IP Address Handoff in the MANET

IEEE INFOCOM 2004

2004.2.19 林佑青

Outline

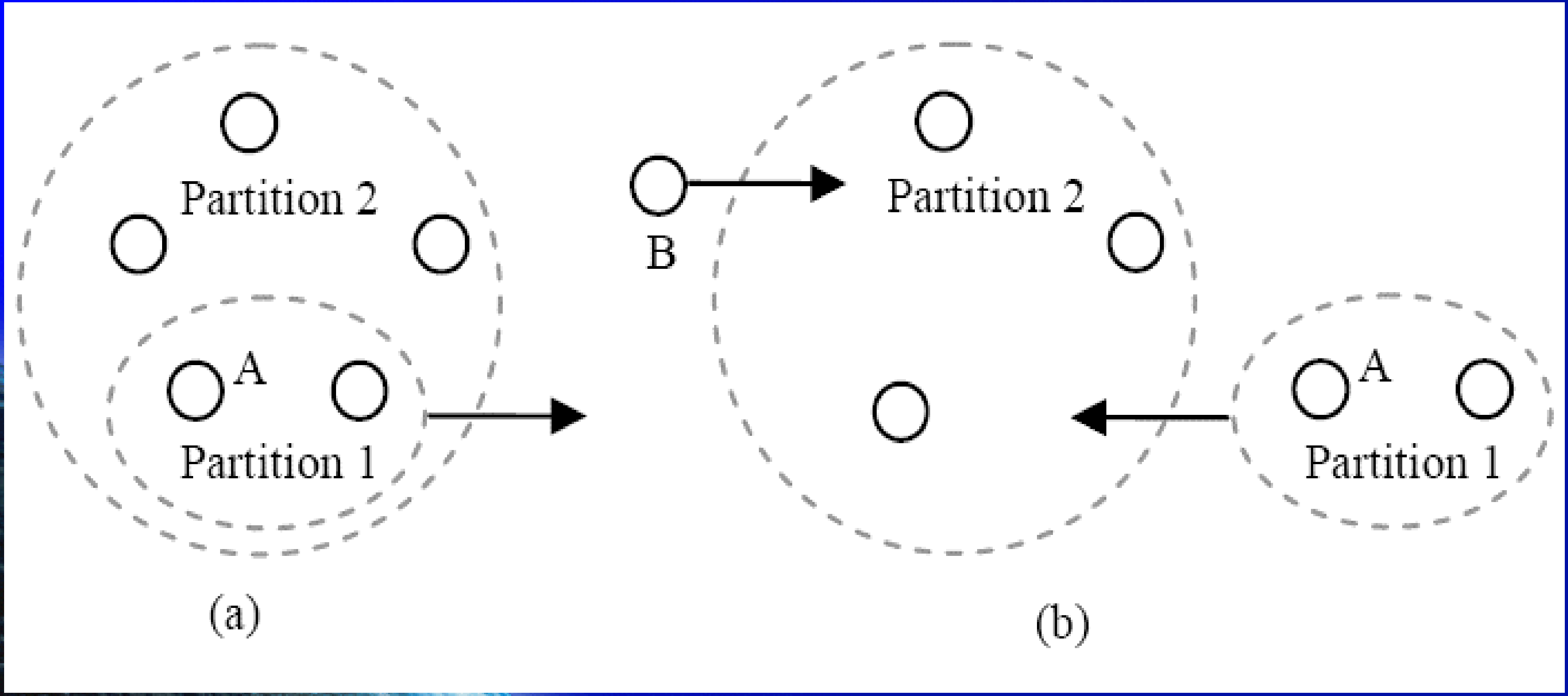
- Introduction
- Issues and related work
- Solutions
  - Broken routing fabrics
  - Broken communications
- Performance evaluation
- Conclusion

Introduction

- A mobile node in the MANET may change its IP address more frequently due to the deployment of autoconfiguration, global connectivity, and hierarchical addressing schemes.
- There are some solutions for IP address change, but the overhead resulting from address changes has not been carefully examined.

