An adaptive strategy for deflection routing in meshes

Thierry Chich

To cite this version:

Thierry Chich. An adaptive strategy for deflection routing in meshes. [Research Report] LIP RR-1997-25, Laboratoire de l'informatique du parallélisme. 1997, 2+18p. hal-02101903

Laboratoire de l'Informatique du Parallélisme

Ecole Normale Supérieure de Lyon
Unité de recherche associée au CNRS n°1398

An adaptive strategy for deflection routing in meshes

Thierry Chich

August 28, 1997

Research Report N^O 97-25

Ecole Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : (+33) (0)4.72.72.80.00 Télécopieur : (+33) (0)4.72.72.80.80
Adresse électronique : lip@lip.ens−lyon.fr

An adaptive strategy for deflection routing in meshes

Thierry Chich

August 28, 1997

Abstract

In this paper, we describe a new adaptive routing algorithm for meshed-topology deflection networks. Our algorithm is based on a local learning method which evolves in order to produce a local spatial representation of the traffic. We first prove that our algorithm is a generalization of the Z^2 routing. Secondly, we prove that we can set the parameters of the learning algorithm such that our adaptive policy cannot create livelock situation. Then we show experimentally the efficiency of our algorithm. First, we compare the routing policies in a grid network, under an uniform load. Second, we create local congestion in order to show that the adaptive routing scheme avoid the overloaded region. Moreover, we propose a more realistic traffic model, and show that our algorithm is valid, even in such context. At last, we show that the algorithm is also efficient in a torus network. These results show the relevance of this method.

Keywords: Deflection routing, adaptive strategy, all-optical networks

Résumé

Nous proposons un algorithme adaptatif pour les réseaux maillés à déflexion. Cet algorithme est fondé sur une méthode d'apprentissage locale qui permet d'obtenir une représentation spatiale du trafic et de gérer ainsi la répartition de la charge. Nous montrons que notre algorithme peut être vu comme une généralisation du routage Z^2 qui est optimal dans les réseaux maillés réguliers. Ensuite, nous montrons que notre algorithme ne peut pas créer d'inter-bloquage dynamique. Puis nous montrons expérimentalement l'efficacité de notre algorithme. D'abord, nous comparons notre politique de routage avec une politique de routage Z^2 dans une grille, sous une charge uniforme. Puis nous créons artificiellement des congestions locales pour montrer que notre algorithme permet d'éviter les zones surchargées. Enfin, nous proposons un modèle de trafic plus réaliste et nous montrons que, et dans la grille et dans le tore, l'algorithme permet l'amélioration des performances par un meilleur partage des ressources.

Mots-clés: Routage par déflexion, routage adaptatif, réseaux tout–optiques

An adaptive strategy for deflection routing in meshes

Thierry Chich

August 28, 1997

1 Introduction

Metropolitan Area Networks (MAN) have been recently introduced (see for instance [13]). The goal of a MAN is to connect several Local Area Networks, and to integrate several services. FDDI [15] and DQDB [13] have been first considered for this purpose. However, we can observe a growing interest of a wide public for the use of communication services. This requires networks with very large bandwidth, that FDDI and DQDB cannot provide. Hence, all-optical networks are gaining increasing attention in this domain. Glass fiber offers immunity from bandwidth-limiting electromagnetic effects, and are capable of higher connection density, and lower loss. The benefit of this large bandwidth forces to use photonic switches to avoid the so–called “electronic bottleneck”.

A fundamental method used for all–optical networks is to provide end-to-end lightpath, as in passive star scheme [1]. However, this solution is not scalable, and cannot be extended to the Metropolitan Area Networks. Indeed, the size of the central switch should increase proportionally to the square of the number of nodes. Multi–hop schemes are therefore necessarily considered. In all–optical multi-hop networks, the current method consists in limiting the conversion in electronic format to the header. The payload remains in optical format from the source to the destination. The photonic switch must be able to extract the header, to resolve the routing problem, and to reinsert the regenerated header as the payload is delayed by crossing some fiber loops (see figure 1, from [6]).

The contention problem in the photonic switches must be processed as simply as possible, and the mechanism must avoid the use of buffer. Therefore, the deflection routing schemes are commonly used and are often implemented in experimental all–optical networks [5, 6, 9]. Deflection routing consists in sending contending packets on free outputs, instead of queuing them. A unique

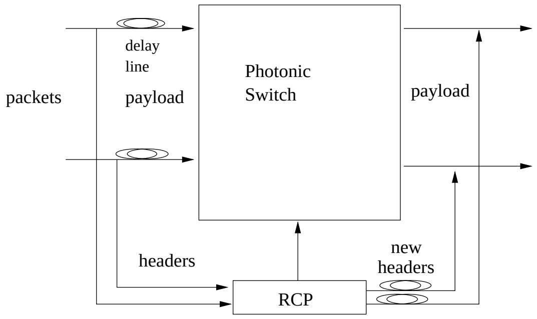


Figure 1: Deflection routing in a 2 × 2 crossbar: the routing control processor in denoted by RCP.

condition on the topology of the network is required for deflection routing. Every router must have equal in–degree and out–degree. Of course, the number of additional switching due to a deflection must be as small as possible. Thus, the number k of shortest paths beyond one node and the other ones is an important topological parameter related to the efficiency of deflection routing, since the probability of deflection decreases in this node as $e^{-O(k)}$. It is known that, for topology as Shuffle Net or meshes, the deflection routing scheme offers some advantages, such as a large flexibility in bandwidth assignment, and the lack of internal congestion [2, 10]. Deflection routing scheme is not only considered for all-optical networks. Indeed, the drawbacks of the store–and–forward scheme are amplified when the network operate at very high transmission rate. In [4], some arguments in favor of eliminating buffers in Gb/s multi-hop packet switching networks are stressed.

The most important part of the studies on deflection networks assumes a global synchronization of the network. With this synchronous approach, the routing decision can be done using more informations than in the asynchronous case. However, it is difficult to synchronize packet–arrivals (see [6]). Moreover, asynchronous networks would be easier and cheaper to build. In [8], the performances of asynchronous deflection routing are studied . Experiments show that the degradation of the performances of asynchronous routing networks compared to the performances of synchronous networks is important in a static context (i.e. every node emits at a fixed rate, randomly towards all the destinations). However, the difficulty of synchronizing multiple streams of optical signals in a node could lead to build asynchronous networks. It is therefore interesting to find methods which can deal with this kind of networks.

In this paper, we present a new routing algorithm that improves the behavior of asynchronous networks. This algorithm is designed to deal with dynamical traffic condition (i.e. every node emits for a fixed time, only towards one destination) in mesh architecture. We will show that our method gives better performances, even for static traffic. First, we explain the principle of the adaptive algorithm, and give some interesting properties of the routing algorithm. Then, we experimentally prove the efficiency of this routing algorithm, for grid and torus topologies, with regular and non-regular traffic.

