CPE 400 Computer Communication Networks

FALL 2018

Project Topics for CPE 400

(1) Dynamic routing mechanism design with focus on throughput

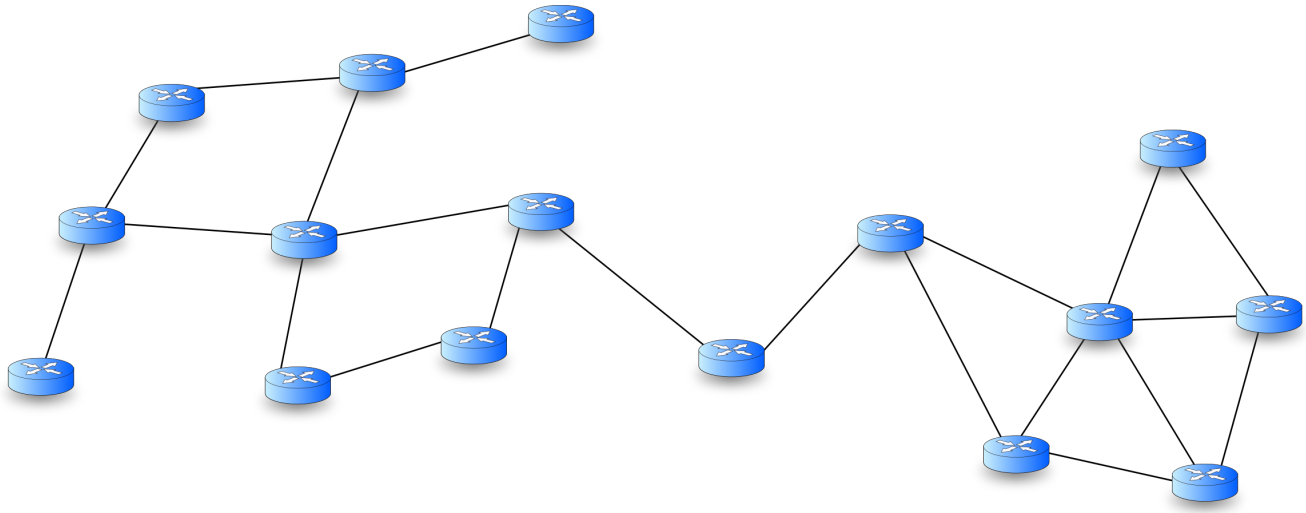


Figure 1 - Generic network of routers

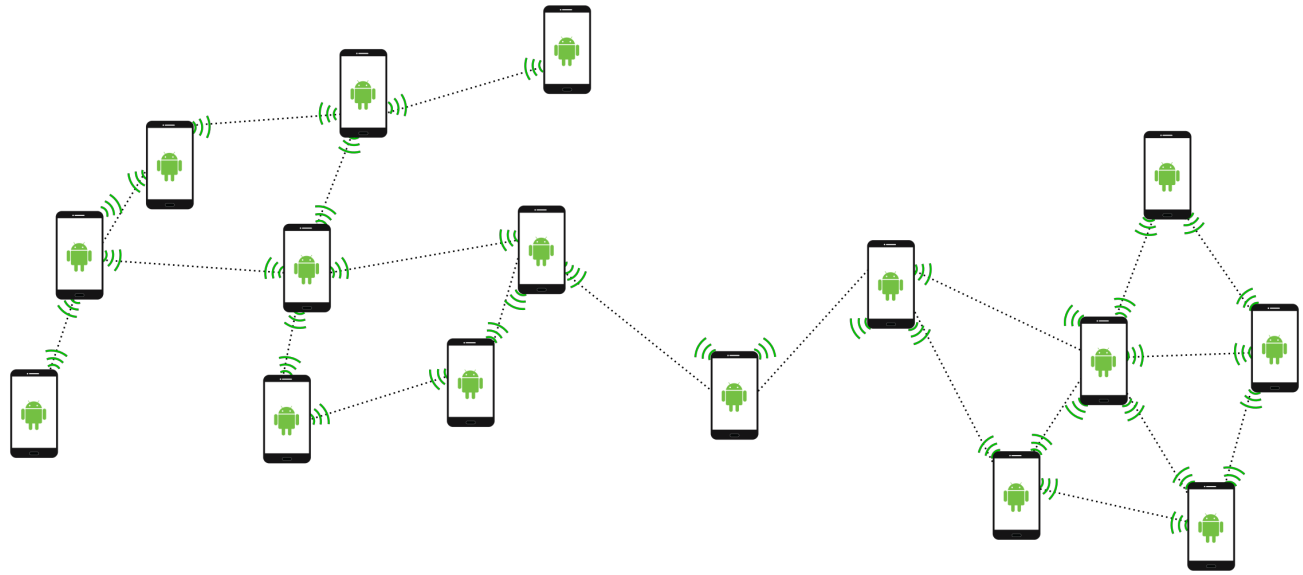


Figure 2 - Same topology as Figure 1, but now all links are wireless, and each device can be both source of traffic and also act as a relay