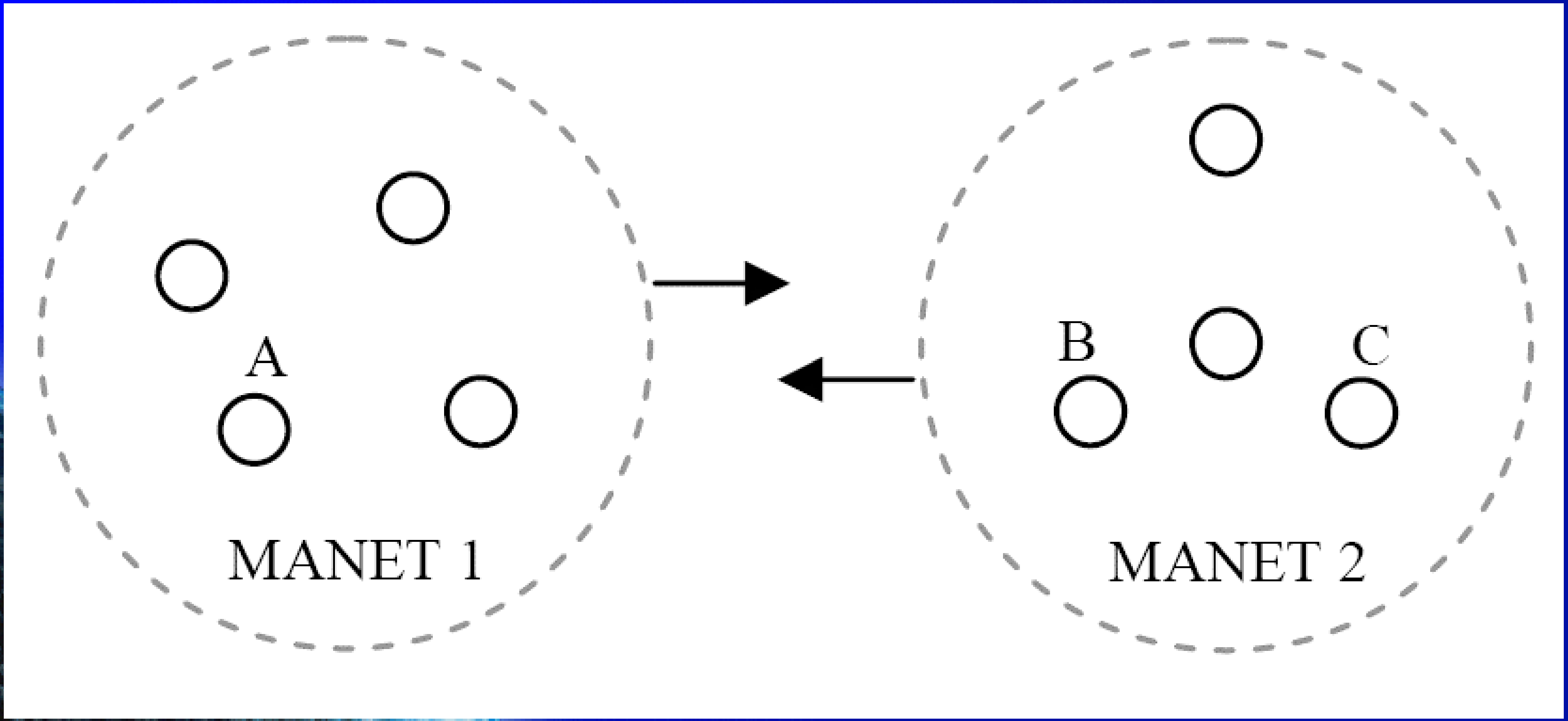
Scenarios of IP address change (1)

