Solution 6

Discussion: 30.1.2008

Solution 6.1: Mobile IP

a.) *What are the requirements for an IP variant supporting mobility? Does mobile IP fulfil them all?*

Transparency is not kept completely: the delay is higher than in fixed networks, the quality of the connection is lower. For best effort traffic, maybe the effects are that small that not much is seen of it, nevertheless you cannot completely hide the influence of mobility. Scalability also can become a large problem is many nodes move around and lot of control messages have to be sent. (Solution: micro mobility.) Security also is a problem – topologically incorrect addresses are not tolerated by firewalls, and if using reverse tunnelling, there is the risk of tunnel hijacking. Furthermore, if path optimization is used, one can find out the movement patterns of certain users.

b.) *Give the steps required for a handover from one foreign agent to another foreign agent including layer 2 and layer 3.*

Even if you can do a handover with mobile IP, each handover starts on layer 2. Thus, first the usual procedure of scanning the medium, detecting other base stations, deciding on one of them, (make reservations, if allowed by the network,) reroute data inside the infrastructure network, release resources with the old base station.

As soon as the registration with the new base station is done, we have to listen for agent advertisements to get a new COA from a new FA, start authentication, and inform the HA about the new address. In the mean time, lot of data are sent to the old base station; here, the FA would have to buffer them to avoid a data loss. The new FA would inform the old one about the new device, the data transferred during handover is redirected. But – mobile IP in pure version is not able to keep the data transmitted during a handover. This usually is left over to layer 4.

Solution 6.2: Routing in ad-hoc networks

a.) *What are the benefits of location information for routing in ad-hoc networks, and which problems do arise?*

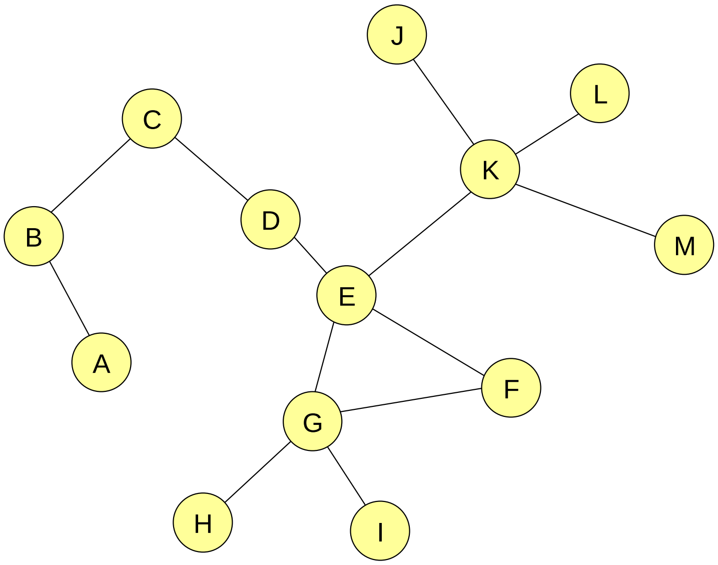
Some routing algorithms use positioning information about destination hosts to decide on a forwarding direction for the packets – simply pass a packet to the neighbour who sits in the right direction. This avoids the use of control information necessary in proactive routing, and also the delay in reactive routing.

But: on one hand, privacy is los: you can construct movement patterns for certain devices. And, maybe worse: nobody guarantees that there is no obstacle in the destination direction so that you have to take a “wrong” path in the beginning and only later on change direction. You can end in dead ends.

b.) *Think of ad-hoc networks with fast moving nodes, e.g., cars in a city. What problems arise even for the routing algorithms adapted to ad-hoc networks? How is the situation changed on highways?*

When cars inside a city move fast, the topology changes too fast for all routing algorithms to adapt to. Flooding could be the only solution. On a highway the situation is astonishingly much better, even if speed is higher: you can think about groups of cars with similar speed and same direction. Forwarding is possible along the roads in driving direction.

c.) *Given is an ad-hoc network which has initially the following topology:*



Connected nodes are in transmission range and can forward messages along the edges. For routing, DSR should be used. The caches of all nodes are empty. Now, D wants to send some packets to H. Give the sequence of messages exchanged for finding a path (also the "unnecessary" messages). For each message, describe sender, receiver, and the list of used nodes as written in the header by the routing protocol. Assume that the paths are symmetrically.

