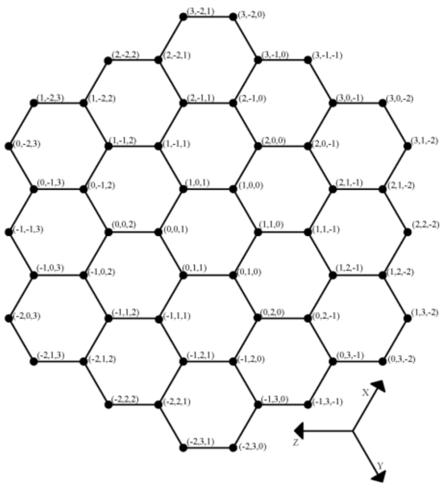

Figure 2. The addressing scheme of the honeycomb structure.

5. Large-scale network demands

The routing scheme introduced has until now been widely used in multiprocessor systems. These systems differ from large-scale networks in several ways. In order to identify what changes should be made to the scheme in order to make it applicable for large-scale networks, five major differences have been identified.

- Where multiprocessor systems can take advantage of cube and hypercube designs to increase connectivity, and can stack nodes close to each other, large-scale networks must follow an essentially two-dimensional surface, and natural formations as well as human structures form the major constraints on placement of the nodes. Node density varies greatly from deserted landscapes to heavily populated areas.
- Large-scale networks today contain thousands of transport nodes and millions of network termination points. Therefore, the structures must scale very well.
- Traffic in large-scale networks is increasingly Internet traffic; this traffic, opposed to many other traffic types, exhibits a non-localised pattern; hence, most of the traffic passing nodes in networks is transit traffic.

- Large-scale networks are dynamic entities, which are extended and upgraded continuously, and in which failures occur, and must be handled while the network is still in operation. In addition, large-scale networks have to be established over time. By nature, only add-on's can be performed to change the structure. Consequently, it needs to be extendable so that its coverage area as well as its node density in already covered areas can be expanded.
- Large-Scale networks support many services where continual operation is essential, and therefore there is a need for independent paths in order to support protection and fast restoration.

6. Revised Schemes

The problems are suggested solved by a number of extensions of the scheme. These results all refer to the 4-regular grid structure, but most of them are extendable to the honeycomb structure as well.

6.1. Hierarchical extension

Hierarchies are introduced in order to deal with scalability issues as well as the need for larger node density in some areas than in other, because the lower hierarchical levels may exist in selected areas only, as shown in fig. 3. This also makes the structure gradually extendable. However, the structure only supports topological routing if all lowest hierarchical layers exist globally, and even in this case it is not supported strongly. In the following, a revised algorithm is presented, which always gives a shortest path, even if parts of the lower hierarchical layers are left out.

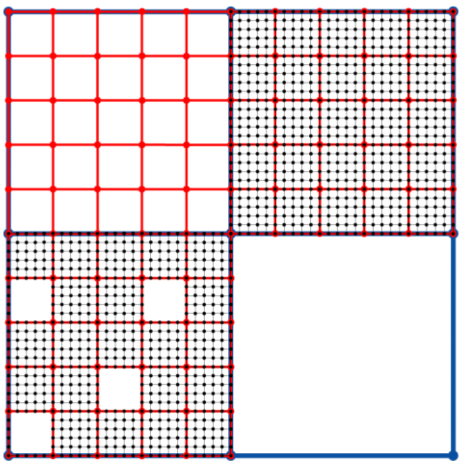

Figure 3. An example of how hierarchies can be used to create structures, which are more scalable than the basic structure. At the same time, it is gradually extendable and allows for varying node density.

Hierarchies are established by adding lines to the basic structure. For constructing just one hierarchical layer, an odd integer $g>1$ is chosen for granularity. Now every node $u=(u_x,u_y)$ such that u_x and u_y are both divisible by g, is connected by new lines to the four nodes (u_x-g,u_y), (u_x+g,u_y), (u_x,u_y-g) and (u_x,u_y+g). More hierarchical layers are constructed in a similar manner. Constructing n layers, these are denoted $H_0,\ldots,H_{n-1},H_n$, where H_n is said to be the highest layer and H_0 the lowest. $H_0=S$ corresponds to S without hierarchical extensions and is said to be the basic structure. In the rest of the section $k\leq n$. In general a node u belongs to H_k if and only if $u\in N$, $u_x\equiv 0 \bmod g^k$ and $u_y\equiv 0 \bmod g^k$. H_k also contains a set of lines, so that every node u is connected to the nodes (u_x-g^k,u_y), (u_x+g^k,u_y), (u_x,u_y-g^k) and (u_x,u_y+g^k), which are all nodes in H_k.The hop distance between two nodes, with the restriction that only lines of $\bigcup H_0,\ldots,H_k$ are used, is written $d_k(u,v)$.

