QOS DRIVEN SURVIVABLE MOBILE AD-HOC IP COMPATIBLE COGNITIVE NETWORKS

E. Gelenbe and R. Lent
School of Electrical Engineering and Computer Science
University of Central Florida

This paper introduces a new survivable self-organizing routing scheme with best-effort QoS end-to-end connections for mobile ad-hoc networks inspired by the Cognitive Packet Network (CPN) approach. This work is conducted under U.S. Army STRICOM Contract via NAWC No. N61339-02-C0117.We will present the novel algorithm as well as a test-bed implementation.

This scheme uses Smart Packets (SP) to learn paths to destinations. As they travel from node to node, they record information about their path in their Cognitive Map, a private area in the packet created for that purpose. After they reach their destinations, an Acknowledgement Packet (ACK) heads back to the source node using the reverse path both to confirm arrival of the packet, and to distribute routing information grabbed by the SP.

Payload is transported using Dumb Packets (DP). They are given a map, discovered previously by a SP, with the path to take to reach the destination. At every hop, they inspect their map to find out where to go next.

Smart Packets use a Random Neural Network (RNN) to take routing decisions at every node. A RNN is constructed at nodes in the path to the destination with as many neurons as next-hop possible decisions at that node. Then, reinforcement learning uses information shared by previous Smart Packets to adjust both excitatory and inhibitory weight matrices in the RNN. An ever possible convergent state of the RNN provides the next-hop decision for the SP. Whenever the use of a RNN is not possible because of lack of surrounding information, Smart Packets use broadcast as a last-resort alternative.

Applications are able to request QoS goals for their connections. For example, they may indicate that they need minimum delay, or maximum dispersion for security reasons. Smart Packets take into account these requirements using an appropriate expression in the reinforcement learning algorithm.

We are conducting simulation experiments with large number of mobile nodes under various network conditions. We have implemented the algorithm for the Linux kernel 2.4.18, and our mobile architecture can be seamlessly connected to IP networks.