Create and simulate a new routing strategy that maximizes the overall throughput of a mesh network. Throughput is affected by many factors that should be considered, such as nodal processing delay, overloaded buffers, loss, etc. The more realistic assumptions you can make for your network, better it is.

(2) Dynamic routing mechanism design with focus on energy conservation

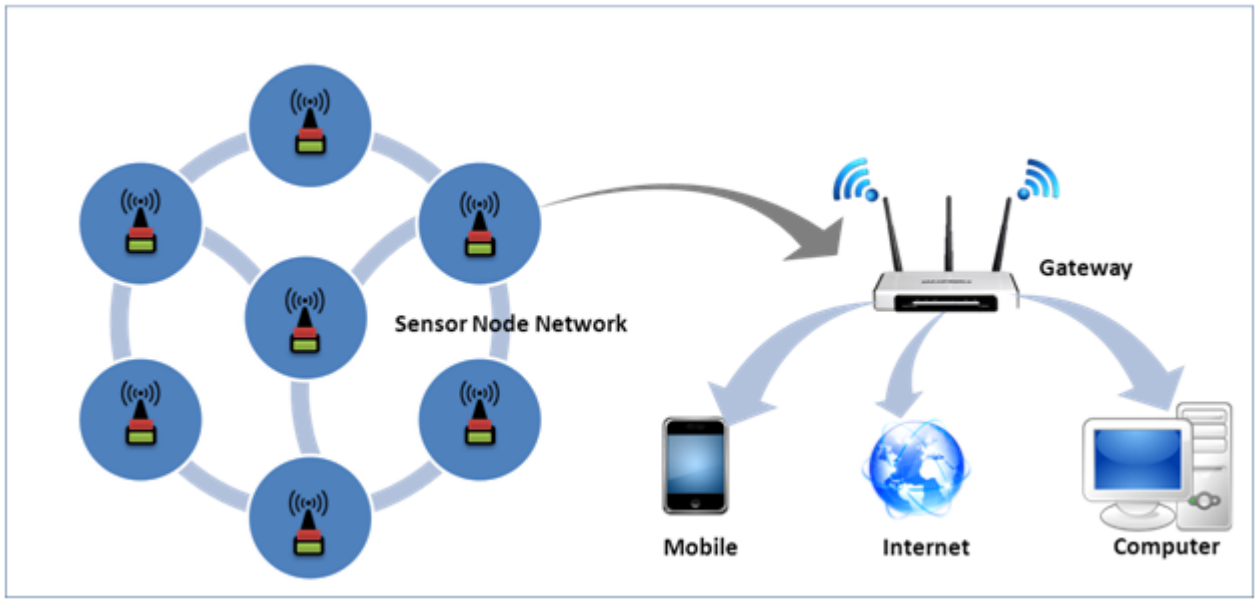


Figure 3 - Wireless Sensor Network example

Create and simulate a sensor network. The objective is to increase the longevity of the network by using dynamic routing. Sensors are energy constrained and are not super powerful like common computers. Every time a sensor transmits some packets, it will decrease the amount of energy left. The team will create a dynamic routing scheme to try and maintain all nodes of the network online for as long as possible.