2 A new routing strategy

A MAN must support many kinds of traffic to provide service integration. It has been shown that the traffic in an Ethernet network is self–similar [12]. On the other hand, the bursty nature of the traffic in Wide Area Networks was also proved [14]. Then, the traffic supported by a MAN would certainly be bursty too. An important advantage of deflection routing scheme is that the network does not suffer of internal congestion. The burstiness just causes more loss in the input queues. It is much more tricky to deal with dynamical traffic flows which create important overload locally. Usual routing algorithms in meshes can even produce local overload. For example, in a grid topology, shortest paths routing implies that the nodes in the middle of the grid are more often crossed by routes than nodes on the border. Such local overloads produce severe degradation of the quality of service. In particular, it implies that the available bandwidth of the servers depends on their position in the grid. Therefore, the solution consisting in using regular graphs only as interconnection topologies is not sufficient. It allows to avoid local overload due to the topology, but it does not allow to deal efficiently with the overload due to the dynamical traffic. Hence, it is important to derive routing schemes that balance the load in the network.

2.1 The standard algorithms

The usual deflection routing scheme consists in sending packets which cannot be sent on shortest paths in a another direction, rather than buffering them. Many variants have been proposed. Each variant is based on the use of different kinds of priority (e.g. oldest packet first or nearest destination principle).

Our algorithm do not use some characteristics of the packets. It is based on the idea that shortest paths are not equivalent. Hence, a value in $[0,1]$ is associated to every link for each destination. This value changes according to the traffic. Given a destination, links are sorted according to this value. The larger the value associated to a link, the better the quality of service of the link is supposed to be. The routing algorithm sends packets on the free link that has the larger value corresponding to their destination.

Formally, we will make use of the following notations :

- $\mathcal{N}_x$ is the set of addresses of the neighbors of node x. When there is no ambiguity, we omit the index.
- $p_{x\to z}[y] \in [0,1]$ is the value attributed to the output link joining node z from node x if node y is the destination. We assume that $\sum_{z\in\mathcal{N}_x} p_{x\to z}[y] = 1$ for all y.
- $n_z[y]$ is the number of shortest paths from node z to node y. This number is easily computable by a variant of the Dijkstra's algorithm.

In the simplest shortest-path routing scheme, the values $p_{x\to z}[y]$ are set to:

$$p_{x\to z}[y] = \begin{cases} \dfrac{1}{\sum_{l\in\mathcal{N}_x} n_l[y]} & \text{if } n_z[y] \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

However, it is more efficient to keep packets on routes on which the probability to be deflected is minimum. Thus, an idea is to route packets to nodes which are crossed by the largest number of shortest paths. Values are then computed as follows:

$$p_{x\to z}[y] = \frac{n_z[y]}{\sum_{l\in\mathcal{N}_x} n_l[y]}$$

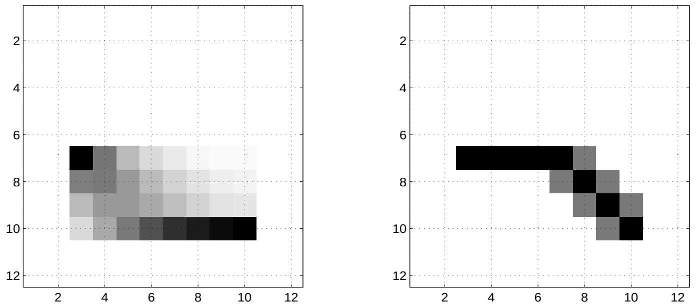
In mesh topologies, this modification gives better performances than the former algorithm. In [3], it is proved that this routing scheme, called "Z^2 routing", is optimal in regular meshed networks. Our experimental studies will always compare our new method with this improvement.

On figure 2, we express the probability that a packet, coming from node $(7,3)$, and going to node $(10,10)$ has to use each node of a 12×12 grid. The left hand side shows the probabilities for the simplest shortest path algorithm. The probabilities are depicted as grey levels (i.e. black is 1, and white 0). The right hand side shows the probabilities for the Z^2 routing algorithm. All the packets are remaining on row 7 until the index of the column become larger than 7. Then the packets are routed to $(10,10)$ on the diagonal.

2.2 A new efficient adaptive algorithm

2.2.1 Routing protocol

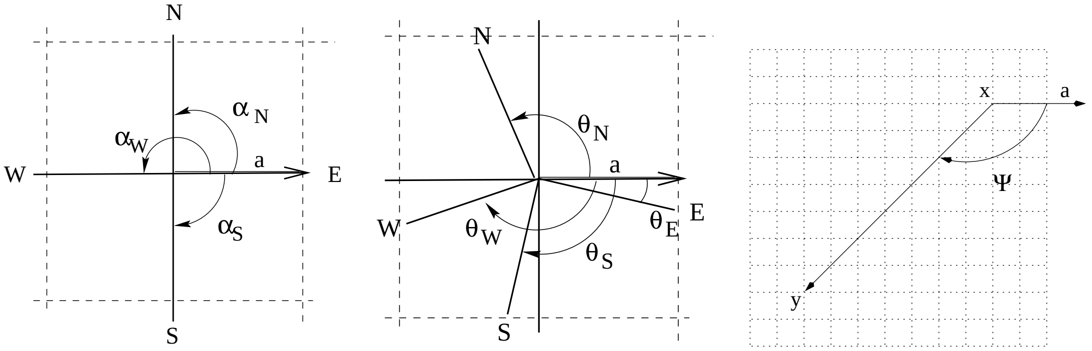
In order to avoid local overload, we will derive a new way of computing the values associated to each link and each destination. We use the following structure and the following notation: in each



(a) simple shortest path routing (b) Z^2 routing

Figure 2: Probabilities that a packet from $(7,3)$ to $(10,10)$ passes in the nodes of a 12*12 grid

router, we use a set of "logical links". Each physical link (figure 3(a)) is related to a logical link (figure 3(b)). These logical links will evolve in order to give a representation of the load. Let $\vec{a}$ be a fixed vector on each node x. For instance, let it be the West-East direction. We denote by $(\alpha_z)_{z\in\mathcal{N}_x}$ the angles between each physical link and the reference axis (see figure 3(a)). These angles will not change. Let $(\theta_z)_{z\in\mathcal{N}_x}$ be the angles between each logical link and the reference axis. All of these angles are normalized to be in $]-\pi,\pi]$. If a message arrives in node x, with destination y, let $\Psi = (\vec{a}, \overrightarrow{xy})$ be the angle between the reference axis and the vector router–destination (figure 3(c)).