An operator, $round$, is defined, which simply returns a closest integer value of a fraction: For two integers a and b the operator $round(a/b)$ determines an integer I such that $|I-(a/b)|$ minimum. In all cases where the operator is applied, b is odd. Thus, never more than one value of I exists.

Before stating the algorithm, three basic properties are listed. Let S be a hierarchical 4-regular grid structure, where g is the granularity chosen, and n is the number of hierarchies. Assume for now that all hierarchical layers exist globally. Property 1: From a given node $u\in\bigcup H_0,\ldots H_k$ the node $u'\in H_{k+1}$ such that $d_a(u,u')$ minimum is unambiguously determined as $u'=(g^{k+1}\cdot round(u_x/g^{k+1}),g^{k+1}\cdot round(u_y/g^{k+1}))$. Property 2: Let u be a node in H_k, and let u' be the node in H_{k+1} such that $d_a(u,u')$ minimum. Let $v\in N$ and v' be the node in H_{k+1} such that $d_a(v,v')$ minimum. If a shortest path between u and v is passing a node in H_{k+1}, a shortest path between u and v, passing u' and v' exists. Property 3: If $d_k(u,v)\leq d_{k+1}(u,v)$ then $d_k(u,v)\leq d_n(u,v)$.

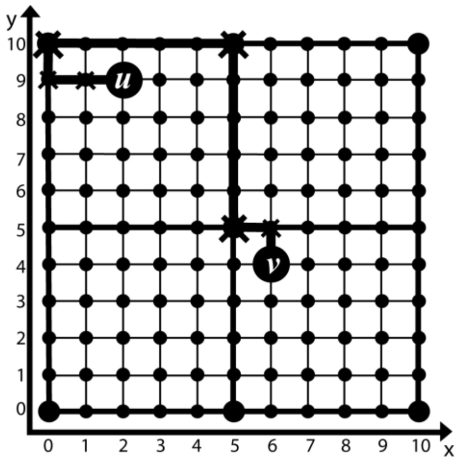

Figure 4. Hierarchical routing of a packet *p* from *u*=(*2,9*) to *v*=(*6,4*). In *u* it is determined that routing should be done through the upper hierarchical layer, and *p* forwarded towards (*0,10*). From (*0,10*) the packet is forwarded to the neighbour closest to *v*, which is (*5,10*). The neighbour of (*5,10*) closest to *v* is (*5,5*). From (*5,5*) the node closest to *v* is (*6,5*), and from this node *p* is forwarded to *v*.

The revised routing algorithm works as follows: When a packet p with destination node v is received in a node u, which is in H_k but not H_{k+1}, it is decided if p should be routed through H_{k+1} (or even higher layers). This is done by deciding if $d_{k+1}(u,v)<d_k(u,v)$, illustrated in fig. 4. Due to property 3 higher hierarchies need not be considered. v might not be in H_k, but due to property 2 it is sufficient to compare the distances between u and v'', v'' being the node in H_k such that $d_a(v,v'')$ minimum.

Since u and v'' are both in H_k, no shortest path uses lines of H_{k-1} or lower layers. Thus, $d_k(u,v'')=\frac{|u_x-v''_x|+|u_y-v''_y|}{g^k}$

If a shortest path exists in H_{k+1} using only lines in H_k and lower layers, this path does also exist in $\bigcup H_0,\ldots H_k$, implying that $d_k(u,v'')=d_{k+1}(u,v'')$. If this is not the case, a line in H_{k+1} is contained in a shortest path. Due to property 2 this implies that such a path exists, passing both u' and v', where u' is the node in H_{k+1} such that $d_a(u,u')$ minimum, and v' is the node in H_{k+1} such that $d_a(v,v')$ minimum: $d_{k+1}(u,v'')=d_k(u,u')+d_{k+1}(u',v')+d_k(v',v'')$.

