Venice PhD course: Lab 4

April 18th 2013

Modelling an OSN

The scenario you are asked to consider is the spread of information in an online social network. It is well-known that some topics or messages go viral in such networks, meaning that the majority of nodes are aware of the message, whilst other topics remain localised and do not spread far. Your task is to build a simple PEPA model that can be used to study this phenomenon.

- We assume that the network consists of a large number of subnetworks, each of which may or may not have the message circulating within it at any given time. Users are represented by nodes and nodes can both receive messages from and publish messages onto any of the networks.
- We assume that we are concerned with the spread of a single tagged message (received from the network in the *message* action); all other messages are treated as *data*.
- It is also assumed that each node has a limited capacity so that the arrival of more data may overwrite the message even before it has been read. You may handle this probabilistically, or you may use different states to capture the state of your node (for example, assuming that the capacity of a node is limited to 5 items (message and/or data arrivals)) but this latter approach will lead to a much larger state space.
- If the message persists long enough at a node to be read, then the node may or may not choose to pass the message on, and in that case they may publish onto a network that already has the message or not. This choice is probabilistically determined by the proportion of subnetworks which currently are carrying the message.
- Whilst a subnetwork may receive the tagged message several times from different (or even the same) nodes; we assume that a node that currently holds the message does not receive it again, although it may receive it again after the message has been overwritten by other data.

Thus, using probabilistic representation of capacity, a node may be in three distinct states:

1. Here it does not have the tagged message; it can receive the message from the network (and move to state 2) or it may receive more data and remain in the same state;

2. Here it has received the tagged message but the message has not yet been read; it can receive more data that may not overwrite the message (remain in state 2) or may overwrite it (return to state 1); alternatively the message may be read moving the node to state 3.

3. Here the message has been received and read and the node may choose to propagate the message, by itself publishing the message to a subnetwork; as in state 2 it will continue to receive other data.

You should similarly think about the different states and transitions for a subnetwork. You will then be ready to build a simple PEPA model consisting of a number of nodes and a number of subnetworks and the interactions between them. Think about the starting state of your model.

From this basic model it is up to you to choose what behaviour you are interested in and how to explore it. For example, the scenario above assumes that some nodes start with the message and is suitable for a transient analysis of how the message propagates over time, but the message will always eventually be lost. Instead you could introduce an action to spontaneously create new copies of the message. Similarly I have not described any direct transfer between subnetworks except via nodes, but this could be added to the model if you felt it was appropriate.

Whichever model you develop, you are asked to write a report on your modelling study which includes

1. A description of the system you have modelled, and any assumptions that you have made in constructing your model;

2. A description of your experimental plan, i.e. what you set out to investigate and how you chose to do it. For example, this would explain which aspects of the model you chose to vary and which you kept the same throughout; it would also detail which analysis techniques you used;

3. A report of your results, including some graphs and/or tables;

4. A brief conclusion explaining what you have learned about the system through your modelling study.

A curious example in PEPA

Consider the simple toy PEPA model given below. In some configurations of the system equation the resulting model will deadlock (enter a state from which it cannot leave) whilst for others it will not.

```
r = 1.0;

// Component P
P1 = (a, r).P2;
P2 = (b, r).P3 + (d, r).P1; // dead code here: activity d is blocked
P3 = (c, r).P4; // P offers only action c
P4 = (d, r).P5;
P5 = (e, r).P1;

// Component Q
Q1 = (a, r).Q2;
Q2 = (b, r).Q3;
Q3 = (c, r).Q4 + (a, r).Q2; // dead code here: activity a is blocked
Q4 = (d, r).Q5;
Q5 = (e, r).Q1;

// no deadlock
P1[4] <a, b, c, d, e> Q1[6]

// P1[2] <a, b, c, d, e> Q1[2] -- deadlock
// P1[2] <a, b, c, d, e> Q1[4] -- no deadlock
```

Enter this example in the PEPA Eclipse Plug-in and study this bimodal behaviour. Do you understand how the deadlock is happening?(The single-step navigator may help you.) Can you discover any pattern to when the deadlock occurs?