On Redundant Multipath Operating System Support for Wireless Mesh Networks

Presented by Raluca Musaloiu-E.
Johns Hopkins University

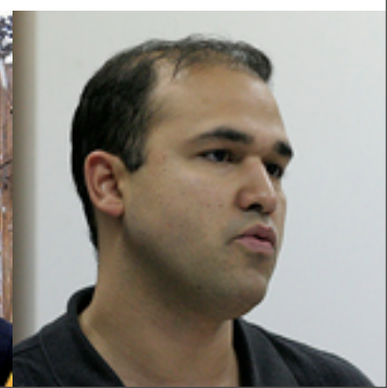
Yair Amir

Claudiu Danilov

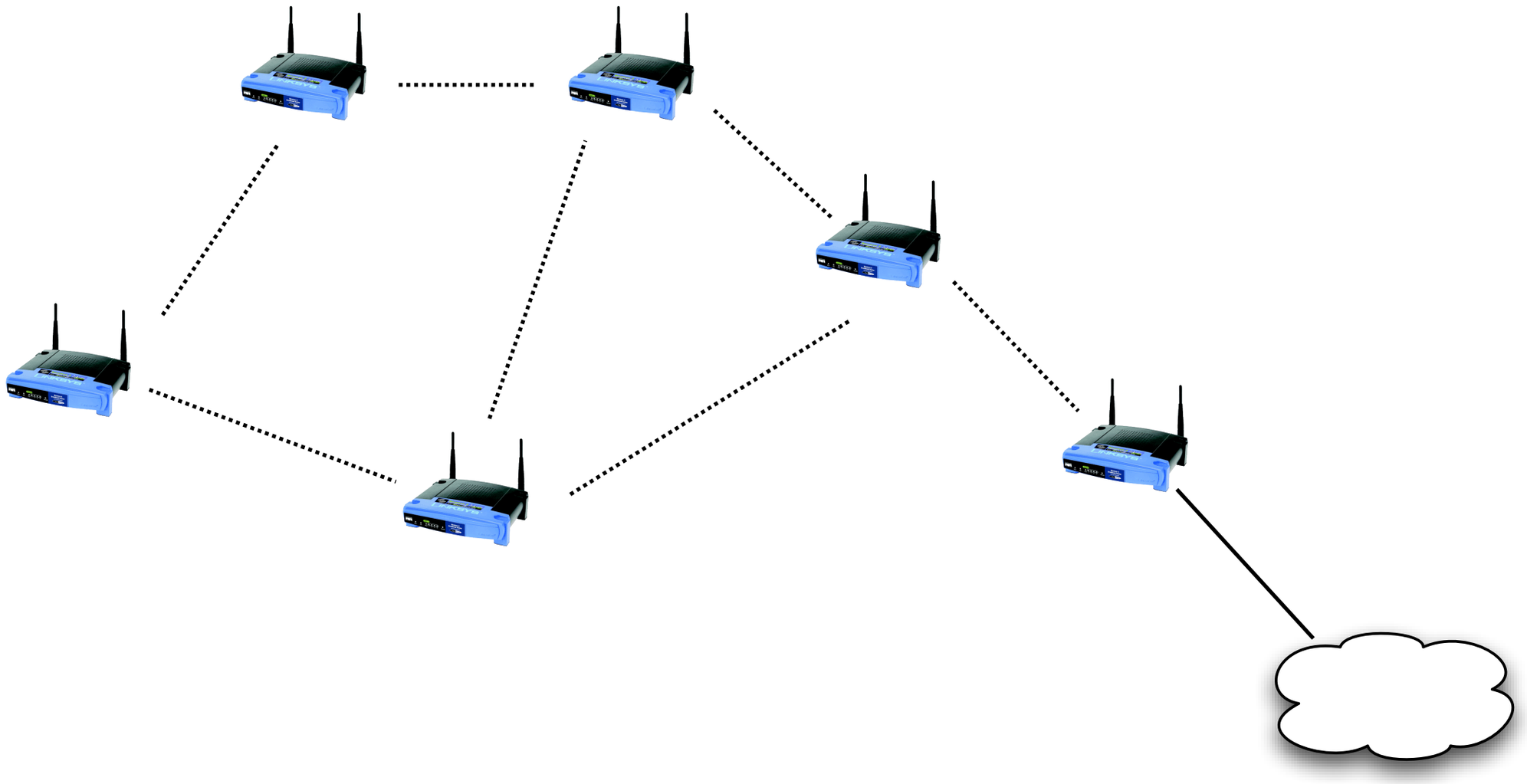
Michael Kaplan

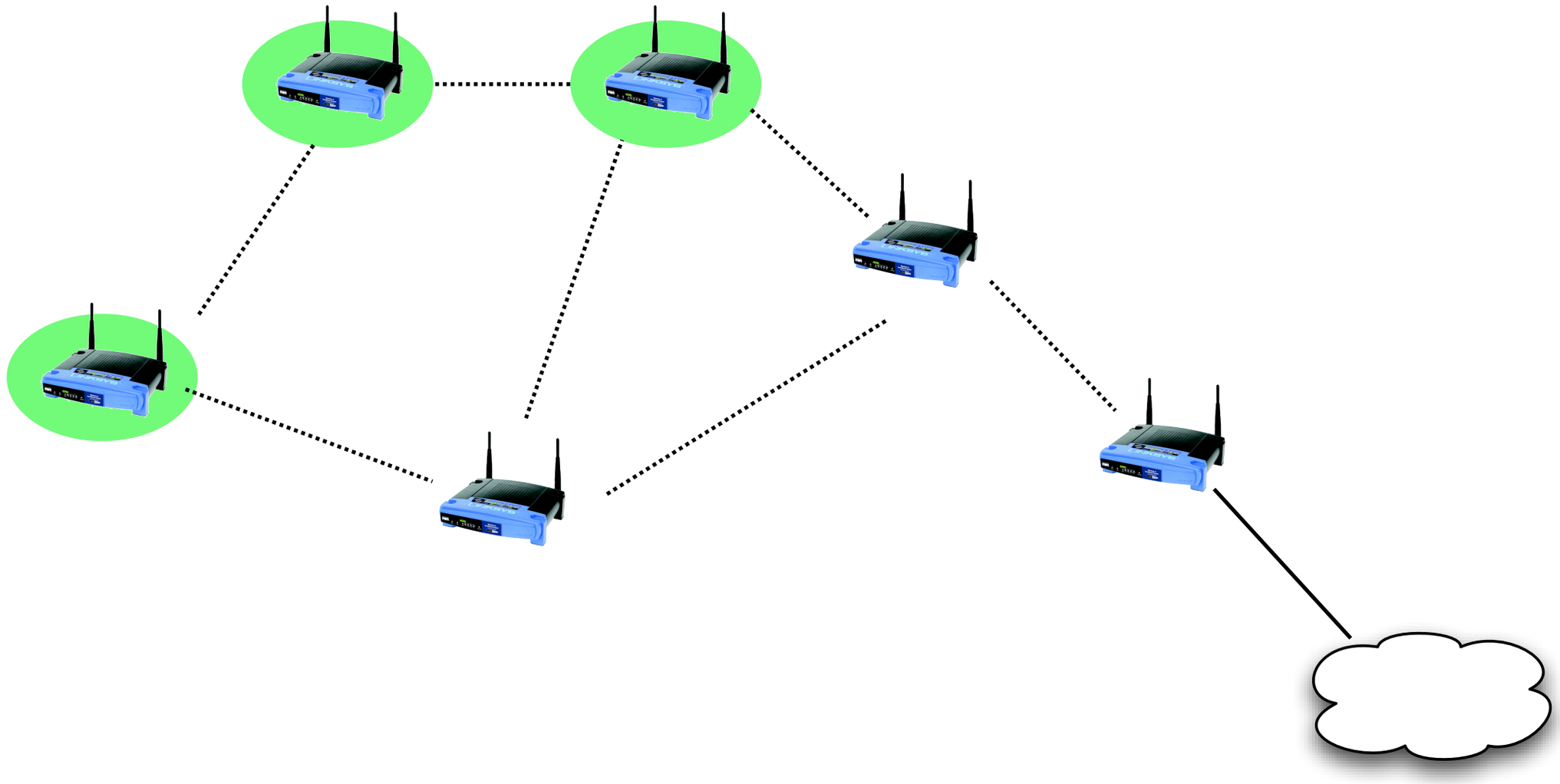
That's Me

