# DSAV Framework: Validating Source Addresses via SAV Tables Generated by a Distributed Control-plane Protocol

Dan Li (Tsinghua University)

Jianping Wu (Tsinghua University)

Mingqing Huang (Huawei)

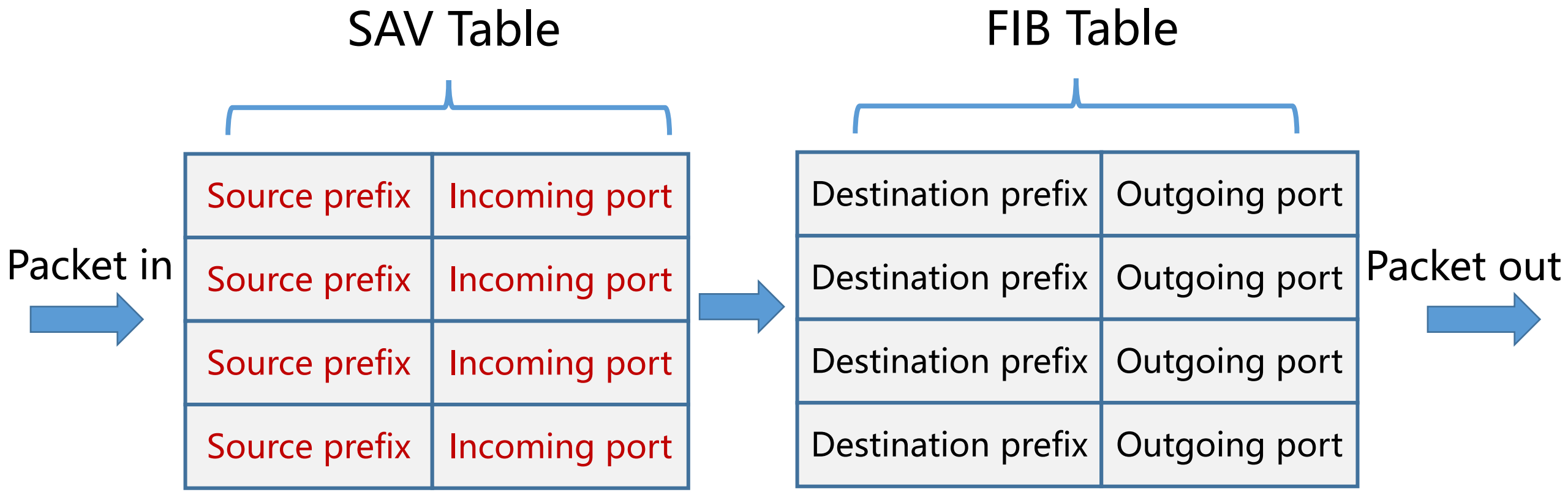
Lancheng Qin (Tsinghua University)

Nan Geng (Huawei)

# DSAV Motivation

- SAVI is not enough and intra-/inter- domain SAV is necessary
  - SAVI is only fully effective when deployed globally
  - However, it is impractical to expect all access networks to deploy SAVI simultaneously
  - If an access network does not deploy SAVI, spoofing traffic from it should have a chance to be blocked before arriving at the destination host (as close to the source as possible)
- Problem of existing intra-/inter- domain SAV technologies
  - uRPF-based technologies may lead to improper block or improper permit
- DSAV Framework
  - Depending on a distributed control-plane protocol to generate accurate SAV table, instead of using uRPF
  - Improving the scalability of the protocol by limiting the computation/communication overhead

# SAV Table in DSAV Routers



# DSAV Protocol to Generate SAV Tables

- Basic idea of DSAV protocol
  - Discovering the real data-plane forwarding path via hop-by-hop prefix notification, and generating SAV tables in routers along the path
  - Separating the protocol into an intra-domain part and an inter-domain part, both sharing the same high-level idea
- Terminologies
  - Node: A router in intra-domain DSAV or an AS in inter-domain DSAV
  - Prefix notification: The process by which a node notifies the incoming direction of its source prefixes to all the other nodes in the network
  - During prefix notification, each node conducts one of the three operations
    - Message origination: A node generates original notification messages
    - Message relaying: A node generates relaying notification messages after receiving a notification message
    - Message termination: A node terminates the received notification message

