Contents

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*

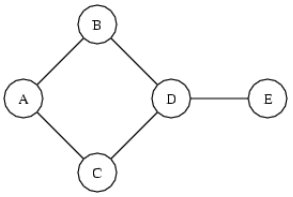


Figure 2: A simple g7aph with one segment and one cycle

Given this g7aph as input the algorithmh ill output:

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

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.

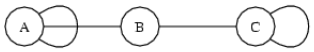
Note that

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 doesn't need synchronization. the CTPs doesn't need to have an ID, thus many nodes can send simultaneously or asynchronously a CTP without creating any problem.

1. If at least one of the primary routes

4. If R

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

$$\cdots \leftrightarrow A \leftrightarrow B \leftrightarrow C \leftrightarrow N$$

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

$$\cdots \rightarrow A \rightarrow B \rightarrow C \rightarrow N \rightarrow C \rightarrow B \rightarrow A$$

In this case A will know the following upload route:

$$A \rightarrow B \rightarrow C \rightarrow N$$

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 in the opposite. The CTP information filter, will allow us to receive only the best routes. However, since the Rtt Delay (12.1.1) is tuned for download routes only, it is possible that some upload paths will be ignored.

As explained in section 9.5 the e□ciency of a route is used as a parameter to evaluate

The figure below shows this result. In the x

where n is the number of nodes of the graph. For example, in random graphs with

[10] A Survey of Two Signature Aggregation Techniques: http://crypto.stanford.edu/ dabo/abstracts/aggsurvey.html

[11] Aggregate and Verifiably Encrypted Signatures from Bilinear Maps: http://crypto.stanford.edu/ dabo/abstracts/aggreg.html

^_^