(a) Physical links (b) Logical links (c) Destination

Figure 3: Notations

The values associated to each link than set to:

$$p_{x\to z}[y] = \frac{|\Psi - \theta_z|}{\sum_{l\in\mathcal{N}_x} |\Psi - \theta_l|}$$

Packets are routed on the physical link for which the corresponding logical link is the closest to y: the physical link chosen z_c is such that $z_c = \text{argmin}(|\theta_z - \Psi|)$.

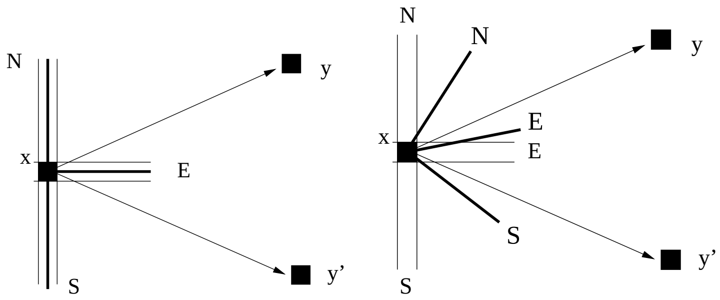
2.2.2 Learning algorithm

In order to adapt the algorithm to the modification of the traffic, angles θ_z are modified dynamically. Each time a packet is routed, the angles are modified before considering the next packet. More precisely, the aim of the learning algorithm is to bring each logical link closer to the last vector router–destination. When no message is crossing the router, the system is relaxed, that is the angles between physical links and logical links are decreased. Our algorithm use two parameters: the learning rate δ, and the forgetting rate, δ_o. The former is used in order to control the magnitude of the learning: the larger is this parameter, the closer θ_z will be to Ψ. The forgetting rate has a similar function: it controls the way θ_z is bring closer to α_z when no packet is currently in the router. The learning rule is using to adapt the routing decision to a spatial configuration of the traffic, and the forgetting rule, to decrease this influence if the specific spatial configuration disappears. One can formalize the learning algorithm as follows:

Algorithm 1 Learning(δ, δ_o)

```
    if there is a packet in the router
        for $z \in \mathcal{N}$
            $\Delta\theta_z = \frac{\delta(\Psi-\theta_z)}{1+3/\pi^2(\theta_z+\delta(\Psi-\theta_z)-\alpha_z)^2}$
        else
            for $z \in \mathcal{N}$
                $\Delta\theta_z = \delta_o(\alpha_z - \theta_z))$
end
```

Figure 4 shows a realistic evolution of the logical links in an arbitrary node if a stream of packets is sent to a destination y. Assume there is an other (but weaker) stream of packets at heading for node y'. In figure 4(a), we can see that both streams are routed on the East link. After learning, we can see that the stream directed to y is still using the East link. But the stream directed to y' use the South link. Thus, the contention risk is minimized.



(a) Before learning (b) After learning

Figure 4: Learning algorithm

2.2.3 Remarks on the adaptive routing

Property 1 *For $t \geq 0$, let $\theta_z(t)$ be the angle of the logical link corresponding to the neighbor z of x. Assume that for all $z \in \mathcal{N}_x \theta_z(0) = \alpha_z$. Then if δ=0, the adaptive routing is equivalent to the Z^2 routing algorithm.*

Proof. Assume that a packet is emitted from a node $(1,1)$ to $y = (y_R, y_C)$, with $y_R < y_C$. The packet will be routed on $(1,2)$ with respect to the Z^2 policy. Indeed, the packet is routed to the node which gives the largest number of shortest path ending in y. We denote $n_z[y]$ the number of shortest paths from $z = (z_R, z_C)$ to y. We have that $n_z[y]$ is related to the binomial coefficient:

$$n_z[y] = \left(\begin{array}{c} y_R - z_R + y_C - z_C \\ y_R - z_R \end{array} \right)$$

$n_z[y]$ becomes maximum as $y_R - z_R$ becomes closest to $\frac{y_R - z_R + y_C - z_C}{2}$. Thus, if $y_R < y_C$, $n_{(1,2)}[y]$ is larger than $n_{(2,1)}[y]$. For all (x_R, x_C) such that $y_R - x_R < y_C - x_C$, the horizontal link will be chosen (i.e $n_{(x_R, x_C+1)}[y] > n_{(x_R+1, x_C)}[y]$). Now, consider the case $y_R - x_R = y_C - x_C$. Then $n_{(x_R, x_C+1)}[y] = n_{(x_R+1, x_C)}[y]$. The packet is routed with a fair coin toss on one of the two links (see figure 2(b)).
In the other hand, if $\delta = 0$, $\theta_z(t) = \alpha_z$ for all t. Then, the routing decision depends on the angle $\Psi(t)$. On figure 5, we have shown an example for which $\vec{a}$ is set to the West-East direction. We can see that $\alpha_E = 0$ and $\alpha_S = -\pi/2$. If $|\Psi(t) - \alpha_S| < |\Psi(t) - \alpha_E|$, then the packet is routed on the East link. If $\Psi(t)$ is exactly between the two angles, the packet is routed on one of the two links with a fair coin toss, and on the South link otherwise. As $\Psi(t) = \arctan(\frac{y_R - x_R}{y_C - x_C})$, it is easy to see that the decision routing is exactly the same that it would be using the Z^2 routing policy.

□

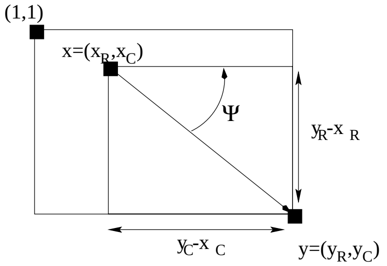


Figure 5: Notations for the proof of the property 1

3 Livelock situation

In deflection routing networks, packets may be delayed for an unbounded time. However, if there is no priority policy, and if packets are routed with shortest paths routing policy, the probability that a packet arrives to its destination grows to 1 as the time spent in the network grows. Our algorithm cannot avoid such probabilistic livelocks. In this section, we will prove that our algorithm does not produce other kinds of livelock.

A packet can be routed by our algorithm on a link which does not belong to a shortest path from the current node to the destination, even if there are free shortest path links. A set of logical links for which this event can occur, is called a "misrouting" set. A livelock involves at least one misrouting set. Indeed, if there is no misrouting set, packets are routed on shortest paths.
We can arbitrarily set the value of the angles in order to create one misrouting set of logical links (for example, if the logical links are identical to the physical links with a rotation of π, all packets

are routed on the opposite physical link than they should go). However, the angles are not fixed randomly. Let us study the routing cases leading to a misrouting set of logical links, and how to avoid these situations. First, we will show that our learning rule preserves the spatial relation between logical links, and physical links (i.e. we can set δ and δ_o such that the angular distance between a logical link and its physical link is bounded).

