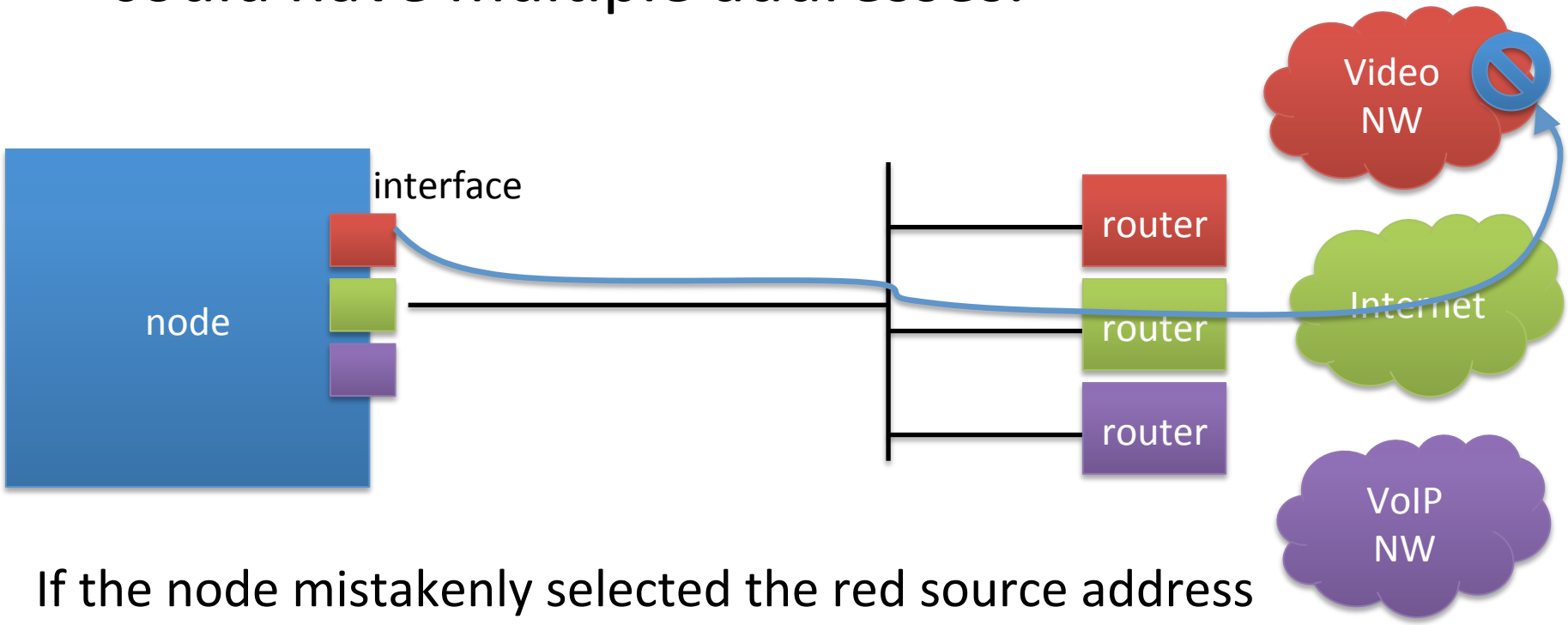
Day2 Morning Session 1 Agenda

- 10:00-11:00 Packet Filtering
  - Review: source address selection
  - router filter
  - home router security
  - bcp38 and source address validation

Multiple addresses and source address selection

- The design of IPv6 is that every node/interface could have multiple addresses.



If the node mistakenly selected the red source address to access Internet (green) the return packet will most likely not get back to the originating node

Multiple addresses and source address selection

- RFC 6724:Default Address Selection for Internet Protocol Version 6 (IPv6)
  - recently revised, but most existing implementation is still old RFC 3484

Prefix	Precedence	Label
::1/128	50	0
::/0	40	1
::ffff:0:0/96	35	4
2002::/16	30	2
2001::/32	5	5
fc00::/7	3	13
::/96	1	3
fec0::/10	1	11
3ffe::/16	1	12

RFC6724

Prefix	Precedence	Label
::1/128	50	0
::/0	40	1
2002::/16	30	2
::/96	20	3
::ffff:0:0/96	10	4

