This paper came up with the joint design of the TCP/IP protocol and Internet architecture is the pioneer of today's Internet. In this article the authors proposed a packet network interworking protocol named TCP, which is extremely important for the interconnection of networks. The network interconnection principle they proposed determines the Internet system. In this article, the authors first introduced the concept of a gateway and described its role and importance, and detailed the components of the protocol, including segment structure, flow control, etc.

A gateway is a "gateway" that connects a network to another network. It is also called an inter-network connector and a protocol converter. In the absence of a router, TCP/IP communication is not possible between the two networks, and to achieve communication between the two networks, it is necessary to pass through the gateway. The gateway is the most complex network interconnection device on the transport layer to implement network interconnection. It is a translator between two systems that use different communication protocols, data formats or languages, and even completely different architectures. It is used for interconnection between two different high-level protocols, which can be used for both WAN and LAN interconnections, and repackages the received information to suit the needs of the destination system.

No transmission can be 100 percent reliable, and TCP will retransmit if it doesn't receive acknowledgement for a particular packet. Any retransmission policy requires some means by which the receiver can detect duplicate arrivals, so here comes the retransmission window strategy. The strategy was to improve the retransmission by defining a part of length of sequence numbers as 'window' and in the different ways the receiver and sender react to different situations.

Both hosts of a TCP connection set the receive buffer for the connection. The data sent by the sender is first placed in the data cache, and the receiver reads the data from the cache. If the receiver reads the data less than the sender sends the data. The speed of the cache will overflow. To avoid buffer overflows, TCP must provide traffic control services to the application process. TCP allows the two sides of the connection to each maintain a dynamic receiving window to provide flow control. Through the receiving window, the sender can know the size of the buffer space available to the receiver to adjust the sending speed. When the receiving window of the receiver is 0, the sender continues to send the segment with only one byte of data to the receiver, and the buffer starts to clear after the receiver confirms, so that the sender can continue to send the data message.

3.4.

S1:

destination	port
A	1
B	2
default	3

S2:

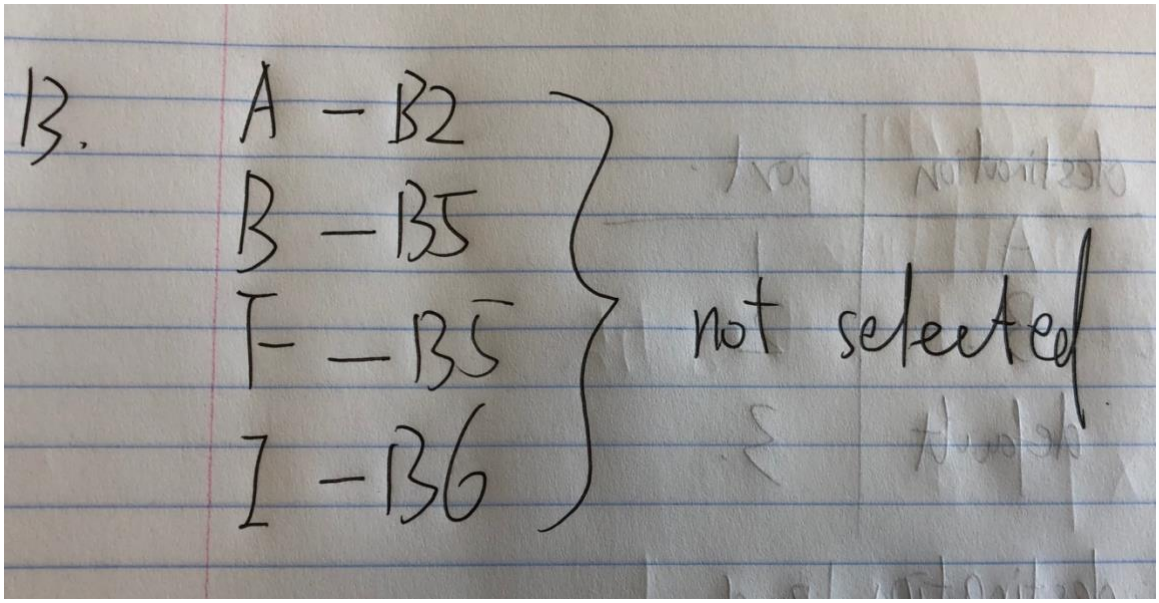
destination	port
A	1
B	1
C	3
D	3
default	2