- Merger of two partitions of a network



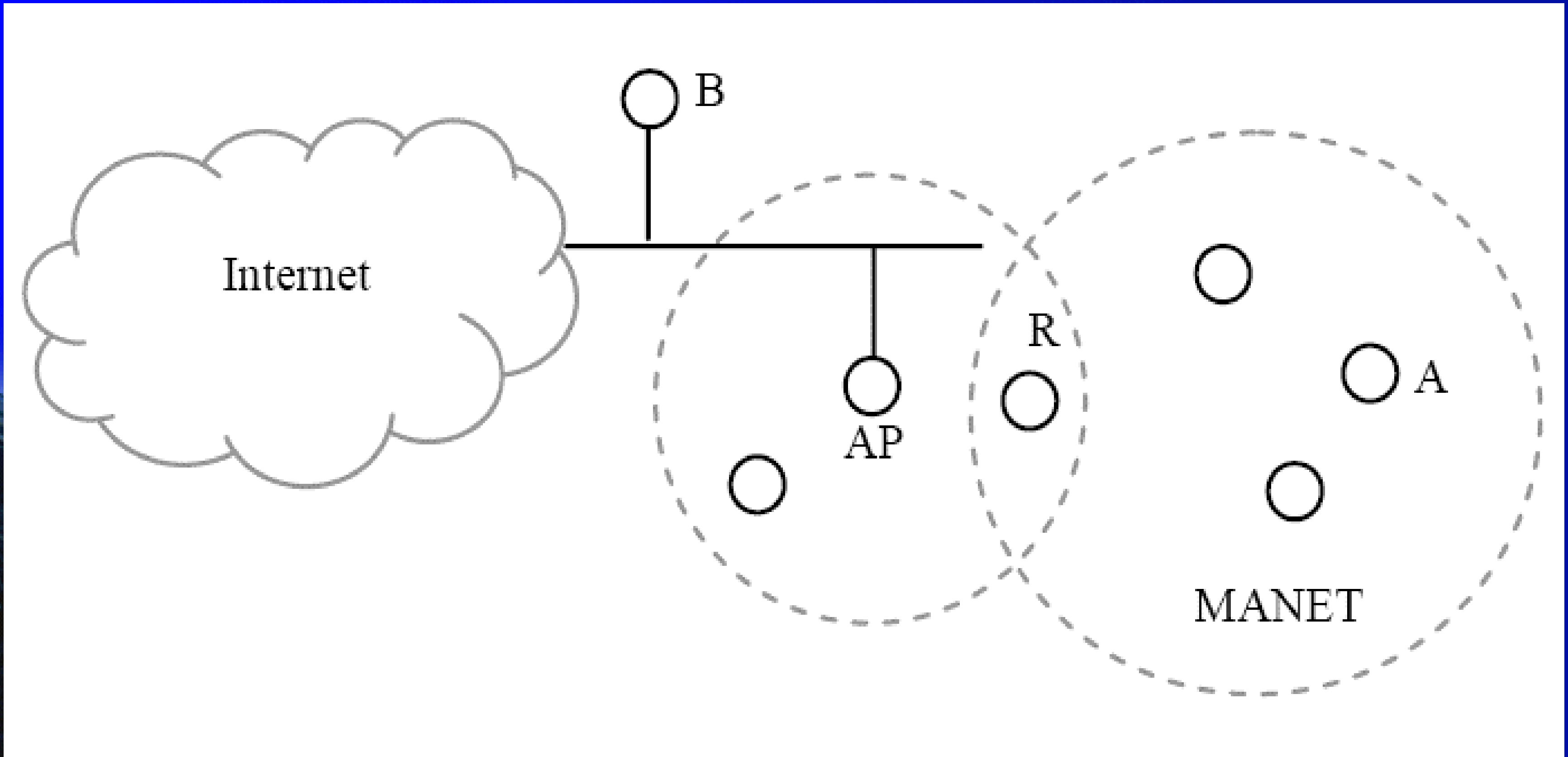
Scenarios of IP address change (2)

