**Name: Gustavo Hammerschmidt.**

**HW4, five questions, 100 points**

**1.** **(20 points)** Assume a 3600 bytes packet (including 20 bytes header) arrives at a router, and the MTU for outgoing link is 700 bytes (including header). Fragmentation occurs (on the payload). Fill out the table below. Does the next router reconstruct the original packet from its fragments? Why? (2 reasons) (2 pts)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Fragment length (including header) | Flag | Offset |
| Fragment1 | 700 bytes (1 pt) | 1 (1 pt) | 0 (1 pt) |
| Fragment2 | 700 bytes (1 pt) | 1 (1 pt) | 85 (1 pt) |
| Fragment3 | 700 bytes (1 pt) | 1 (1 pt) | 170 (1 pt) |
| Fragment4 | 700 bytes (1 pt) | 1 (1 pt) | 255 (1 pt) |
| Fragment5 | 700 bytes (1 pt) | 1 (1 pt) | 340 (1 pt) |
| Fragment6 | 200 bytes (1 pt) | 0 (1 pt) | 425 (1 pt) |

**The next router does not reconstruct the original packet because it would increase the complexity over the router’s functionality; and because, in a network, the path a packet may take depends on the amount of traffic on the routers, so the router does not reconstruct the original packet because it may not receive all of the fragmented packets: a packet may take another route.**

**2.** **(20 points) IP protocol uses 1’s complement for IP header checksums. Suppose you have the following three 16-bit words: 0101011001100011, 1110001100000110, 0111001010101100. What is the 1’s complement of the sum of these 16-bit words? Show all work step by step in the worksheet given below. Note: first find sum of first two words, then find sum of that and the 3rd word, then find its 1’s complement. Why is it that IP takes the 1’s complement of the sum; that is, why not just use the sum? (2 points) With the 1’s complement scheme, how does the receiver detect errors? (4 points) Is it possible that a 1-bit error (one bit is flipped) will go undetected? If yes, give an example. (4 points) How about a 2-bit error? If yes, give an example. (4 points)**

**0101 0110 0110 0011 (1st word)**

**+ 1110 0011 0000 0110 (2nd word)**

**1 0011 1001 0110 1001**

**0011 1001 0110 1001**

**+1 (if required)**

**0011 1001 0110 1010 (2 points)**

**0011 1001 0110 1010**

**+ 0111 0010 1010 1100 (3rd word)**

**1010 1100 0001 0110**

**+1 (if required)**

**NOT REQUIRED (2 points)**

**1010 1100 0001 0110 (SUM OF WORDS)**

**0101 0011 1110 1001 (1’s complement)**

**(2 points)**

**Why is it that IP takes the 1’s complement of the sum; that is, why not just use the sum? (2 points)**

* **Because the value of the sum of the words within the header could be altered on the transmission; but, if the router takes the 1’s complement of the sum and sums it with the sum, it will obviously get a result of only ones, due to the mathematical property that every binary number plus its complement will result on a binary number of only ones. So, if the router took the 1’s complement on the sum, there would be no error detection, proving that the IP needs to add the 1’s complement of the sum to the header before forwarding it to the first hop router, providing means to an error detection, once it’s conceived that there is no error in the data before the transmission of itself.**

**With the 1’s complement scheme, how does the receiver detect errors? (4 points)**

* **It sums the checksum field on the header with the sum of the other words in the header, if it detects a zero in the sum result, it knows that an error has occurred during the transmission.**

**Is it possible that a 1-bit error (one bit is flipped) will go undetected? If yes, give an example. (4 points)**

* **No, because, for an odd number of bits, there will be at least one bit that will not have a bit-correspondent location on another word, so there will be an error in the result sum of the checksum field in the header and the sum of the words. So, no, it is not possible for a 1-bit error to go undetected. The example given below just confirms my answer.**

**Example(for an odd number, suppose 3, green indicates an error):**

**0101 1100 1001 1000**

**+ 1010 0011 0110 0110**

**----------------------------**

**1111 1111 1111 1110**

**Error was still detected because, with an odd number of errors, there will be at least one bit of error with no matching location.**

**How about a 2-bit error? If yes, give an example. (4 points)**

**- Yes, it is possible for a 2-bit number to go undetected, as it is for any even number if the location of every two errors match.**

**Example(for an even number, suppose 2, green indicates an error):**

**0101 1100 1001 1001**

**+ 1010 0011 0110 0110**

**------------------------------**

**1111 1111 1111 1111**

**An error passed without detection.**

**3.** **(20 points)** Assume you are in a private network connected to the Internet via a router. Your node’s IP address is 192.168.45.121/28.

