Project Notes **Spring 2018**

RL+MANETs

Professor Peter Stone *Students: Justin Lewis, Jacob Winick*

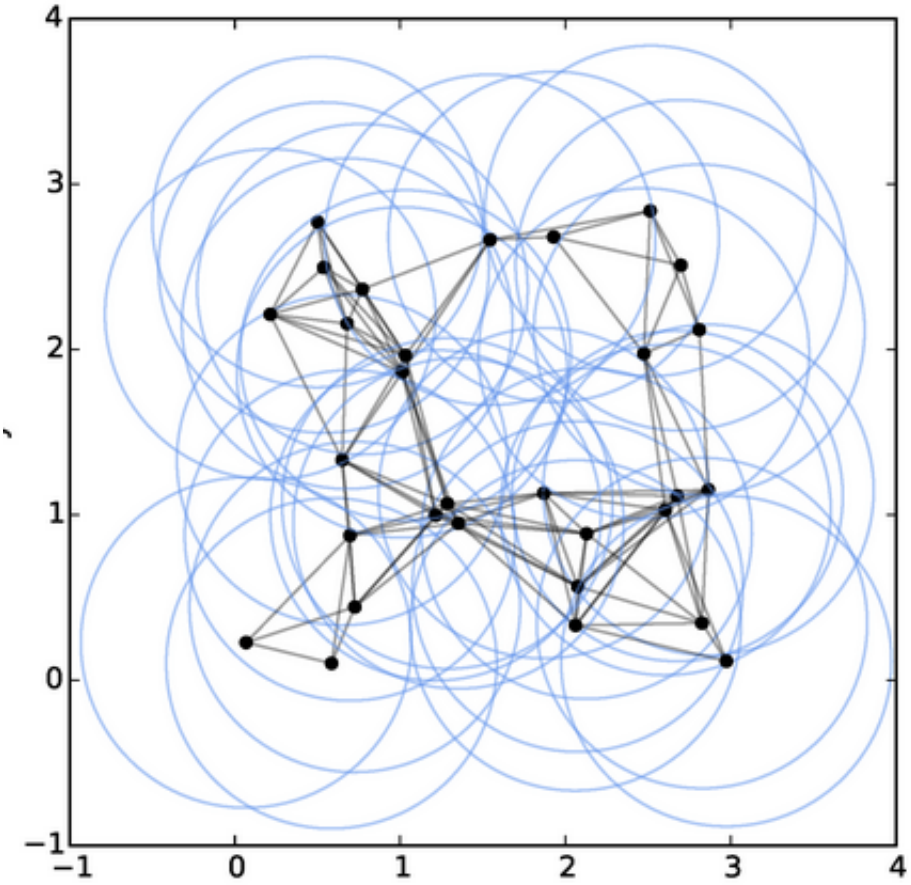
1 Research Problem Summary:

- *Overall goal:* apply Reinforcement Learning to packet route optimization for a class of Wireless Ad Hoc Network known as Mobilized Ad Hoc Networks (MANETs)
- Route optimization for MANETs is difficult
  - network topology is dynamic
  - generally decentralized
- Additional problem difficulties:
  - transmission cost / interference
  - power constraints
  - scalability

2 MANETs:

- decentralized network structures
- nodes are mobile:
  - can move in space
  - may join/leave any time
- *applications*:
  - mobile internet (e.g: project LOON)
  - mobile phone networks / SPANs
  - mobile sensor networks
  - vehicular networks / VANETs
  - military applications
  - smart cities

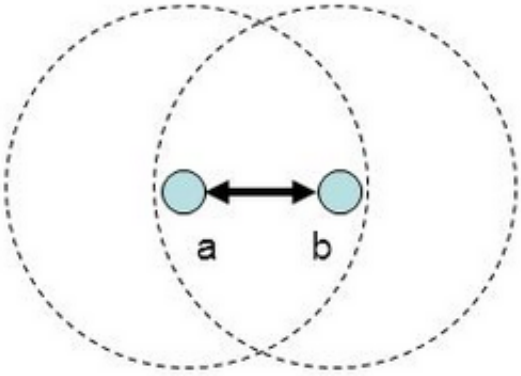
2.0.1 MANET Example:



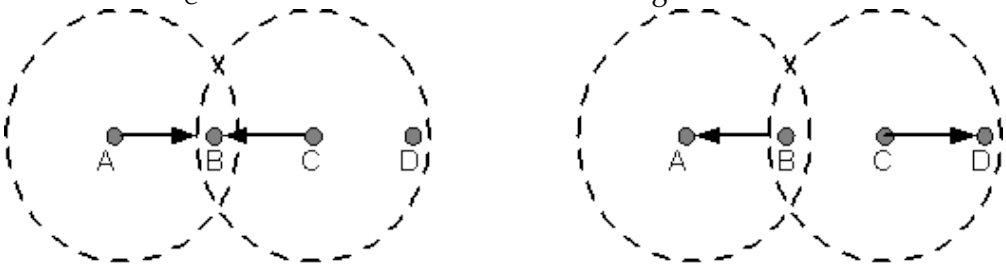
Range of communication demonstrated by surrounding circle

3 Communication Scheme:

Each device uses radio waves to communicate packets to other devices. The broadcasted message is transmitted between nodes over a certain frequency channel. If two devices use the channel simulatenously, a "collision" occurs. This nullifies both broadcasts.