RFC3484

default source selection policy

1. Prefer same address
   - Use 2001:db8::1 to communicate with 2001:db8::1 instead of fe80::1
2. Prefer appropriate scope
   - Use fe80::1 to communicate with fe80::2 instead of 2001:db8::2
3. Avoid deprecated addresses
   - SLAAC renews IPs. Prefer addresses that are “preferred”
4. Prefer home addresses
   - Prefer home address over care/of address
5. Prefer outgoing interface
   - If outgoing interface is specified(or set) for destination, use that interface’s address

default source selection policy

5.5 Prefer addresses in a prefix advertised by the next-hop

- Most IPv6 implementations do not track which prefix is advertised by which router

6. Prefer matching label
   - Use 6to4 address to communicate with 6to4 destinations, etc
7. Prefer temporary addresses
   - Prefer privacy addresses over static addresses
8. Use longest matching prefix
   - works like a router

[review] Privacy extensions and corporate administration

- If you proxy all communications to/from the internet, privacy extensions may be a security risk
  - it becomes hard to locate which node is compromised
- If you do not, privacy extensions will provide some security, but internal accountability is a different question
- If you are a system admin for a corporate network, this is something you have to take into consideration
  - use DHCP to enforce policy? external methods?

Hint:
Privacy extensions and Cryptographically Generated Addresses(CGA) are NOT the same thing. They are both randomly generated identifiers aimed towards different goals. CGAs verify ownership of an address and prevent spoofing.

Considerations

- Where node privacy is not required, or network policy requires full IP address accountability, the options would be to
  - turn off privacy address
  - configure source address selection to prefer static address (this is allowed in RFC)
  - use DHCP to assign IP address, and also to distribute source address selection policy

Unfortunately, there is no Best Practice yet

Recommendation

- If possible, I would recommend avoiding multiple GUA prefix network architecture
- This simplifies troubleshooting, and is also better for filtering security
- If there is an absolute need, you will probably have to compromise (such as being restricted to specific vendors etc.), considering the current implementation status

Router filter examples

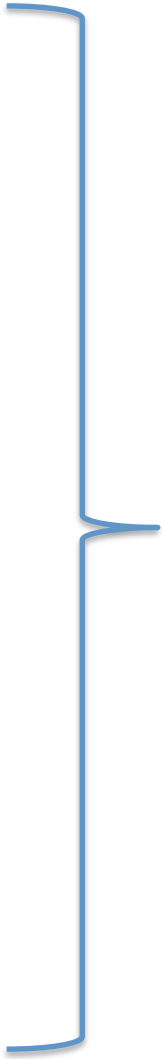
```
family inet6 {
    filter loopback-filter {
        term ospf {
            from {
                next-header ospf;
            }
            then accept;
        }
        term bgp {
            from {
                source-address {
                    [neighbor address]/128;
                }
                next-header tcp;
                destination-port bgp;
            }
            then accept;
        }
```

Allow routing protocol packets based on header type (or port)

example based on Juniper

Router filter examples

```
term snmp {
        from {
            source-address {
                [NOC IP];
            }
            next-header udp;
            destination-port snmp;
        }
        then accept;
    }
    term ntp {
        from {
            source-address {
                [NOC IP];
            }
            next-header udp;
            port ntp;
        }
        then accept;
    }
    term telnet {
        from {
            source-address {
                [NOC IP];
            }
            next-header tcp;
            destination-port telnet;
        }
        then accept;
    }
    term dns {
        from {
            source-address {
                [NOC IP];
            }
            next-header udp;
            source-port domain;
        }
        then accept;
    }
```



Allow management communication

example based on Juniper

Router filter examples

```
term tcp-established {
            from {
                next-header tcp;
                tcp-established;
            }
            then accept;
        }
        term icmp {
            from {
                next-header icmpv6;
            }
            then accept;
        }
        term fragment {
            from {
                next-header fragment;
            }
            then accept;
        }
        term default {
            then discard;
        }
    }
  }
}
```

Allow other headers essential for communication

- tcp-est
- icmp
- fragment

Than discard rest of packet directed toward router

example based on Juniper

Other Router filtering

- Previous example is applied to loopback on Juniper. With Cisco, it would be an infrastructure ACL
- Packets you have to discard for legal reasons
  - Usually applied to applicable interfaces
    - OP25B (spam protection)
    - access to illegal sites
- Restricting communication
  - Usually applied to applicable interfaces
    - source IP, dest IP, port numbers, etc.
- Security Features
  - BCP38
  - Protection against scanning (usually at edge devices)