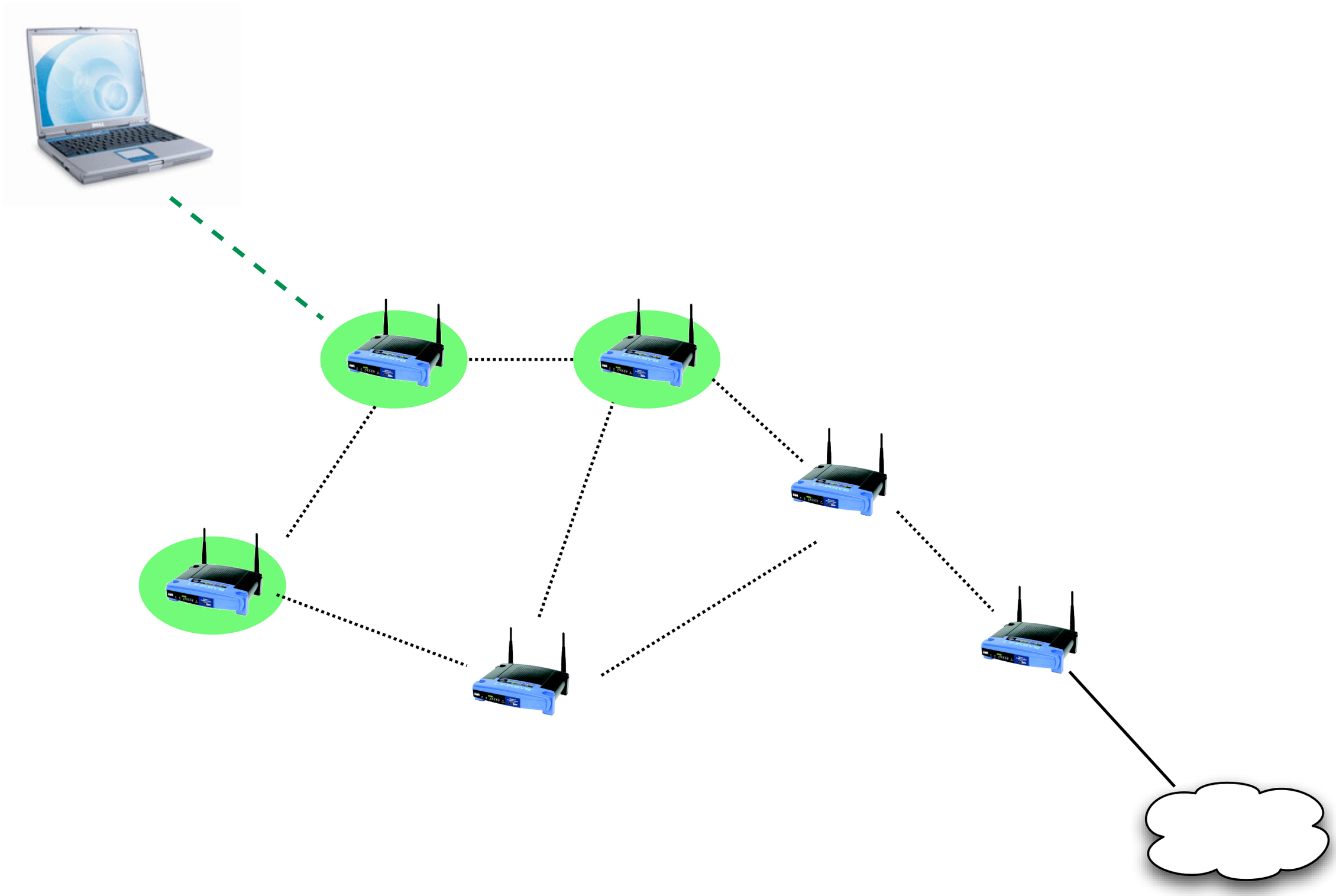
Nilo Rivera

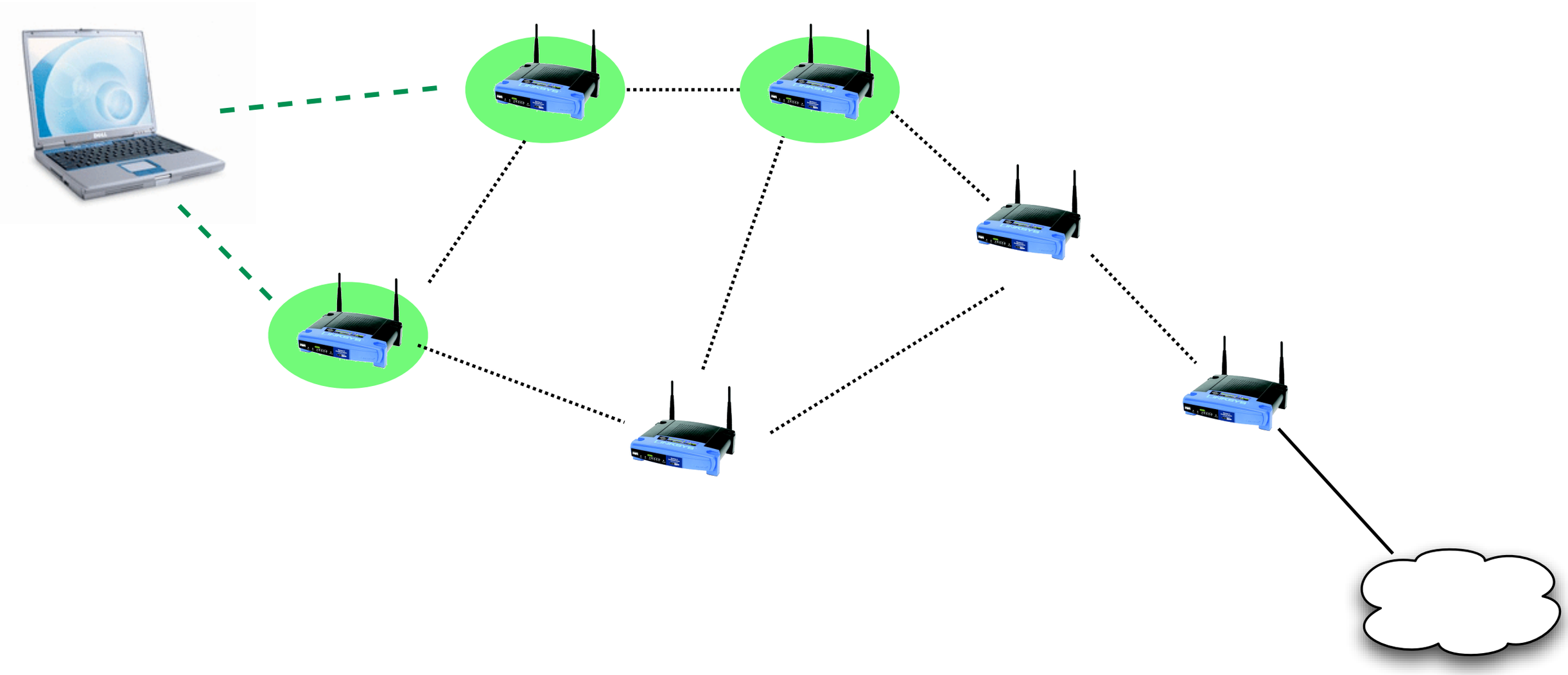


SMesh story







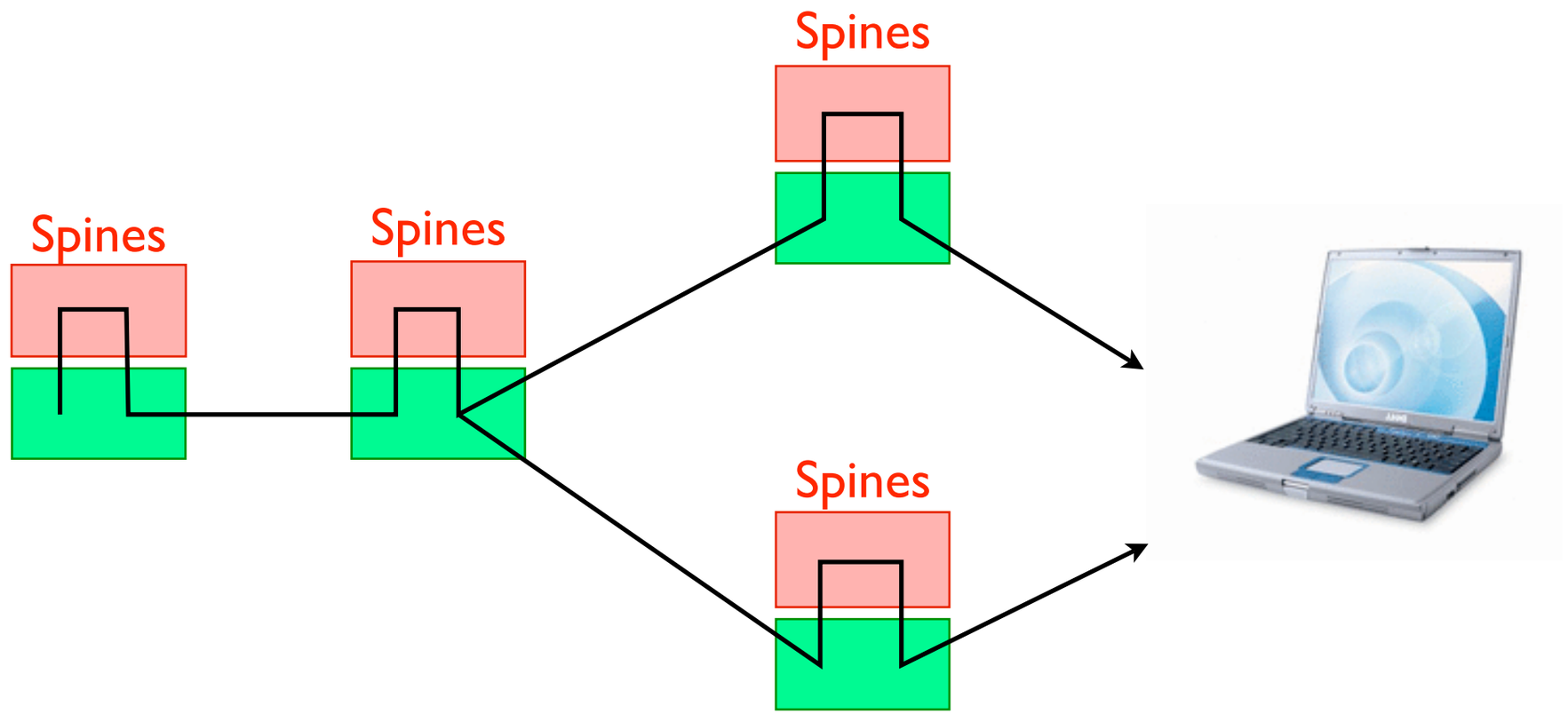


Redundant multipath routing is an essential service for increasing reliability in wireless mesh networks.

Redundant multipath is not natively **supported** by current operating systems.

