Peer-to-peer networks – (due till March 25, 2009)

Exercise 4.1: Random Linear Fountain Codes (part 2/2)

Extend the application from the last exercise:

- It should be possible to define a (short) message.
- This message should be XORed with random vectors generating a sequence of packets.
- After a sufficient number of packets have been generated, the original message should be reconstructed.

Solution:

(see sample solution on our homepage)

Peer-to-peer networks

Exercise 4.2: Gnutella

The simplest kind of flooding can generate a large number of packets.

1) In which way does the Gnutella protocol try to mitigate this problem?

a) What is the purpose of the DescriptorID
b) How would you compose/generate it and why?

Solution:

a) Each node stores an identifier for packets it has seen before. After a new packet has been sent to all neighbors (except to the one the packet came from), the packet's DescriptorID is stored. This way, all duplicates can be identified as such an need not be sent once more.

b) The DescriptorID should be unique for every message in the network. Since it is specified as a 16-byte value, a random number of this size will not likely occur twice. In addition, the sender's IP address could be stored.

Peer-to-peer networks

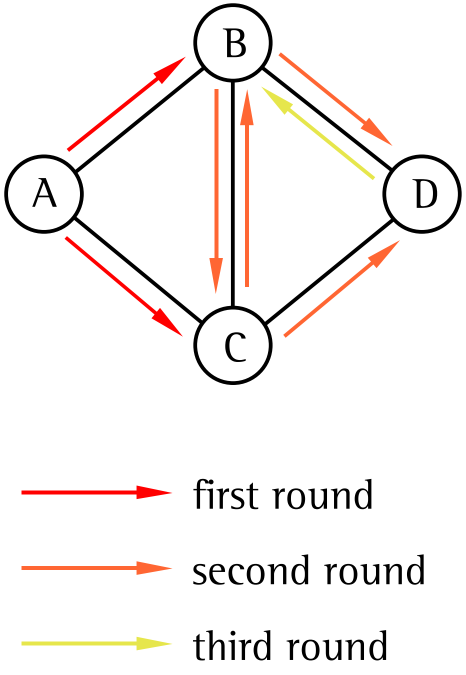
Exercise 4.2: Gnutella

2) Identify situations in which unnecessary packets still occur.

Solution:

(To understand this solution, keep in mind that the nodes are only connected directly in the overlay network. The actual Internet connections below can both buffer and delay a packet in a random fashion).

In the first round, A sends to B and C. The second round generates the most packets even though only a single packet to D would be necessary. B may not know that C got the packet so far, so it sends a copy. The same is true for C who sends a packet to B (we might be able to save one of these packets if either B or C gets the packet from A significantly faster). In the third round, even D will likely send out an unnecessary packet to one of its slower neighbors.



Peer-to-peer networks

Exercise 4.2: Gnutella

The simplest kind of flooding can generate a large number of packets.

3) The protocol uses two values named „TTL“ and „Hops“.

a) What is the purpose of the „TTL“?
b) For which reason is „Hops“ stored in addition? Do both values carry redundant information? Is the additional storage of „Hops“ beneficial for the protocol?

Solution:

a) The TTL (Time to Live) is set to the initial live-time of a packet and is usually measured in hops (= nodes to be visited). Upon the arrival at a new node, the TTL is decremented. An incoming packet with a TTL of zero is still processed by the node but not forwarded to a neighbor. The purpose is to avoid endlessly circulating packets in a network.

b) The specification of Gnutella says that the sum TTL+Hops must not exceed the maximum TTL of a packet. But if a nodes wanted to cheat, it could manipulate both values. One of the few advantages of the Hops-counter could be that a node knows how long the packet traveled so far. However, it is not obvious how Gnutella could make use of this information.