MANIAC Challenge: UNC Charlotte Team

Lin Li Fan Li Yu Wang
Department of Computer Science
The University of North Carolina at Charlotte, USA

According to the rules of MANIAC Challenge, a team gains ten points for each packet belonging to one of their flows to reach its intended destination and loses one point for each packet it transmits. Therefore, to achieve higher points, two straightforward strategies are: (1) forwarding fewer packets for other teams, and (2) receiving more packets destinated to ourselves. The second strategy is out of our control with current MANIAC implementation, thus the best strategy is dropping all packets passing through our team nodes to avoid point deduction. However, if every team behaves like that, no packets can be transmitted in such non-cooperative network. Therefore, we investigate the tradeoff between forwarding packets and dropping packets by designing a new randomized forwarding strategy.

The basic idea of our randomized forwarding strategy is to drop or redirect packets with an adaptive probability, which depends on the statistics of local traffic (here, the traffic of a neighbor is the number of packets which has been forwarded to that neighbor via current node). Each of our team nodes maintains the statistics of how many packets forwarded to each neighbor. When a packet arrives, our strategy acts as following: (1) if the packet is for our team member, we forward it without any hesitation; (2) if the packet is for our direct neighbor, we drop it to prevent the neighbor from gaining 10 points; otherwise, (3) we redirect the packet to the neighbor who has less traffic in the past, and then drop the packet with a probability p. Here, the dropping probability p is adaptively proportional to the ratio of the number of packets has been forwarded to this neighbor to the total number of packets has been forwarded.

We implement our strategy using MANIAC API, however, we build our own simulation environment to verify our strategy on a single machine. In the simulation environment, we randomly generate traffic passing through the machine. Simulation results prove that our strategy can work properly.

We also intend to include more elements into our forwarding strategy, such as traffic type (real time or not), node movement (by team members), node distribution, etc. However, following the K.I.S.S. (Keep It Sweet & Simple) Principle, we stick with our simple randomized forwarding strategy, and leave other solutions as possible future work.