- Merger of two independent MANETs



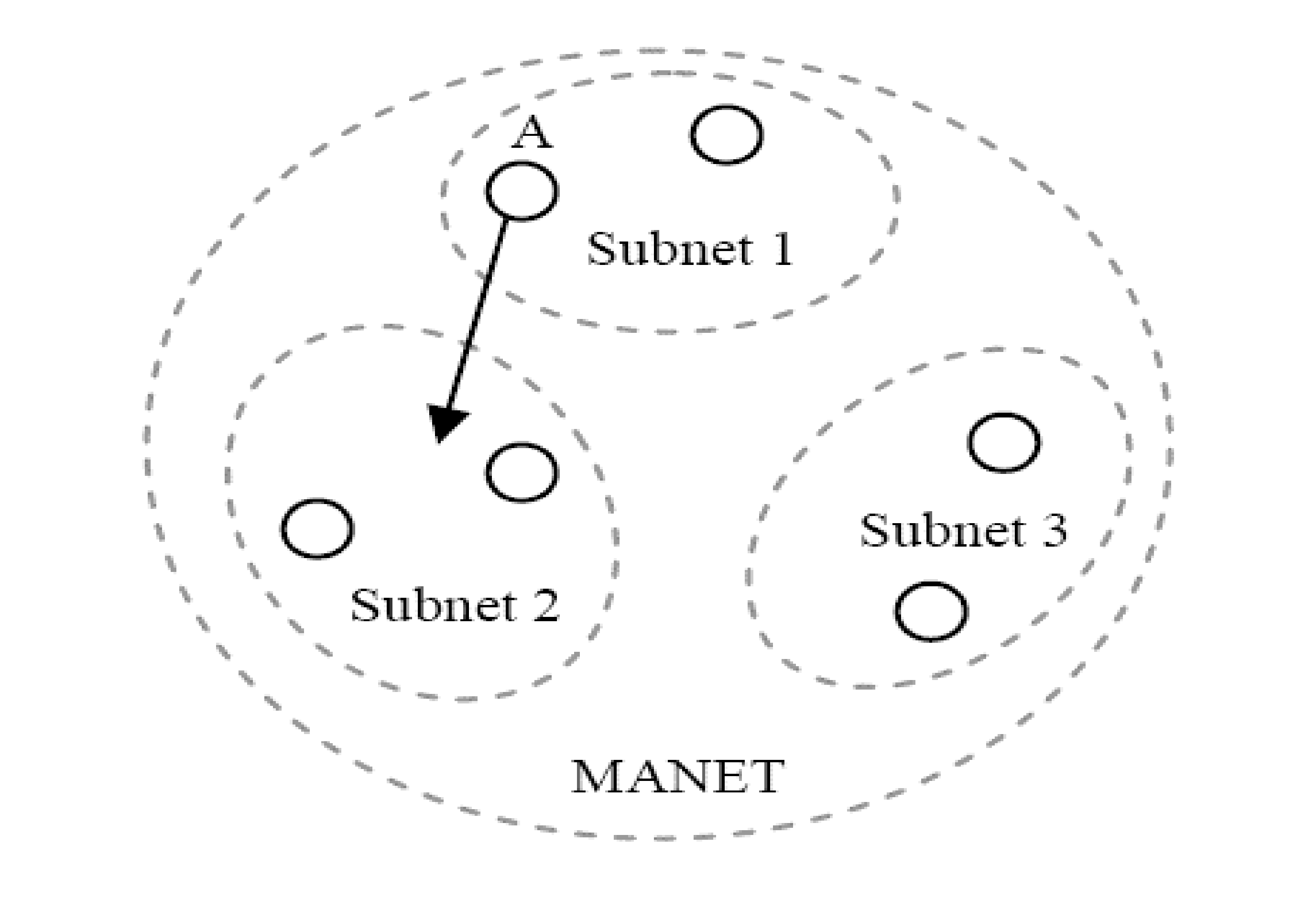
Scenarios of IP address change (3)

