Technische Universität München
Lehrstuhl Informatik VIII
Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.
Stephan M. Günther, M.Sc.

Master Course Computer Networks

Homework 4
(submission until December 17 into INBOX located in front of 03.05.052)

Note: Subproblems marked by * can be solved without preceding results.

Traceroute and routing paths

The lecture discussed topics regarding route selection in the Internet and presented *traceroute*, a tool to trace the path packets might take to a given destination. In this problem you will investigate how traceroute works and what its limitations are.

a)* Briefly explain the basic principle behind traceroute.

A node issues probes with increasing TTL values starting at 1. The TTL is decreased at every router. When reaching zero, the router should return an ICMP type "time exceeded" code "TTL exceeded in transit" to the sender. The source address of this ICMP packet is one of the respective router's IP addresses, ideally the address of the interface the original probe entered the router. The process terminates when the final destination is reached.
Note that ICMP may use different types of probes, e.g. ICMP, UDP, and TCP.

b) Traceroute uses by default ICMP or UDP payloads, depending on implementation. Argue why using TCP might be a good / bad idea.

TCP might be a good idea since this it is not obvious that someone tries to traceroute a path. ICMP packets, although bad behavior, might be filtered. However, TCP might also be a bad idea since a firewall near the target will probably filter most destination ports.
So, TCP might be a good alternative to ICMP or UDP probes if you know that the target you are tracerouting is a web server. In such a case it might be a good idea to issue probes destined for TCP 80, which will likely traverse any firewall.

One might expect to see the same IP addresses in reversed order when mutually issuing traceroutes from two endpoints. However, it turns out that the paths seem to be quite different.

c)* Record the route between two systems under your control from *both* sides and sketch the paths. You can use, for instance, your VM and your local computer. Be sure to use your public IP address when recording the path to your local computer.

The traceroute will most likely reveal very asymmetric paths.

d) Have a close look at the IP addresses of both routes you just recorded. In particular, look for IP addresses that are close to each other in the address space. Is there any basis to assume that two different IP addresses belong to the same router?

Consider the traceroutes depicted in Table 1. The first column shows a pretty complete traceroute

Hop	131.159.20.11 → 83.133.105.60	83.133.105.60 → 131.159.20.11
1	131.159.20.11	83.133.105.1
2	131.159.252.149	217.71.107.49
3	*	217.71.96.165
4	129.187.0.149	80.81.192.222
5	188.1.37.89	188.1.145.169
6	188.1.144.110	188.1.145.158
7	188.1.144.149	188.1.37.90, 188.1.146.138
8	188.1.145.234, 188.1.145.230	188.1.37.90
9	193.178.185.39	131.159.252.1
10	217.71.96.181	131.159.252.1
11	217.71.96.169	131.159.252.150
12	217.71.107.50	*
13	83.133.105.60	*

Table 1: Traceroute between from 131.159.20.11 to 83.133.105.60 and vice versa.

except for hop 3 which did not answer. The second traceroute, denoting the reverse direction, did not reach the target at all.
If we try to match addresses of both directions we have to keep in mind that the host that initiated the traceroute is not visible in the traceroute. The first hop appearing is that node's default gateway. Thus, address 83.133.105.1 is most likely the default gateway of 83.133.105.60, which is reasonable considering a /24 subnet. This would mean that most probably the addresses 83.133.105.1 and 217.71.107.50 belong to the same node but two different interfaces. Unfortunately, it is very difficult to be sure about this as ingress traffic might enter a network via another node than egress traffic is leaving this network.
Another anomaly is shown for the reverse direction between hop 7 and 8: Obviously, there are two different possible nodes at hop 7. This is not uncommon an may be an indication of some kind of load balancing. However, the unusual thing is that 188.1.37.90 appears at both hop 7 and 8. This would means that there are at least two paths between 83.133.105.1 and 131.159.20.11 that have different lengths.

e)* Assume that you have a suspicion that two IP addresses belong to the same router. Explain a concept to prove your suspicion.
Hint: Think about the identifier field in the IP header which is not chosen at random by most routers. You should also have a look at the paper *IP Alias Resolution Techniques* by Ken Keys (CAIDA).

See the paper stated in the problem.

Routing protocols

In the lecture you learned about different routing protocols. These can be grouped in different ways. For instance, there are *Interior* and *Exterior Gateway Protocols* (IGPs and EGPs). Another classification could be by means of the functioning principle, i. e., *distance vector* and *link-state protocols*. The protocols you should have some idea about are in particular RIP, OSPF, and BGP.

a)* Discuss the differences between RIP and OSPF. To which of the above mentioned classes do they belong?

RIP is a distance vector protocol, OSPF a link state protocol. Distance vector protocols only know the respective next hop and the approximate distance (cost) to a respective target (compare to a traffic sign). Link state protocols have rather complete maps of a network or parts of a network.
Both RIP and OSPF are IGPs, i. e. are used within autonomous systems.

b)* How does BGP differ from usual distance vector protocols?

First, BGP is a path vector protocol which means that routing updates contain the whole path to a destination. This is unlike normal distance vector protocols.
Second, BGP differs in the kind of metrics used: The best route in terms of BGP is not necessarily the shortest one in terms of hops (although this is one aspect considered by BGP) or in terms of a cost factor that is calculated by physical link parameters.
In particular it allows for policy-based routing, i. e., administrators specify which routes are advertised to which peers. This allows to define complex customer-provider relationships between peering autonomous systems. (see lecture slides!)

