UNIVERSITA' DEGLI STUDI DI MILANO
Dipartimento di Scienze dell'Informazione

RAPPORTO INTERNO N° 106-93

A Probabilistic Model to Evaluate the Performances of Deflection Networks

Elena Pagani   Gian Paolo Rossi

A Probabilistic Model to Evaluate the Performances of Deflection Networks

Elena Pagani[†]
Gian Paolo Rossi[‡]

Technical Report
106/93

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico, 39/41
I-20135 Milano (Italy)

ABSTRACT

In this report we present a probabilistic model to evaluate the performances of networks that use deflection routing on mesh topology. By considering each deflection network node as a system of queues and by assigning to each input traffic pattern the probability to enter one of the output links according to deflection routing policy, it is possible to estimate performance indexes for different topologies and traffic loads.

To this purpose, we run the analytical model by using the Mathematica software. The results have been compared with those previously obtained through simulations. The comparison is encouraging and indicates that the analytical model properly describes the deflection network node's behaviours.

---

*This work has been supported by CNR, the National Research Council, under grant PFT, 1992.

†pagae@ghost.sm.dsi.unimi.it

‡rossi@hermes.mc.dsi.unimi.it

1 Introduction

In this report we present a probabilistic model for the analytical evaluation of the performances of networks that use deflection routing on mesh topology [3]. By starting from the model presented in [2], we have generalized the node structure, thus obtaining a more practical and useful representation of the deflection network node and its behaviours. This provides a simple tool to estimate performance indexes, e.g. transmission delay, queue length and waiting time, for different network topologies and for varying traffic load and composition.

We are currently studing deflection networks with the aim of deriving the protocols that provide added value services on top of the basic datagram service [4, 7, 8]. The availability of an analytical model allows to predict the performances of these protocols and to analyze the way different traffic patterns affect one another. By finding the relationships existing among the parameters that influence the protocol and by solving the related optimization problem, we expect to be capable of performing the proper and fine tuning to achieve the best performances.

Aim of this report is to describe the analytical model, provide its correctness analysis and to present the results obtained from the initial execution of the model that has been performed by using Mathematica software package. The obtained results have been compared with those we obtained from simulations. The first compared analysis is encouraging and indicates that the analytical model properly describes the behaviours of the deflection network nodes and that the applied routing policy forces traffic flows to follow the shortest available path to destinations.

The report is organized as follows: in Section 2 we describe the deflection network, together with the assumptions we have made in the model construction. In Section 3, the probabilistic model is described, and in Section 4 we present the correctness analysis. In Section 5 we discuss the results obtained by implementing the model.

2 System Model

In this Section, we briefly describe the behaviours of basic deflection networks [3], that will be considered throughout this report, and the assumptions we have done about the nodes and the network in the construction of our model.

We consider the network composed of n nodes. Each node has 4 input links and 4 output links that connect it to the adjacent nodes. Two further links, one for the input and one for the output, provide the communication with the local upper user entity (Figure 2.1). Links have the same transmission speed. We will demonstrate that this model can be easily generalized to a node with L I/O links.

For every input link, I/O operations are performed as those usually encountered in slotted rings. A packet addressed to the node is extracted. A packet in the local user queue enters the network if an empty slot is available, i.e. a slot has

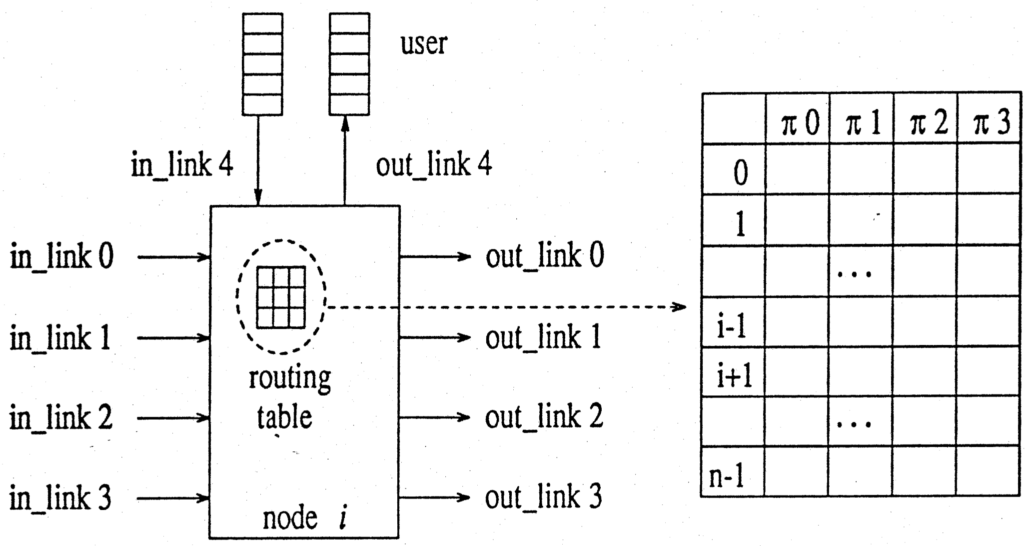

Figure2.1: Deflection network node model.

been received empty or a packet has been extracted. Switching and transmission processes proceed in time slots. We can generalize this model by assuming that the delay of a packet on the output link j of the node i is τ_{ij}.