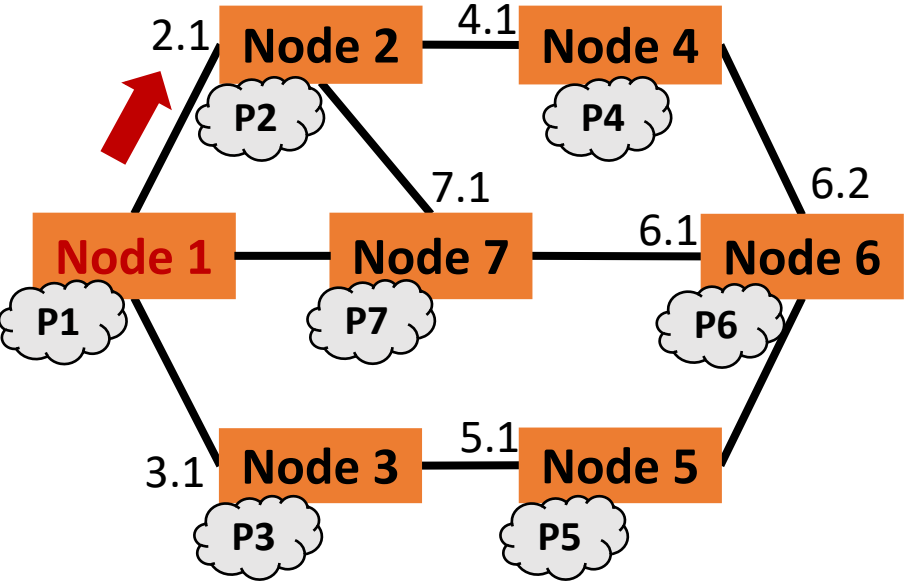
# DSAV Notification Message Format

The DSAV notification message contains two main fields

- Source prefix field
  - This field contains the source prefixes of the initial node
  - When receiving a message, the node generates SAV rules for the source prefixes
  - This field remains unchanged during the prefix notification process
- Propagation scope field
  - This field contains a list of destination prefixes which take the neighboring node as the next hop (from FIB)
  - This field is used to discover the real data-plane forwarding path
  - This field changes hop by hop during the prefix notification process

# An Example of DSAV Workflow (1)

| FIB for Node 1 | |
|---|---|
| Dest Prefix | Next hop |
| P2 | Node 2 |
| P3 | Node 3 |
| P4 | Node 2 |
| P5 | Node 3 |
| P6 | Node 2 |
| P7 | Node 2 |