Clear, also this distance is easily calculated given the addresses as well as values of g and n:

$$d_{k+1}(u,v'')=\frac{|u_x-u'_x|+|u_y-u'_y|+|v'_x-v''_x|+|v'_y-v''_y|}{g^k}+\frac{|u'_x-v'_x|+|u'_y-v'_y|}{g^{k+1}}$$

The value of k is easily derived from the address of u.

If $d_k(u,v'')<d_{k+1}(u,v'')$, routing is done following the normal routing scheme: $w\in N(u)$ so that $d_a(v,w)$ minimum is determined, and p forwarded to w. If $d_k(u,v'')>d_{k+1}(u,v'')$ routing along the shortest path happens by forwarding p to a node $w\in N(u)$, such that $d_a(u',w)$ minimum. If $d_k(u,v'')=d_{k+1}(u,v'')$ either scheme may be applied.

If lower layers are omitted in parts of the structure, the above may not hold, since only paths in the highest hierarchical layer can be assumed to exist globally. However, such left out parts of the lower hierarchies are complete squares as shown in figure 3. A problem may only arise if a path should be established between nodes in two different squares, passing one or more left out squares. However, in this case another path of same or shorter length can be established using higher layer nodes. Therefore, the only revision of the scheme needed in case of left out parts, is that routing should be done through H_{k+1} if $d_k(u,v'')=d_{k+1}(u,v'')$.

The algorithm presented ensures that routing upwards and downwards hierarchies are done, so that a shortest path is always chosen. The scheme is easily extended to support different granularity in different layers.

6.2. Path restoration and Topological Routing

Algorithms for routing in incomplete structures are necessary in order to deal with network failures. Such algorithms have been evolved for the 4-regular grid structure [5], [6]. However, these algorithms need to be revised to deal efficiently with failures of arbitrary shape in packet-switched networks. It is of great importance that the routing is efficient in the sense that short paths are chosen, and the use of tables minimised. At the same time, routing should be possible between any pair of nodes between which a path exists, even in case of failures. Therefore, a new algorithm is proposed that works by constructing small tables in nodes incident to lines, which are unavailable. These tables require update information from the given area of failure only. The scheme presented here deals with a basic 4-regular grid structure.

In a 4-regular grid structure $S=N\bigcup L$, with a standard (x,y) addressing scheme, suppose that a set of nodes $N''\subseteq N$ and a set of lines $L''\subseteq L$ is removed, and let $S''=N''\bigcup L''$. Throughout this paper S'' is assumed connected, but this is easily generalised.

If $S-S''$ is connected then $S'=S-S''$. Otherwise, choose a node $u\in S-S''$. The nodes of S' are then u and all nodes v such that a path between u and v exists in $S-S''$. The lines of S' are the entire set of lines incident with two nodes in S'. $S-S'$ is said to be a lake of S and denoted S_l.

If $S-S''$ is not connected, the definition clearly depends on the choice of u. This reflects the fact that if the network is split into more components, from a node in one such component, all other components seem to be failing.

A node is said to be a border node (BN) of S_l if it is incident with lines in both S' and S_l. The set of all nodes of S_l and all lines of S_l not incident with any node in S' is said to be the interior of S_l.

In this paper, only the handling of one such lake is considered. The algorithm is easily extended to deal with any number of lakes. This corresponds to allowing S'' not connected.

The presented algorithm, called the lake algorithm, works in the following way: When a lake S_l is detected, all BNs of S_l maintain a table of the border of S_l. Whenever a packet p with destination v could be forwarded by the normal scheme only along lines in S_l, a lookup in the table is made. p is then routed around S_l, towards a node w on the border so that $d_a(w,v)$ minimum. From here, the standard Topological Routing scheme is again applied. The principle is illustrated in fig. 5.