At each slot a node can extract from or put into the network as many packets as its I/O links. Packets access the network independently of their destination, as soon as a free slot exists. At a locally synchronized instant, each node switches to the preferred link the packets it has in input buffers according to addressing and local routing information. Should two or more packets be routed to the same link, they collide. Collided packets are *deflected* over non preferred links and they are not locally buffered. Deflections may waste the network capacity and reduce the maximum throughput attainable under ideal conditions. However, it has been shown that this waste can be very limited and that its impact decreases as the network size increases [3]. The network behaviour is stable under any load, i.e. the throughput cannot collapse due to the saturation of some resources. In fact, packets cannot be accumulated into nodes neither be blocked awaiting for the availability of transmission buffers. Packets are kept travelling in the network until they reach their destination.

The routing function assumes that nodes know their shortest-path distances to all possible destinations and Backward Learning technique, [1], is used to dynamically estimate the required distances. To this purpose, each node maintains a *routing table*. The table has as many entries as many nodes in the network. Each entry contains the output links in preference order according to the increasing distance, expressed in number of hops, between the current node and the destination node. These information describe the network topology. In the sequel, we assume that no failures occur, hence routing tables never change.

When collisions occur, non-preferred paths have to be choosen with the aim to minimize the total amount of hops a packet has to follow to reach its destination. To this purpose, we associate to each packet a priority function in the use of an output link. The closer the packet is to its destination, the higher its priority. More formally, let $D(i,s,k)$ be the distance (in number of hops) between a given node i and a destination node s through the output link k of the node i. If we indicate with $p(i,s,k)$ the priority of the packets addressed to s on the output link k of the node i, then for two given destinations s and t we can say $p(i,s,k) > p(i,t,k)$ iff $D(i,s,k) < D(i,t,k)$. In the next section, we will describe the use of the priority function to determine the traffic distribution in the network. Informally, the higher the priority associated to a link, the lower is the probability of being deflected from that link. In the simulation model, the number of hops already done by a packet is also considered and packets with number of hops greater than a given treshold cannot be deflected, whereas packets with number of hops equal to a maximum are *filtered*, i.e. extracted from the network. In the probabilistic model we do not consider single packets, but rather flows of packets; so these information are not available.

3 Probabilistic Model

3.1 Node Model

In this section, we present the model of the deflection network node we have derived to evaluate the network performances.

We indicate with γ_i the generation rate of packets of the user i, and with μ_i the input rate from the user input queue (Figure 3.1). In the sequel, we will consider the input rate to an output link j of the node i, μ_{ij}. We indicate with λ_{ijk} the packet flow on the internal link of the node i that connects the input link j with the output link k, and, in particular, $\lambda_{ijk:t}$ is the portion of packets with destination the node t. Similarly, at a node i, $\gamma_{i:t}$ is the generation rate of packets with destination the node t. If Φ_{ij} is the input rate of the node i, link j, then:

$$\Phi_{ij} = \sum_{k=0}^{4} \lambda_{ijk} \tag{3.1}$$

where $k = 4$ is the output link towards the upper user entity. By assuming that at most one packet a time can arrive from one link, then $\Phi_{ij} \leq 1$. Moreover, it should be $\lambda_{i44} = 0$, since a node never sends packets to itself.

The packets generated by a node i can access the network by entering the output link k with rate:

$$\mu_{ik} = 1 - \sum_{l=0}^{3} \lambda_{ilk} \tag{3.2}$$

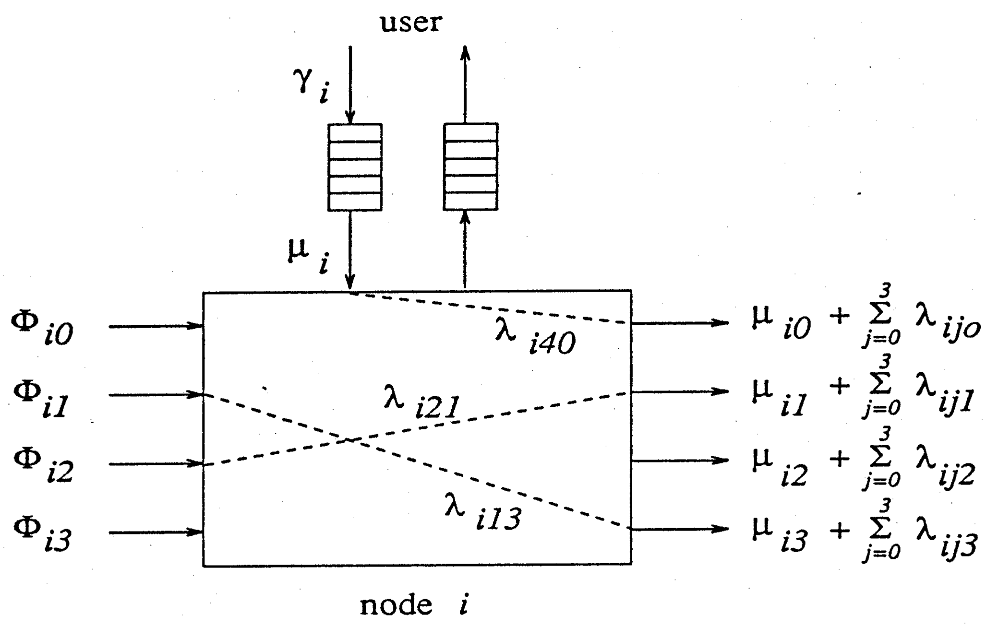

Figure3.1: Probabilistic deflection network node model.

with $k \neq 4$; i.e. μ_{ik} is the probability that anyone of the input flows uses the same link. The access probability of a single packet equals the probability to enter one of the links [6], that is:

$$\sum_{h=0}^{3} \mu_{ih} - \sum_{h} \sum_{g>h} \mu_{ih}\mu_{ig} + \sum_{h} \sum_{g>h} \sum_{f>g} \mu_{ih}\mu_{ig}\mu_{if} \quad - \quad \mu_{i0}\mu_{i1}\mu_{i2}\mu_{i3} \tag{3.3}$$

To determine the flows distribution within the network we should analyze the propagation of the packet traffic from a given node to the adjacent ones. The allocation of the input flows to the output links has to be defined according to the routing policy and the collision events. The flow of packets with destination t entering the node i from the input link j can be deflected from the output link k, if a flow of packets with destination $s \neq t$ and entering the node i from the input link $\bar{\jmath} \neq j$ exists and:

- $p(i, s, k) > p(i, t, k)$ and k is the preferred output link for the packets with destination s;
- $p(i, s, k) > p(i, t, k)$, packets with destination s have link k as second choice and have been deflected from the preferred ones.

In the case $p(i, s, k) = p(i, t, k)$, both flows have the same deflection probability, equal to 1/2; by discretizing, this is the same as to choose by means of a fair coin toss the packet that has to be deflected (i.e. the deflection probability has a uniform distribution).

For all the packets with destination t, we indicate with $\pi_0(t)$, $\pi_1(t)$, $\pi_2(t)$ and $\pi_3(t)$ the output links respectively in the order from the best to the worst choice.

Furthermore, let $\mathcal{P}$ be a proposition. We can define the characteristic function I as follows:

$$I[\mathcal{P}] = \begin{cases} 1 & \text{if } \mathcal{P} \text{ holds} \\ 0 & \text{otherwise} \end{cases}$$

The probability $\delta_{ijk:t}$ to deflect from the output link k the flow of packets in input at the link j of the node i and destination t is:

$$\begin{aligned} \delta_{ijk:t} \; = \; & \sum_{\bar{j} \neq j} \sum_{s}^{p(i,s,k)>p(i,t,k)} \Phi_{ij:t}\Phi_{i\bar{j}:s}(I[\pi_0(s)=k] + \delta_{i\bar{j}\pi_0(s):s} I[\pi_1(s)=k] + \\ & \delta_{i\bar{j}\pi_0(s):s}\delta_{i\bar{j}\pi_1(s):s} I[\pi_2(s)=k] + \delta_{i\bar{j}\pi_0(s):s}\delta_{i\bar{j}\pi_1(s):s}\delta_{i\bar{j}\pi_2(s):s} I[\pi_3(s)=k]) + \\ & \sum_{\bar{j} \neq j} \sum_{s}^{p(i,s,k)=p(i,t,k)} \frac{1}{2}\Phi_{ij:t}\Phi_{i\bar{j}:s}(I[\pi_0(s)=k] + \delta_{i\bar{j}\pi_0(s):s} I[\pi_1(s)=k] + \\ & \delta_{i\bar{j}\pi_0(s):s}\delta_{i\bar{j}\pi_1(s):s} I[\pi_2(s)=k] + \delta_{i\bar{j}\pi_0(s):s}\delta_{i\bar{j}\pi_1(s):s}\delta_{i\bar{j}\pi_2(s):s} I[\pi_3(s)=k]) \end{aligned}$$

According to the last term of both sums, a flow to a node s that is routed to its last output choice could partially deflect another flow directed to a node t. This is the consequence of considering flows of packets instead of single packets. In fact, in a discrete system, a packet that arrives on its last choice link does not compete with other packets, because they have been already routed on their better choices.