Definition 1 *A* packet stream $(M_t)_{t\geq 0}$ *is denoting a sequence of packets crossing a* 4*-input node, since time* 0 *to the present time, denoted* t*. Each packet has the average length (i.e. 642 ns). The packets are sent to possibly different destinations. A sequence* $(\Psi_t)_{t\geq 0}$ *can be associated to the packet stream.*

Property 2 *Let* x *be an arbitrary* 4*-input node. Let* $(M_t)_{t\geq 0}$ *be a stream of packets traversing node* x*. For* $t \geq 0$*, let* $\theta_z(t)$ *be the angle of the logical link corresponding to the neighbor* z *of* x*. Assume that for all* $z \in \mathcal{N}_x \theta_z(0) = \alpha_z$*. Then, for any* $\epsilon \in [0,1]$*, there exists* δ *and* δ_o *such that,*

$$\forall t \geq 0, \forall z \in \mathcal{N}_x, |\theta_z(t) - \alpha_z| \leq \epsilon\pi$$

Proof. We will create the "worst" packet stream. The destination (more precisely, the corresponding angle $\Psi(t)$) of each packet will be calculated such that $\Delta\theta(\Psi(t), \theta(t), \alpha, \delta)$ be maximum. We must precise the normalization condition in order to know the derivative domain of $\Delta\theta$. The normalization implies that $|\Psi - \theta| \leq \pi$ and $|\theta + \delta(\Psi - \theta) - \alpha| \leq \pi$. Under these conditions, $\Delta\theta$ is continuous and derivable. The roots of

$$\frac{\partial \Delta\theta}{\partial \Psi}(\Psi(t), \theta(t), \alpha, \delta) = 0$$

are

$$\Psi_1(\theta,\, \alpha,\, \delta) = \theta + \frac{\sqrt{\pi^2 + 3\,(\theta-\alpha)^2}}{\sqrt{3}\,\delta} \quad \text{and} \quad \Psi_2(\theta,\, \alpha,\, \delta) = \theta - \frac{\sqrt{\pi^2 + 3\,(\theta-\alpha)^2}}{\sqrt{3}\,\delta}$$

If δ is small, i.e. smaller than $\frac{1}{2\sqrt{3}}$, which is a realistic assumption, then $|\Psi_1 - \theta|$ and $|\Psi_2 - \theta|$ are both larger than π. The normalization conditions we set previously, imply that this value cannot be reached. Thus, the function $\Delta\theta$ is a growing function for the variable Ψ in its domain. Then, for each realistic value of δ, $\Delta\theta$ is maximum for $\Psi = \pi + \theta$, and by symmetry, is minimum for $\Psi = -\pi + \theta$. The worst packet stream is therefore composed of packets all pointed to the opposite direction of the logical link. Assume that the link we study is the link East. Assume also that $\vec{a}$ is oriented as West-East direction, such as $\theta_E(0) = 0$. Each packet of the stream $(M_t)_{t\geq 0}$ is such that $\Psi(t) = \theta_E(t-1) - \pi$.

As $(M_t)_{t\geq 0}$ is defined as the "worst" packet stream, this sequence of packets implies that the maximum value of $|\theta_E(t) - \alpha_E|$ is reached. This maximal value cannot be greatest than π. Indeed, if $\theta(t) + \Delta\theta(t) \geq \pi$, then the normalization is applied and $\theta(t+1)$ is set back to $\theta(t) + \Delta\theta(t) - 2\pi$. Then, if this maximal value is π, the logical link evolving from this stream packet will turn infinitely. If this value is less than π, then the $\theta_E(t)$ has an asymptotic value $\theta_E^{\max}$ such that $\theta_E(t) \to \theta_E^{\max}$ for $t \to \infty$. We will show that we can set δ and δ_o such that $\theta_E^{\max}$ is as little as we want. An analytical proof is not easy to do. However, we can compute the values of $\theta_E^{\max}$ for fixed values of δ and δ_o. Figure 6 shows the values of $\theta_E^{\max}$ as a function of $\delta_o \in [0, 0.005]$ and $\delta \in [0, 0.01]$. Areas which are not depicted, represent the values of δ_o and δ for which there is no stabilization i.e. the logical link turn infinitely.

□

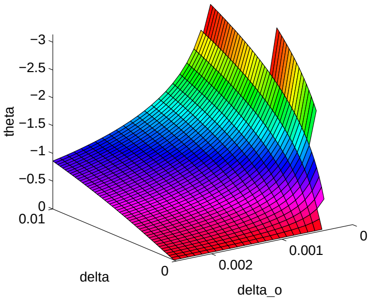


Figure 6: Asymptotic values of θ_E as a function of δ_o and δ

Corollary 1 *Let x an arbitrary 4-input node. Let $(M_t)_{t\geq 0}$ be a stream of packets crossing node x. For $t \geq 0$, let $\theta_z(t)$ be the angle of the logical link corresponding to the neighbor z of x. Assume for all $z \in \mathcal{N}, \theta_z(0) = \alpha_z$. Then there exists δ and δ_o such that $(\theta_z(t))_{z\in\mathcal{N}}$ is not a misrouting set.*

Proof. Without loss of generality, we consider the North and East links of an arbitrary node x. $\vec{a}$ is in the West-East direction ($\alpha_E = 0, \alpha_N = \pi/2$). Assume that y is a destination in the South-East region of x. Let Ψ the angle relative to node y. If the logical links are such that a packet headed for node y is sent on the North link, then $(\theta_z(t))_{z\in\mathcal{N}}$ is a misrouting set (situation illustrated on figure 7). The packet is misrouted if

$$|\Psi - \theta_N| < |\Psi - \theta_E| \tag{1}$$

Note that Ψ is a real in $]-\pi/2, 0]$, because y is in the South-East region. Assume that $\Psi - \theta_E$ is negative. Then equation 1 become

$$\begin{aligned} & \Psi - \theta_N > \Psi - \theta_E & (2) \\ \Rightarrow \quad & \theta_N < \theta_E & (3) \end{aligned}$$

Note that if, in the property 2, δ and δ_o are set such that $\epsilon = 1/4$, then

$$\forall z \in \mathcal{N}, |\theta_z(t) - \alpha_z| < \pi/4 \tag{4}$$

and then, $\theta_N > \pi/4$, and $|\theta_E| < \pi/4$. Then equation 3 is impossible. Assume now that $\Psi - \theta_E$ is positive. Equation 1 become

$$\begin{aligned} & -\Psi + \theta_N < \Psi - \theta_E & (5) \\ \Rightarrow \quad & \theta_N < 2\Psi - \theta_E & (6) \end{aligned}$$

$2\Psi - \theta_E$ is maximal for the larger value of Ψ, that is 0. Then equation 6 implies that $\theta_N < -\theta_E$, that is impossible if equation 4 is satisfied. Therefore, if equation 4 is satisfied, the packet cannot be misrouted. Since this argument can be applied for all the region, the corollary is proved.

