Algorithms 2500 – Section 1A and Section 1B
Second Project
Deadlines: First report 4/25 and Second report 5/2

Algorithms for optimizing attacks on communication networks

Maximum points 100

General information

Maximum group size 3.

Any programming language can be used for the implementation.

The project reports should be submitted by email to the TA the day of the deadline and in hardcopy to the Instructor the first class after the deadline. Additionally, the reports and the code should be submitted to the TA electronically the day of the deadline.

The project execution should be demonstrated to and discussed with the TA on the week of the deadline of the second report. i.e. (First week of May)

Project goals

Communication networks are the basis of our everyday life. The Internet is probably the most common communication network today in use. It is composed by millions of nodes. Such nodes are servers or data centers, that store and elaborate the data, and routers that are used to relay the information to the end user. Servers and routers are connected through high capacity fiber communication cables. The data follows a path in the network which is determined by the routing algorithm in use.

The cyber-physical nature of the Internet makes it susceptible to both physical and cyber-attacks. In this project, we will focus on physical attacks at the link level, i.e., an attacker will try to disrupt communications by attacking (destroying) links in the network.

The goal of this project is to design and compare the performance of different attack strategies under different routing strategies.

Network model and problem formulation

Communication networks can be represented using undirected graphs $G = (V,E)$. In our project, we consider one node $S$ as source of information and another node $D$ as the destination. They need to communicate through the network.

Links in the network have a maximum capacity. This capacity should be randomly selected in the range $[1,20]$ for each link.
We assume that time is slotted in intervals. The attacker can attack one link for each slot.

The network uses MAXFLOW as a routing algorithm, that is the amount of data transferred from S to D is given by the MAXFLOW between these two nodes and the amount of flow in each link is also given by this algorithm.

**GOAL:** The goal of the attacker is to **disrupt as much communication, as soon as possible** by selecting an appropriate link to destroy at each interval.

We will consider two routing strategies, both based on MAXFLOW:

- The first routing is **static**, i.e. even when under attack the routing WILL NOT change. The network will keep using the same paths for the flow from S to D. If a link is attacked, then all paths through that link will become unavailable. The set of paths used can be calculated using the set of augmenting paths of the Ford–Fulkerson algorithm.
- The second routing is **reactive**, that is the MAXFLOW is recalculated at each step after the attack has happened. This way the network can re-optimize the dataflow as more links become unavailable.

**Attacker algorithms**

You should design two attack strategies, one designed for the static routing and the other for the reactive routing. You can assume that the attacker knows everything, i.e., the topology of the network, the capacity of the links, source and destination, the residual capacity of the links, the paths used by the routing algorithm, etc.

The goal of your algorithms is to reduce the amount of flow from S to D as soon as possible. To this purpose the attacker should carefully select the links to attack.

As a baseline comparison, you should implement a random attack strategy that at each iteration attacks a random link in the network.

**Implementation and evaluation**

You should download the topology of Kentucky Datalink available at this link:

http://www.topology-zoo.org/files/Kdl.gml

Note that the topology/data is in GML format to represent the graph. You should develop a parser to read the file and import it in your program. Nodes have ids, which can be used to determine the endpoints of the links. Possible inconsistencies can be fixed as appropriate.

You should add two nodes S and D to the graph. Each of these should be connected to K nodes in the network selected at random. The capacity of the links between S or D, to the K selected nodes should be 20. Such links cannot be attacked by the attacker.

You should perform three sets of experiments with static routing and reactive routing.

In both cases you should make three graphs.
The first graph focuses on the flow over time between S and D. It should have the time slot on the x-axis, and the flow at that time slot on the y-axis. On this graph, you should have three lines, one for each of your algorithms and one for the random attack approach. You can set K = 10.

The second graph should find the time at which 50% of the original flow has been disrupted with the different approaches. You should make a histogram that compares such time for each approach. Also in this case the graph should have three bars, one for each algorithm. You can set K = 10.

The third graph studies the impact of K. The x-axis should be the value of K, i.e. the number of nodes to which S and D are randomly connected. K may go from 1 to 30. The y-axis should be number of rounds needed by the attacker to make the maximum flow equal to zero. You should compare all three approaches.

Overall 6 graphs should be produced. Make sure graphs are readable when printed.

Reports

Two reports should be prepared. The first report does not require implementation and should include at least the following sections:

- Abstract: Summary of the following sections
- Introduction and motivation: Overview of the problem
- Proposed solutions: Sufficiently detailed pseudo-code, description and complexity analysis of the proposed algorithms
- Plan of experiments: Description of the methodology which is planned to be used in the experiments, significance and expected results.

The second report requires implementation and describes the implementation and results. It should include at least the following sections:

- Abstract: Summary of the following sections and results
- Implementation: Description of the methodology used for implementing the proposed solutions (e.g. relevant classes and data structures, structure of the program, etc.).
- Experiments: Description of the experiments performed and obtained results. Results should be represented in a graphical form as well as discussed in the writing.
- Conclusions: Summary of the work and final considerations.

Similar sections can be merged if a single report is provided. In this case the deadline will be that of the first report. However, the description of the implementation and experimental results should be well developed and discussed.