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 $C \quad B$ to reach A is $C \quad B \quad A$, and finally to reach S is $C \quad B \quad A \quad S$. The same also applies for all the other nodes which received the TP, f.e, B
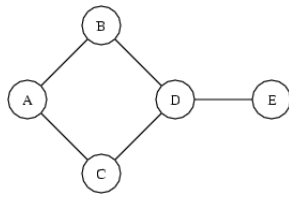
Figure 2: A simple g7aph with one segment and one cycle

Given this g7aph as input the algorithmh ill output:

$$
\begin{array}{llllll}
A & B & D & C & & \\
A & B & D & E & & \\
A & C & D & B & & \\
A & C & D & E & & \\
B & A & C & D & E & \\
B & D & & E & & \\
\end{array}
$$

$$
\begin{array}{llllll}
B & & E & C & & \\
A & B\,D & C & & D & C \\
& & D & E & E & \\
B & D & E & & & \\
B & & & & & \\
\end{array}
$$

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.
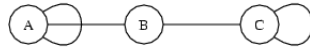
Note that

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.

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 $S \to D$ 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)

1. *B* creates the set *R*:

$$R = \left\{ \tilde{r} \mid \overline{M} \right\}$$

3. For all r

4. It sends the ETP to B

1.

If N doesn't erase the route received in the CTP, A wie CTP,A

routes. If m 2, then the route is exactly the copy of a previous route. If m 2, then the route is approximately a copy of a previous route. If $m = 1$, the route is definitely

13   Simulating the QSPN v2

As a proof of concept, we've a written the q2sim.py[6], a simulator implementing the
core of the QSPN v2.

The q2sim is an event-oriented Discrete Event Simulator. Each (event,time) pair is
pushed in aorityueue. Theainopfmeves theue

the event having the lowest 'time' value. This "popped" event is executed. In this
case, the events are the packets sent on the network.

   In this paragraph we'll analyze the results of va.ious simulations.

TP flux   The *TP flux* of a node *n*

By increasing the number of starter nodes the mean TP flux increases slightly until it reaches $m = k$. Indeed, if all the nodes of the graph are starters, then each of them will send a TP to all the other nodes. The increasing of the number of starter nodes

The same happens for a mesh graph. The mesh graph used in the simulations is a
square grid where every intersection represents a node.

If the graph is uniformly connected, as in the case of a mesh graph, then the distribu-
tion ofonsingleonTPonfluxes is uniformontoo. Theonmajority of singleonTP1nfluxes areonnearonto
the mean TP, while highest TP are shared between di erent nodes, i.e.

there aren't few unlucky nodes which have to bear a high TP flux. In the following
figure, we can ob1(ect)rve the TP flux distribution of two dit Q(ectSPN)-307(explorati)1(on)-307(of)-307(a)]T0-10.95
urroundwing hmgnod50,i ab(the)-472(highest)448((TP)448((flux.)7371(The)-48((TP)]T0-10.959Td[(flux)-321(d)-1(e)
is a 11 ×

^_^