□

4 Characteristics of the simulator

In order to study the efficiency of this adaptive algorithm, we have written a network simulator. In this section, we describe its characteristics. In the following sections, the results are presented.

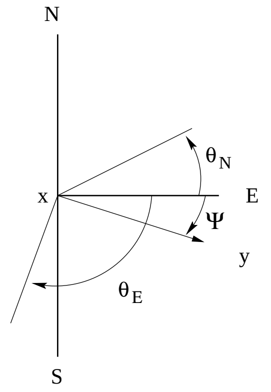


Figure 7: A misrouting set

Topological characteristics The results we present are generated for a 12x12 grid. The links are bidirectional. The distance between two nodes is 2 km. The time slot (i.e the maximum size of a packet) is 1 μs. As the light speed in a fiber is $2/3C$, a link can contain 10 full packets. Furthermore, all input links are preceded by a one–slot length fiber in order to be able to introduce without loss. Hence, each link has a capacity of 11 full packets.

Router characteristics Each router needs to store arriving packets from the users of the network (application, LAN, etc.). The size of this input buffer is fixed to 100 packets. The insertion strategy is an asynchronous version of the classic insertion in deflection routing networks [7]. A packet is inserted in the network when there is a free place for it. Thus, losses are due to the overflow of the queues.

Packet characteristics In an asynchronous routing scheme, the packets could have different sizes. This ability is a great advantage, because the size of the packet is adapted to the need of the application. We have considered that the length of the packets L follows a bimodal law polarized both at (1) the length of the acknowledgment packet, and (2) the length of the slot. We have fixed the minimum size of a packet at $200ns$ (the size of a header). We set $P(L = 200ns) = 0.3$, and $P(L = 1\mu s) = 0.4$. The other message lengths are chosen as multiple of $0.1\mu s$, uniformly in the range $[300, 900]$. The average of such a law is $642ns$ (which implies that each link has a capacity close to 15 packets).

We have also define three packet types :

- **background :** the destination of each packet is randomly chosen in the set of the other nodes
- **spy :** this kind of packets is used to have some informations, such as the number of deflection, the number of waiting messages or the waiting time in the input queue. The destination of these packets is fixed. The traffic generated by these spy packets must be low, in order to not disturb the global behavior. For instance, in our measurements, the spy traffic is generated by the node $(3, 3)$ to the node $(10, 10)$ (see figure 16). The emission frequency is set to 0.01.
- **hot–spot :** as in the former case, the traffic is directed to one destination. But the purpose is

to generate heavy traffic, analog to the traffic generated by a bandwidth-requiring application, as FTP. The emission frequency is typically set to 0.8.

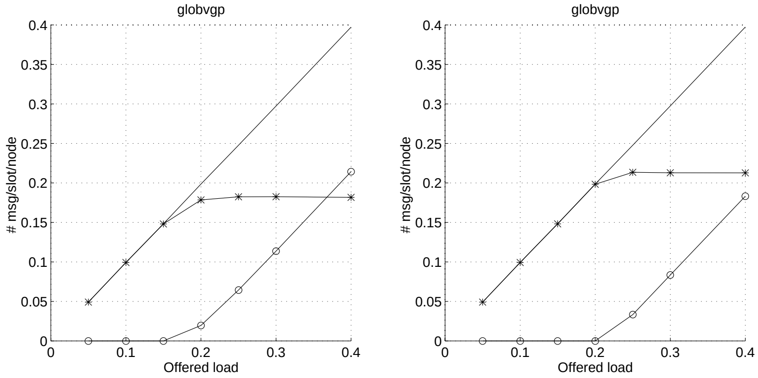
All router can emit the three kind of packets, independently, at the same time.

5 Regular traffic

5.1 Throughput

For all the following experiments, we have set the parameters of the adaptive algorithms such as $\delta = 0.002$ and $\delta_o = 0.0005$. Note also that each experiment has been performed for 10^6 slots (e.g. for an offered load of 0.2, each node has emitted 200,000 packets in each router).

Figure 8 presents the throughput and the loss in the same network (described in section 4), with the two different routing modes. We measure the throughput (number of arrived packets per slot and per node, i.e. the curve with the stars) and the loss (number of lost packets per slot and per node, i.e. the curve with the circles) as the offered load varies (number of emitted packets per slot and per node).



Z^2 routing

adaptive routing

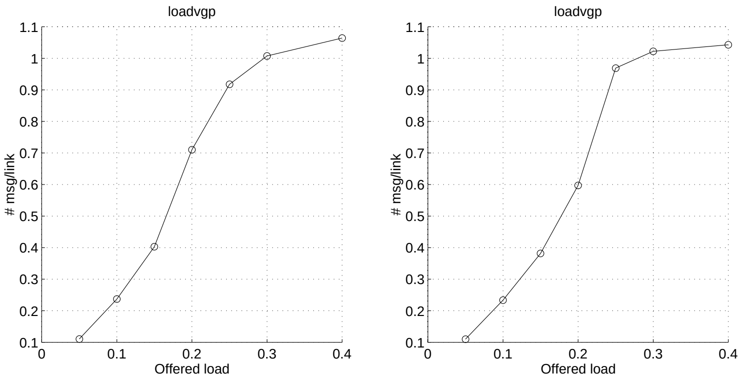
Figure 8: Throughput and loss under an uniform offered load

Even in this standard context, the adaptive routing scheme is better than the Z^2 routing scheme. This is due to the topology. A grid is not a regular meshed-topology. There is an overloading in the center of the grid. The adaptive routing scheme increases the use of the border of the network, allowing to avoid the deflection produced by the overload in the center (as we will see later).

The figure 9 shows the evolution of the internal load in both networks. The average number of packets per slot and per link is presented. The saturated state is apparent. For an offered load below 0.25 (saturation threshold), the internal load is lower in the adaptive routing network than in the Z^2 routing network. Less packets are deflected. For more intensive offered load, the internal load could be more important in the adaptive routing network than in the Z^2 routing network. This is due to the improvement of the throughput, which allows to insert more packets in the network.