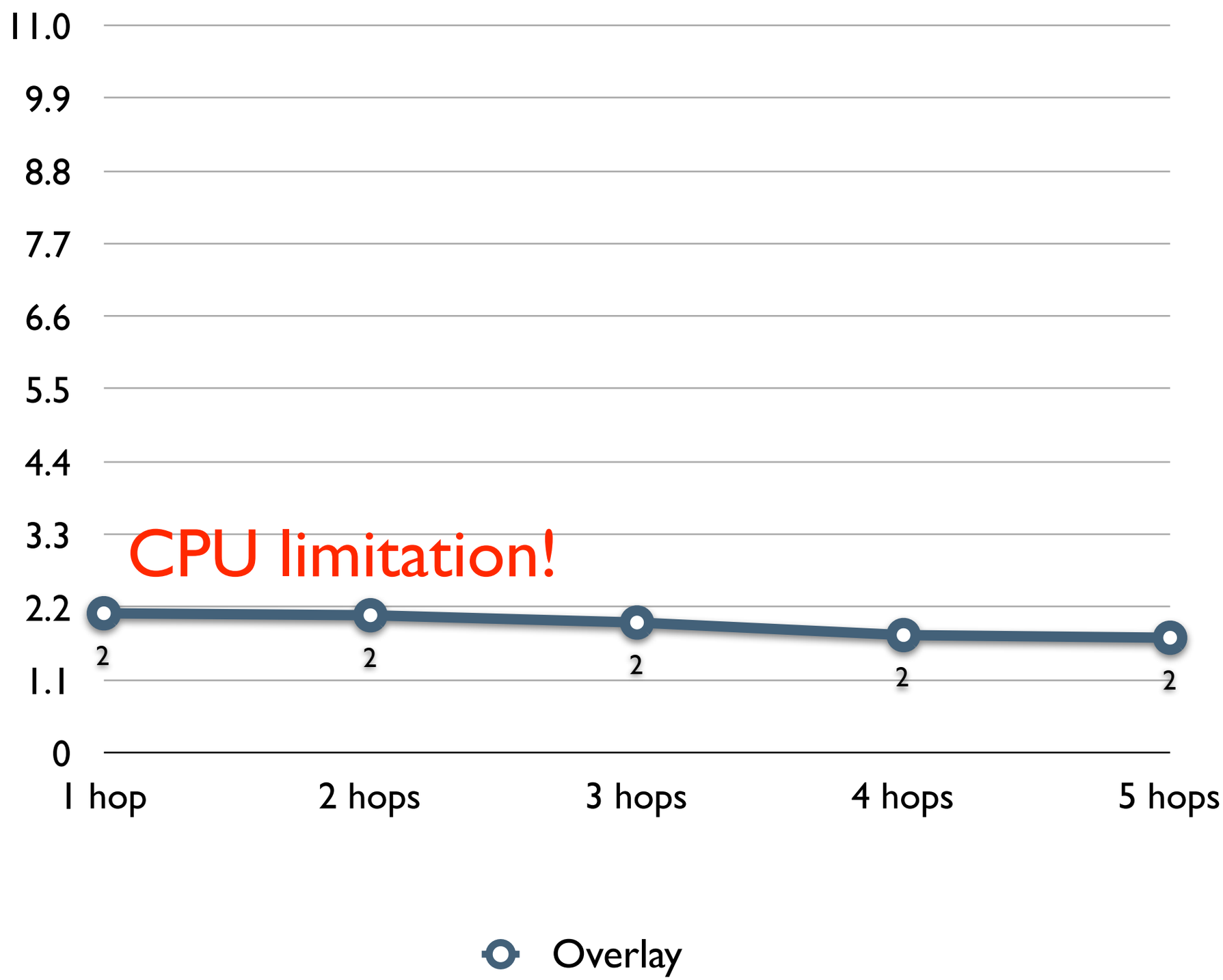
User space

Kernel space

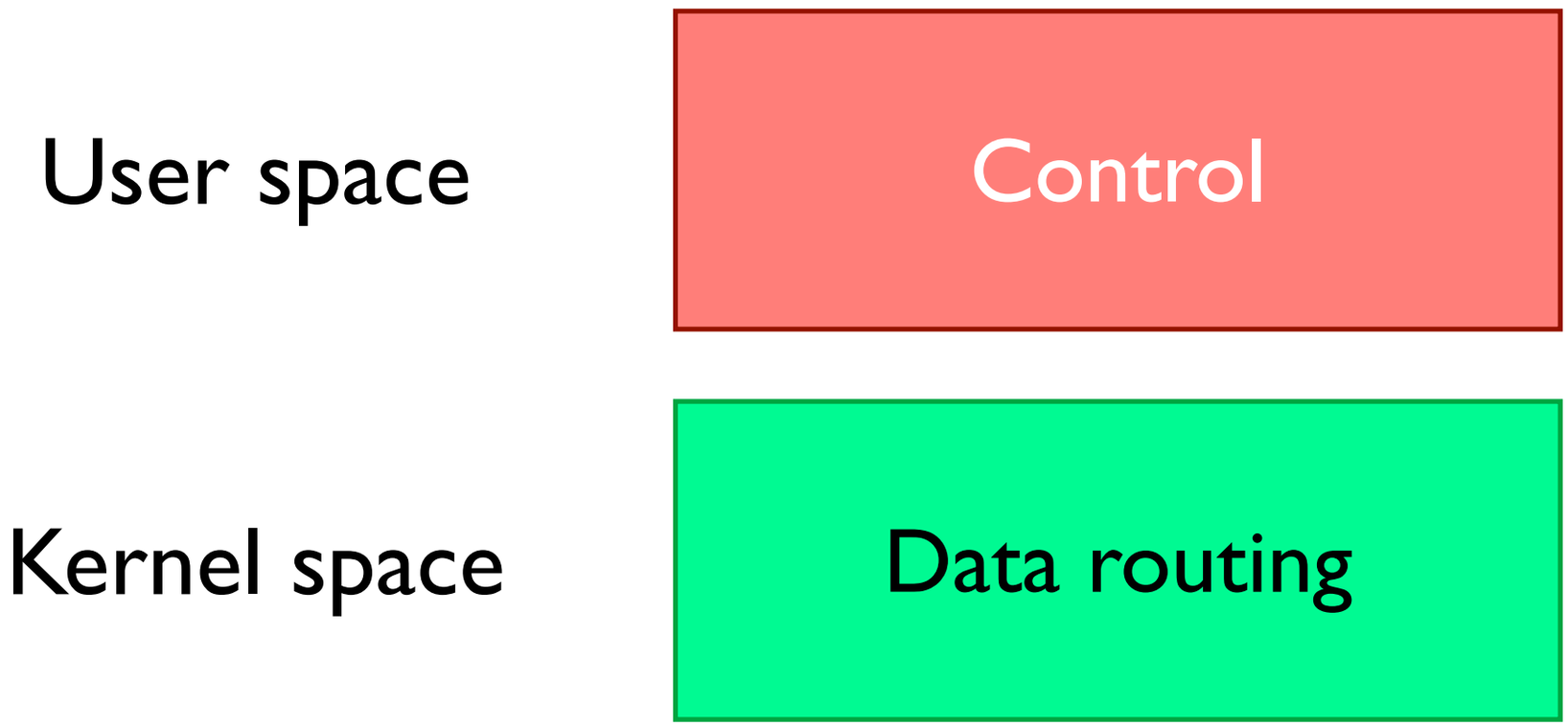


A cost effective wireless mesh deployment requires **low-cost** mesh nodes.

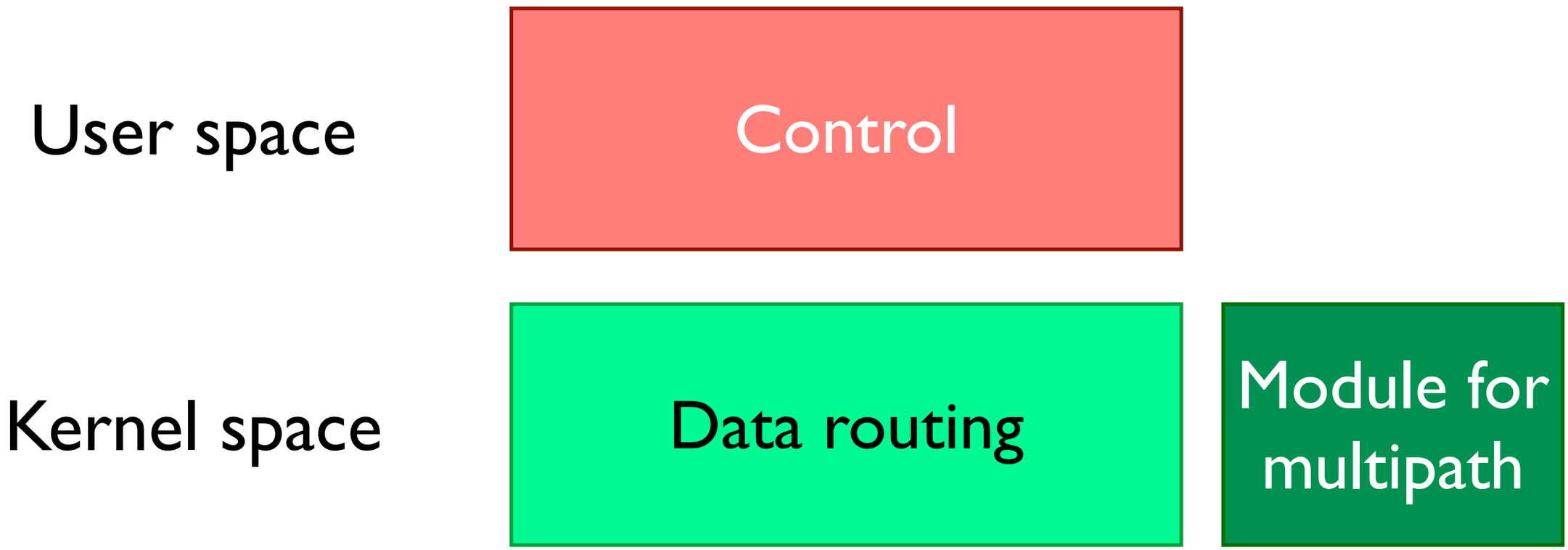
TCP Throughput (Mbps)



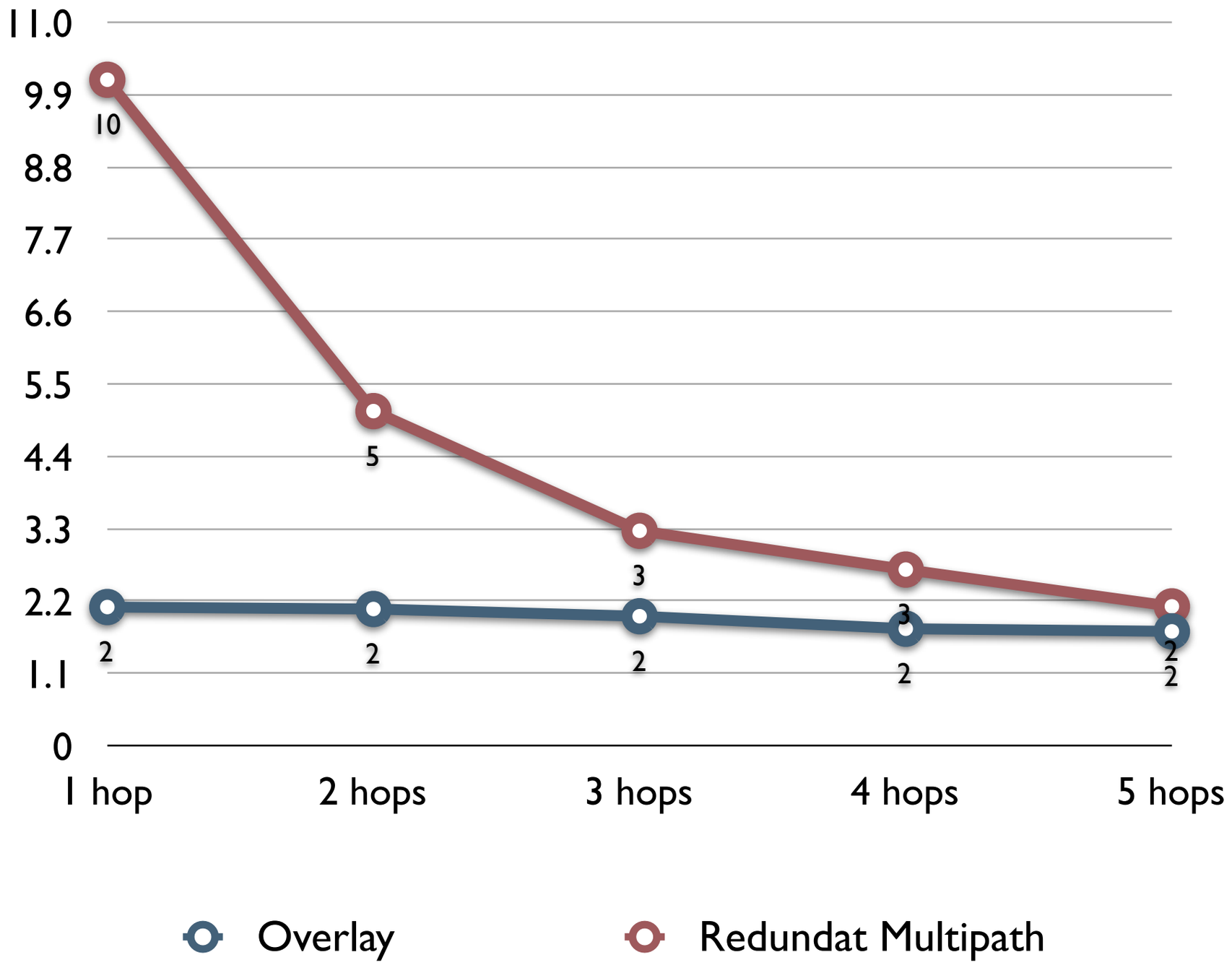
We present a **minimally invasive** mechanism to support redundant multipath routing in kernel-space.



We present a **minimally invasive** mechanism to support redundant multipath routing in kernel-space.

