

序次	实验内容与进度安排																																																					
1	<p>RIP 协议的模拟与编程(python) (预计 2 次上机)</p> <p>要求:</p> <p>(1) 使用 python 列表(list)/集合 set/字典 dictionary/元组 tuple 等类型来表达路由表</p> <p>(2) 实现 old 路由表和 RIP message 的计算功能, 从而形成新的路由表</p> <p>(3) old 路由表和 RIP message 的表项的输入, 采用程序输入的方式, 避免硬编码输入的方式。</p> <div><div><p>RIP message from C</p><table><tr><td>Net2</td><td>4</td></tr><tr><td>Net3</td><td>8</td></tr><tr><td>Net6</td><td>4</td></tr><tr><td>Net8</td><td>3</td></tr><tr><td>Net9</td><td>5</td></tr></table></div><div><p>RIP message from C after increment</p><table><tr><td>Net2</td><td>5</td></tr><tr><td>Net3</td><td>9</td></tr><tr><td>Net6</td><td>5</td></tr><tr><td>Net8</td><td>4</td></tr><tr><td>Net9</td><td>6</td></tr></table></div><div><p>Old routing table</p><table><tr><td>Net1</td><td>7</td><td>A</td></tr><tr><td>Net2</td><td>2</td><td>C</td></tr><tr><td>Net6</td><td>8</td><td>F</td></tr><tr><td>Net8</td><td>4</td><td>E</td></tr><tr><td>Net9</td><td>4</td><td>F</td></tr></table></div><div><p>Updating algorithm</p></div><div><p>New routing table</p><table><tr><td>Net1</td><td>7</td><td>A</td></tr><tr><td>Net2</td><td>5</td><td>C</td></tr><tr><td>Net3</td><td>9</td><td>C</td></tr><tr><td>Net6</td><td>5</td><td>C</td></tr><tr><td>Net8</td><td>4</td><td>E</td></tr><tr><td>Net9</td><td>4</td><td>F</td></tr></table></div><div><p>Net1: No news, do not change</p><p>Net2: Same next hop, replace</p><p>Net3: A new router, add</p><p>Net6: Different next hop, new hop count smaller, replace</p><p>Net8: Different next hop, new hop count the same, do not change</p><p>Net9: Different next hop, new hop count larger, do not change</p></div></div>	Net2	4	Net3	8	Net6	4	Net8	3	Net9	5	Net2	5	Net3	9	Net6	5	Net8	4	Net9	6	Net1	7	A	Net2	2	C	Net6	8	F	Net8	4	E	Net9	4	F	Net1	7	A	Net2	5	C	Net3	9	C	Net6	5	C	Net8	4	E	Net9	4	F
Net2	4																																																					
Net3	8																																																					
Net6	4																																																					
Net8	3																																																					
Net9	5																																																					
Net2	5																																																					
Net3	9																																																					
Net6	5																																																					
Net8	4																																																					
Net9	6																																																					
Net1	7	A																																																				
Net2	2	C																																																				
Net6	8	F																																																				
Net8	4	E																																																				
Net9	4	F																																																				
Net1	7	A																																																				
Net2	5	C																																																				
Net3	9	C																																																				
Net6	5	C																																																				
Net8	4	E																																																				
Net9	4	F																																																				
2	<p>OSPF 协议的模拟与编程(python) (预计 2 次上机)</p> <p>要求:</p> <p>(1) 使用 python 列表(list)/集合 set/字典 dictionary/元组 tuple 等类型来表达链路状态分组;</p> <p>(2) 实现某个节点在收到所有的链路状态分组, 使用 Dijkstra 算法, 从而形成该节点的路由表;</p> <p>(3) 链路状态分组的表项的输入, 采用键盘输入的方式, 避免硬编码输入的方式。</p> <div><div><p>(a)</p></div><div><table><tr><th colspan="2">Link</th><th colspan="2">State</th><th colspan="2">Packets</th></tr><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr><tr><td>Seq.</td><td>Seq.</td><td>Seq.</td><td>Seq.</td><td>Seq.</td><td>Seq.</td></tr><tr><td>Age</td><td>Age</td><td>Age</td><td>Age</td><td>Age</td><td>Age</td></tr><tr><td>B 4</td><td>A 4</td><td>B 2</td><td>C 3</td><td>A 5</td><td>B 6</td></tr><tr><td>E 5</td><td>C 2</td><td>D 3</td><td>F 7</td><td>C 1</td><td>D 7</td></tr><tr><td></td><td>F 6</td><td>E 1</td><td></td><td>F 8</td><td>E 8</td></tr></table><p>(b)</p></div></div>	Link		State		Packets		A	B	C	D	E	F	Seq.	Seq.	Seq.	Seq.	Seq.	Seq.	Age	Age	Age	Age	Age	Age	B 4	A 4	B 2	C 3	A 5	B 6	E 5	C 2	D 3	F 7	C 1	D 7		F 6	E 1		F 8	E 8											
Link		State		Packets																																																		
A	B	C	D	E	F																																																	
Seq.	Seq.	Seq.	Seq.	Seq.	Seq.																																																	
Age	Age	Age	Age	Age	Age																																																	
B 4	A 4	B 2	C 3	A 5	B 6																																																	
E 5	C 2	D 3	F 7	C 1	D 7																																																	
	F 6	E 1		F 8	E 8																																																	

3	<p>UDP 编程 (预计 1 次上机)</p> <p>要求:</p> <p>(1) “UDPClient.java” 转化为 python 编程</p> <p>(2) “UDPServer.java” 转化为 python 编程</p> <p>重点/难点: 利用 Wireshark 捕获程序所发送的分组并分析。</p>
4	<p>TCP 套接字编程(预计 2 次上机)</p> <p>要求:</p> <p>(1) “CompteTCPClient.java” 转化为 python 编程</p> <p>(2) “CompteTCPServer.java” 转化为 python 编程</p> <p>(3) 用 Wireshark 捕获程序所发送的分组</p> <p>重点/难点: 通信双方的套接字 API 的使用和三次握手的编程, 并实现利用 Wireshark 捕获程序所发送的分组。</p>