The internal flows for all the destinations $t \neq i$, i.e. the flows on the internal links, are described by:

$$\begin{aligned} \lambda_{ij\pi_0(t):t} &= \Phi_{ij:t} - \delta_{ij\pi_0(t):t} \\ \lambda_{ij\pi_1(t):t} &= \delta_{ij\pi_0(t):t} - \delta_{ij\pi_0(t):t} \cdot \delta_{ij\pi_1(t):t} \\ \lambda_{ij\pi_2(t):t} &= \delta_{ij\pi_0(t):t} \cdot \delta_{ij\pi_1(t):t} - \delta_{ij\pi_0(t):t} \cdot \delta_{ij\pi_1(t):t} \cdot \delta_{ij\pi_2(t):t} \\ \lambda_{ij\pi_3(t):t} &= \delta_{ij\pi_0(t):t} \cdot \delta_{ij\pi_1(t):t} \cdot \delta_{ij\pi_2(t):t} \end{aligned} \tag{3.4}$$

The generalization to a node with L input/output links is obtained as follows:

$$\lambda_{ij\pi_x(t):t} = \prod_{y=0}^{x-1} \delta_{ij\pi_y(t):t} - (\prod_{y=0}^{x} \delta_{ij\pi_y(t):t}) \cdot I[x < (L-1)]$$

We can observe that the internal flows are normalized. In fact:

$$\forall t \neq i \quad \sum_{x=0}^{L-1} \lambda_{ij\pi_x(t):t} = \Phi_{ij:t}$$

that is, input flows keep.

From the above equation, we can define the output configuration $\mathcal{C}$ for the node i as $\mathcal{C}_i = \{\lambda_{ijk:t} \; \forall j, k, t\}$[1]. Its *cost function* $f_\mathcal{C}$ is:

$$f_\mathcal{C}(\mathcal{C}) = \sum_j \sum_k \sum_t \lambda_{ijk:t} \cdot D(i,t,k) \tag{3.5}$$

[1]To simplify the notation, in the following, we will omit the index i.

Then, the cost function is the total path length of the packets on the output links. In the simulation model, the output configuration is built by calculating the cost for all possible configurations and by choosing the one having the lowest cost. The f_C defined in (3.5) equals the one obtained through simulations; in fact, all $\lambda_{ijk:t}$ terms are null but those corresponding to the packets that have been actually received by the node; for them $\lambda_{ijk:t} = 1$. As a consequence, f_C reduces to the sum of the paths that these packets have to follow to reach their destinations. In Section 4, we will show that our definition of deflection probability allows to determine the minimum cost configuration.

3.2 Queueing Model

The deflection network model allows to evaluate the network performances through the derivation of significative indexes such as the node service rate for user packets (i.e. the mean time spent to access the network), or the mean waiting time in the user input queues.

We assume that each slot is independent of the next one, thus the probabilities for packet extraction and packet generation have a geometric distribution. The service time in the user output queue is deterministic, whereas the service time for the user input queue has an arbitrary distribution that depends on the traffic load at each node. We assume that the user input queue is served according to a FIFO policy; waiting packets access the network as soon as there is a free slot and independently of their preferred output links. The same holds also for the user output queue, thus allowing to model these queues respectively as a Geom/G/4 and a Geom/D/4 (in both cases up to 4 packets can be served at each slot). It follows that at a node i a packet waits for a time interval Δt before being served with probability $P[B_i = \Delta t] = \mu_i \cdot (1 - \mu_i)^{\Delta t}$, with μ_i the service rate. The mean service time is $(1 - \mu_i)/\mu_i$. Reminding Section 3.1, the mean service time μ_i for a packet is given by (3.3).

The moment generating function f_M is $f_M = (\mu_i)/(1 - z - \mu_i z)$, from which we obtain the second moment $E[B_i(B_i - 1)]$. From this and by using both the most general version of the Pollaczek-Kinchin formula [5] and the Little's Theorem, we can derive the mean user input queue length and the mean waiting time of a packet. From the Pollaczek-Kinchin formula, the mean waiting time $E[W_i]$ for the packets in the node i is given by:

$$E[W_i] = \frac{1 - \mu_i}{\mu_i} + E[W_{qi}] = \frac{1 - \mu_i}{\mu_i} + \frac{\gamma_i E[B_i(B_i - 1)]}{2(1 - \gamma_i \mu_i)}$$

where $E[W_{qi}]$ is the mean waiting time of a packet in the user input queue, μ_i is the mean service rate for the node i, and γ_i is the packets generation rate. By the Little's Theorem, the mean number of packets in the node i, $E[M_i]$, is:

$$E[M_i] = \gamma_i \cdot W_i = \gamma_i \frac{1-\mu_i}{\mu_i} + \frac{\gamma_i^2 E[B_i(B_i-1)]}{2(1-\gamma_i\mu_i)} \quad (3.6)$$