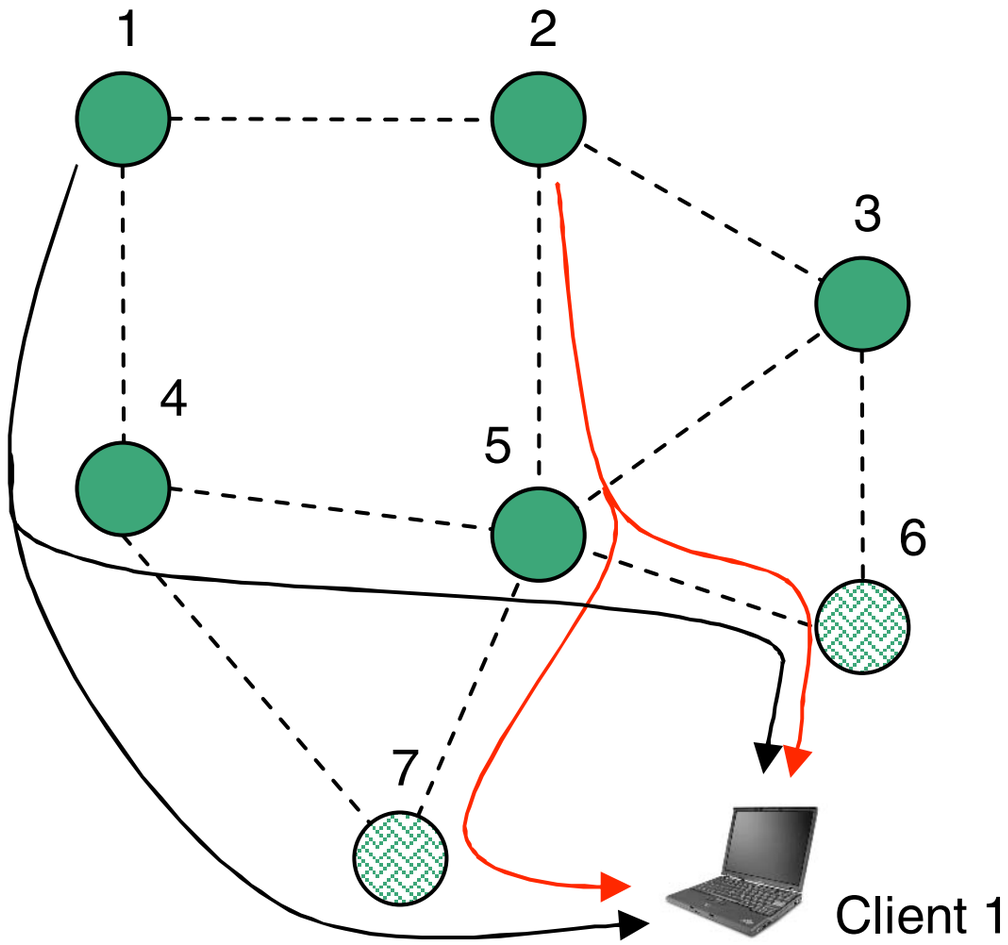


TCP Throughput (Mbps)

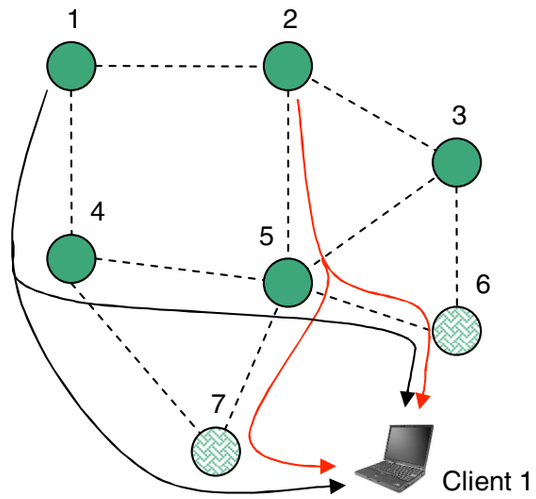


Architecture

To route, consider **entry point**, in addition to destination address.



We use **multiple routing tables**.



Node 5

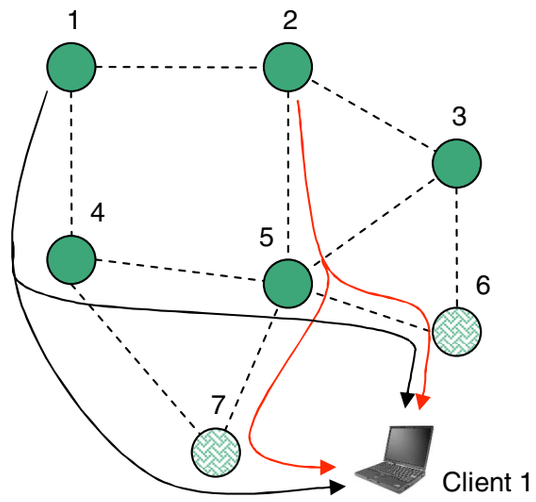
Node 3

Node 2

Node 1

client 1
client 2
....

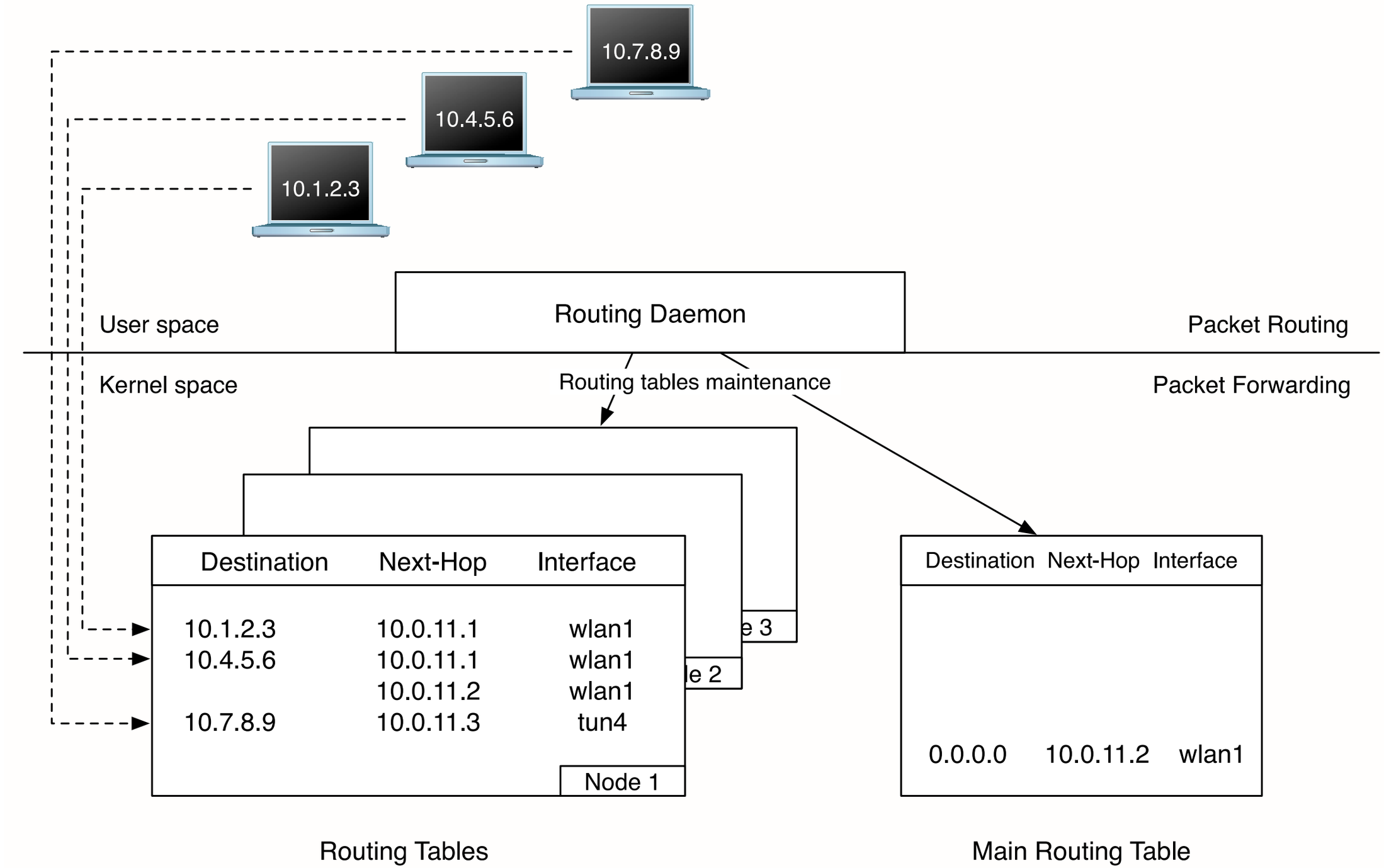
Each route may have **multiple next-hops**.



Node 5

Node 1

Destination	Next-hops
client 1	6, 7
client 2	3
...	...


10.7.8.9
10.4.5.6
10.1.2.3
User space
Kernel space
Routing Daemon
Routing tables maintenance
Packet Routing
Packet Forwarding
Destination Next-Hop Interface
10.1.2.3 10.0.11.1 wlan1
10.4.5.6 10.0.11.1 wlan1
10.0.11.2 wlan1
10.7.8.9 10.0.11.3 tun4
Node 1
Routing Tables
Destination Next-Hop Interface
0.0.0.0 10.0.11.2 wlan1
Main Routing Table

Implementation

Encode entry node in the packet's **IP header**.

0							7	8							15	16							23	24							31
Version				IHL				TOS								Total length															
Identification (IPID)																Flags	Fragment offset													
TTL								Protocol								Header checksum															
Source IP																															
Destination IP																															
Options and padding																															

Encode entry node in the packet's **IP header**.

0							7	8							15	16							23	24							31
Version				IHL				TOS								Total length															
Identification (IPID)																Flags	Fragment offset													
TTL								Protocol								Header checksum															
Source IP																															
Destination IP																															
Options and padding																															

Encode entry node in the packet's **IP header**.

0							7	8							15	16							23	24							31
Version				IHL				TOS								Total length															
Identification (IPID)																Flags	Fragment offset													
TTL								Protocol								Header checksum															
Source IP																															
Destination IP																															
Options and padding																															

Use
policy routing
and define
multiple routing
tables.

```
# iptables -A PREROUTING -t mangle
-m u32 --u32 "2&0xFFFF=35"
-j MARK --set-mark 35
```

```
# ip rule add fwmark 35 table 35
```