This can cause complex interactions amongst the communication nodes. For example in the *"hidden node problem"* (see below), node N_A can see N_B, but cannot see N_C. If N_A and N_C both communicate to N_B simultaneously, then a collision will occur; furthermore, node N_A cannot attribute this collision to N_C unless information is shared amongst the nodes.



Hidden node problem Exposed node problem

These interference problems coupled with a decentralized control scheme can lead to inefficient routing. A large portion of time can be spent rebroadcasting messages which previously collided.

4 Traditional Solutions (abridged):

1. Reactive routing (AODV):

    - This method is reactive because nodes do not save information regarding the entire network topology. They only maintain connections to local neighbors.
    - Nodes periodically send "hello" messages to gain information about their immediate neighbors.
    - Whenever a node wishes to send a packet through the network, it sends a "route request" (RREQ) to find a path to the destination. This request is passed along the network following a fixed protocol until the destination is found. Then the information is passed back through the network to the sending node.

2. Proactive routing (OLSR):

    - This method is proactive as the nodes maintain a local "image" of the network's full topology.
    - Nodes periodically send "hello" messages to gain information about their immediate neighbors.
    - Additionally, nodes broadcast their local connections to the rest of the network through "flooding". Essentially, the information is spread to every connected node in the network by a fixed policy. "Flooding" can cause unnecessary transmission and cause packet collisions.
    - With a local image of the network, a version of shortest path algorithm is applied to find a suitable path to the destination.

5 Why Reinforcement Learning?:

- More direct, global combinatiorial optimization in a distributed fashion seems unlikely to scale effectively (finding the exact optimum is NP-hard I would imagine)
- The "best" strategy given limited computation time and limited power usage is difficult to derive directly
- RL agents can learn how to best encorporate available information at a node to improve routing (while maintaining constraints such as power usage)
- RL agents can learn how to most effectively cooperate (direct analysis is difficult). Ex: how often and how much information should be shared amongst neighbors.
- The computation time complexity of a DRL solution could potentially be smaller (i.e. faster) than a more straightforward optimization algorithm (latency is important)

6 RL Problem Constraints:

1. Decentralized control
2. Multi-agent learning scheme
3. Full cooperation (protocol enforced)
4. Arbitrary intercommunication (with associated cost)

7 Physical / Environmental Constraints

1. Number of unique frequency channels N_c is variable and will be analyzed to see achievability of specific data rates
2. Range of unique frequency channels is proportional to transmission frequency

8 RL Agent State Space:

- there are a number of potential state types which may prove useful in deriving the best policy. Below is a list of state classes and specifics.

1. Routing Information:
    - Packet source / destination
2. Environment Information:
    - Current estimate of topology of network
    - Strength of network connections
    - Physical location and heading of nodes (possible?)
3. System Information:
    - number packets in local buffer
    - battery state
4. History
    - time of day
    - time elapsed between certain actions

9 RL Agent Action Space:

- similar to the state space, there are a number of distinct actions which can be considered.
- I would imagine that the action space should be structured as a sequence of decisions within time slots (viewed as a time series)

1. Packet Related

    - Forward packet to neighbor (on specific channel)
    - Drop packet
    - Save packet to buffer
    - WAIT (do nothing)

2. Environment related

    - Sense channels (local connections)
    - Measure local node locations (may not be physically feasable)

3. Cooperation related

    - Proactively / reactively share information with nodes in network
    - Proactively / reactively request information from network

10 Reinforcement Signal / Reinforcement Protocol

- the reinforcement signal should be related to the following:

    - Quality of service (QoS)
    - Transmission slowdown / delay
    - Power usage
    - Transmission collisions

- How the reinforcement signal should be exactly formulated will require more consideration.
- The way in which the reinforcement signal is provided to an agent is also an open question.

11 Initial RL Formulation:

Assume constraints from sections 6 and 7. The following is an initial attempt at formulating the state space, action space, and learning mechanism.

11.1 State Space

Let the state space be broken into two primary sections:

1. Environment state

   Let a node's estimate of the true environment state be represented by a directed graphical model $G(N, E)$ with nodes N and edges E.

   Each node $n \in N$ represents an agent within the assumed collaborative scheme. (Let n_ℓ represent the local agent). Each edge $e \in E$ represents the communication channel connecting two nodes.

   Each node $n \in \{N \diagup n_\ell\}$ will hold the following information:

   - node ID
   - flag if node is a destination of packet in buffer
   - counter for time elapsed since last received topology information from node
   - information regarding collisions between n_ℓ and $n \in \{N \diagup n_\ell\}$

2. Local buffer state

11.2 Action Space

11.3 Learning Method

12 References:

13 Random Thoughts:

- nodes can "sniff" packets and infer the graph structure
- nodes could proactively probe the network to learn its structure (causal graph inference)
- sharing information with neighboring nodes comes at a cost
  - how can the agent's local information be shared at different sizes?
  - essentially a variable-sized embedding?
- perhaps it is wise to share some information at times which are less busy? (offline learning)
- it seems that the model complexity of each node in the network could be variable and dependent on its position in the network
  - less connected nodes probably are less important in the global optimization and thus require less information
  - spend power most efficintly based on network position
- if a core model is shared amongst the agents, then it may improve coordination
  - each node can anticipate the actions of the others
  - maybe some portion of the model is shared and another is not (role assignment)
- what about fairness?
  - for example, in a mobile phone network the users closest to the bay stations will receive higher traffic loads
  - perhaps these methods make more sense in settings where fairness isn't as relevant (team strategy vs. mutual cooperation)
- perhaps some information is always passed along as part of the packet header? Inspiration from ant colony optimization.