- Merger of a MANET with a LAN

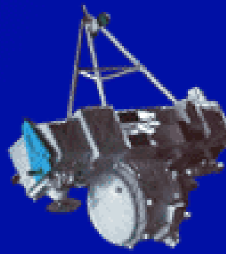


Scenarios of IP address change (4)

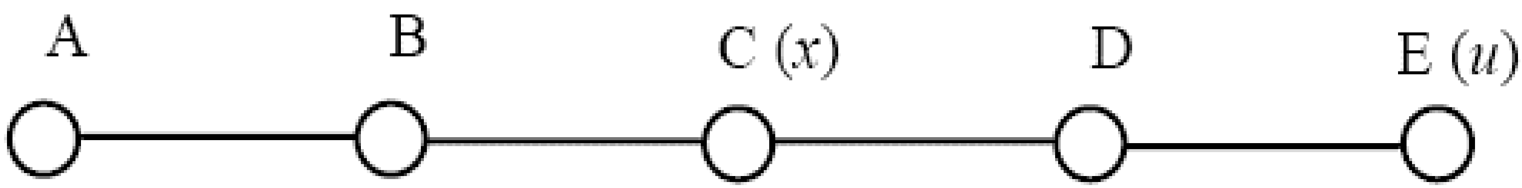
- A MANET with a hierarchical addressing



Issues



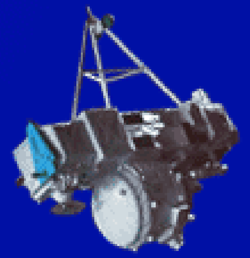
- Broken routing fabrics
- Broken on-going communications



Dest	Next
D	C (x)
E	C (x)

Dest	Next
B	C (x)
A	C (x)

Related work



- Mobile IP
  - The nodes in the MANET are mobile and instable, none of them can be designated as the home agent or foreign agent for another node.
- Tunneling mechanism
  - IP-in-IP
  - Overhead
  - DoS attack

Solutions to broken routing fabrics (1)

- Assume AODV as the routing protocol.
- Route Shift packet
  - Contains the source's old address and new address.
  - On receipt of the packet, the neighbors change the next hop from x to y.
  - It is vulnerable to IP spoofing attacks.

Solutions to broken routing fabrics (2)

- Cryptographic method
  - Node C signs the Route Shift packet with its private key.
  - All neighbors contact the CA to get the certificate for node C's public key and validate it.
- Disadvantage
  - Delay and communication overhead.

Solutions to broken routing fabrics (3)

- Node C chooses a random number for address x, and puts the hash value of the number in the Route Shift packet.
- All its neighbors store the hash value in the neighbor tables and routing tables.

Solutions to broken communications (1)