Use

MULTIHOP

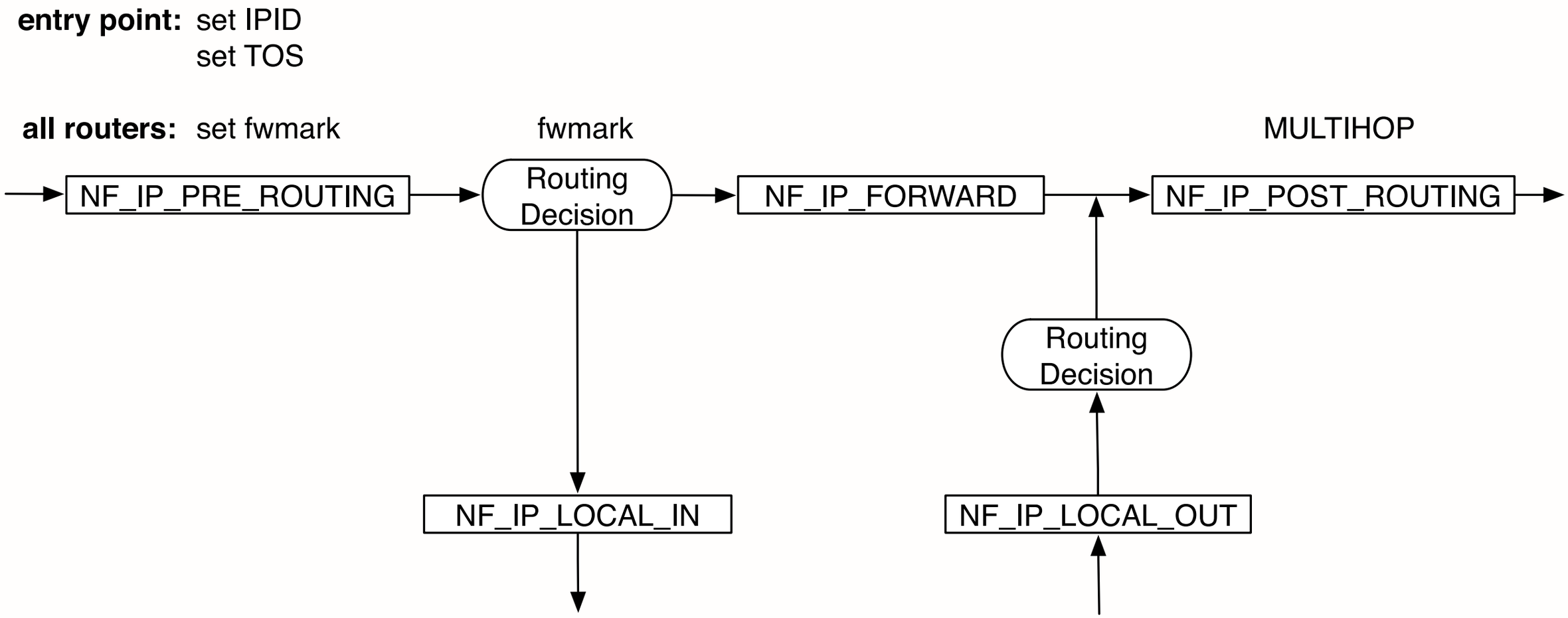
Netfilter

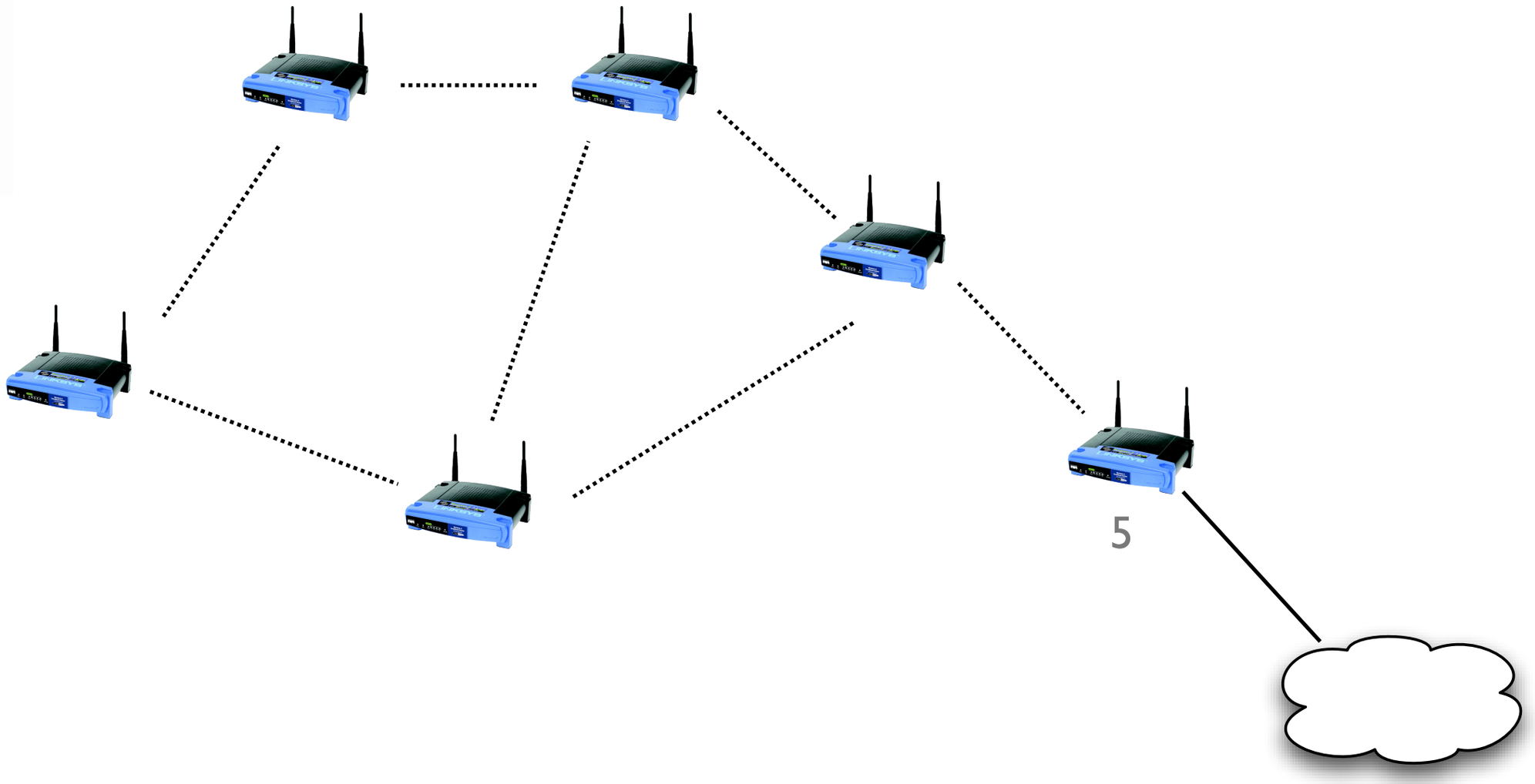
module.

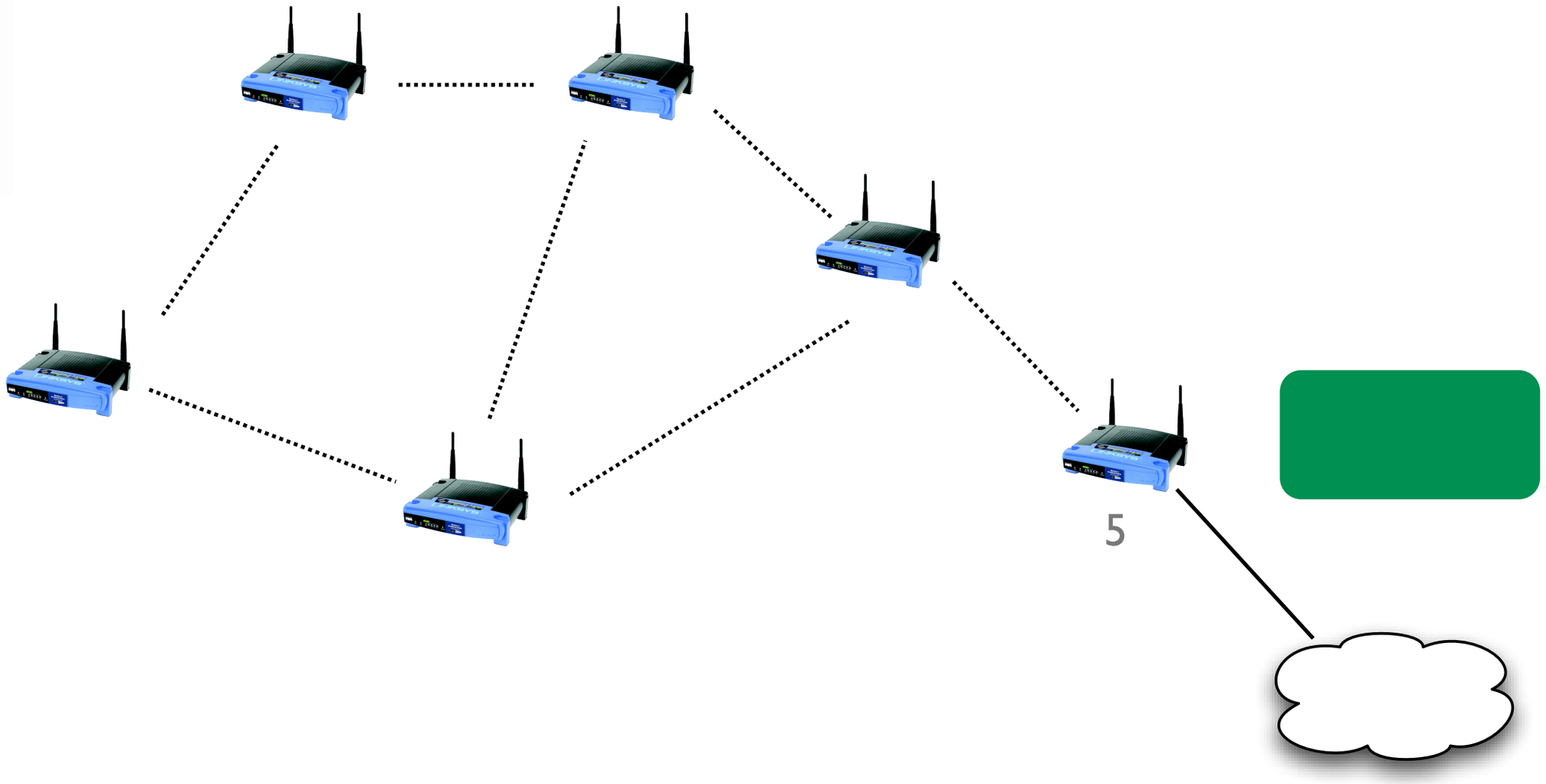
CONFIG_IP_ROUTE_MULTIPATH

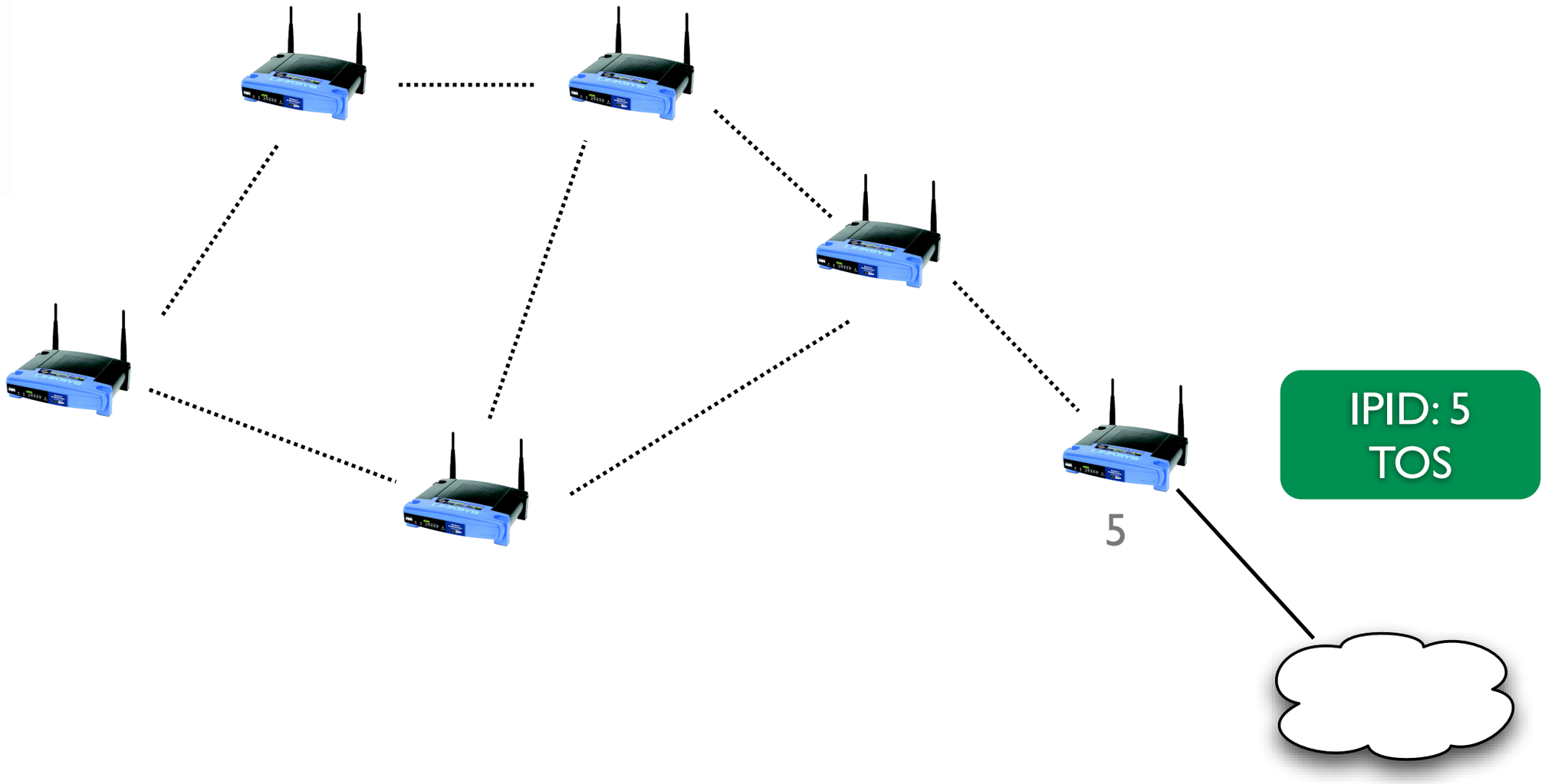
```
# ip route add 10.233.59.169/32 table 35
        nexthop via 10.0.11.32 dev eth1
        nexthop via 10.0.11.33 dev eth1
```

