Peer-to-Peer Systems
Exercise 1
Published: 23.10.2014
Presentation date: 30.10.2014

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Peer-to-Peer Systems – Exercise
Winter Term 2014/2015

General Remarks

Welcome to the exercise for the lecture Peer-to-Peer Systems.

Please follow the general remarks regarding the organization of the exercise.

- The lecture's website is to be found here:
`http://tsn.hhu.de/teaching/lectures/2014ws/p2p.html`
- For further inquiries, please contact the lecturer under the following email address: `graffi@cs.uni-duesseldorf.de`

Problem 1.1 - Gnutella Messages

Please consider the Gnutella topology as depicted in the Figure 1. All peers have a local view on the topology which they maintain. The peers further maintain a list of messages they forwarded recently in order to avoid unnecessary retransmissions. In this exercise we assume a round-based communication, thus messages are received in parallel, sent in parallel and received in the next round.

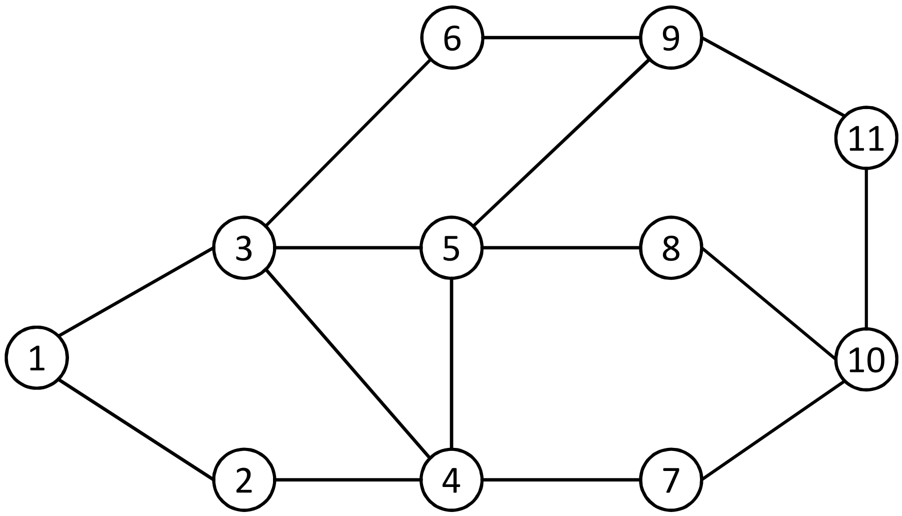

Abbildung 1: Gnutella Topology

a) Query Forwarding in Gnutella

Peer 1 attempts to search for the keyword „P2P“ using a TTL of 2. The Hop count is increased by one each time the query is received. The queries are propagated in the network until the Hop > TTL runs out. List all messages transferred in the network according to the message template in following table (Table 1). The hop count denotes the value at the receiver side.

Round	**Sender**	**Receiver**	**Hop**
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

Tabelle 1: Message Format

Solution:
The messages are transferred according to Figure 2. The figure shows the remaining hops. The numbers at the edges mark the rounds in which messages are sent along these edges. This results in the following transmission table (Table 2).

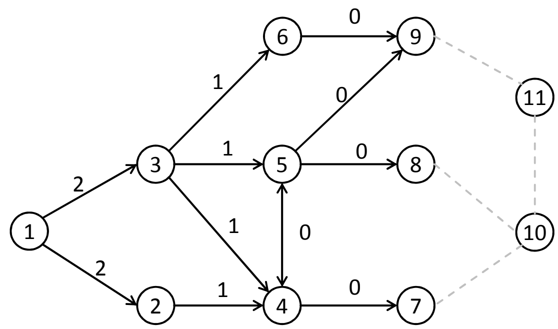

Abbildung 2: Solution Figure: Query Forwarding in Gnutella

Round	**Sender**	**Receiver**	**Hop**
1	1	2	1
1	1	3	1
2	2	4	2
2	3	4	2
2	3	5	2
2	3	6	2
3	4	7	3
3	4	5	3
3	5	4	3
3	5	8	3
3	5	9	3
3	6	9	3

Tabelle 2: Solution Table: Query Forwarding in Gnutella

b) Routing of QUERYHIT Messages in Gnutella

Now assume that the nodes 2, 3, 8 and 11 have files matching the keyword „P2P“. List the QUERYHIT messages that occur after the query in a) has been stated. Use the same message type (except the Hop) and use also the same rounds.

Solution:
The QUERYHIT messages are routed back on the paths that the corresponding Queries took.