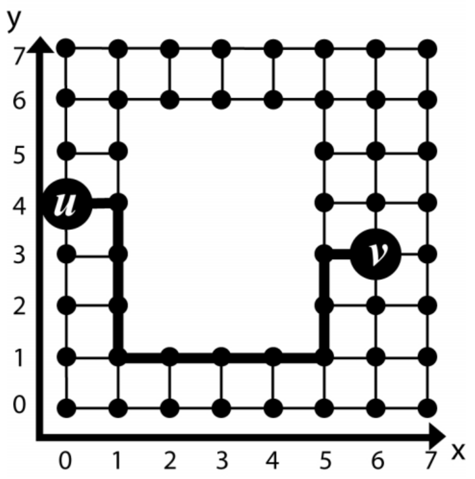

Figure 5. Routing a packet *p* from *u* to *v* in a structure with a lake. In (*1,3*) no path exists to a node closer to *v*, and thus the lake algorithm is applied. (*5,3*) is the node on the border of the lake, closest to *v*, and *p* is forwarded to this node, following a shortest path around the lake.

In the following, consider a structure S containing a lake S_l. For every node, u, the neighbours of it are ordered as a sequence: $(u_x+1,u_y),(u_x,u_y-1),(u_x-1,u_y),(u_x,u_y+1,),(ux+1,u_y),\ldots$

An element is said to be to the right of the preceding element. See fig. 6.

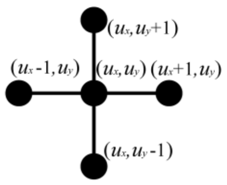

Figure 6. From u the neighbour nodes are ordered (u_x+1,u_y), (u_x,u_y-1), (u_x-1,u_y), (u_x,u_y+1). (u_x,u_y+1) is said to be the first node on the left-hand side of (u_x+1,u_y), and (u_x,u_y-1) the first node on the right-hand side of (u_x+1,u_y).

When a node u detects that it cannot establish contact to a neighbour node, it becomes aware that a lake has appeared, and u has become a BN. The next step is to detect this lake, and collect the information necessary for running the lake algorithm. This is done by sending a left-control-packet q_u: From u it is send to the first available node on the left hand side of the detected unavailable node. When a node v receives q_u it is first checked if $u=v$. If this is not the case, v is added to a list carried by q_u, storing the addresses of all nodes passed, in that particular order. It is then send to the first available node on the left hand side of the node from which it is was received. If $u=v$ the information of nodes passed is stored in a table T_u and q_u terminated. The nodes stored in T_u are the nodes defining the border of S_l: The BNs are only a subset of these.

Both right-control-packets and left-control-packets can be used. They provide the same information, the only difference being the ordering of the nodes. By sending such control packets in specified intervals, T_u is kept up to date. In the following, S_{T_u} denotes the set of nodes listed in T_u as well as the set of lines connecting all pairs of nodes listed preceding each other in T_u. Clearly S_{T_u} is connected. The shortest path between two nodes u,v both in S_{T_u} using only lines of S_{T_u} is written $d_{S_{T_u}}(u,v)$. When it is experienced that the failure has been recovered from, the table is deleted and routing is done following the normal Topological Routing scheme.

The same scheme is followed in all other BNs. In an implementation of the scheme some optimisation may be done in order to reduce the number of control packets, by letting control packets carry information for use in all BNs, and not only their origin. Now, assume that every BN maintains a table as described. When a packet p with destination v is received in u, the following happens:

- It is determined if any node $w \in S' \cap N(u)$, satisfies that $d_a(w,v)<d_a(u,v)$. If this is the case p is forwarded to such a node.
- If no $w \in S' \cap N(u)$ satisfies $d_a(w,v)<d_a(u,v)$, p would in S be sent on a line which does not exist in S'. Now $w \in T_u$ is found such that $d_a(w,v)$ minimum. If $d_a(w,v)=d_a(u,v)$, then v is in the interior of S_l, so no appropriate path can be established. In this case, p is discarded. Otherwise p is forwarded to a node $z \in T_u$ such that $d_a(z,v)=d_a(w,v)$. Among these possible nodes z is chosen as to minimise $d_{S_{T_u}}(u,z)$. A shortest path in S_{T_u} from u to z is determined, and p is routed along this, keeping explicitly specified route information, so that any node passed simply check this and routes accordingly. Assuming that the lake does not change during the routing process, p is forwarded to a node z such that $d_a(z,v)<d_a(u,v)$. It is easy to see that a path between z and v of length $d_a(z,v)$ exists in S', and thus p will reach its destination. Some optimisation can be done in this scheme, in order to shorten the path between u and z. This can be done by using lines not in S_{T_u}. Nodes not in S_{T_u} can also be used, but care must be taken, especially if more lakes exist.