MULTIHOP


entry point: set IPID
set TOS
all routers: set fwmark
fwmark
MULTIHOP
NF_IP_PRE_ROUTING
Routing Decision
NF_IP_FORWARD
NF_IP_POST_ROUTING
NF_IP_LOCAL_IN
Routing Decision
NF_IP_LOCAL_OUT

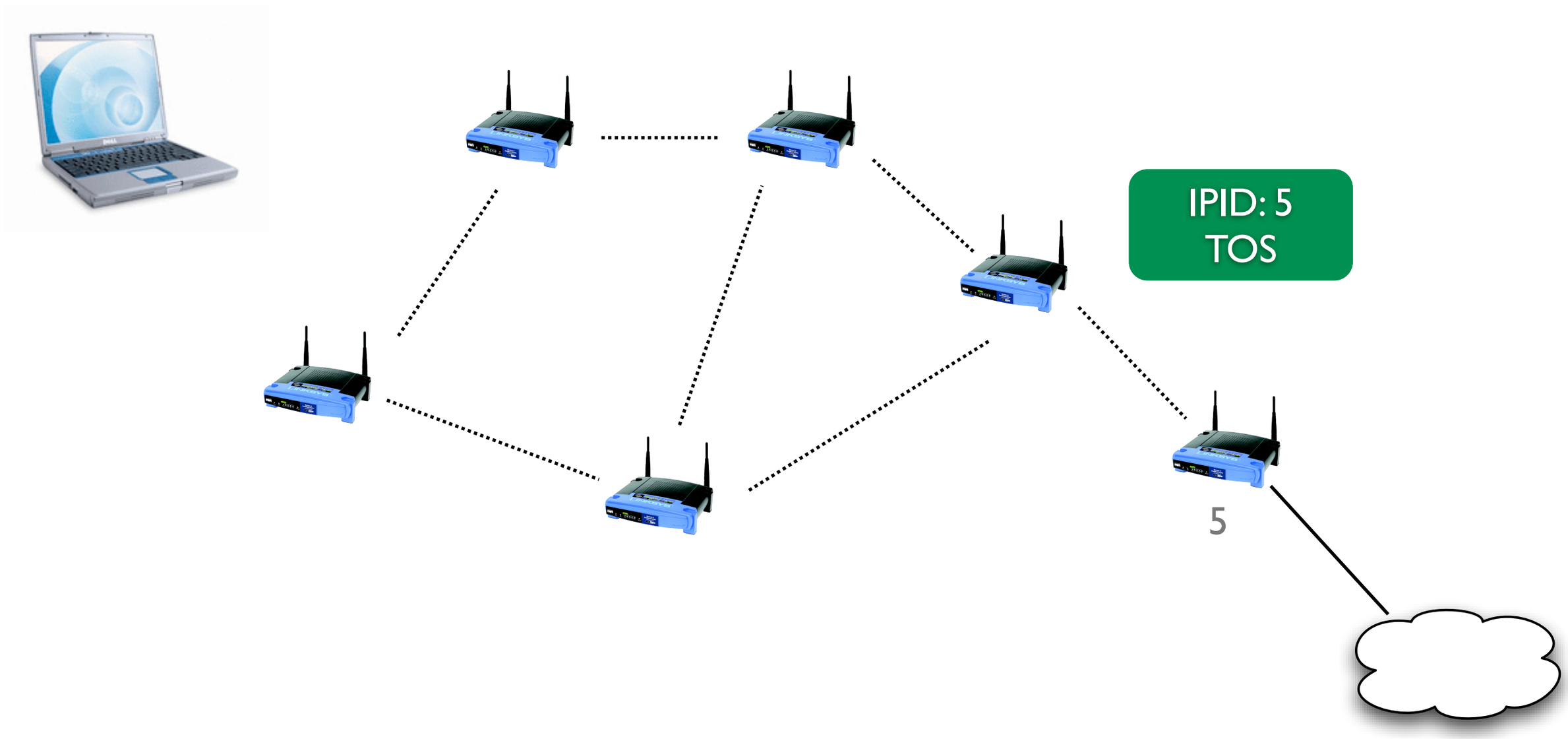




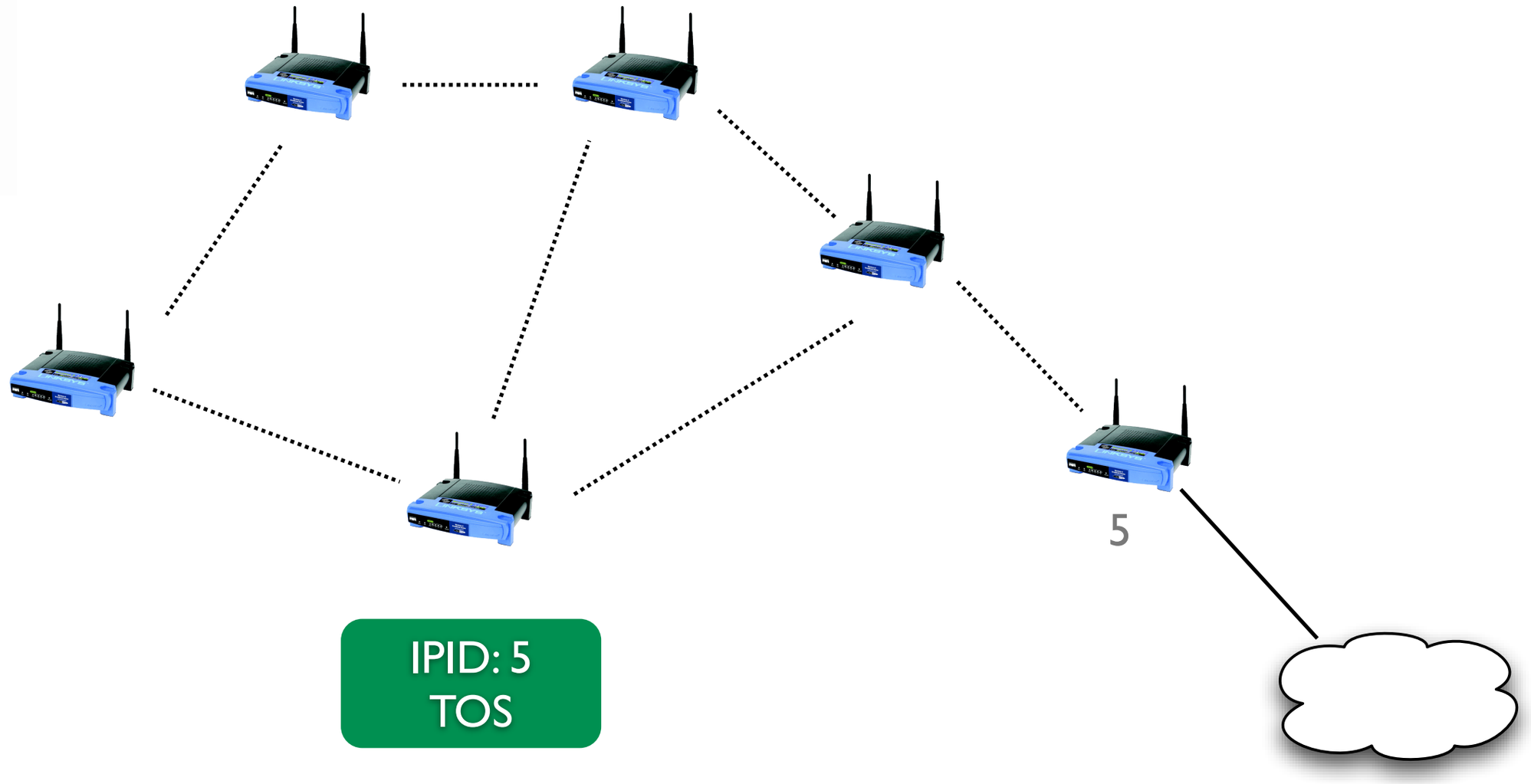


IPID: 5
TOS

5


IPID: 5
TOS
5

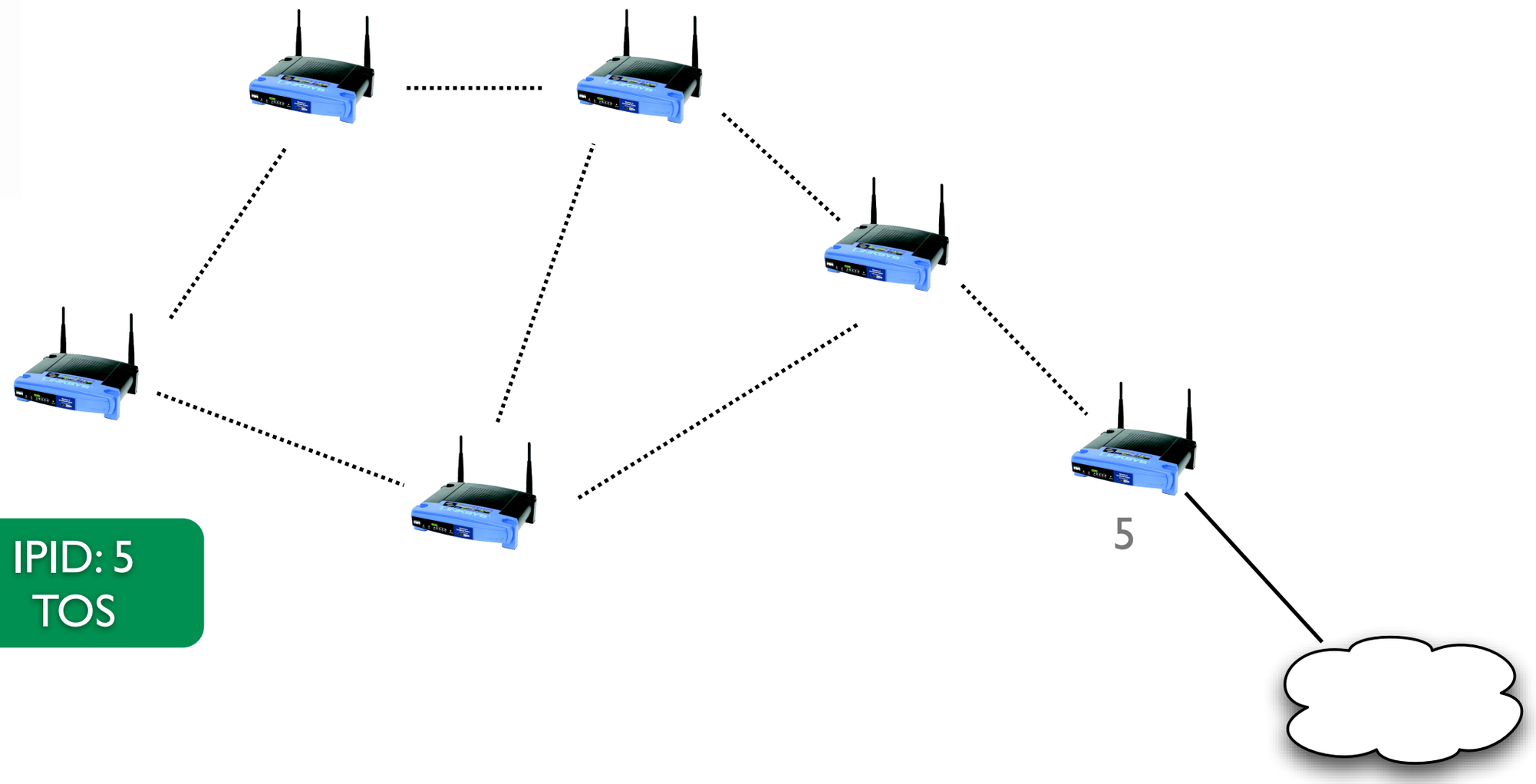
IPID: 5
TOS



5

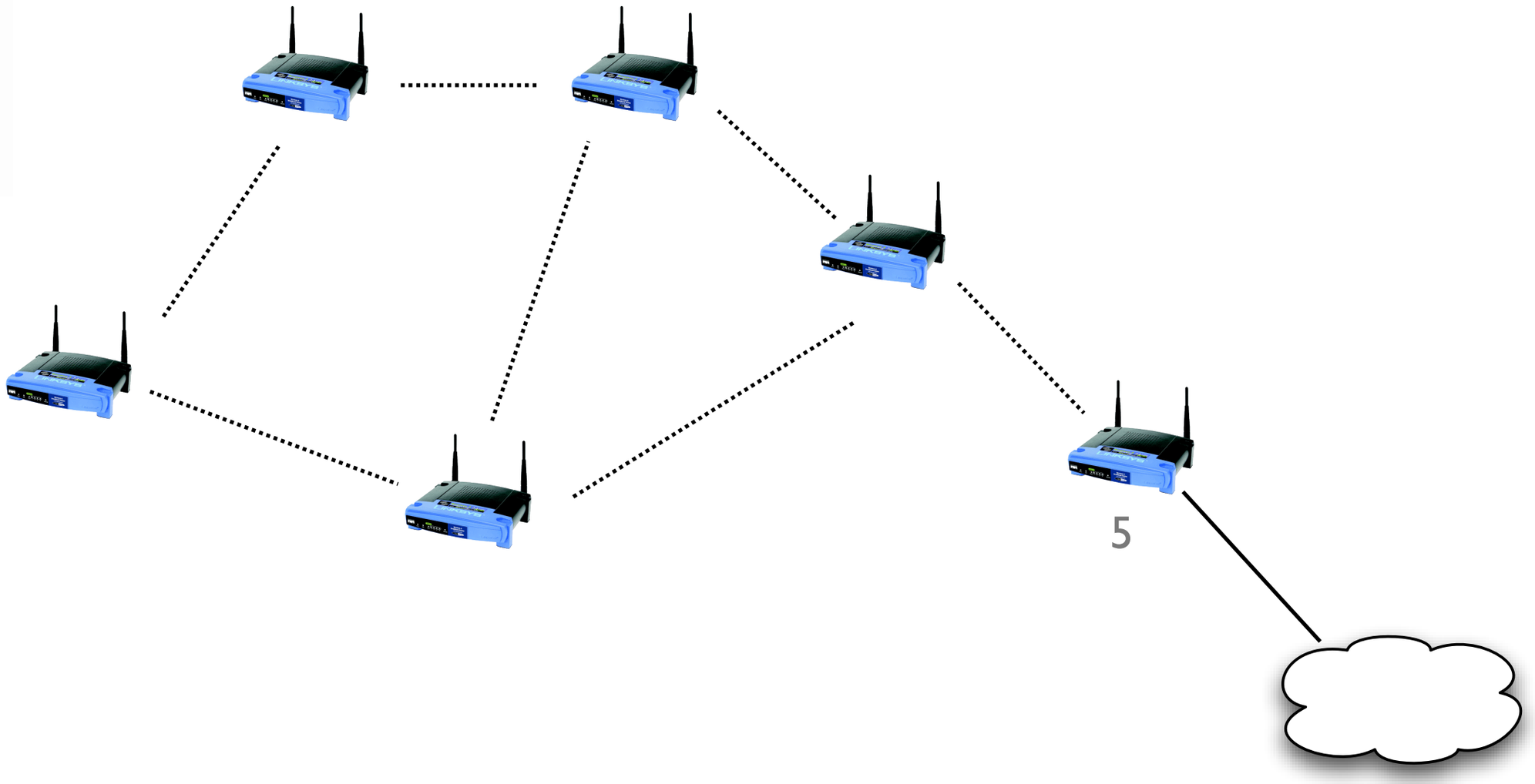
IPID: 5
TOS

IPID: 5
TOS



5