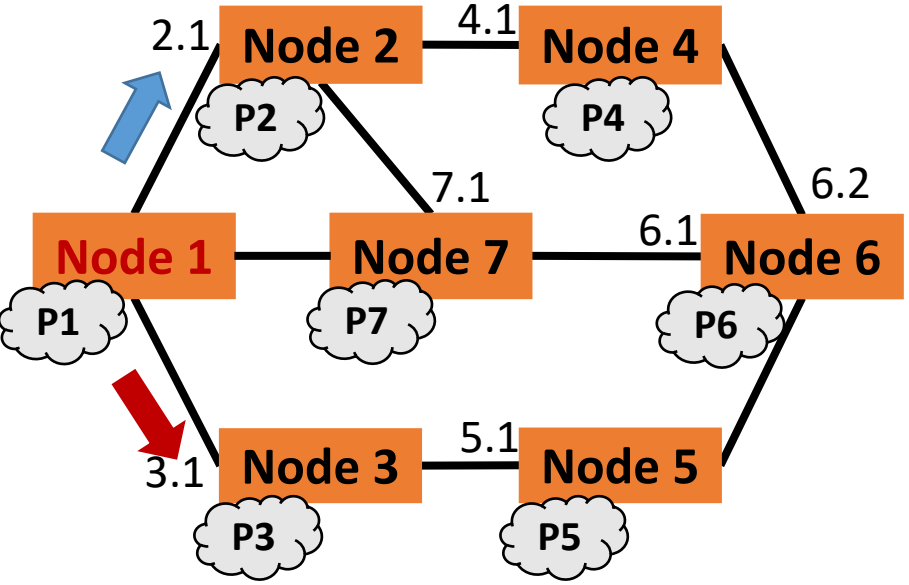

The process of prefix notification for P1

Node 1 conducts message origination since P1 is the source prefix of Node 1

- From Node 1's FIB, P2, P4, P6, P7 take Node 2 as the next hop, so Node 1 generates an original notification message to Node 2
  - Message from Node 1 to Node 2
    - Source prefix → P1
    - Propagation scope → P2, P4, P6, P7

# An Example of DSAV Workflow (1)

| FIB for Node 1 | |
|---|---|
| Dest Prefix | Next hop |
| P2 | Node 2 |
| P3 | Node 3 |
| P4 | Node 2 |
| P5 | Node 3 |
| P6 | Node 2 |
| P7 | Node 2 |



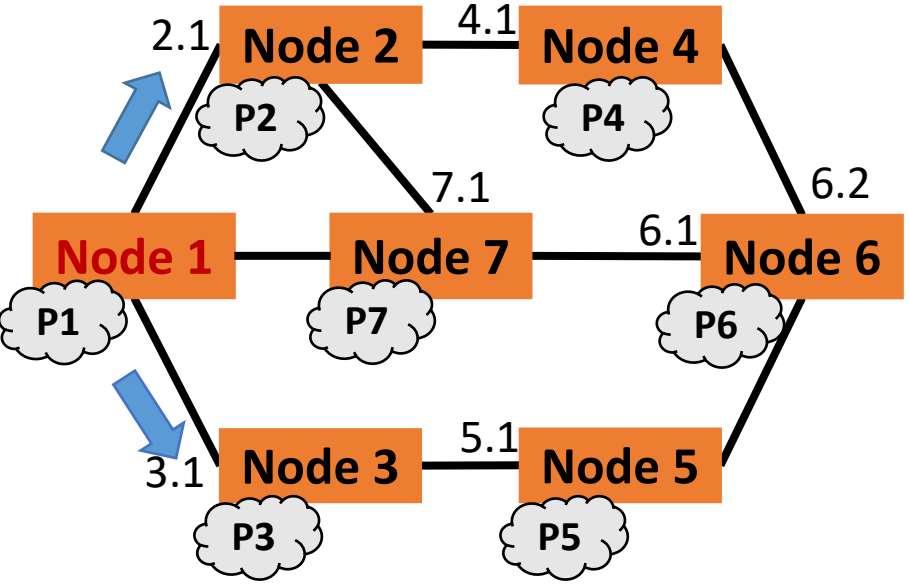

The process of prefix notification for P1

Node 1 conducts message origination since P1 is the source prefix of Node 1

- From Node 1's FIB, P3, P5 take Node 3 as the next hop, so Node 1 generates an original notification message to Node 3
  - Message from Node 1 to Node 3
    - Source prefix → P1
    - Propagation scope → P3, P5

# An Example of DSAV Workflow (1)

| FIB for Node 1 | |
|---|---|
| Dest Prefix | Next hop |
| P2 | Node 2 |
| P3 | Node 3 |
| P4 | Node 2 |
| P5 | Node 3 |
| P6 | Node 2 |
| P7 | Node 2 |