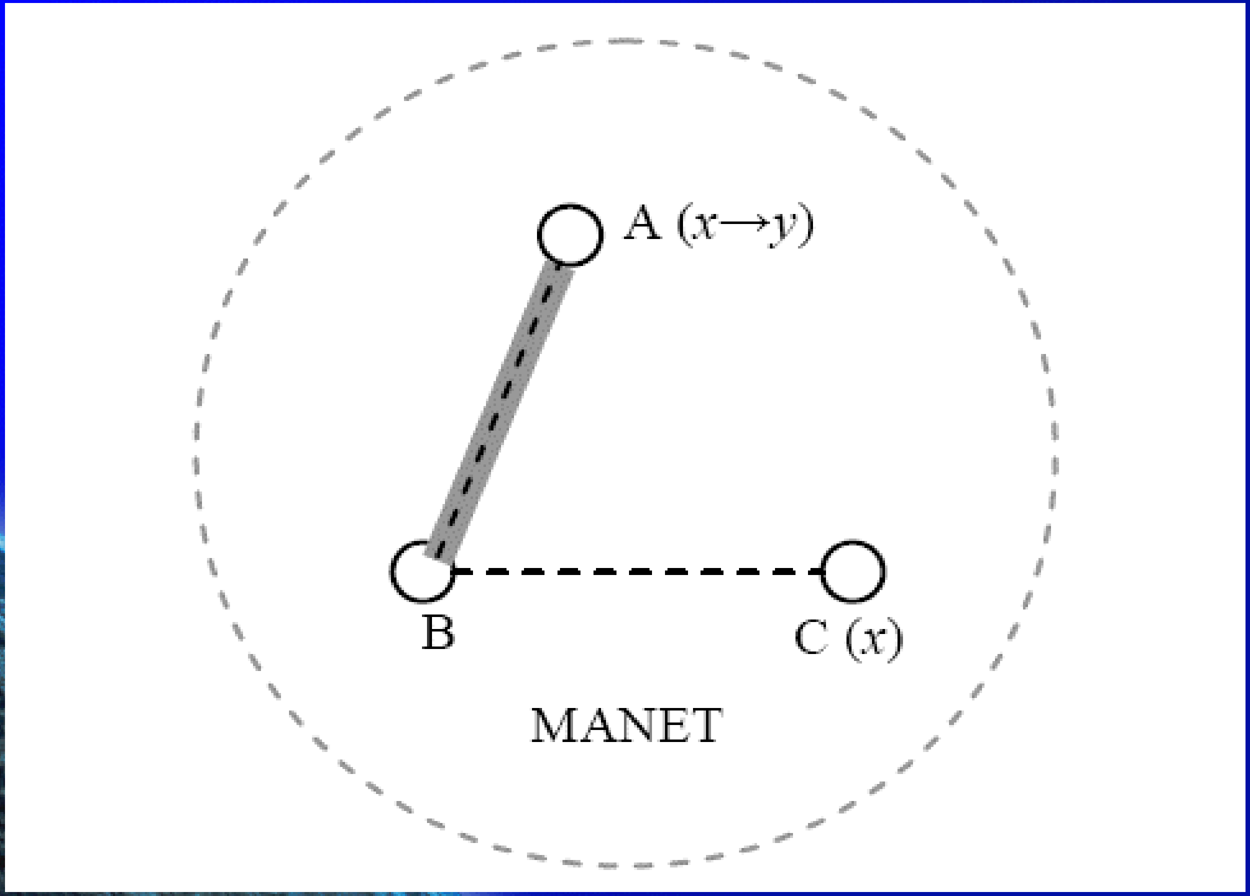
- Suppose that node A is communicating with node B, and node A change its address from x to y.
- Route rebuilding
  - Broadcast of RREQ to build the path
- NAT
  - Node A：new destination address of y → x
    old source address → new address
  - Node B：new source address of y → x
    old destination address → new address

Solutions to broken communications (2)

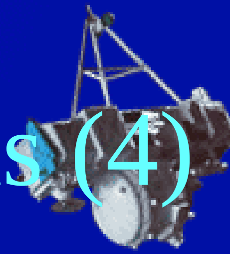
- Advantages
  - The overhead of a second IP header is saved.
  - Only one address in the IP header is modified in NAT, faster when applied with the improved computation of IP checksum.
  - The tunneling scheme brings a "DoS" problem.

Solutions to broken communications (3)

- DoS problem caused by IP tunneling



Solutions to broken communications (4)



- Extends NAT to utilize both port numbers and sequence numbers to distinguish different connections at node B.