Let L_{ik} be the random variable that measures the number of packets on the output link k of the node i. By reminding that at most one packet may be on one output link, and by using $P[L_{ik} = l]$ to indicate the probability that l packets are on the output link k of the node i, the mean value of L_{ik} is:

$$E[L_{ik}] = \sum_{l=0}^{1} l P[L_{ik} = l] = P[L_{ik} = 1] = \sum_{j=0}^{4} \lambda_{ijk} \quad (3.7)$$

If we indicate with K_{ik} the random variable that measures the number of packets that are in the network and have been sent from the output link k of the node i, its mean value is :

$$E[K_{ik}] = \sum_{j=0}^{4} \lambda_{ijk} \cdot \tau_{ik}$$

where τ_{ik}, as defined in Section 2, is the time a packet spends in the output link k of the node i to reach the next adjacent node. From the initial assumption that $\forall i$, $\forall k\ \tau_{ik} = 1$:

$$E[K_{ik}] = \sum_{j=0}^{4} \lambda_{ijk} \quad (3.8)$$

From (3.6), (3.7), (3.8), the total number of packets in the system $E[M]$ is given by:

$$E[M] = \sum_{i=1}^{n} (E[M_i] + \sum_{k=0}^{3} (E[L_{ik}] + E[K_{ik}])) \quad (3.9)$$

from which we calculate, by using the Little's Theorem, the mean delay of the datagram packets

$$E[T] = \frac{E[M]}{\gamma} \quad (3.10)$$

with γ the mean generation rate, i.e. $\gamma = \sum_{i=1}^{n} \gamma_i / n$.

It is easy to extend all the above relations to the general case of nodes with L I/O links.

4 Correctness Analysis

In this section, we demonstrate that the described method allows to identify the minimum cost output configuration.

Let $\mathcal{C}$ be the configuration determined by our model, and let $\mathcal{C}\prime$ be the optimal configuration, and let's suppose that $f_C(\mathcal{C}) - f_C(\mathcal{C}\prime) > 0$.

Since the input configuration is the same in both cases, the following flows preservation rules hold:

$$\Phi_{ij:t} = \Phi_{ij:t'} \;\Rightarrow\; \forall j, \forall t \sum_k \lambda_{ijk:t} = \sum_k \lambda_{ijk:t'} \tag{4.1}$$

$$\forall j \sum_k \sum_t \lambda_{ijk:t} = \sum_k \sum_t \lambda_{ijk:t'} \tag{4.2}$$

$$\forall t \sum_j \sum_k \lambda_{ijk:t} = \sum_j \sum_k \lambda_{ijk:t'} \tag{4.3}$$

$$\sum_j \sum_k \sum_t \lambda_{ijk:t} = \sum_j \sum_k \sum_t \lambda_{ijk:t'} \tag{4.4}$$

where $\Phi_{ij:t'}$ and $\lambda_{ijk:t'}$ are related to the optimal configuration.

Should $\mathcal{C}$ not to be optimal, with higher probability the flows follow their non-preferred paths, thus increasing the total path length. From the previous definition of the internal flows (3.4) and of the deflection probability, the flow with a given destination can be deflected to a worse link only when a flow with higher priority on that link exists. In the sequel, we will show that this hypothesis contains a contradiction.

Lemma 4.1: *$f_C(\mathcal{C})$ and $f_C(\mathcal{C}\prime)$ cannot differ in all terms.*

Proof. If this would hold, then the flow to each destination would be deflected from the first choice. Since the priority function cannot identify a cyclic relation among a group of nodes, this is an absurd.

Lemma 4.2: *If $f_C(\mathcal{C})$ and $f_C(\mathcal{C}\prime)$ differ in terms of two destination s and t, and two links k_1 and k_2, then $f_C(\mathcal{C}) \leq f_C(\mathcal{C}\prime)$.*

Proof. We can suppose, without loss of generality, that the flow to (with destination) s is deflected from the link k_1 to k_2 by the flow to t. If $D(i,s,k_1) < D(i,s,k_2)$, the difference $f_C(\mathcal{C}) - f_C(\mathcal{C}\prime)$ is:

$$\textstyle\sum_j(\lambda_{ijk_1:s}D(i,s,k_1) + \lambda_{ijk_2:s}D(i,s,k_2) + \lambda_{ijk_1:t}D(i,t,k_1) + \lambda_{ijk_2:t}D(i,t,k_2)) -$$
$$\textstyle\sum_j(\lambda_{ijk_1:s'}D(i,s,k_1) + \lambda_{ijk_2:s'}D(i,s,k_2) + \lambda_{ijk_1:t'}D(i,t,k_1) + \lambda_{ijk_2:t'}D(i,t,k_2))$$

Let's suppose that $f_C(\mathcal{C}) - f_C(\mathcal{C}\prime) > 0$. The previous hypothesis lead to obtain $\sum_j \lambda_{ijk_1:s'} > \sum_j \lambda_{ijk_1:s}$ and $\sum_j \lambda_{ijk_21:s'} < \sum_j \lambda_{ijk_2:s}$; moreover, for the flows preservation rule (14):

$$\sum_j(\lambda_{ijk_1:s'} - \lambda_{ijk_1:s}) \;=\; \sum_j(\lambda_{ijk_21:s} - \lambda_{ijk_21:s'}) > 0$$

