MANIAC Challenge

The Mongoose Strategy

AbdelRahman Mohamed, Maen Khedr

Department of Computer Engineering
College of Engineering and Technology
The Arab Academy for Science and Technology
Arab Republic of Egypt
Tel: +2035622615 switch 221
Email: abdelrahman.aast@gmail.com
Maan.khedr@gmail.com

Basically, we head to turn the famous MANETs forwarding and dropping dilemma to a classical game theory problem. After reaching this point, we can then apply all, well known, theories in solving such kinds of games.

Game theory, at large, depends on the knowledge of all possible strategies in any player's strategy profile. Each player can build up the game matrix, fill in the payoffs, calculate the oddments and finally mix the strategies available to him in accordance to certain ratios (oddments) for his own maximum profit.

Researches have been made exhaustive in that area in which our strategy depends on making intensive use of them. So as to do so, we first have to define what we mean by *the knowledge of all possible strategies in any player's strategy profile*. Of course the number of ideas for making up a strategy could clearly term to infinity. In spite of that, a strategy in the MANIAC challenge could be defined with a finite set of factors. These factors might be: probability, randomness, punishment for uncooperative nodes, network distance (in terms of the number of hops), etc.

Being convinced with this fact, we are currently implementing an algorithm that fills in values in all possible factors that we could come up with. And according to these, configurable (with respect to the current state of the game), values, the decision of what is to be done with the packet is made.

For example, if our algorithm finds out that a node make its forwarding and dropping strategy totally random, we always drop its packets. This node simply doesn't care about the reputation issue, then why should we even bother!

Rules:

Before the sneaking algorithm (the one responsible to execute what is discussed above) works, there are some rules that are applied no matter what the situation was. These rules are:

- If our decision to forward that packet will deliver it to its intended destination, the packet is dropped with a relatively high probability.
- If the packet is destined to our partner, it is blindly forwarded.
- We keep track with a rough figure of the current score for each team so that if a team's score boosts, the future decisions of forwarding its packets is tuned down.

Tools to be used in our strategy:

- Promiscuous will be needed in our algorithm.
- We found that building a communication protocol between our team's nodes will help with increasing the precision of the values obtained by the sneaking algorithm.
  Jobs of the communication protocol may include:
  - Inform about any new UNDEFINED forwarding strategies by aiding in the factor configuration process.
  - Inform the other team's node that it should shortly receive an indirectly forwarded packet (that the current node just forwarded). If the packet was not received, an intermediate node is being uncooperative.
  - Inform about any node that was found to be uncooperative under any circumstances.

Implemented Successfully:

Of what we have discussed, we are currently working on finishing and fine tuning the sneaking algorithm. The set of rules and the communication protocol are already done.

In parallel to the implementation scheme, we finished installing the MobiEmu emulator for testing after completion of the final implementation.

Finally, the whole Mongoose strategy will be coded on the MANIAC API provided for getting ready to the competition.