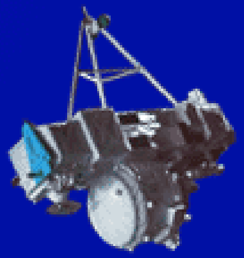
TABLE I. NAT TABLE AT NODE B

1	2	3	4	5	6
Old remote address	**New remote address**	**Local port**	**Remote port**	**Remote sequence number**	**Next remote sequence number**
x	y	80	2030	228743	22884312
…	…	…	…	…	…

Solutions to broken communications (5)

- Address Change Message (ACM)
  - Includes the old address, new address, protocol, local port, remote port, and sequence number.
- To save communication overhead, the message can be combined with the RREP packet.
- To prevent IP spoofing attacks, the ACM packet must be signed with node A’s private key, which can be validated with A’s public key at node B.

Performance evaluation



- Overhead of broken routing fabrics
  – 2pl packets
- Overhead of broken communications
  – 2ml packets

N nodes, l links

K connections with m nodes and on p active paths.

Implementation

cap_handoff1 - Ethereal

File Edit Capture Display Tools

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.155	192.168.1.173	TCP	10000 > 10000 [SYN] Seq=2716095825 Ack=0 Win=5840
2	0.001809	192.168.1.173	192.168.1.155	TCP	10000 > 10000 [SYN, ACK] Seq=787063896 Ack=271609
3	0.001870	192.168.1.155	192.168.1.173	TCP	10000 > 10000 [ACK] Seq=2716095826 Ack=787063897
4	2.941862	192.168.1.155	192.168.1.173	TCP	10000 > 10000 [PSH, ACK] Seq=2716095826 Ack=78706
5	2.943534	192.168.1.173	192.168.1.155	TCP	10000 > 10000 [ACK] Seq=787063897 Ack=2716095830
6	2.944257	192.168.1.173	192.168.1.155	TCP	10000 > 10000 [PSH, ACK] Seq=787063897 Ack=271609
7	2.944282	192.168.1.155	192.168.1.173	TCP	10000 > 10000 [ACK] Seq=2716095830 Ack=787063901
8	28.331967	192.168.1.140	192.168.1.173	TCP	10000 > 10000 [PSH, ACK] Seq=2716095830 Ack=78706
9	28.333859	192.168.1.173	192.168.1.155	TCP	10000 > 10000 [PSH, ACK] Seq=787063901 Ack=271609
10	28.333918	192.168.1.140	192.168.1.173	TCP	10000 > 10000 [ACK] Seq=2716095834 Ack=787063905
11	39.547260	192.168.1.140	192.168.1.173	TCP	10000 > 10000 [PSH, ACK] Seq=2716095834 Ack=78706
12	39.550315	192.168.1.173	192.168.1.155	TCP	10000 > 10000 [PSH, ACK] Seq=787063905 Ack=271609
13	39.550367	192.168.1.140	192.168.1.173	TCP	10000 > 10000 [ACK] Seq=2716095838 Ack=787063909
14	41.621408	192.168.1.140	192.168.1.173	TCP	10000 > 10000 [FIN, ACK] Seq=2716095838 Ack=78706
15	41.623089	192.168.1.173	192.168.1.155	TCP	10000 > 10000 [FIN, ACK] Seq=787063909 Ack=271609
16	41.623156	192.168.1.140	192.168.1.173	TCP	10000 > 10000 [ACK] Seq=2716095839 Ack=787063910

⊞ Frame 1 (74 bytes on wire, 74 bytes captured)

⊞ Ethernet II, Src: 00:02:2d:2d:6f:67, Dst: 00:10:dc:56:58:49

⊞ Internet Protocol, Src Addr: 192.168.1.155 (192.168.1.155), Dst Addr: 192.168.1.173 (192.168.1.173)

⊞ Transmission Control Protocol, Src Port: 10000 (10000), Dst Port: 10000 (10000), Seq: 2716095825, Ack: 0, Len: 0

Conclusion

- Introduced the Route Shift packet, address Change Message, and NAT scheme to solve the address change problem.
- Future work
  - Simulation of the schemes needs more effort.
  - Implementation only aims at one node’s address change.
  - The scenarios of more connections and more address changes .