5.2 Spy traffic

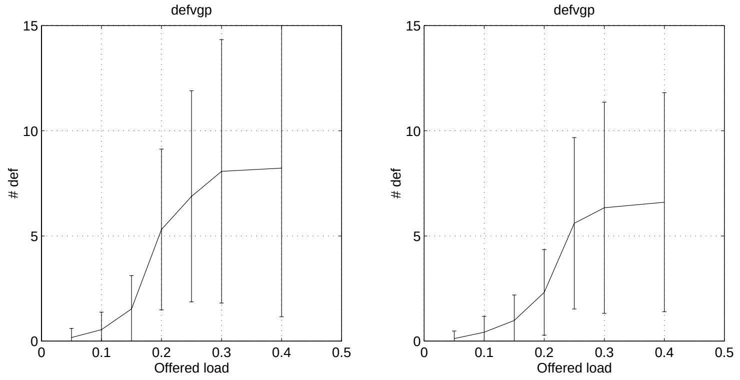
The spy traffic is a Poissonian flow emitted from the node $(3, 3)$ to the node $(10, 10)$, at frequency 0.01. Figure 10 shows the average number of deflection for a spy packet as the offered load grows.



Z^2 routing adaptive routing

Figure 9: Number of packets per slot and per link

Vertical bars represent the standard deviation, which is a very important parameter. Indeed, one of the most important drawback of the deflection routing scheme is the disorder in the packet arrivals. Larger is the standard deviation, larger must be the output buffer to sort the arrivals.
It is clear that the spy traffic is less deflected and less disordered in the adaptive routing network than in the Z^2 routing network.

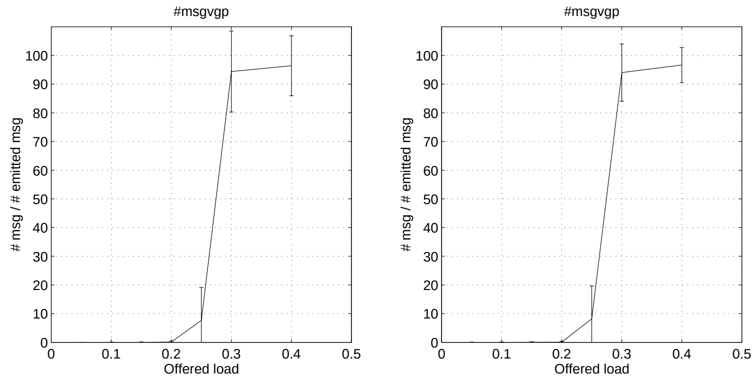


Z^2 routing adaptive routing

Figure 10: Average number of deflections during the time recquired to reach $(10, 10)$ from $(3, 3)$ as function of the offered load

Figures 11 and 12 characterize the behavior of the queue. Figure 11 shows the average number of packets waiting in the spy queue as a function of the offered load. As the saturation arises (offered load : 0.3), the number of packets in the queue becomes close to the maximum (the size of the queue). This behavior is typical of the queuing systems [11]. It occurs when the rate (classically noted ρ) becomes larger than 1.

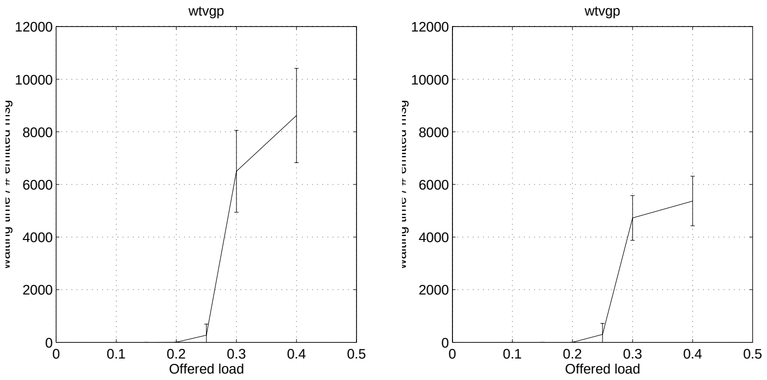
In both cases (adaptive or Z^2 routing), curves are very similar. It is interesting to observe the behaviors for the waiting time in the spy queue (figure 12). This figure shows the average waiting time for the packets in the queue (3,3). The unit time is 100 ns. For example, for an offered load



Z^2 routing adaptive routing

Figure 11: Average number of packets waiting in the $(3,3)$ queue at the insertion of a spy packet

of 0.3, the average waiting time in this queue for a Z^2 routing network is $650,000$ ns, i.e. 600μs. In the adaptive routing network, for the same offered load, the waiting time in the queue is close to 450μs. Whereas the number of packets waiting in the queue are nearly the same in both cases, the waiting time in the queue for each packet is quite different. This phenomenon prove a difference of internal behavior. Every packet is waiting for a longer time in the queue of the Z^2 routing network than in the adaptive one.



Z^2 routing adaptive routing

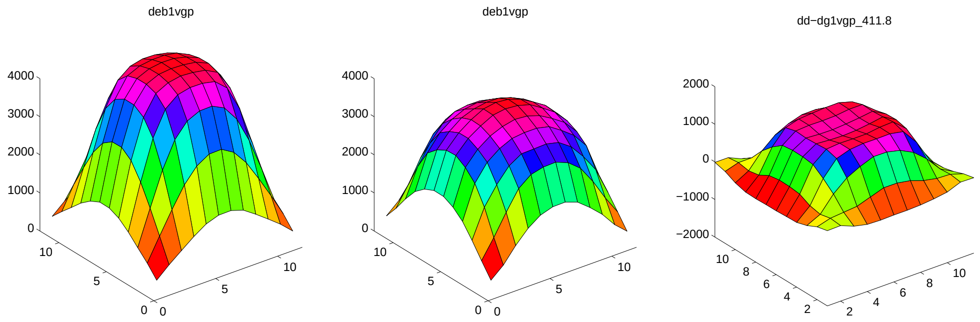
Figure 12: Average waiting time (*100ns) for the spy packets in the $(3,3)$ queue

5.3 Local load

In this section, our purpose is to illustrate the difference of the load repartition for one routing scheme to another. Figures 13 and 14 represent the values of the internal load in each node of the network. Since the network is a grid, the xy-axis represents the topology, and the z-axis is the average number of packets crossing the node for 1000 slots. As the number of links by node is limited to 4, and the average length of packets is 0.642 slot, the maximum local load is near 6500.

However, we never observe local load beyond 4500 packets (because of the inter-packet space).

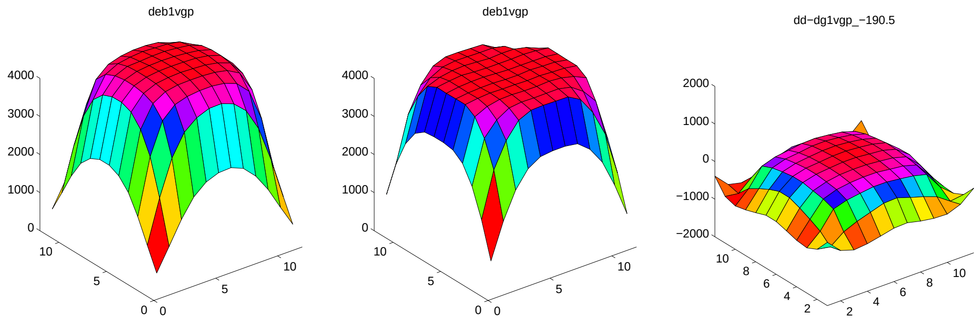
Figure 13 shows respectively the local load in the Z^2 routing context, the local load in the adaptive routing context, and the difference between the two previous measurements. For an offered load of 0.2, the internal load is lower for the adaptive routing. The adaptive routing uses more intensively the border links.