Round	Sender	Receiver
2	2	1
2	3	1
4	8	5
5	5	3
6	3	1

Tabelle 3: Solution Table: Routing of QUERYHIT Messages in Gnutella

c) No State Information in Gnutella

Assume that in order to save storage on peers, no state information for query forwarding is hold. Describe at one example in Figure 1 the problem that arises.

Solution:
Queries are then forwarded to peers, from which they were received. Thus a query is always reinitiated with Hop+2 in every node, see for that Figure 3. The figure shows the remaining hops. Messages could run in circles and reinitiate further queries. Overall, the query would generate a large amount of traffic. Additionally, QUERYHITs cannot be routed back the way they came from, thus they have to be flooded again, at least the hop count is known.

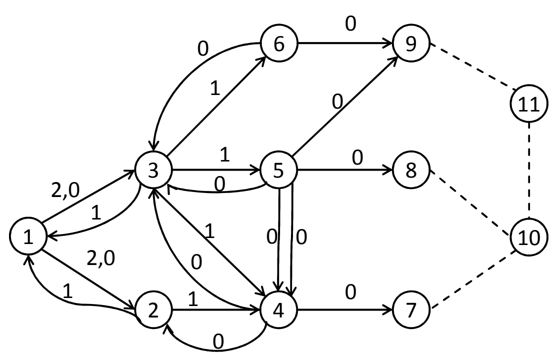

Abbildung 3: Solution Figure: Routing of QUERYHIT Messages in Gnutella

Problem 1.2 - Scalability of Gnutella

In the following task we will have a closer look on the scalability of Gnutella. First we are going to calculate the number of reachable nodes for a single request. Afterwards, we will have a look on the total traffic caused by a single query and its response. Please use the notation as shown in the table below in the following task.

Symbol	Description
n	Number of connections each client has to its neighbors
t	Time-to-live counter
s	Size in bytes of single request
a	Mean percentage of users who typically share content
b	Mean percentage of users who typically have responses to search queries
l	Mean amount of data of a query response
w	total bandwidth of the network in MBps
$R(a, b, l)$	A function representing the Response Factor, a constant value that describes the product of the percentage of users responding and the amount of data generated by each user.
$f(n, x, y)$	A function describing the maximum number of reachable users that are at least x hops away, but no more than y hops away.
$h(n, t, s)$	A function describing the maximum amount of traffic generated by relaying a transmission of s bytes given any n and t. Generation is defined as the formulation and outbound delivery of data
$k(n, t, R)$	A function describing the maximum amount of traffic generated in response to a search query, including relayed data, given any n and t and Response Factor R.

a) Formula for Number of Users

Given a node A with n outgoing links. Please derive a formula $f(n, x, y)$ for the maximum number of reachable users that are at least x hops away, but no more than y hops away from node A. Assume that the number of links each user maintains to its neighbors is n and that we do not discover duplicate nodes on each of the n outgoing links in case of a query request.

Solution:
The first node reaches n nodes, all further nodes in the flood forward the query $n-1$, as they do not send the query back to the node from which they received it. For each outgoing link a node has the number of reachable nodes can be calculated similar to the number of nodes of a complete n-ary tree with height h and node degree n as shown in Equation 1.

$$\sum_{i=0}^{h} n^i \tag{1}$$

Similar to n the n-ary tree, the calculation of the reachable nodes can be done. Given the number of outgoing links with n that each node has, the topology of the Gnutella network can be regarded as an initial node reaching n neighboring nodes which each become the root of a (n-1)-ary tree. For the (n-1)-ary trees after the first hop, we have a TTL $t-1$. We get the number of nodes on a (n-1)-ary tree as:

$$\sum_{t=x}^{y} (n-1)^{(t-1)} \tag{2}$$

Due to the fact that each node has n outgoing links, we end up with the following formula for the number of reachable nodes, which are at least x hops away, but no more than y hops away.

$$f(n,x,y) = n * \sum_{t=x}^{y} (n-1)^{(t-1)} \tag{3}$$

b) Calculation of Number of Users

Please calculate the number of reachable users for $t=7$ and $n=4$ as well as for $t=7$ and $n=8$.

Solution:

$$f(n,x,y) = f(4,1,7) = 4 * \sum_{t=1}^{7} (4-1)^{(t-1)} = 4,372 \text{ users} \tag{4}$$

$$f(n,x,y) = f(8,1,7) = 8 * \sum_{t=1}^{7} (8-1)^{(t-1)} = 1,098,056 \text{ users} \tag{5}$$

c) Formula for Query Traffic for a Single Query Request

Please derive a formula $h(n,t,s)$ for the total traffic generated by a single query request assuming a query size of s, a TTL counter of t, and an out-degree of n. Please take into account all outbound and inbound transmissions. Hint: A request sent from a Node A to a Node B causes outbound traffic at Node A as well as inbound traffic at Node B.