Net ID in decimal? **192.168.45.112**

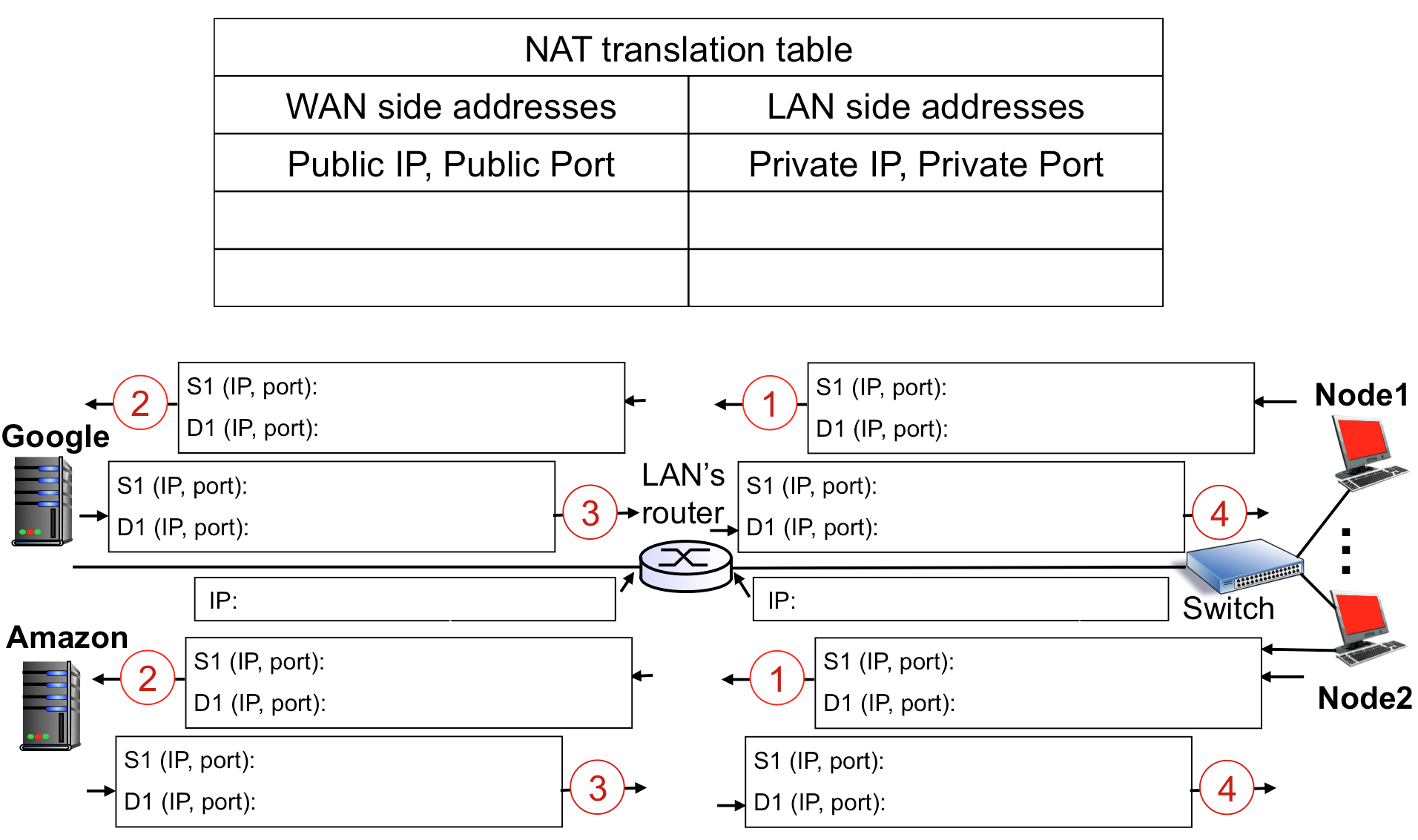
Range of valid IPs? **192.168.45.113** to **192.168.45.126**

Which IP address is used for the router interface? **192.168.45.113**

The router has another interface connected to the Internet. Its public IP address is 144.37.12.45.

Node1 (192.168.45.121 port 3003) sends an HTTP request to Google server whose IP address is 54.23.130.30 (port 80), and Google server replies. Node2 (192.168.45.125 port 3003) sends an SMTP request to Amazon mail server whose IP address is 145.23.170.92 (port 25), and Amazon mail server replies. Create port numbers at router starting from 6200.

Fill in the blanks in the picture below. Enter the IP addresses and ports for packets’ sources and destinations at each step and also in NAT table. Enter the IP addresses asked in the picture (22 blanks).