Z^2 routing adaptive routing difference

Figure 13: Load on each node under an offered load of 0.2

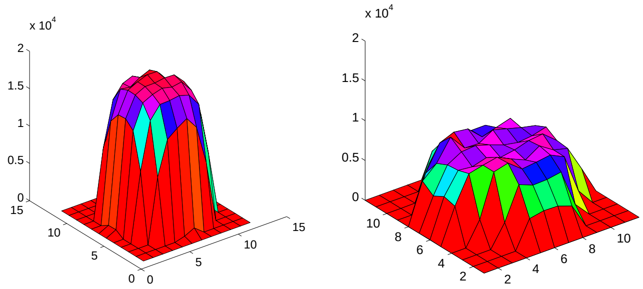
In figure 14, the offered load is 0.25. We have seen that the internal load is slightly larger in the case of adaptive routing than in Z^2 routing. However, performances decreasing with the load such as the deflection rate is still better in the adaptive routing network than in the Z^2 routing network. Indeed, even with this additional load, the adaptive routing spread efficiently the load in the network.



Z^2 routing adaptive routing difference

Figure 14: Load on each node under an offered load of 0.25

Figure 15 shows the spatial distribution of the losses arising for an offered load of 0.25. The surface of the loss is more spreads for the adaptive routing network than for the Z^2 routing network (although the global loss is larger in this latter case).



Z^2 routing

adaptive routing

Figure 15: Loss on each node under an offered load of 0.25

6 Irregular traffic

6.1 Central hot-spots

In this section we study what occurs when an uniform load of the grid is disturbed by an important local overload. We have chosen to generate an heavy overloading in the center of the grid as shown on figure 16. Node $(6,6)$ emits a 0.9 rated poissonian traffic to $(7,7)$. Node $(8,5)$ emits at the same rate to $(5,8)$. As the overload generated by these two traffics is in the center of the grid, the spy traffic emitted from $(3,3)$ to $(10,10)$ will be analyzed.

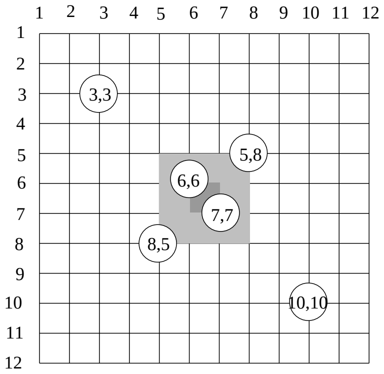
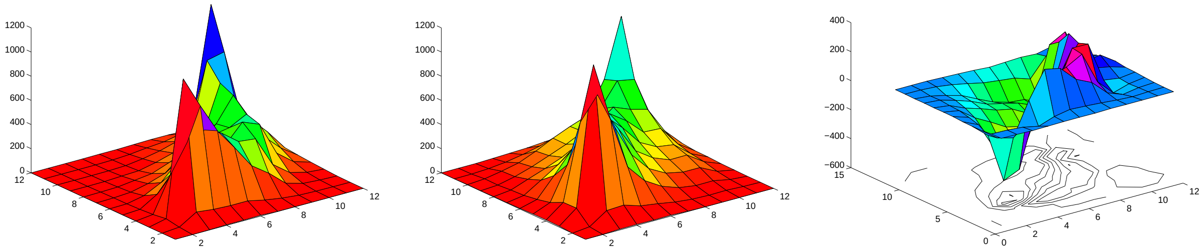


Figure 16: Hot–spot and spy traffics

Figure 17 shows the distribution of the spy packets in the network under a uniform load 0.2, with the two hot–spots. The comparison of the two pictures shows what are the effects of the adaptive routing. The packets avoid the region where the destinations of the two hot-spots are situated. The packet stream goes on the right side. As the hot-spots generated seem symmetrical, this behavior seems strange. However, insertion and reception are not symmetrical processes in deflection networks. If a node cannot emit, the queue could become overloaded. But there is overloaded region such that the packets headed for a node in this region cannot arrive, the packets try to join their destinations, and contribute to overload the region. Thus, hot-spots can absolutely not be considered symmetrical, and it is more efficient to avoid the destination region than the source.



Adaptive routing Z^2 routing difference

Figure 17: Number of spy packets crossing the nodes (under an offered load of 0.2)

6.2 A chaotic traffic

A regular traffic is not sufficient to conclude about the efficiency of the adaptive algorithm in realistic context. Indeed, realistic situations are characterized by the non-uniformity of the traffic, spatially and temporally. Is our algorithm able to deal with heavy variations ? However, it is difficult to exhibit the "typical realistic traffic". Hence, we have build a model of realistic traffic, and experiment our algorithm on this model.

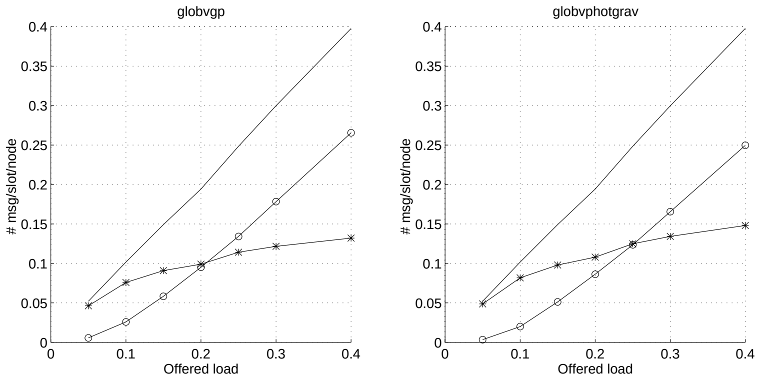
We create a non-uniform traffic such that we can control the average of the offered load, and such that the average length of the path source–destination is the average length of the path in the graph. Then, we create an arbitrary couple source–destination, with several characteristics, chosen such that we can control the offered load $\overline{F}$ in sake of uniformity. The parameters are:

- N_n is the number of nodes in the graph.
- $\overline{T}$ is the average time of emission to a direction
- $\overline{f}$ is the average frequency of the emission from the source to the destination.