S3:

destination	port
C	2
D	3
default	1

S4:

destination	port
D	2
default	1


13.
A – B2
B – B5
F – B5
I – B6
not selected

15. B1:

A-interface A B2-interface: C

B2:

B1-interface: A B3-interface: C

B4-interface D

B3:

B2-interface: A, D C-interface: C

B4:

B2-interface: A D-interface: D

3.46

(a)

	A	B	C	D	E	F
A	0	X	3	8	X	X
B	X	0	X	X	2	X
C	3	X	0	X	16	
D	8	X	X	0	2	X
E	X	2	1	2	0	X
F	X	X	6	X	X	0

(b)

	A	B	C	D	E	F
A	0	X	3	8	4	9
B	X	0	3	4	2	X
C	3	3	0	3	16	
D	8	4	3	0	2	X
E	4	2	1	2	0	7
F	9	X	6	X	7	0

(c)

	A	B	C	D	E	F
A	0	6	3	6	4	9
B	6	0	3	4	2	9
C	3	3	0	3	16	
D	6	4	3	0	2	9
E	4	2	1	2	0	7
F	9	9	6	9	7	0

3.54. (a). A:

dest	cost	next hop
B	2	-
C	3	C
D	∞	-
E	∞	-
F	9	C

B:

dest	cost	next
A	2	-
C	∞	-
D	4	E
E	2	E
F	∞	-

D:

dest	cost	next hop
A	∞	-
B	4	E
C	∞	-
E	2	E
F	∞	-

F.

dest	cost	next hop
A	9	C
B	∞	-
C	6	C
D	∞	-
E	∞	-

(b). A:

dest	cost	nexthop
B	12	D
C	3	C
D	8	D
E	10	D
F	9	C

D.