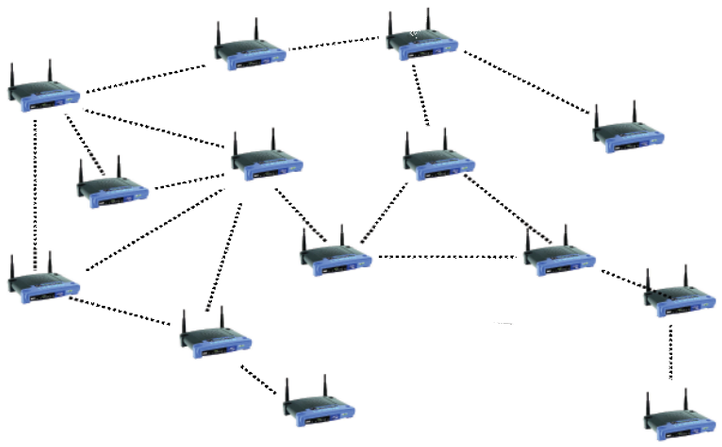
IPID: 5
TOS



Evaluation

One router

5 nodes
wireless
“line” setup

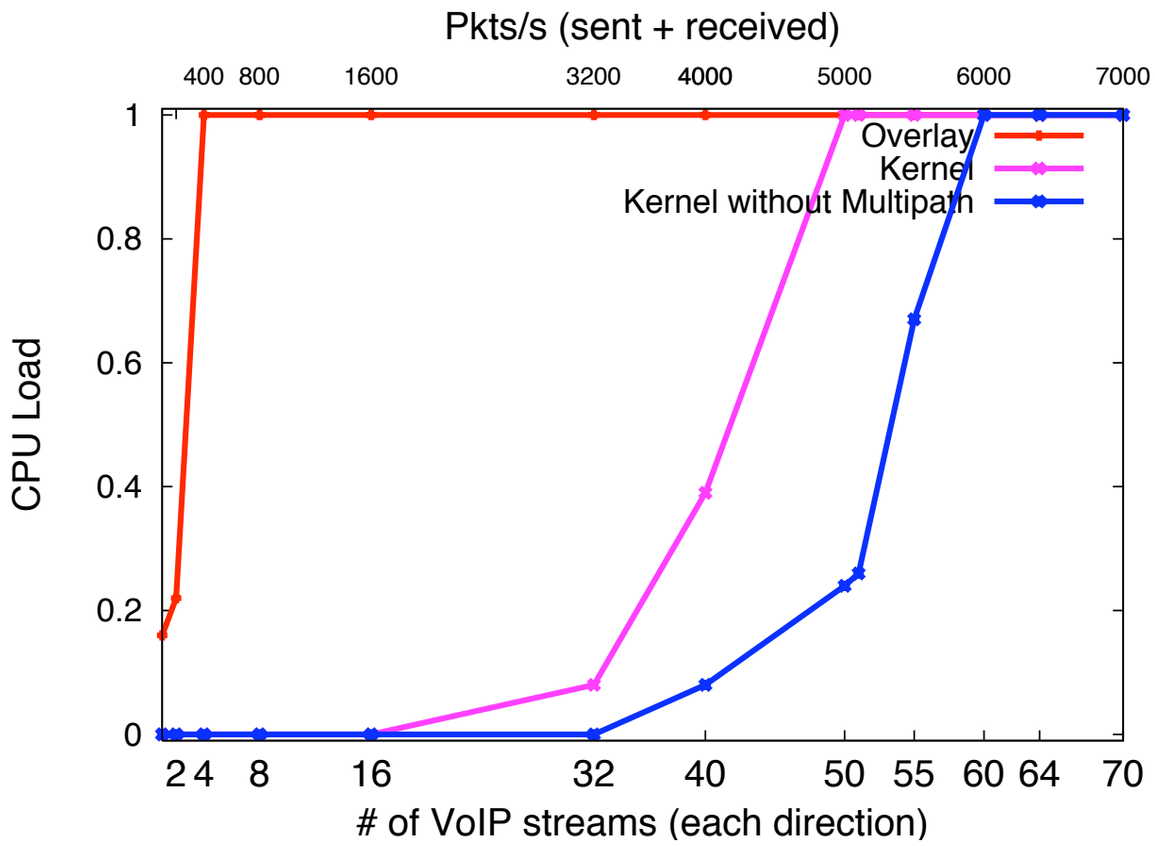


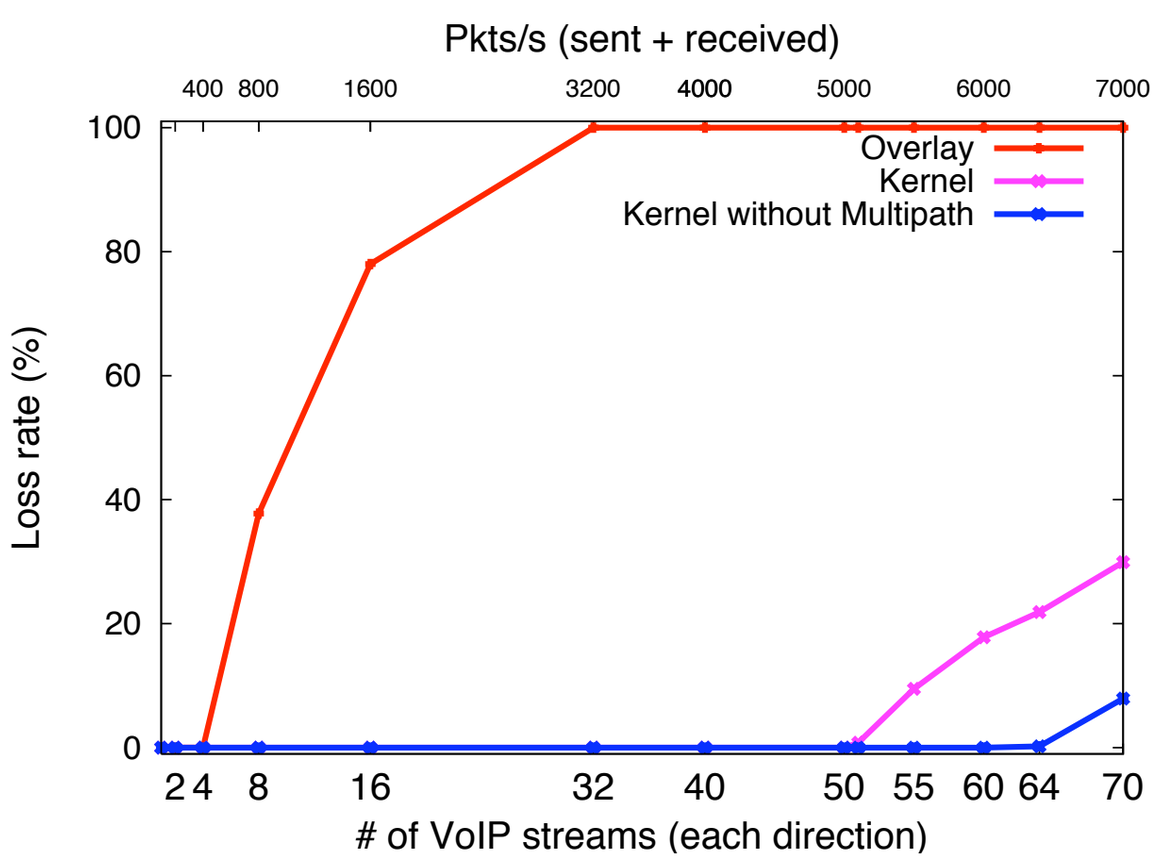
17 nodes
wireless
testbed

Rate	24 Mbps
Transmission power	50 mW
Retransmission limit	7
VoIP stream	64 Kbps

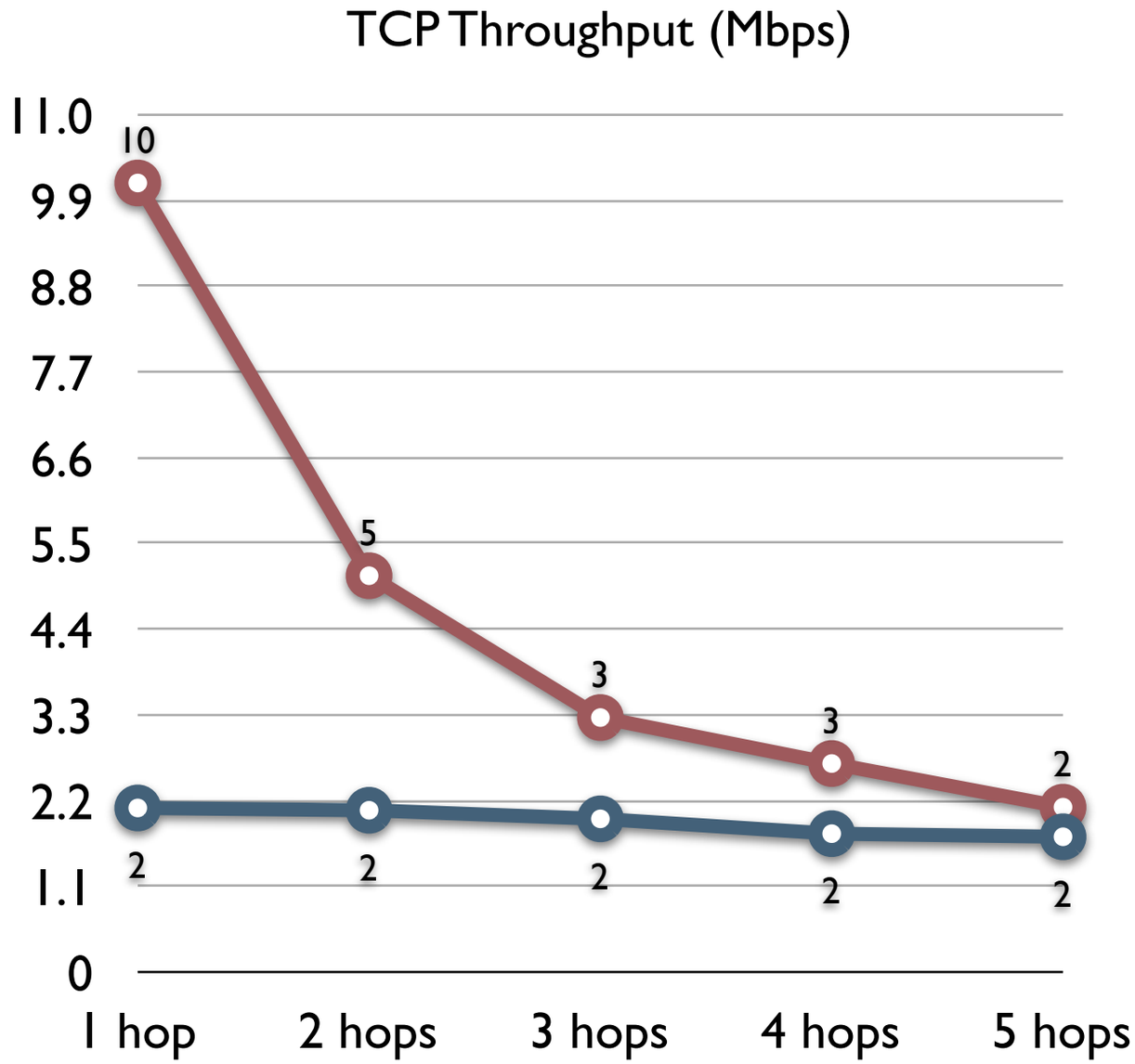
We route up to **50** duplex VoIP streams
before the CPU is saturated.

Overlay	4 streams	512 Kbps
Kernel	50 streams	6.4 Mbps