(3) Hybrid dynamic routing mechanism design

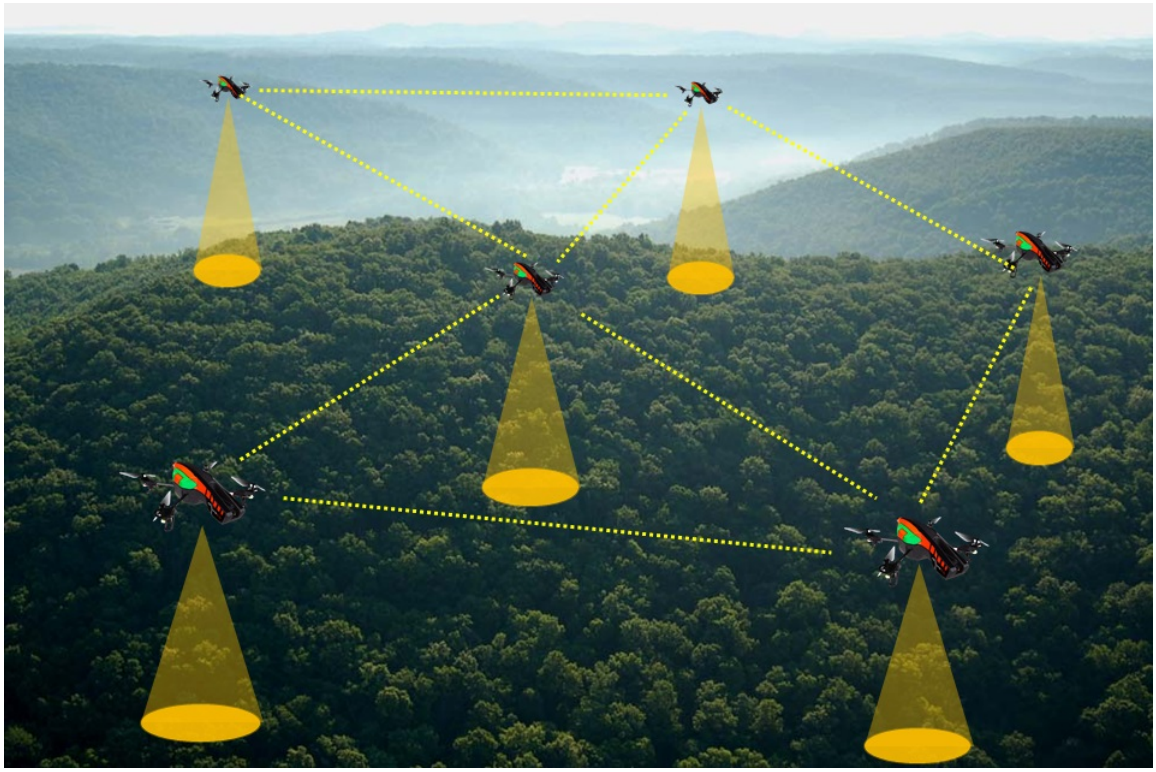


Figure 4 - Network of drones, energy AND throughput are points of concern

Imagine if some drones are collecting some data and sending it to a command and control base (Figure 4). If the application at each drone has low data rate requirement, it does not require too much throughput, instead they could be better off having longer delays in the end-to-end transmission, and saving energy at the same time (Figure 4). For high data rate requirement, the focus is more on throughput. In this project the team will simulate such scenario, combining both throughput and energy. Given that each node only requires a certain amount of throughput, how can the packets be routed so that the network of drones can stay "alive" longer, and the communication to the C&C is not broken?