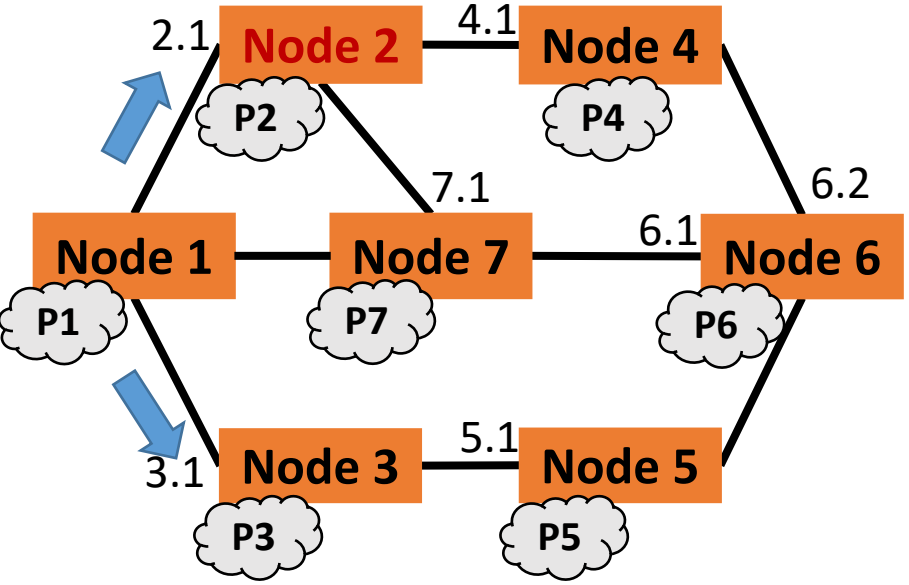

The process of prefix notification for P1

Node 1 conducts message origination since P1 is the source prefix of Node 1

- From Node 1's FIB, no prefix takes Node 7 as the next hop, so Node 1 does not send any notification message to Node 7

# An Example of DSAV Workflow (2)

| FIB for Node 2 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 1 |
| P3 | Node 1 |
| P4 | Node 4 |
| P5 | Node 4 |
| P6 | Node 4 |
| P7 | Node 7 |



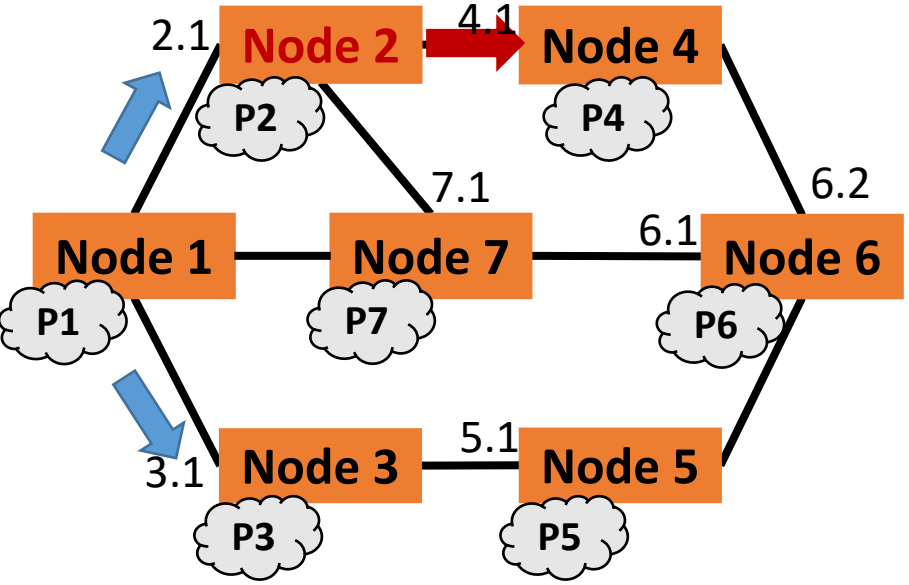

The process of prefix notification for P1

When Node 2 receives the message from Node 1 at port 2.1

- ◆Message from Node 1 to Node 2
  - ➢Source prefix → P1
  - ➢Propagation scope → P2, P4, P6, P7
- ◘Node 2 generates the SAV rule for source prefix P1
  - ◆<source prefix P1, incoming port 2.1>