What happens if the connection between E and G breaks down?

Written as triples (from node – to node – path):

- *D* starts searching: (*D C* -) (*D E* -)
- *C* and *E* forward: (*C B C*) (*E K E*) (*E F E*) (*E G E*)
- *B*, *F*, *G*, *K*: (*B A CB*) (*K J EK*) (*K L EK*) (*K M EK*) (*F G EF*) (*G H EG*) (*G I EG*)
- No more hosts forward – no further neighbors exist fort he receivers of the updates in the previous round.

H sends a route reply: (*H D EG*)

Breakdown of the connection between *E* and *G*: *E* generates a route error, thus *D* starts all over with flooding a route request. It is not recognized that all information to know a path over *F* was exchanged before.

Exercise 6.3: TCP in wireless networks

a.) *Can the problems using TCP in wireless networks be solved by replacing TCP with UDP? Where could this be useful and why is it quite often dangerous for network stability?*

Using UDP, in fact a better throughput can be achieved. But, it only works for a few users doing so. If a higher number of users would transmit data over UDP, the missing congestion control would lead to a high packet loss rate. Furthermore, UDP does not provide reliable data transfer. Thus: for only a small number of users, using UDP brings advantages, if the application layer takes over control functionality for reliable data transfer.

b.) *Assume a fixed Internet connection with a round trip time of 20 ms and an error rate of 10^{-10}. Calculate the upper bound on TCP's bandwidth for a maximum segment size of 1000 byte. Use the relation $BW \le \frac{0.93 * MSS}{RTT * \sqrt{p}}$ for this calculation (BW = bandwidth, RTT = round trip time, MSS = maximum segment size, p = loss probability).*

Now two different wireless access networks are added. A WLAN with 2 ms additional one-way delay and an error rate of 10^{-3}, and a GPRS network with an additional RTT of 2 s and an error rate of 10^{-7}. Redo the calculation ignoring the fixed network's error rate. Compare these results with the ones derived from the second formula (use RTO = 5 RTT). Why are some results not realistic?

A bit tricky at the beginning: the given error rates are bit error rates (BER) – assuming independence of bit errors, the packet loss rate *p* can be calculated as $p = 1 - \left((1-BER)^{packetsize}\right)$. Ignoring FEC and ARQ, *p* and in the following the bandwidth can be calculated to:

- Fixed connection: BER = 10^{-10}, MSS = 1000 Byte = 8000 Bit

  $\rightarrow p = 1 - \left(\left(1-10^{-10}\right)^{8000}\right) \approx 8 \cdot 10^{-7}$

  Using a RTT of 20 ms, we have: $BW \le \frac{0.93 \cdot MSS}{RTT \cdot \sqrt{p}} = \frac{0.93 \cdot 8000}{0.02 \cdot \sqrt{8 \cdot 10^{-7}}}\,\text{bit/s} \approx 416\ \text{MBit/s}$

- WLAN: The same calculation with a BER = 10^{-3} and additional delay of 2 ms gives a packet loss rate of *p* = 0.99966 and a maximum bandwidth of BW = 338 kBit/s. This is a good example why large packets in WLAN cause problems and why FEC and ARQ are needed.

- GPRS: Using additional delay of 2 s and a BER of 10^{-7}, the maximum bandwidth is reduced to BW = 130 kBit/s! This is no problem for the usual GPRS data rate but shows the negative influence of the high delay.

In practice, the performance of wireless networks heavily depends on the error correction capabilities of the lower layers. When FEC and ARQ on layer 2 are efficient, the higher BER is hidden from TCP. But, the bandwidth at the same time is reduced. The slow start mechanism in TCP would have to consider large RTTs – see GPRS. For GPRS you can see, that only increasing the data rate in the network not necessarily gives higher user data rates…