

Quantum Shortest Path Netsukuku

<http://netsukuku.freaknet.org>
AlpT (@freaknet.org)

December 19, 2006

Abstract

This document describes the QSPN, the routing discovery algorithm used by Netsukuku. Through a deductive analysis the main proprieties of the QSPN are shown. Moreover, a second version of the algorithm, is presented.

1 Preface

1. Mobiles node aren't supported by Netsukuku algorithms. ¹

2.

For the sake of simplicity, in this paper, we will assume to operate on level 0 (the level formed by 256 single nodes).

4 Tracer Packet

A *TP* (Tracer Packet) is the fundamental concept on which the QSPN is based: it is a packet which stores in its body the IDs of the traversed hops.

4.1 Tracer Packet flood

A TP isn't sent to a specific destination but instead, it is used to flood the network. By saying "the node A sends a TP" we mean that "the node A is starting a TP flood".

A TP flood passes only once through each node of the net: a node which receives a TP will forward it to all its neighbours, except the one from which it

At the end, A will know the route $A \rightarrow B \rightarrow C \rightarrow D$ and F will know the route $F \rightarrow E \rightarrow D$.

```

        current branch can't be explored anymore, therefore it is a
        valid route. Print it */
    print branch
}

```

A proof of concept of the above algorithm has been implemented in Awk [4].

Example

Consider this graph:

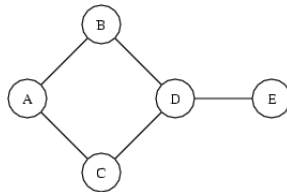


Figure 2: A simple graph with one segment and one cycle

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

```

6 Raw Tracer Packet flood

We can consider each route given by the output of the above algorithm as a

singT310(341-o)-31(o)t40-334(g42(s)-1ot4(g42(r)-1(r)-3st(thm)(t,)-4525()41-like)-420()41-in0()41-a0()41-n(it

node doesn't forward the RTP to the neighbour from which it has received the packet itself.

$Xc \dots c + XcY \rightarrow Xc \dots cY$ Example:

$123ABCD A + 123A987 \rightarrow 123ABCD A987$

$c \dots cZ + YcZ \rightarrow Yc \dots cZ$ Example:

$ABCD A123 + 987A123 \rightarrow 987ABCD A123$

$c \dots c + YcZ \rightarrow Yc \dots cZ$ Example:

$ABCD A + 987A123 \rightarrow 987ABCD A123$

Invalid route A route must not be in the form of:

$XacaY$

where a and c are two nodes. A simplification, which gives a route of this

3. In a cycle, just two TP are needed, and one is the reverse of the other. The first can be constructed in this way:

- Choose a node of the cycle, this will be the pivot node.
- Start from one neighbour of the pivot and write sequentially all the other nodes until you return to the pivot (but do not include it). Call this string C .
- The TP will be:

$$CpC$$

where p is the pivot node.

Example: if we choose the node D as the pivot, we can write the TP as:

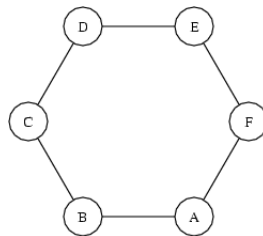


Figure 4: A cycle

$EFABCDEFABC$

and its reverse:

$CBAFEDCBAFE$

These two TPs will give all the routes to all the nodes of the cycle.

7.3 The question

Can we implement a “live” version of the Simplify Route algorithm like we did with the Generate Route one?

The reply is ahead.

8 Continuous Tracer Packet

A Continuous Tracer Packet (CTP) is an extension of the TP flood: a node will

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.

9.3 Cyclicity

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 underlined routes are the new route for G . As you can see, in the CTP (10) G doesn't find any new route, so it drops the packet and doesn't

5. \mathcal{Q}^2 is easier and simpler than \mathcal{Q} to be implemented. In general this means

because the old routes can contain interesting information about upload routes. For example, consider this segment:

... *A* *B* *C* *N*

If *N* doesn't erase the route received in the CTP, *A* will receive the following CTP:

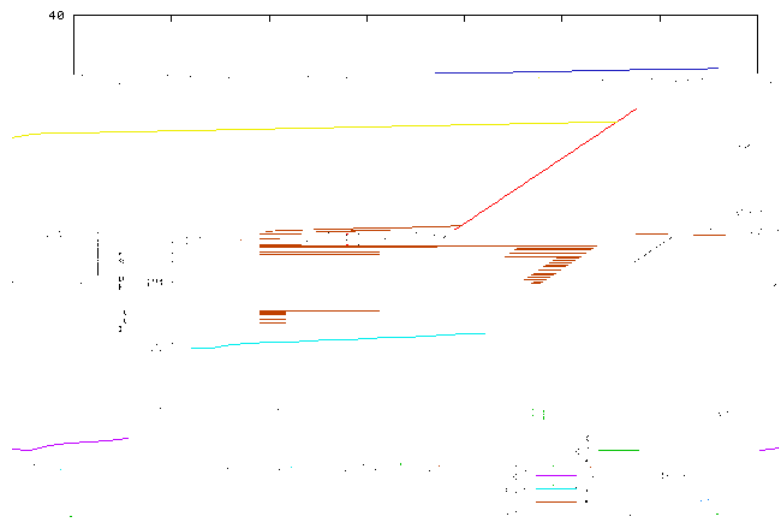
... *A* *B* *C* *N* *C* *B* *A*

In this case *A* will know the following upload route:

A *B* *C* *N*

When parsing a CTP, a node will recognize the part of the routes which are in the form of $XacaY$, where *a*, *c* are two nodes and *X* and *Y* are two generic routes. The packet will then be split in Xac and caY .

At this point we've finished. In fact, we are sure to receive at least one upload route per node because a CTP traverses each path first in one sense and then



The complete graph is the worst meor the]TJF7119.963Tf201.01450Td[Qe]TJF76.9743Tf7.8763.6145Td[2e]

different nodes, i.e. there aren't few unlucky nodes which have to bear a high

[2] Netsukuku topology document: topology.pdf

[3] Depth-First Search: http://en.wikipedia.org/wiki/Depth-first_search

[4] Generate Routes in Awk: generate _

$$\frac{\wedge \wedge}{-}$$