We consider the following cases:

$\sum_j \lambda_{ijk_1:t} = \sum_j \lambda_{ijk_1:t'}$, $\sum_j \lambda_{ijk_2:t} = \sum_j \lambda_{ijk_2:t'}$ Since the distances are the same, it has to be $f_C(\mathcal{C}) = f_C(\mathcal{C}\prime)$, otherwise the preservation rule (14) does not hold. Moreover, if the flow to t is not deflected, then the flow to another node, say r, should exist, that deflects the flow to s. Then, the same considerations have to be applied to the pair of nodes r and s.

$\sum_j \lambda_{ijk_1:t} < \sum_j \lambda_{ijk_1:t'}$, $\sum_j \lambda_{ijk_2:t} > \sum_j \lambda_{ijk_2:t'}$ It means that the flow to t is deflected from the link k_1 as well. One or more nodes should exist that deflect both the flows to s and t from the link k_1. This contrasts with the hypothesis (this case will be considered in the sequel).

Hence, it should be $\sum_j \lambda_{ijk_1:t} > \sum_j \lambda_{ijk_1:t'}$ and $\sum_j \lambda_{ijk_2:t} < \sum_j \lambda_{ijk_2:t'}$.

Because $f_C(\mathcal{C}) - f_C(\mathcal{C}') > 0$ holds, the following has to hold too:

$$\begin{aligned} D(i,s,k_1)\sum_j(\lambda_{ijk_1:s} - \lambda_{ijk_1:s'}) \quad &+ \quad D(i,s,k_2)\sum_j(\lambda_{ijk_2:s} - \lambda_{ijk_2:s'}) + \\ D(i,t,k_1)\sum_j(\lambda_{ijk_1:t} - \lambda_{ijk_1:t'}) \quad &+ \quad D(i,t,k_2)\sum_j(\lambda_{ijk_2:t} - \lambda_{ijk_2:t'}) > 0 \end{aligned} \tag{4.5}$$

Moreover, by the flow preservation rules:

$$-\sum_j(\lambda_{ijk_1:s} - \lambda_{ijk_1:s'}) = \sum_j(\lambda_{ijk_2:s} - \lambda_{ijk_2:s'}) > 0 \tag{4.6}$$

$$\sum_j(\lambda_{ijk_1:t} - \lambda_{ijk_1:t'}) = -\sum_j(\lambda_{ijk_2:t} - \lambda_{ijk_2:t'}) > 0 \tag{4.7}$$

One of these cases holds:

- $D(i,t,k_1) = D(i,t,k_2)$ This implies that the flow to s is deflected from the link k_1 by the flow to another destination $r \neq t$; in fact, otherwise the deflection probabilities $\delta_{ijk_1:s}$ and $\delta_{ijk_1:t}$ would be the same and the flows would fairly divide both links. If another node r exists, whose flow deflects the flow to s from k_1, the same arguments hold for r and s.
- $D(i,t,k_1) > D(i,t,k_2)$ From our definition of internal flows, it would be $\sum_j \lambda_{ijk_2:t} > \sum_j \lambda_{ijk_1:t}$, that implies that the flow to t is deflected from the link k_2 by the flow to another node r. This contrasts with the hypothesis that all other flows are the same for both configurations.

Hence, it has to be $D(i,t,k_1) < D(i,t,k_2)$. Moreover, from our hypothesis it has to be $D(i,t,k_1) < D(i,s,k_1)$, since the flow to t deflects the flow to s from the link k_1, and $D(i,s,k_2) < D(i,t,k_2)$, since the same does not hold for link k_2, otherwise there should be other unpreferred links on which the flow to s can be allocated. However, this would contrast with our hypothesis (this case will be considered by the following Lemmas). Then:

$$D(i,t,k_1) < D(i,s,k_1) < D(i,s,k_2) < D(i,t,k_2) \tag{4.8}$$

From this and from (15), (16), (17), it should be:

$$D(i,t,k_1)\sum_j(\lambda_{ijk_1:t} - \lambda_{ijk_1:t'}) > D(i,s,k_1)\sum_j(\lambda_{ijk_1:s} - \lambda_{ijk_1:s'})$$

$$D(i,s,k_2)\sum_j(\lambda_{ijk_2:s} - \lambda_{ijk_2:s'}) > D(i,t,k_2)\sum_j(\lambda_{ijk_2:t} - \lambda_{ijk_2:t'})$$

From this and from (4.8) we obtain:

$$\sum_j(\lambda_{ijk_1:t} - \lambda_{ijk_1:t'}) > \sum_j(\lambda_{ijk_1:s} - \lambda_{ijk_1:s'})$$

$$\sum_j(\lambda_{ijk_2:s} - \lambda_{ijk_2:s'}) > \sum_j(\lambda_{ijk_2:t} - \lambda_{ijk_2:t'})$$

By adding the above relations:

$$\sum_j(\lambda_{ijk_1:t} - \lambda_{ijk_1:t'} + \lambda_{ijk_2:s} - \lambda_{ijk_2:s'}) > \sum_j(\lambda_{ijk_1:s} - \lambda_{ijk_1:s'} + \lambda_{ijk_2:t} - \lambda_{ijk_2:t'})$$

that does not satisfy the flow preservation rules. Hence, in this case, $\mathcal{C}$ has to be at least as good as $\mathcal{C}'$.