# An Example of DSAV Workflow (2)

| FIB for Node 2 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 1 |
| P3 | Node 1 |
| P4 | Node 4 |
| P5 | Node 4 |
| P6 | Node 4/7 |
| P7 | Node 7 |




The process of prefix notification for P1

When Node 2 receives the message from Node 1 at port 2.1

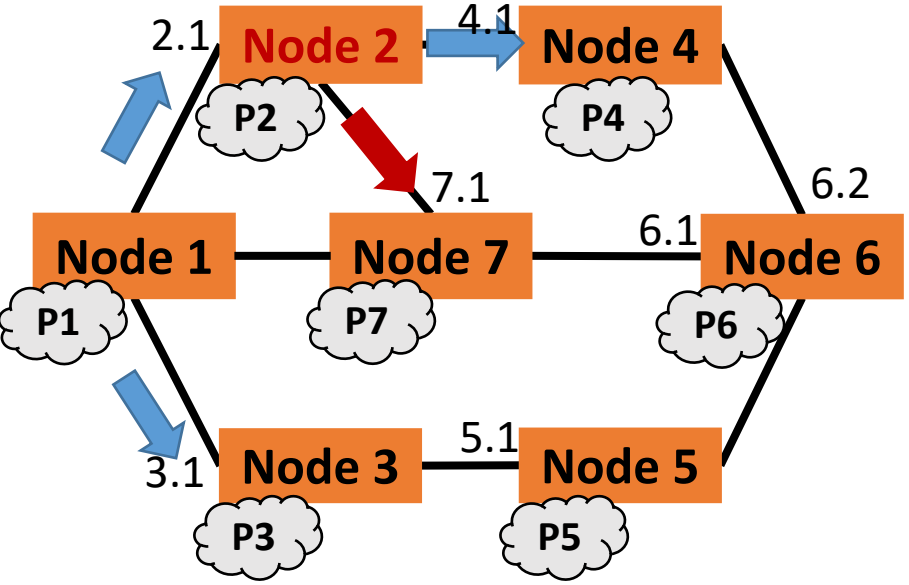
- ◆Message from Node 1 to Node 2
  - ➢Source prefix → P1
  - ➢Propagation scope → P2, P4, P6, P7

□From Node 2's FIB, P4, P6 take Node 4 as the next hop, so Node 2 conducts message relaying and generates a relaying notification message to Node 4

- ◆Message from Node 2 to Node 4
  - ➢Source prefix → P1
  - ➢Propagation scope → P4, P6

# An Example of DSAV Workflow (2)

| FIB for Node 2 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 1 |
| P3 | Node 1 |
| P4 | Node 4 |
| P5 | Node 4 |
| P6 | Node 4/7 |
| P7 | Node 7 |




The process of prefix notification for P1

When Node 2 receives the message from Node 1 at port 2.1

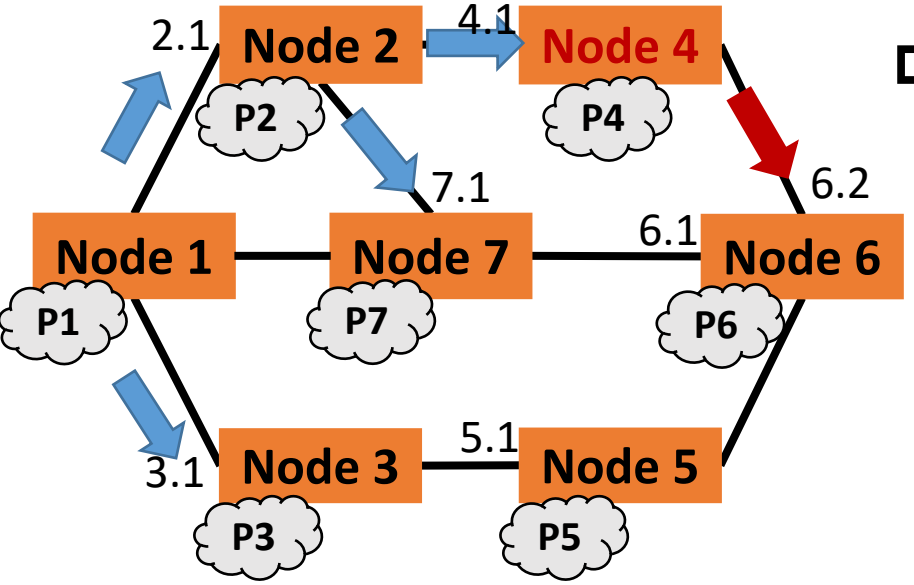
- ◆Message from Node 1 to Node 2
  - ➢Source prefix → P1
  - ➢Propagation scope → P2, P4, P6, P7