If a packet with an explicitly specified route is received in a node, and the next node listed on the route is unavailable, the explicitly defined route is discarded, and a new route set up using the scheme above.

It might happen that a node u detects a failure, but has to route a packet p with destination v before a table has been set up. One suggestion for handling this case is to forward p as a right- or left packet, being routed as the corresponding control packets. At some point, e.g. when p reaches a node z such that $d_a(z,v)<d_a(u,v)$, p is in z routed according to the general scheme. In u it is possible to copy p, and send it as both right- and left packets ito provide faster transmission.

The scheme provided could be revised by enlarging the set of nodes, which are maintaining tables: Larger tables, kept in a larger set of nodes, can reduce the length of paths chosen. This trade-off between table sizes and path lengths comprises an interesting field for further research.

Handling of failures in hierarchical extensions of the structure is not supported by this algorithm. If a packet needs to be routed through higher hierarchies, it is send towards the nearest higher hierarchy node. If this node is unreachable, it has to be determined in which direction the packet should be routed. Further research is needed clarifying how this should be done.

6.3. Mappings

Large-scale networks must, as mentioned, conform to physical landscapes and can not be built entirely regularly; therefore, a mapping from the structure represented by the addressing scheme onto the physical network is necessary. Such mappings, while not arbitrary, can give considerably freedom in placement of nodes and lines in the landscape, as long as two conditions are fulfilled:

First, the two dimensions x, y in the addressing must be preserved as general physical directions in the network; distances measured as a given number of hops, though, need not be of similar length in kilometres. Therefore what is represented as a grid of squares in the addressing scheme may be mapped such that the four nodes comprising the corners of a square are placed as the corners of an arbitrary trapezoid; the lines may follow any path through the landscape, as long as they do not overlap.

Second, no two distinct lines or nodes in the addressing scheme may be placed in such proximity in the physical network, that they become a singe point of failure.

It is possible to add nodes of degree one or two to the physical structure, even though they do not exist in the addressing scheme; in this case, such nodes are simply associated with the closest node(s) in the structure.

6.4. Independent paths for protection

In general, the schemes proposed offer fast line restoration. However, in order to support applications not tolerating any restoration latency, it is necessary to send data through two or more independent paths, an approach recognized as protection. In the following, a scheme allowing for this is introduced. It is first introduced for the 4-regular grid and then extended to the hierarchical extension.

In general, the 4-regular grid offers four independent paths between any pair of nodes. Paths are considered independent only if they share no nodes.

Assuming the structure is complete, the routing scheme can be made deterministic and predictable, depending on how lines are chosen when different shortest paths are available. If several different such predictable routing schemes are chosen carefully, a corresponding number of independent paths between any pair of nodes can be established. For a rather simple example, assume that a packet p has origin u and destination v. Two copies of p, namely p and p' are sent from u.

In u as well as all other nodes passed until v is reached, the routing follows the general scheme. Whenever a choice of next hop has to made between two nodes z and z' so that $dist_a(z,v)=dist_a(z',v)$, p and p' are routed differently: In this case, p is routed along the x direction while p' is routed along the y direction.

If $u_x \neq v_x$ and $u_y \neq v_y$ p and p' are send on two independent paths, and if $u_x = v_x$ or $u_y = v_y$ the same path is followed. This scheme always uses as many short paths as are available in the structure. If more paths exist, they cannot be shortest, which makes it necessary to revise the general scheme of always forwarding packets to a node closest to the destination. Two cases must be dealt with:

- If $u_x \neq v_x$ and $u_y \neq v_y$ two paths can be established as stated above. Two more paths are established by routing via three intermediate nodes, v'_q, v''_q, v'''_q and $v'_{q'}, v''_{q'}, v'''_{q'}$ respectively. The routing between these is done using the normal Topological Routing scheme. When v'_q has been reached, routing is done towards v''_q, from where routing is done towards v'''_q. From here q is routed towards v. A similar scheme is used for q'. Let $r_x=(v_x-u_x)/|v_x-u_x|$ and $r_y=(v_y-u_y)/|v_y-u_y|$. Then q is routed via $v'_q=(u_x-r_x, u_y)$, $v''_q=(u_x-r_x, v_y+r_y)$ and $v'''_q=(v_x, v_y+r_y)$, while q' is routed via $v'_{q'}=(u_x, u_y-r_y)$, $v''_{q'}=(v_x+r_x, u_y-r_y)$ and $v'''_{q'}=(v_x+r_x, v_y)$. The length of the paths of q and q' are four hops longer than the two shortest paths. No four independent paths from u to v in this structure can have lower maximum, average, or minimum lengths than the set of paths of p, p', q and q'.
- If $u_x = v_x$ or $u_y = v_y$ (assume the latter) only one shortest path p can be established. Three more packets, q, q' and q'' are routed in a way similar to the former case. The paths of q and q' are both 2 hops longer than the path of p, and constructed by routing through intermediate nodes v'_q, v''_q and $v'_{q'}$, $v''_{q'}$ respectively, where $v'_q=(u_x,u_y+1)$, $v''_q=(v_x,u_y+1)$, $v'_{q'}=(u_x,u_y-1)$ and $v''_{q'}=(v_x,u_y-1)$. The path of q'' is eight hops longer than the shortest, with four intermediate nodes $v'_{q''}=(u_x-r_x,u_y)$, $v''_{q''}=(u_x-r_x,u_y+2)$, $v'''_{q''}=(v_x+r_x,u_y+2)$, $v''''_{q''}=(v_x+r_x,v_y)$. $r_x=(v_x-u_x)/|v_x-u_x|$.

This approach will clearly yield four independent paths, such that the paths are shortest possible. However, this approach is not directly applicable in case of hierarchical structures. If a packet has to be routed in only one hierarchical layer, the approach will work, but consider that this is not the case. If a packet has to be routed through higher hierarchies, it is encapsulated and routed towards the nearest higher hierarchy node. Therefore, all copies of it are routed towards this node. The approach provided can be used for the routing towards this node, making the paths independent, except for the nodes where a change in hierarchy occurs. The same is true for packets travelling the opposite way in the hierarchies. This results in four line-independent paths, of which at least one is shortest possible. Only nodes, which involve hierarchical shifts, are dependent. This might be dealt with by making such nodes more reliable, for example by doubling equipment and power supply.

7. Conclusion

The fundamentals of a new promising routing scheme for large-scale networks, Topological Routing, have been presented. It has several advantages over the schemes used today: Tables are only needed in case of network failures, and even in that case small tables are sufficient, and only little communication is needed to keep the tables updated. Fast restoration is supportable, and protection paths are easily set up. The scheme relies on the network structures satisfying certain global constraints. However, by establishing hierarchies and using appropriate mapping schemes, a high degree of freedom is provided and gradual extension of networks supported. Further research is needed before Topological Routing can be applied, especially in the fields of hierarchies, restoration, protection, QoS and load balancing, Here the need for a unified standard for construction and classification of network capabilities is crucial.

REFERENCES

[1] Ole Brun Madsen, Jens Dalsgaard Nielsen and Henrik Schiøler, “Convergence”, RTLIA 2002, Wien

[2] Ole Brun Madsen, Thomas Phillip Knudsen, Jens Myrup Pedersen, “SQoS as the base for the Next Generation Global Infrastructure”, IT&T Concerence 2003, October 22-23 2003, Letterkenny, Ireland.

[3] Ivan Stojmenovic, “Honeycomb Networks: Topological Properties and Communication Algorithms”, IEEE Transaction on Parallel and Distributed Systems, Vol 8, NO. 10, October 1997.

[4] Ming-Syan Chen, Kang G. Shin, Dilip D. Kandlur, “Adressing, Routing and Broadcasting in Hexagonal Mesh Multiprocessors”, IEEE Transactions on Computers, Vol. 39, No.1, January 1990.

[5] Jie Wu, “A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model”, IEEE Transactions on Computers, Vol.52, No.9, September 2003.

[6] Ming-Jer Tsai, Sheng-De Wang, “Adaptive and Deadlock-Free Routing for Irregular Faulty Patterns in Mesh Multicomputers”, IEEE Transactions in Parallel and Distributed Systems, Vol. 11, No.1, January 2000.