The random variables T et f follow respectively the distributions $\mathcal{U}(2*\overline{T})$ and $\mathcal{U}(2*\overline{f})$ (where $\mathcal{U}$ denotes the uniform law). The average number of stations emitting at each time is given by $\overline{N_e} = \frac{\overline{F}N_n}{\overline{fT}}$. The random variable N_e is following $\mathcal{U}(2*\overline{N_e})$. It is clear that these choices are somewhat arbitrary. However, they determine a totally non-uniform traffic, composed of long communications between couple of points, what is more realistic than a traffic composed of sequences of independent packets.

Figure 18 shows the performance of both routing policies in this context. Comparing this figure to figure 8, we can see that the non-uniform traffic, for the same value of the offered load, is less inserted than the uniform traffic. The input queues are frequently overloaded. Even in this context, the adaptive routing is more efficient than the Z^2 routing. More packets can be inserted in the network. Hence, the throughput is higher.

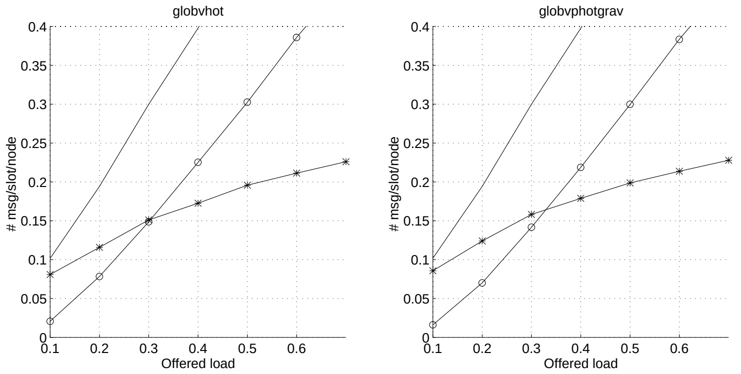
In order to prove that the difference between Z^2 and adaptive routing policies are not only due to the topology (we have shown that the adaptive policy is better on a grid), we have tested this algorithm on a torus 12×12. The gain is less visible than in the grid. Then, we have changing the scale in order to show the differences (figure 19) . In a torus, the Z^2 routing policy is optimal,



Z^2 routing adaptive routing

Figure 18: Throughput and loss under a non–uniform traffic

because the torus is a regular meshed-topology. However, under non-uniform traffic, the adaptive routing give again better performances (e.g., for the 0.3 rate, the adaptive routing network has inserted 22,780,002 packets, for 21,761,163 packets for the Z^2 routing network: the gain is slightly higher than 5%). This proves that the adaptive routing policy is efficient to deal with the spatial variations of the traffic.



Z^2 routing adaptive routing

Figure 19: Throughput and loss under a non–uniform traffic in a torus 12×12

7 Conclusion

In this work, we have expose a new adaptive routing strategy for deflection meshed networks. In [3], Badr and Poder explain that the shortest paths are not all equivalent. Thus, they introduce the Z^2 routing and prove its optimality in regular meshed-topology. For non-regular meshed-topology or non-uniform load of the network, however, the optimality is no longer true. We have generalized this idea in order to consider that the better path is neither the least loaded, nor the more offering choice path, but the path which offers a compromise between these features.

Our algorithm is based on a simple spatial representation of the traffic in the network. The routing decision is a function of the spatial distribution of the traffic. We have proved that this routing can be understood as a generalization of the Z^2 routing. We have also proved that the parameters of the learning algorithm can be set in order to control the influence of the traffic on the routing decision. Hence, we have proved that there exist values for the learning parameters such that this routing scheme cannot involve livelocks.

We have implemented the adaptive routing algorithm to show the efficiency of this scheme. Different experiments have been performed. Adaptive routing, dealing with uniform or non-uniform traffic, offers better performances than Z^2 routing.

The adaptive algorithm has been developed on meshed-topologies. However, the principle is independent of the topology. We intend to extend this idea to a larger class of graphs.

References

[1] Acampora, A. S., and Karol, M. J. An Overview of Lightwave Packet Networks. *IEEE Network* (Jan. 1989), 29–41.

[2] Acampora, A. S., and Shah, S. I. Multihop Lightwave Networks : A Comparison of Store–and–Forward and Hot–Popato Routing. *IEEE Trans. on Communication 40*, 6 (June 1992), 1082–1089.

[3] Badr, H., and Poder, S. An optimal shortest-path routing policy for network computers with regular mesh-conected topologies. *IEEE Trans. on Computers 38*, 10 (Oct. 1989), 1362–1371.

[4] Baransel, C., Dobosiewicz, W., and Gburzynski, P. Routing in Multihop Packet Switching : Gb/s Challenge. *IEEE Network* (May 1995), 38–61.

[5] Blumenthal, D. J., Feuerstein, R., and Sauer, J. R. First Experimentation of Multihop All–Optical Packet Switching. *IEEE Photonics technology letters 6*, 3 (Mar. 1994), 457–460.

[6] Blumenthal, D. J., Prucnal, P. R., and Sauer, J. R. Photonic Packet Switches : Architectures and Experimental Implementations. In *Proc. of the IEEE* (Nov. 1994), vol. 82(11), IEEE, pp. 1650–1667.

[7] Bononi, A., and Prucnal, P. R. Analytical evaluation of improved access techniques in deflection routing networks. *IEEE/ACM Transaction on Networking 4*, 5 (Oct. 1996).

[8] Chich, T., and Fraigniaud, P. An extended comparison of slotted and unslotted deflection routing. Research Report 97-07, LIP/ENSLyon, 46, Allée d'Italie, 69364 Lyon Cedex 07, Mar. 1997.

[9] Fehrer, J., Sauer, J., and Ramfelt, L. Design and Implementation of a Prototype Optical Deflection Network. In *SIGCOMM* (1994), vol. 24, ACM, pp. 191–200.

[10] Greenberg, A. G., and Goodman, J. Sharp Approximate Models of Adaptative Routing in Mesh Networks. In *Teletraffic Analysis and Computer Performance Evaluation*. Elsevier Science Publisher B.W., 1986, pp. 255–270.

[11] Kleinrock, L. *Queuing systems*, vol. 1. John Wiley & Sons, 1976.

[12] Leland, W., Taqqu, M., Willinger, W., and Wilson, D. On the Self–Similar Nature of Ethernet Traffic. *IEEE/ACM Transactions on Networking 2*, 1 (Feb. 1994), 1–15.

[13] Mollenauer, J. F. Standards for Metropolitan Area Networks. *IEEE Communications Magazine 26*, 4 (1988), 15–19.

[14] Paxson, V., and Floyd, S. Wide Area Traffic : The Failure of Poisson Modelling. *IEEE/ACM Transactions on Networking 3*, 3 (June 1995), 226–244.

[15] Rolin, P. *Réseaux haut débits.* Hermes, 1995.