145.23.170.92, 25

145.23.170.92, 25

145.23.170.92, 25

145.23.170.92, 25

192.168.45.125, 3003

192.168.45.125, 3003

144.37.12.45, 6201

144.37.12.45, 6201

144.37.12.45, 6201

192.168.45.125, 3003

**192.168.45.113**

144.37.12.45

192.168.45.121, 3003

144.37.12.45, 6200

144.37.12.45, 6200

144.37.12.45, 6200

54.23.130.30, 80

54.23.130.30, 80

54.23.130.30, 80

192.168.45.121, 3003

54.23.130.30, 80

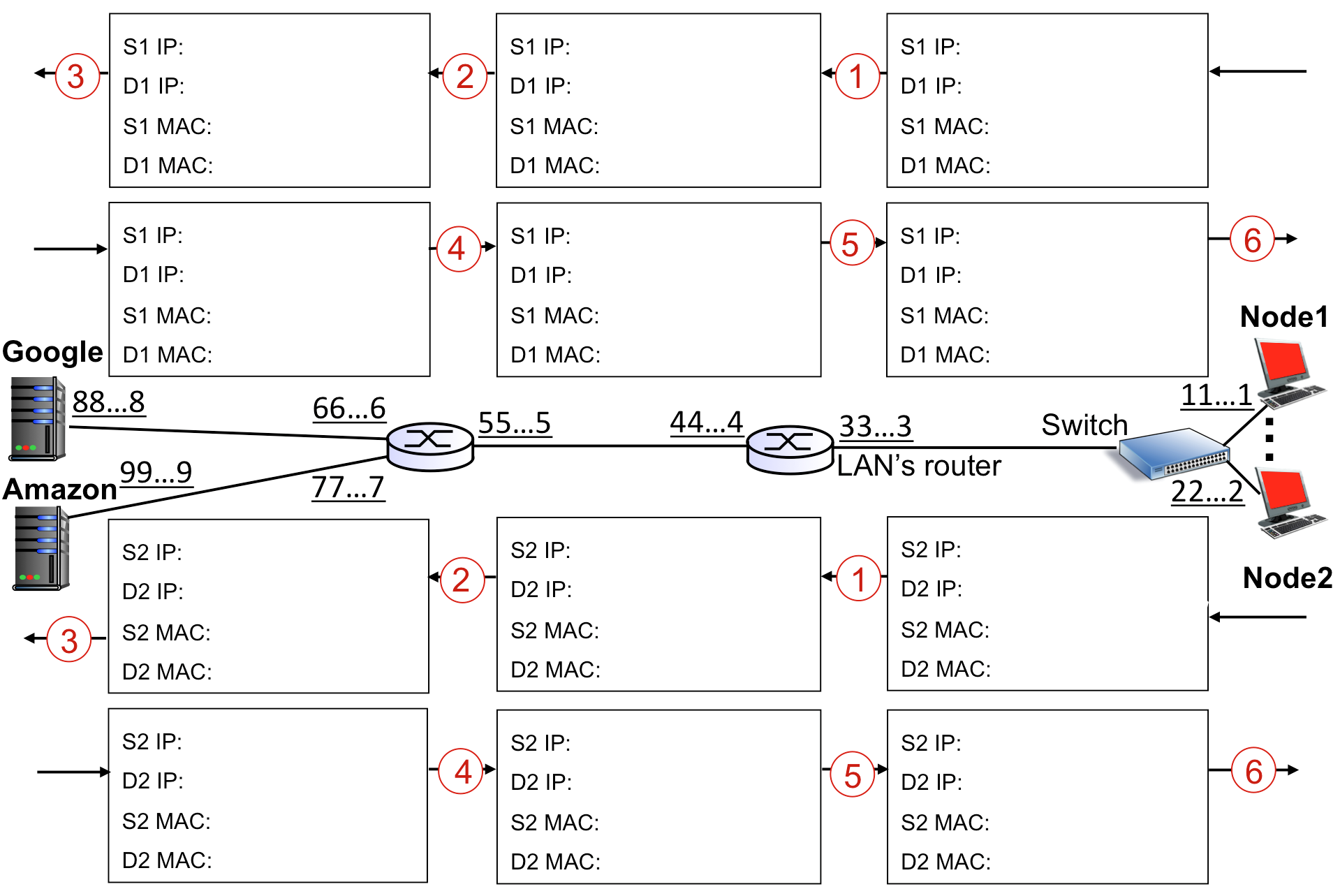
192.168.45.121, 3003

**4.** **(20 points)** Using the configuration of last question, now assume packets pass two routers to reach destination. The picture below contains your private network, the servers, and the routers in between. Node1 sends a message to Google server. (Pay attention: who is the source, who is the destination?). A frame is created at node1 to send the packet to the router. Then the frame is processed and updated in each router to forward the packet to the next hop. Node2 also sends a message to Amazon mail server. Fill in the blanks in the picture. Specify the source and destination **IP addresses** and **MAC addresses** for frames and their enclosed IP packets. (48 items need to be filled). Assume the ARP tables contain all the required mappings.