Native kernel routing	60 streams	7.6 Mbps





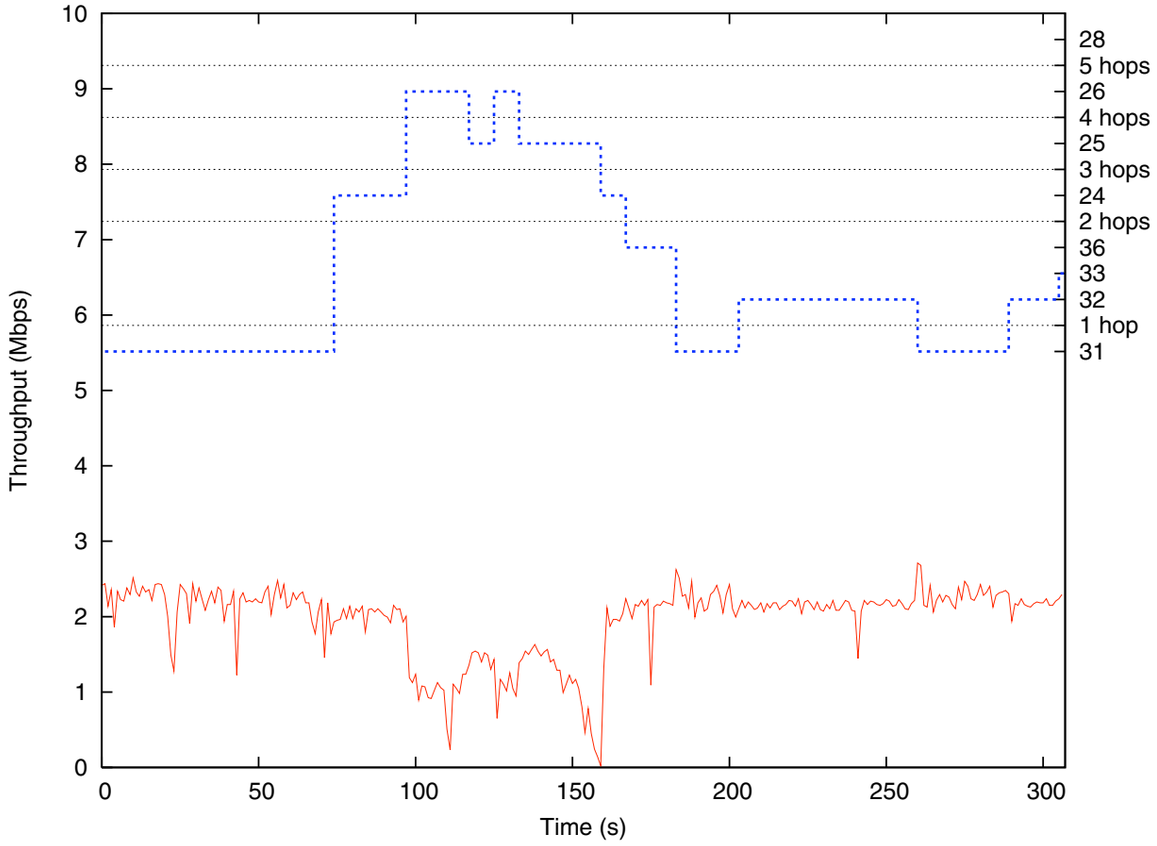
2



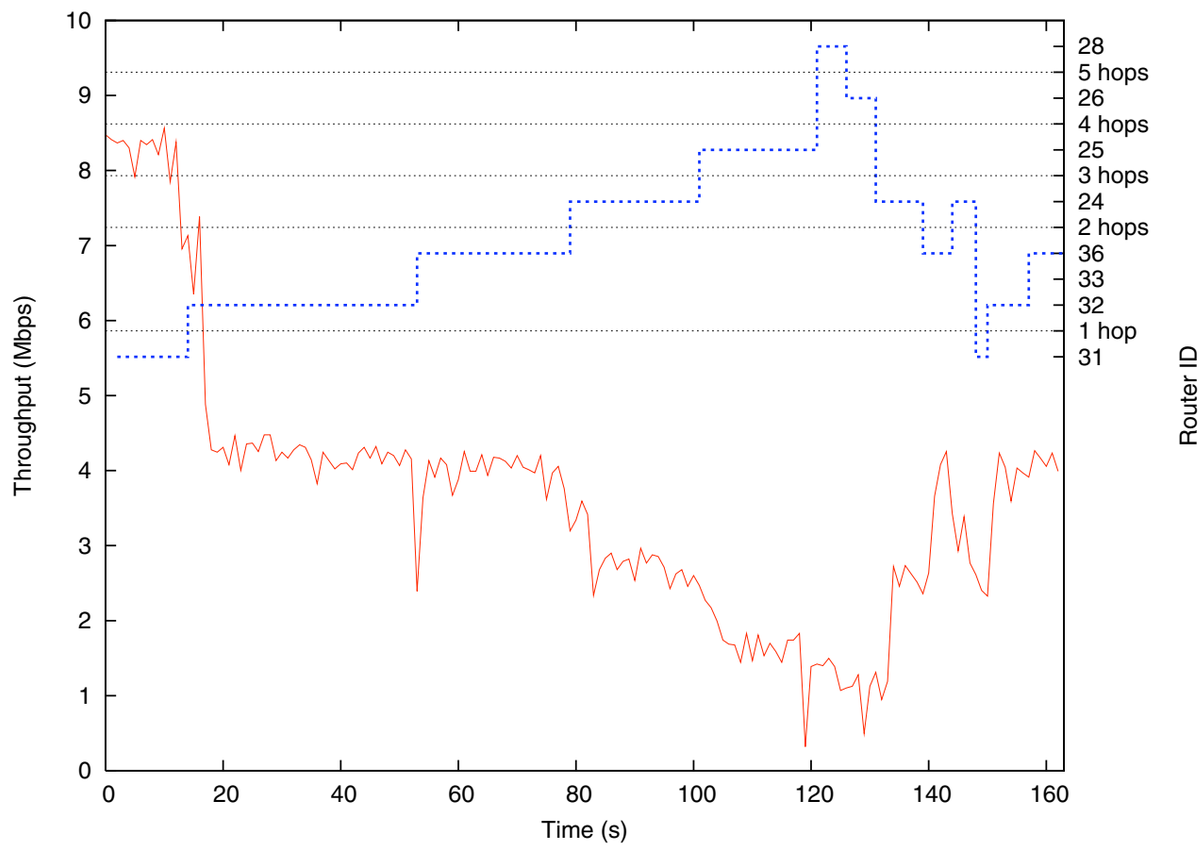
With **one** wireless hop, we get **10 Mbps** in a "line" setup.

3

Overlay



Kernel multipath



Results are close to the “line” topology

(8.5 Mbps for one hop).

Redundant multipath routing is important in WMN.

We can support it with minimal changes in Linux kernel.

SMesh is available as open-source at www.smesh.org.

Thanks

Questions