(4) Dynamic routing mechanism design in faulty network

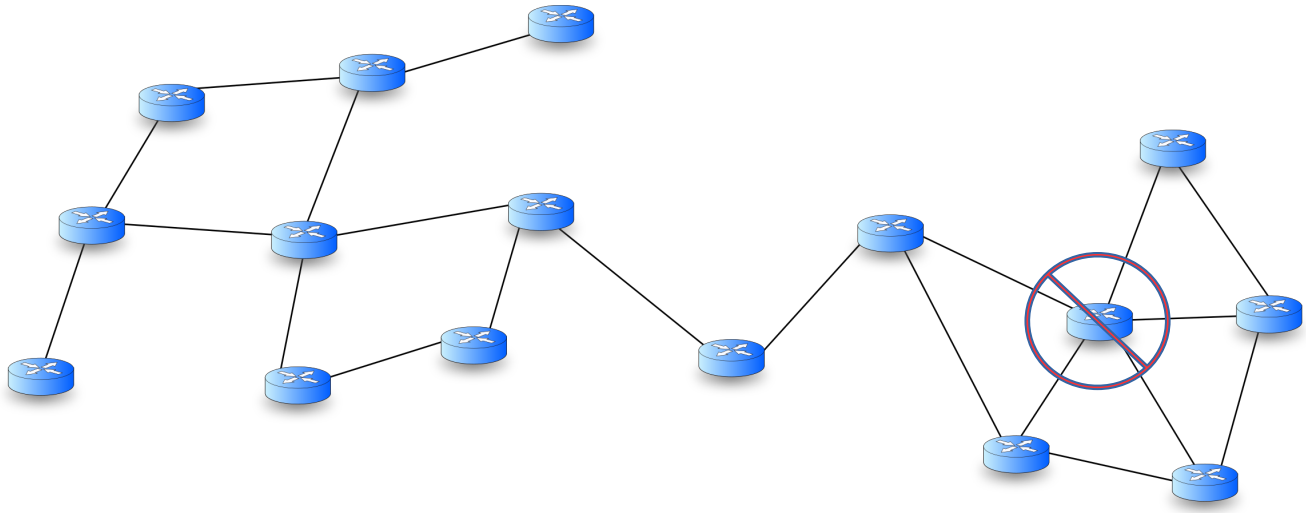
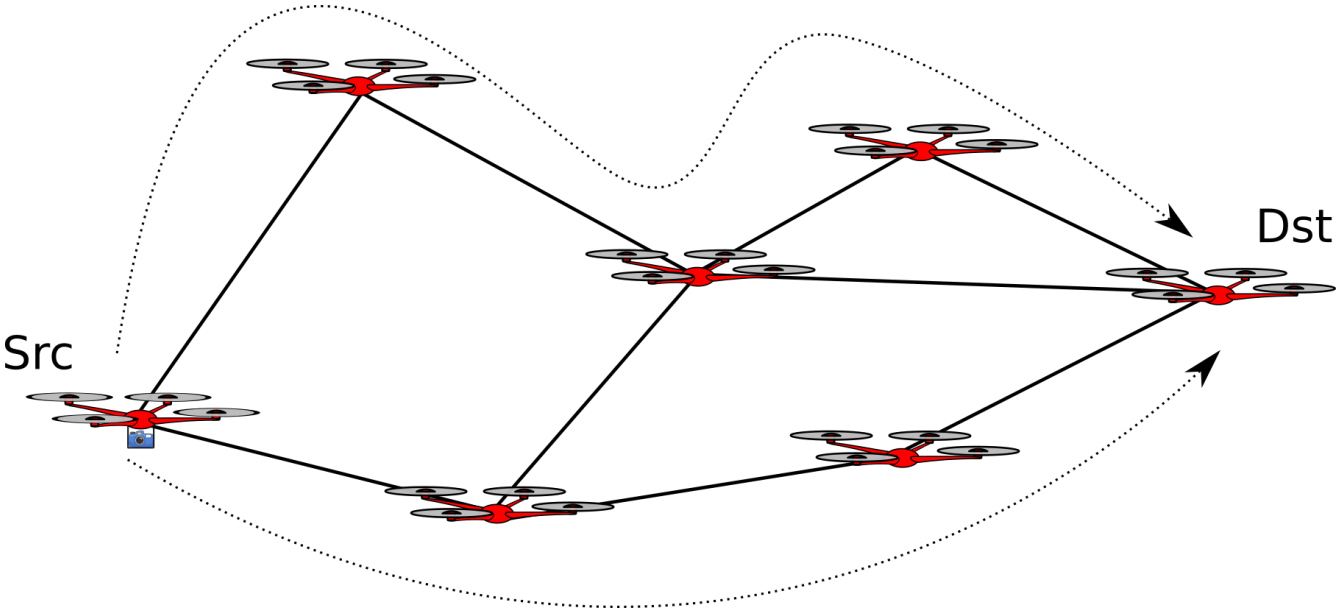


Figure 5 - Faulty network: how to route if a node suddenly stops working for a while?

The team will simulate a mesh network where nodes and links may fail (Figure 5). Nodes and links may fail intermittently, as an input to the simulation, each node and link will have a certain probability to fail. When such failure occurs, the network must adapt and re-route to avoid the faulty link/node.

(5) Application-aware routing



The team will simulate a mesh network where the routing strategy must consider the type off application messages being generated. For example, an MPEG streaming app might send different types of video frames (frame types I, P or B). Each frame type (or application message type) has its use, however, they may not be as important as others (I frames contain more information than B or P). If the routing is aware of the type of message that is being sent/relayed, how can it pick the best next-hop? Is it better to send redundant messages with the same information? The team is expected to design their own routing strategy. Each link in the network has a probability of failing, and the final demonstration should account for that.

(6) Implement a real multi-hop network

The team will develop an Android application using open-source hardware (Raspberry Pi), and open software (Linux: Android/Ubuntu/Raspbian/etc) to create a mesh network (Nearby Connections) among the devices and transfer some data in a multi-hop manner.