dest	cost	nexthop
A	8	A
B	4	E
C	11	A
E	2	E
F	17	A

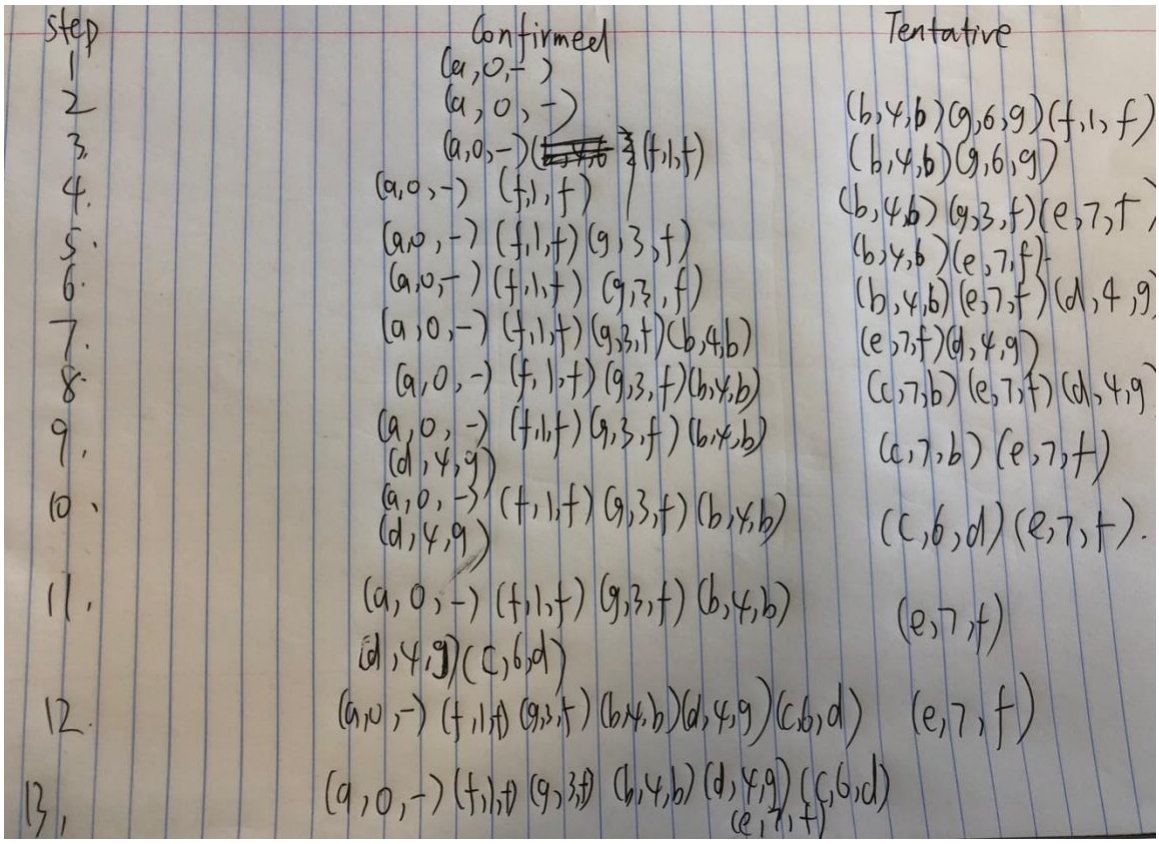
(c). C:

dest	cost	nexthop
A	3	A
B	15	A
D	11	A
E	13	A
F	6	F

3.73.

(a) F (b) B (c) E

(d) A (e) D (f) C.


step
Confirmed
Tentative
1
(a,0,-)
2
(a,0,-)
(b,4,b)(g,6,g)(f,1,f)
3
(a,0,-)(f,1,f)
(b,4,b)(g,6,g)
4.
(a,0,-) (f,1,f)
(b,4,b)(g,3,f)(e,7,f)
5.
(a,0,-) (f,1,f)(g,3,f)
(b,4,b)(e,7,f)
6.
(a,0,-) (f,1,f) (g,3,f)
(b,4,b)(e,7,f)(d,4,g)
7.
(a,0,-) (f,1,f)(g,3,f)(b,4,b)
(e,7,f)(d,4,g)
8.
(a,0,-) (f,1,f)(g,3,f)(b,4,b)
(c,7,b)(e,7,f)(d,4,g)
9.
(a,0,-) (f,1,f)(g,3,f)(b,4,b)
(d,4,g)
(c,7,b)(e,7,f)
10.
(a,0,-) (f,1,f)(g,3,f)(b,4,b)
(d,4,g)
(c,6,d)(e,7,f).
11.
(a,0,-) (f,1,f)(g,3,f)(b,4,b)
(d,4,g)(c,6,d)
(e,7,f)
12.
(a,0,-) (f,1,f)(g,3,f)(b,4,b)(d,4,g)(c,6,d)
(e,7,f)
13.
(a,0,-)(f,1,f)(g,3,f)(b,4,b)(d,4,g)(c,6,d)
(e,7,f)

```
import socket

s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

s.connect(('cse224.sysnet.ucsd.edu',5555))

context = []

arr = "A53271697\r\n".encode(encoding='ascii')

s.send(arr)

while True:

    d = s.recv(1024)

    if d:

        context.append(d)

    else:

        break

data = b''.join(context)

data = data.decode("utf-8")

print(data)

s.close()
```

End of Project Gutenberg Etext of King Henry IV, Part 1 by Shakespeare

keyword is: 3B8A