Solution:
The total bandwidth consumption can be calculated as follows: On the n outgoing links of the querying node a traffic of $n*s$ is caused. On all other links which are a least one hop and at most t hops away that amount of traffic is $f(n,1,t-1)*(n-1)*s$. As outgoing traffic is simultaneously inbound traffic at the receiving node, we have to multiply the total amount of traffic by 2. Finally, we get the formula as shown in Equation 6.

$$h(n,t,s) = 2*(n*s + f(n,1,t-1)*(n-1)*s) \tag{6}$$

d) Query Traffic for a Single Query Request

Please calculate the total amount of traffic caused for a singe query request given the size of $s = 100$ bytes given $t = 7$ and $n = 4$ as well as for $t = 7$ and $n = 8$.

Solution:
For $s = 100$, $t = 7$, and $n = 4$:

$$h(n,t,s) = h(4,7,100) = 2*(4*100+f(4,1,6)*(4-1)*100) = 2*(400+1,456*300) = 874,400 \text{ bytes} \tag{7}$$

For $s = 100$, $t = 7$, and $n = 8$:

$$h(n,t,s) = h(8,7,100) = 2*(8*100+f(8,1,6)*(8-1)*100) = 2*(800+156,864*700) = 219,611,200 \text{ bytes} \tag{8}$$

e) Total Traffic for a Single Query-Response-Cycle

Given the mean percentage of users who typically share content $a = 0.5$, mean percentage of users who typically have responses to search queries $b = 0.3$, and the mean size of a query response $l = 300$ bytes, the response factor R can be calculated as follows:

$$R(a,b,l) = a*b*l \tag{9}$$

Then the total amount of traffic generated by the answering nodes is given as:

$$k(n,t,R(a,b,l)) = R(a,b,l)*\sum_{i=1}^{t} f(n,i,i)*i \tag{10}$$

Please calculate the total amount of traffic caused for a complete query-response-cycle given the size of $s = 100$ bytes given $t = 7$ and $n = 4$ as well as for $t = 7$ and $n = 8$.

Solution:
First we calculate the response factor R as $R(0.5,0.3,300) = 0.5*0.3*300 = 45$ bytes.
The amount of traffic generated for the query response for $s = 100$, $t = 7$, and $n = 4$ is given as:

$$k(n,t,R) = k(4,7,45) = 45*\sum_{i=1}^{7} f(4,i,i)*i = 45*(4*3^0*1+4*3^1*2+...+4*3^6*7) = 1,279,440 \text{ bytes} \tag{11}$$

Together with the traffic generated for flooding the query through the network as calculated in Task D, we get a total amount of traffic of: $1,279,440+874,400 = 2,153,840$ bytes or about 2 MB..

The amount of traffic generated for the query response for $s = 100$, $t = 7$, and $n = 8$ is given as:

$$k(n,t,R) = k(8,7,45) = 45*\sum_{i=1}^{7} f(8,i,i)*i = 45*(8*7^0*1+8*7^1*2+...+8*7^6*7) = 337,652,640 \text{ bytes} \quad (12)$$

Together with the traffic generated for flooding the query through the network as calculated in Task D, we get a total amount of traffic of: $337,652,640 + 219,611,200 = 557,263,840$ bytes or about 557 MB.

f) Total Bandwidth Consumption

Finally, let us assume that we have 10 query requests per second in a Gnutella network. How much total bandwidth w is required in the whole network to solve all the queries assuming the same values for t, n, s, and R as in Task E?

Solution:
Given the number of queries per second as 10 qps, we calculate the total amount of traffic that needs to be transfered as:
Given the values $s = 100$ bytes, $t = 7$ hops, $R = 45$ bytes, and $n = 4$ links:

$$w = \frac{2,153,840 \text{ bytes} * 10 \text{ qps}}{10^6} = 21,538,400 \approx 21 MBps \quad (13)$$

Given the values $s = 100$ bytes, $t = 7$ hops, $R = 45$ bytes, and $n = 8$ links:

$$w = \frac{557,263,840 \text{ bytes} * 10 \text{ qps}}{10^9} = 5,572,638,400 \approx 5,572 MBps \quad (14)$$

g) Conclusions

Which conclusions can you draw from the results above? How can the scalability of the network be improved?

Solution:
The results clearly show, that the flooding based approach does not scale. To overcome this problem a hybrid approach has been developed.