This document is part of Netsukuku.
Copyright c 2007 Andrea Lo Pumo aka AlpT <alpt@freaknet.org>. All rights reserved.

Contents

1 Preface 1

2 The general idea 1

12 QSPN optimisations 20
12.1 Rtt and bandwidth 20
12.1.1 Rtt delay . . . . . .

1 Preface

The first part of the document describes the reasoning which led us to the construction of the current form of the QSPN v2. If you are just interested in the description of the QSPN v1 and v2 and you already know the concept of

2.3 The QSPN

Netsukuku implements its own algorithm, the *QSPN* (

4.2 Proprieties of the tracer packet

1. A node *D* which received a TP, can know the exact route covered by the TP. Therefore, *D* can know the route to reach the source node *S*, which sent the TP, and the routes to reach the nodes standing in the middle of the route.

   For example, suppose that the TP received by *D* is: *{S, A, B, C, D}*. By looking at the packet *D* will know that the route to reach *B* is $C \rightarrow B$ to reach *A* is $C \rightarrow B \rightarrow A$, and finally to reach *S* is $C \rightarrow B \rightarrow A \rightarrow S$. The same also applies for all the other nodes which received the TP, f.e, *B*

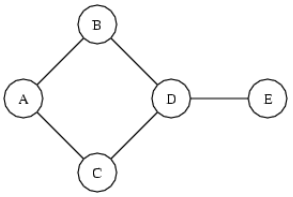


Figure 2: A simple g7aph with one segment and one cycle

Given this g7aph as input the algorithmh ill output:

$$A \rightarrow B \rightarrow D \rightarrow C$$
$$A \rightarrow B \rightarrow D \rightarrow E$$
$$A \rightarrow C \rightarrow D \rightarrow B$$
$$A \rightarrow C \rightarrow D \rightarrow E$$
$$B \rightarrow A \rightarrow C \rightarrow D \rightarrow E$$
$$B \rightarrow D \rightarrow E$$
$$B \rightarrow E \; \omega \; C$$
$$A \rightarrow B \rightarrow D \; \omega \; C \rightarrow D \; \omega \; C \rightarrow$$
$$D \rightarrow E \; \omega \; E$$
$$B \rightarrow D \rightarrow E$$

B

7 Routes

9.1 Interesting information

A node considers a received CTP interesting when its body contains at least a new route, i.e. a route that the node didn't previously know. In other words, if a CTP contains routes already known by the node, it is considered uninteresting.

When a node receives an interesting CTP, it forwards the packet to all its neighbours, excepting the one from which it has received the CTP. If, instead, the CTP is uninteresting, it will drop the packet.

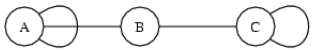
Note that

nodes. This is because an uninteresting CTP contains
previously received, memorised and forwarded by the
the other nodes already know the same routes too.

uppose 30(S)-28a.4.e. aleample

9.3 Cyclicity

When a CTP reaches the extremity of a segment, it is back forwarded, thus it's as if the extreme nodes had a link with themselves.



called *extreme nodes*

3. Q^2

1. If at least one of the primary routes[4] saved in the map of *B* and di□erent from the route $B \rightarrow A$, uses the link *l*, then *B* creates an empty ETP, otherwise the algorithm halts, i.e. *B* won't do anything.
2. If the empty ETP has been created, *B*

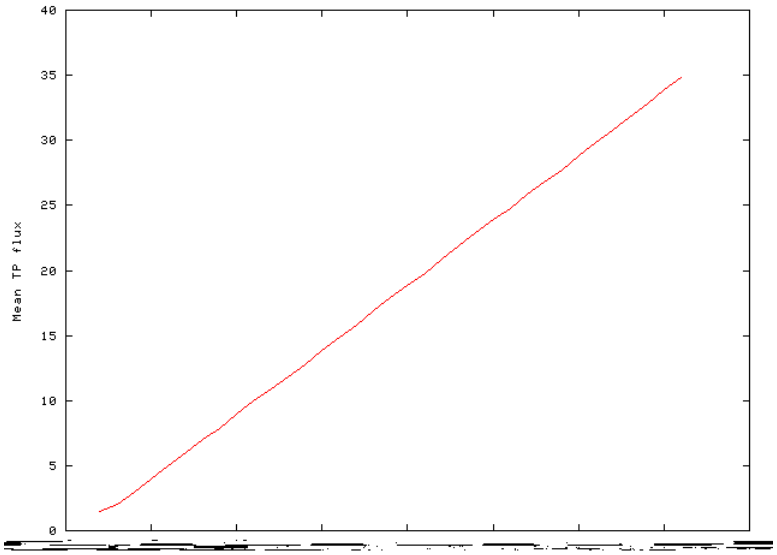
(a) if $r^{\square}$ is a better alternative to at least one primary route $m^{\square}$ M such that dst($r^{\square}$) = dst($m^{\square}$), then $r^{\square}$ is saved in the map of C (note [7]),

(b)

12 QSPN optimisations

12.1 Rtt and bandwidth

since the Rtt Delay (12.1.1) is tuned for download routes only, it is possible that some upload paths will be ignored.



By increasing the number of starter nodes the mean TP flux increases slightly until it reaches $\Phi_m = k$

^_^