(7) Routing in multi-hop network (emulation edition)

The goal here is to create a network emulation and your own routing strategy. There will be an input file with parameters for nodes to read from and start building their routing table. Think of it as having some Terminal windows open, in each one you will run a command (the Node!), with different arguments in each window. The program will use socket programming to transfer messages from node to node, and a simple text-based GUI should be provided where we can have some commands executed (i.e. "send message M to node A"). The team can choose any routing approach (DV, LS, SR), and should be able to demonstrate the functionality of the emulation (we want to see messages when a message is routed through multiple hops until it reaches its destination). We will provide a simple Python example showing socket programming. The emulation should be able to read any input file provided it's properly formatted.

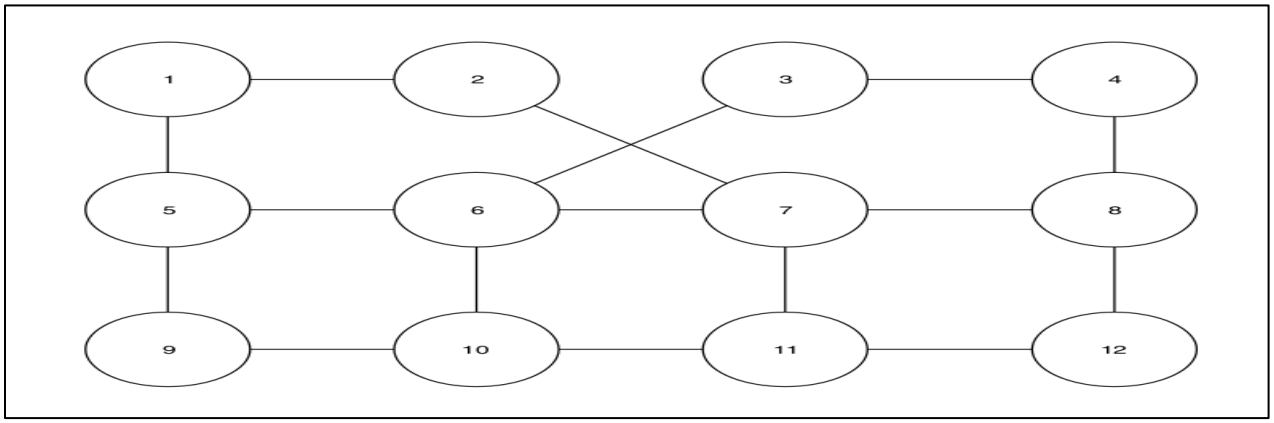


Figure – A simple network topology

In this project you will be given the basic Python code to create and initialize a network of nodes in an actual network using socket programming. Your assignment will be to implement one of the routing protocols covered in class, and extend this code so that any node in the network is able to communicate with any other node by constructing routing information that is shared among the nodes. This algorithm should be dynamic, so that if one of the nodes is turned off, the network recognizes this and adjusts by propagating updated routing information and making that information available to all nodes.

Rubrics for the project grading

CPE 400: 20% of the total grade

Project Submission due by Wednesday, Dec 5th

Deliverables:

1) **Code:**
   a) **[5] Turning in code that compiles and runs properly**
      i) Code compiling properly and providing results as explained in the report: 5;
      ii) Code compiling but not providing results as explained in the report: 2;
      iii) Code not compiling: 0
   b) **[2] Documenting the code**
      i) Clear explanation of the entire code: 2;
      ii) Partial explanation of the code: 1;
      iii) No explanation of the code: 0

2) **Technical Report**
   a) **[5] A report explaining the functionality of the protocol**
      i) Protocol/functionality addressed extensively in details with as many realistic aspects as possible + error handling scenarios: 5;
      ii) Protocol/functionality addressed with some details and some error handling but not extensive: 3;
      iii) Unclear explanation of the protocol: 1;
      iv) No explanation: 0
   b) **[3] Out of the box thinking: novel contribution**
      i) Important Novel Protocol/functionality contributed with details: 3;
      ii) Important Novel Protocol/functionality contributed but explanation not clear: 2;
      iii) Novel Protocol/functionality mentioned but no explanation provided: 1;
      iv) No Novel contribution provided: 0
   c) **[5] Results and analysis of the results**
      i) Results and analysis contributed with details: 5;
      ii) Results and analysis contributed but explanation not clear: 3;
      iii) Results mentioned but no analysis provided: 1;
      iv) No results provided: 0