CHAPTER 29

PRACTICE SET

Questions

Q29-1.

a. In a centralized network, the directory system uses the client-server paradigm, but storing and downloading of the files are done using peer-to-peer paradigm.

b. A decentralized network does not depend on a central directory system.

Q29-3. A decentralized network generates less traffic and is less vulnerable to attacks, but is more difficult to build.

Q29-5. The number of points is 2^{10} or 1024.

Q29-7. The two strategies are called direct and indirect.

a. In the direct strategy, the object is stored in a peer whose ID is somehow *closest* to the object ID.

b. In the indirect strategy, the object is stored in a peer that owns the object, but a reference is stored in a peer whose ID is somehow *closest* to the object ID.

Q29-9.

a. In the direct method, the file is stored in node 20.

b. In the indirect method, the file is stored in node 4, but a reference is given in node 20.

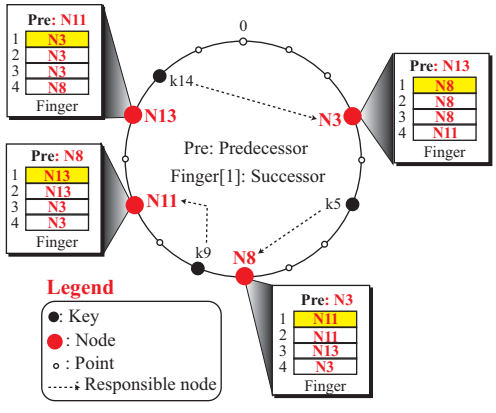
Q29-11. For this network, $m = \log_2 1024 = 10$. This means each identifier has 10 bits. The height of the tree is also 10. The number of leaves is 1024. Each node has 10 subtrees. Each routing table also has 10 rows.

Problem

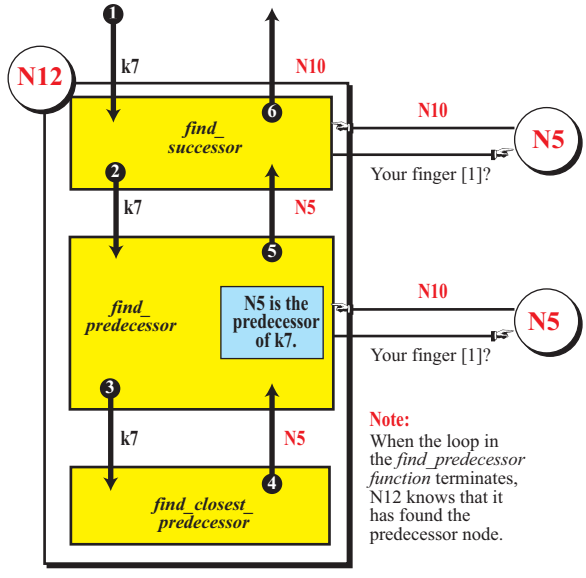
P29-1. The number of rows in the finger table is $\log_2 16 = 4$. The rows in the table for node N6 show the successors of target keys $(6 + 2^{m-1})$ for $m = 1$ to 4 as shown below:

m	Target key	Successor
1	6 + 1	N8
2	6 + 2	N8
3	6 + 4	N12
4	6 + 8	N3

P29-3. The following shows the ring.



P29-5. Let us go through the event as we did in Example 2.16 in the text. The following shows how N12 with some help from N5 goes through the sequence of events to find the successor of k7:



a. Event 1: Since N12 is not the responsible node for k7, it calls its *find_successor* function.

b. Event 2: N12 calls the *find_predecessor* function.

c. Event 3: Since N12 is not the predecessor node of k7, N12 calls the *find_closest_predecessor* function. This function checks the finger table of N12 from bottom to top. It finds that N5 is the closest predecessor because $5 \in (12, 7)$, N5 is between N12 and k7 when going around the ring in the clockwise direction. This can also be seen if we think of 5 as 37 (in modulo 32 arithmetic) and 7 as 39. Definitely $37 \in (12, 39)$. Note that we need to add 32 to 5 and 7 because we move from 12 to these two points as we pass point 0.

d. Event 4: The *find_closest_predecessor* function returns N5 to the *find_ predecessor* function as the closest predecessor of k7. N12 finds that N5 is actually the predecessor of k7 (by remotely calling the N5.finger[1]) because $7 \in (5, 10]$.

e. Event 5: The *find_predecessor* function returns N5 to the *find_successor* function as the predecessor of k7.

f. Event 6: Node N12 finds that the successor node of k7 is N10 by remotely asking N5 to send its finger [1].

P29-7. We have $m = \log_2 16 = 4$. The number of digits in each identifier is $n = m/b$ or 2. The identifiers are in base $2^2 = 4$ as follows: 00 … 03, 10 … 13, 20 … 23, 30 … 33.

P29-9. We have $m = \log_2 16 = 4$ and $n = m/b = 4/2 = 2$. The routing table has n rows (here 2) and 2^b columns (here 4). The identifiers are 00 to 33 in base 4. The following shows a possible routing table for node N21. In row 0, there should be no common digits with the node identifier; the first digit defines the column number. In row 1, the first digit should be 2 (one common digit with node N21); the second digit defines the column. Note that some cells can have no entry.

	0	1	2	3
0	00	10		31
1	20		22	23

P29-11. We use the algorithm in the text:

a. Key k24 is not in the range of the leaf set of N02. The routing table should be used. In this case $p = 0$ and $v = 2$, which means we need to look at the routing table entry [0, 2] = N20. The query is passed to node N20. N20 is not responsible for k24; the key k24 is not in the range of leaf set of N20; the routing table is used. In this case, $p = 1$ and $v = 4$. Since table entry [1, 4] does not exist, the query is passed to a node sharing a prefix as long as the current node, but numerically closer to the key, which is N23. The node N23 is responsible for k24.

b. Key k12 is in the range of the leaf set of N20. The closest node in the leaf set to k12 is the node N11. The query is passed to N11, which is responsible for k12.

P29-13. The following takes place:

a. The length of the common prefix between N0 (N0000) and k12 (k1100) is 0. Node N0 sends the query to the node specified in row 0 of its routing table, node N8.

b. The length of the common prefix between N8 (N1000) and k12 (k1100) is 1. Node N8 sends the query to the node specified in row 1 of its routing table, node N15.

c. Node 15 is responsible for k12. It responds to the query.