The MAC addresses in this network are given in underlined numbers at each interface. For example 11…1 (meaning 11-11-11-11-11-11) is the MAC address of node1. You can use the same notation in your answer.

IP addresses of left router’s interfaces are given in the table below. Note that you may or may not need these IP addresses. The other IP addresses (like Google) are the same as the ones given in the last question.

|  |  |
| --- | --- |
| MAC address | IP address |
| 55…5 | 56.84.62.79 |
| 66…6 | 54.23.130.1 |
| 77…7 | 145.23.170.1 |

****

144.37.12.45

144.37.12.45

144.37.12.45

144.37.12.45

144.37.12.45

144.37.12.45

144.37.12.45

144.37.12.45

99…9

99…9

77…7

77…7

22…2

22…2

33…3

44…4

55…5

55…5

44…4

33…3

145.23.170.92

145.23.170.92

145.23.170.92

145.23.170.92

145.23.170.92

145.23.170.92

192.168.45.125

192.168.45.125

33…3

11…1

44…4

55…5

66…6

88…8

88…8

66…6

55…5

44…4

33…3

11…1

54.23.130.30

54.23.130.30

54.23.130.30

54.23.130.30

54.23.130.30

54.23.130.30

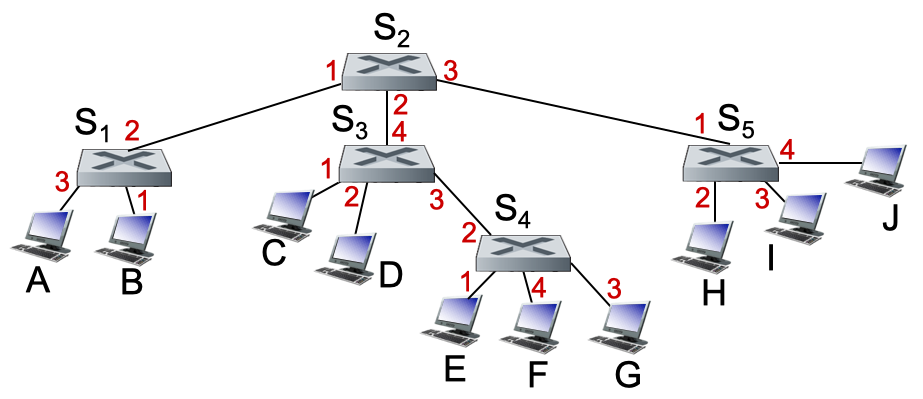
192.168.45.121

192.168.45.121

**5.** **(20 points)** In the picture below assume node B sends a message to node E, and then node E replies to B. Assume all ARP and Switch tables are empty at the beginning. Fill out the following table to show what will be stored at each switch at each step.

Note: at each step, a switch may learn and store one or no new entry. Some rows of the table may remain blank at the end. Figure out how many steps are required. If two broadcasts occur at the same time, you may name the steps as step-a and step-b, for instance step 4-a and 4-b.

Note: at each step, a switch may learn and store one or no new entry in its table. You may not need all the given rows in the below table.



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Sender 🡪 Receivers | | S1 switch table | | S2 switch table | | S3 switch table | | S4 switch table | | S5 switch table | |
|  | Sender | Receivers | MAC | Interface | MAC | Interface | MAC | Interface | MAC | Interface | MAC | Interface |
| 1 | B | S1 | B | 1 |  |  |  |  |  |  |  |  |
| 2 | S1 | A, S2 |  |  | B | 1 |  |  |  |  |  |  |
| 3 | S2 | S3, S5 |  |  |  |  | B | 4 |  |  | B | 1 |
| 4-a | S3 | C,D, S4 |  |  |  |  |  |  | B | 2 |  |  |
| 4-b | S5 | H, I, J |  |  |  |  |  |  |  |  |  |  |
| 5 | S4 | E, F, G |  |  |  |  |  |  |  |  |  |  |
| 6 | E | S4 |  |  |  |  |  |  | E | 1 |  |  |
| 7 | S4 | S3 |  |  |  |  | E | 3 |  |  |  |  |
| 8 | S3 | S2 |  |  | E | 2 |  |  |  |  |  |  |
| 9 | S2 | S1 | E | 2 |  |  |  |  |  |  |  |  |
| 10 | S1 | B |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |  |