



This document is part of Netsukuku.

Copyright c 2007 Andrea Lo Pumo aka AlpT <alpt@freaknet.org>. All rights reserved.

# Contents

<b>1</b>	<b>Preface</b>	<b>1</b>
<b>2</b>	<b>The general idea</b>	<b>1</b>
2.1	The network model . . . . .	1
2.2	The routing algorithm . . . . .	1
2.3	The QSPN . . . . .	2

<b>12 QSPN optimisations</b>	<b>19</b>
12.1 Rtt and bandwidth	19
12.1.1 Rtt delay	19
12.1.2 Asymmetry in $Q^2$	19
12.2 Disjoint routes	20
12.3 Cryptographic QSPN	

# 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,  $B$





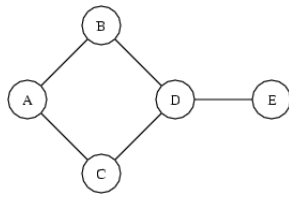


Figure 2: A simple graph with one segment and one cycle

Given this graph as input the algorithm will output:

```

      A  B  D  C
      A  B  D  E
      A  C  D  B
      A  C  D  E
      B  A  C  D  E
      B  D      E
    B      E  C
  A  B  D  C      D  C
          D  E  E
    B  D  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.

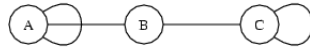
Note that

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

Suppose 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.



rtt or the bandwidth capacity. If the node has reached the *MaxRoutes* limit, it will substitute the old route with the more efficient one.

Note that this definition is more general than the previous. Indeed, if the node *S* doesn't know the route to reach *D*, the efficiency of the route *S* → *D* is equal to 0.

A node can also keep in memory more than *MaxRoutes* limit applies only to the number of routes which will be used to evaluate the



won't even be able to escape from the subcycle . This also means that all the



called *extreme nodes*. When a node becomes an extreme node, it will send another type of tracer packet, called qspn-open (which is also the name of the second phase)

3. This is a consequence of the propriety described above: every time a node joins

2.  $A$  creates the following set:

$$R = \{r \in \overline{M} \mid \text{gw}(r) = B\} \text{ where } r \in \overline{M}$$

The reason for the above procedure is simple:  $C$  considers uninteresting some of the routes contained in the received ETP, then  $C$  will send back its better routes, which have the same destination of those belonging to  $R$ , hoping that they will be useful to the previous interested nodes.

5. In every case, let

$$\overline{R} = \{r \in R$$

### **11.1.1 Real time issues**

When different events happen almost simultaneously, a node can receive contradictory

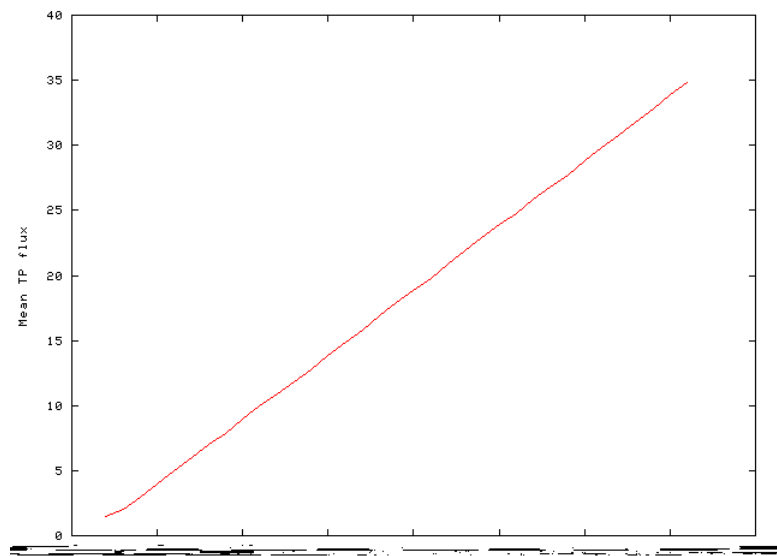






## 12.3 Cryptographic QSPN

A node could easily forge a TP, injecting in the network false routes and links infor-









$$\hat{\phantom{x}} \hat{\phantom{x}} \\ \text{—}$$