If not happened yet, make yourself familiar with the concepts behind BGP routing and AS peering.

c)* Explain the term *policy based routing*.

Routes for packets are not (only) selected by means of a packet's destination address. Instead, additional characteristics of the packets (or a flow of packets) are considered, e.g. origin, type/content, size, customer contracts etc. Policies are defined on those criteria by a network administrator and take precedence over destination-based decisions.

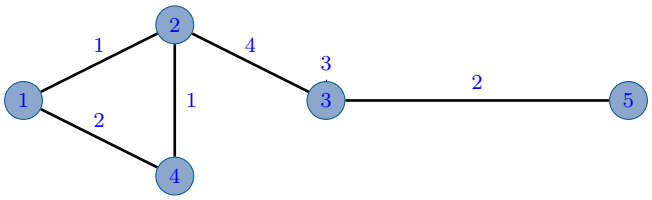

Figure 1: Sample topology

We now consider (generalized) distance vector protocols from a formal point of view. Have a look at the network depicted in Figure 1 which consists of a set of nodes $\mathcal{N}$ and a set of (undirected) edges $\mathcal{E}$. If two nodes $i, j \in \mathcal{N}$ are connected, then $(i, j) \in \mathcal{E}$. Since the edges are undirected, we have that $(i, j) = (j, i)$. The weight w_{ij} of an edge connecting $i, j \in \mathcal{N}$ represents the costs to send a packet, i. e., lower is better. The cost of a path between two non-adjacent nodes is the sum of the edge weights used. Let n denote the total number of nodes in the network. We define the one-hop *distance matrix*

of the network as

$$\boldsymbol{D} = \begin{bmatrix} d_{11} & \dots & d_{1n} \\ \vdots & \dots & \vdots \\ d_{n1} & \dots & d_{nn} \end{bmatrix} \in \mathbb{N}_0^{n \times n}, \quad \text{with } d_{ij} = \begin{cases} w_{ij} & \text{if } \exists (i,j) \in \mathcal{E}, \\ 0 & \text{if } i = j, \\ \infty & \text{otherwise.} \end{cases}$$

Obviously, the element d_{ij} in the i-th row and j-th column of $\boldsymbol{D}$ denotes the distance between two neighboring nodes over exactly one hop and becomes ∞ if j is not reachable within a single hop.

d)* Cast $\boldsymbol{D}$ for the network depicted in Figure 1.

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & \infty & 2 & \infty \\ 1 & 0 & 4 & 1 & \infty \\ \infty & 4 & 0 & \infty & 2 \\ 2 & 1 & \infty & 0 & \infty \\ \infty & \infty & 2 & \infty & 0 \end{bmatrix}$$

We define the m-th power of $\boldsymbol{D}$ with respect to the *min-plus product* as

$$\boldsymbol{D}^m = \boldsymbol{D}^{m-1}\boldsymbol{D} \text{ where } d_{ij}^m = \min_{k \in \mathcal{N}} \left\{ d_{ik}^{m-1} + d_{kj} \right\}. \tag{1}$$

Note, that Equation (1) represents the Bellman-Ford equation discussed in the lecture.

e) Calculate $\boldsymbol{D}^m$ for $m \to \infty$. (can be done by hand)

$$\boldsymbol{D}^2 = \begin{bmatrix} 0 & 1 & 5 & 2 & \infty \\ 1 & 0 & 4 & 1 & 6 \\ 5 & 4 & 0 & 5 & 2 \\ 2 & 1 & 5 & 0 & \infty \\ \infty & 6 & 2 & \infty & 0 \end{bmatrix}$$

$$\boldsymbol{D}^3 = \begin{bmatrix} 0 & 1 & 5 & 2 & 7 \\ 1 & 0 & 4 & 1 & 6 \\ 5 & 4 & 0 & 5 & 2 \\ 2 & 1 & 5 & 0 & 7 \\ 7 & 6 & 2 & 7 & 0 \end{bmatrix}$$

$$\boldsymbol{D}^4 = \boldsymbol{D}^3$$

f) Give a general proof that, for $m \to \infty$, the element in row i and column j of $\boldsymbol{D}^m$ yields the distance between $i, j \in \mathcal{N}$.

Of course, we assume a network with a finite number of n nodes. Furthermore we assume positive integral edge weights.

Proof by induction over m:

- Hypothesis: The element d_{ij}^m in row i and column j of $\boldsymbol{D}^m$ denotes the cost of a shortest path of length at most m between node i and j.

- Basis: The distance matrix $\boldsymbol{D}^1$ contains the shortest distance between any two nodes over exactly one hop (length 1). Consequently, d_{ij} is a shortest path of length at most 1 between i and j.

- Induction step: Consider the element d_{ij}^m denoting the costs of some shortest path p between two nodes i and j over at most m hops. The costs of some other shortest path p' of length at most $m+1$ is found by considering the costs from node i to all other nodes reachable via m hops (which is given by the i-th row of $\boldsymbol{D}^m$) and by the costs to j from each of those nodes over a single hop (which is given by the j-th column of $\boldsymbol{D}$). This yields a total of n potential costs for a new shortest path. Taking the pairwise minimum of the old and new cost as done in Equation (1) discards more expensive (longer) paths and yields the (not necessarily) unique cost of a shortest path of length at most m.

  As the network is assumed to be finite, the longest loop-free path is of length $n-1$ (n denote the number of nodes in the network). Thus, m is also limited by n which closes the proof.