Lemma 4.3: *If $f_C(\mathcal{C})$ and $f_C(\mathcal{C}')$ differ in terms of more than two destinations and two links, k_1 and k_2, then $f_C(\mathcal{C}) \leq f_C(\mathcal{C}')$.*

Proof. It follows by induction from Lemma 4.2. In fact, let's suppose that $f_C(\mathcal{C}) - f_C(\mathcal{C}') > 0$, and let's consider the node $\overline{s}$ so that the flow to $\overline{s}$ is deflected from the best link that has been choosen between k_1 and k_2, let's say $\overline{k}$, without deflecting another flow to another destination. This node $\overline{s}$ should exist, because, as we have observed in Lemma 4.1, the priority function cannot identify a cyclic relation among a group of nodes. Let $\overline{t}$ be the node whose flow deflects the flow to $\overline{s}$ from $\overline{k}$, so that for all nodes t whose flows deflect the flow to $\overline{s}$, it satisfies $min_t\{D(i,\overline{s},\overline{k}) - D(i,t,\overline{k})\}$; (if for all t $D(i,t,\overline{k}) > D(i,\overline{s},\overline{k})$, then, from our definition of internal flows, no one t can deflect the flow to $\overline{s}$). From this choice, all the other nodes t do not compete on link $\overline{k}$; thus, as it has been shown in Lemma 4.2, the flow to $\overline{t}$ cannot deflect the flow to $\overline{s}$ from $\overline{k}$. By repeating this proof on the set of nodes obtained by excluding $\overline{t}$, we eliminate all the nodes. Hence, also in this case, $\mathcal{C}$ has to be at least as good as $\mathcal{C}'$.

Lemma 4.4: *If $f_C(\mathcal{C})$ and $f_C(\mathcal{C}')$ differ in terms of two destinations, s and t, and more than two links, then $f_C(\mathcal{C}) \leq f_C(\mathcal{C}')$.*

Proof. It follows by Lemma 4.2, by considering the links on which the flow to t deflects the flow to s, and also by considering the links on which the flow to s deflects the flow to t, by exchanging the destinations.

Lemma 4.5: *If $f_C(\mathcal{C})$ and $f_C(\mathcal{C}')$ differ in terms of more than two destinations and more than two links, then $f_C(\mathcal{C}) \leq f_C(\mathcal{C}')$.*

Proof. It follows easily, by noticing that this is an extension to more than two links of the case considered in Lemma 4.3.

Theorem 4.1: *Let $\mathcal{C}$ be the output configuration determined by our method. Then $\mathcal{C}$ is the optimal configuration.*

Proof. The proof follows by the above lemmata. In fact, for each other configuration $C\prime$, that can be obtained from the same input configuration of C, C has to be at least as good as $C\prime$.

The above proof demonstrates that the described model of the deflection network node succeeds in the construction of the minimum cost output configuration. Since in the simulation model the output configuration is choosen amongst all the admissible configurations, then the model derive the same output configuration. But this also implies that the analytical model produces the same traffic distribution of the simulation model, under the same offered load. This is confirmed by the results we have obtained by running the model, as shown in the next Section.

By the definitions and the proofs, it is also evident that the presented model could be easily extended to the general case of a node having L input links and L output links. The correctness analysis can be derived from the same arguments.

5 Model Execution

In this section, we present the model implementation that has been realized with the aim to verify whether or not the probabilistic model describes the deflection network behaviour with sufficient accuracy. We also present the first results obtained by running the model.

The analytical model has been implemented by using Mathematica [9] that provides several pre-defined libraries to cope with probabilistic distribution. The model is currently running on HP 9000 Unix machines.

As in [2], the evaluation of the system performances can be obtained by determining the traffic distribution in the network, for a given network topology and for the initial generation rates. When the traffic patterns reach the steady state, we can measure the mean waiting time to access the network and the mean input queue length.

To this purpose, a two phases algorithm is performed. Firstly we set the generation rates $\gamma_{i:t}$ for each source node i and to each destination node t. This allows to define the traffic configuration and to reproduce different traffic conditions for both uniform and non-uniform packet generation. Secondly, we calculate the initial traffic flows $\lambda_{ijk:t}$, by supposing that each flow is routed to the best path to destination; that is, $\lambda_{i4\pi_0(t):t} = \gamma_{i:t}$. This flow is then propagated from one node to the adjacent one by choosing the best output link according to the local routing tables. The packet delay is initially set to ∞. In this phase we do not consider possible collisions, thus the traffic distribution could be unstable, that is some link could be used more than its capacity. Anyhow, collisions are solved by trying to maintain the final cost of the obtained configuration close to the optimal one.

The second phase of the algorithm determines the flow configuration within the network with the aim of stabilizing the system. An iterative procedure is executed, that is composed of the following steps:

1. The flows are propagated from one node to the adjacent ones, and Φ_{ij} is calculated for each input link j of each node i, as $\Phi_{ij} = \sum_{\bar{j}} \lambda_{l\bar{j}k}$ for each node l so that a link exists that connects the output link k of the node l to the input link j of the node i.
2. Once the input flow composition is defined, the deflection probabilities $\delta_{ijk:t}$ are computed according to the Section 3.1.
3. By the δ's, the nodes' output configuration is derived as in the equations (3.4).
4. By adding the $\lambda_{ijk:t}$ for each destination t, the flow on each output link k of each node i is obtained.
5. Then, the service rate μ_{ij} for each output link k of each node i is computed as defined in (3.2).
6. By using the queueing model presented in Section 3.2, the mean service rates and the new delay time are obtained.
7. Finally, if the traffic pattern is stable, then we halt, else we go to 1.

The iterative procedure terminates when the mean delay becomes stable, that is, when its variation is smaller than a constant ϵ. ϵ defines the required accuracy in the approximation of the steady state.

It may happen that, during the iteration, the flows distribution does not get stable, the packets cannot enter the network with the desired rate, that is, $\mu_{ik} \leq \lambda_{i4k}$ and the input queues overflow. In this case, the simulation has to be interrupted and the initial rates have to be scaled by a factor α [2].

We have exercized the model on a 5×5 torus deflection network, with 4 I/O links for each node. We also assumed that no failures occur, that is, the network topology (and thus the routing tables) never changes. We set $\epsilon = 0.100$.

Initially, we have measured the performances under a uniform traffic conditions. All the nodes generate packets at the same rate, and the destinations are uniformily distributed amongst the network destinations, that is, all nodes have the same probability to be destination of a datagram packet.

	Probabilistic Model		Simulation Model	
Pck Gener. Rate	0.2 *pck/slot*	0.5 *pck/slot*	0.2 *pck/slot*	0.5 *pck/slot*
Queue Length (pck)	0.190	0.454	0.2	0.5
Waiting Time (slot)	0.951	0.907	1.0	1.0

Table5.1: Comparison between the two evaluation models.

In Table 5.1 we report the results obtained by considering 0.2 pck/slot, and 0.5 pck/slot generation rates. In particular, we have evaluated the mean waiting time for a packet and the queue length for each node. The simulation results, obtained under the same conditions, seem to confirm that the model properly approximates the deflection network behaviour.

6 Concluding Remarks

In this report we have presented a new evaluation model of Deflection Network performances, useful to analytically evaluate and optimize protocol parameters and adaptable to different network topology and node configuration. We also have shown that our model computes the same routing function as a deflection network, thus producing the same traffic distribution in the network. We are currently running the model and improving the Mathematica implementation although the first results seem to confirm that it properly approaches the deflection network behaviour we obtained through simulations.

References

[1] P. Baran, *"On Distributed Communications Networks"*, IEEE Transactions on Communication Systems, Vol. CS.12, March 1964, pp. 1-9.

[2] J. Bannister, F. Borgonovo, L. Fratta, M. Gerla, *"A Versatil Model for Predicting the Performance of Deflection-Routing Networks", Performance Evaluation: an International Journal*, Vol. 16, No. 1, (1992).

[3] F. Borgonovo, E. Cadorin, *"Locally-Optimal Deflection Routing in the Bidirectional Manhattan Network"*, IEEE INFOCOM '90, S. Francisco. CA, June 1990.

[4] L. Fratta, F. Borgonovo, *"Deflection Networks. Architectures for Metropolitan and Wide Area Network", Computer Networks and ISDN Systems: The International Journal of Computer and Telecommunications Networking*, Vol. 24, No. 2, (1992).

[5] L. Kleinrock, *"Queueing Systems"*, J. Wiley, 1976.

[6] A. M. Mood, F. A. Graybill, D. C. Boes, *"Introduction to the Theory of Statistics"*, McGraw-Hill, 1974.

[7] E. Pagani, G. P. Rossi, *"Providing Circuit Service over a High Speed Deflection Network"*, Proc. of Euromicro '93 Conference, Barcelona, Sep. 6-9 1993, pp.733-739.

[8] E. Pagani, G. P. Rossi, *"A Ring Based Multicast Service over a High Speed Deflection Network"*, Tech. Report 103-93, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, October 1993.

[9] S. Wolfram, *"Mathematica: A System for Doing Mathematics by Computer"*, Addison-Wesley, 1988.