□From Node 2's FIB, P6, P7 take Node 7 as the next hop, so Node 2 conducts message relaying and generates a relaying notification message to Node 7

- ◆Message from Node 2 to Node 7
  - ➢Source prefix → P1
  - ➢Propagation scope → P6, P7

# An Example of DSAV Workflow (3)

| FIB for Node 4 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 2 |
| P2 | Node 2 |
| P3 | Node 2 |
| P5 | Node 6 |
| P6 | Node 6 |
| P7 | Node 2 |




The process of prefix notification for P1

When Node 4 receives the message from Node 2 at port 4.1

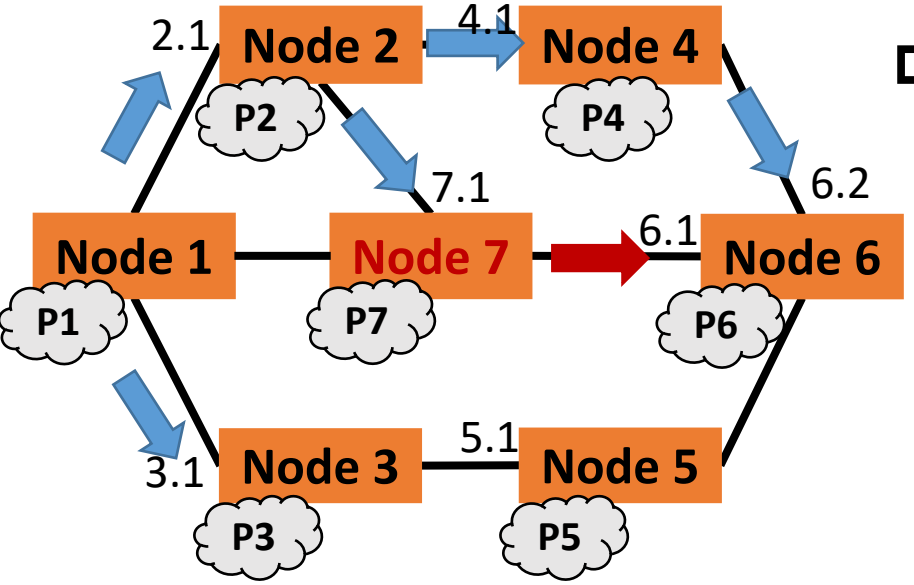
- Message from Node 2 to Node 4
  - Source prefix → P1
  - Propagation scope → P4, P6

- Node 4 generates the SAV rule for source prefix P1
  - <source prefix P1, incoming port 4.1>

- From Node 4's FIB, P6 takes Node 6 as the next hop, so Node 4 conducts message relaying and generates a relaying notification message to Node 6
  - Message from Node 4 to Node 6
    - Source prefix → P1
    - Propagation scope → P6

# An Example of DSAV Workflow (4)

| FIB for Node 7 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 1 |
| P2 | Node 2 |
| P3 | Node 1 |
| P4 | Node 2 |
| P5 | Node 6 |
| P6 | Node 6 |




The process of prefix notification for P1

When Node 7 receives the message from Node 2 at port 7.1

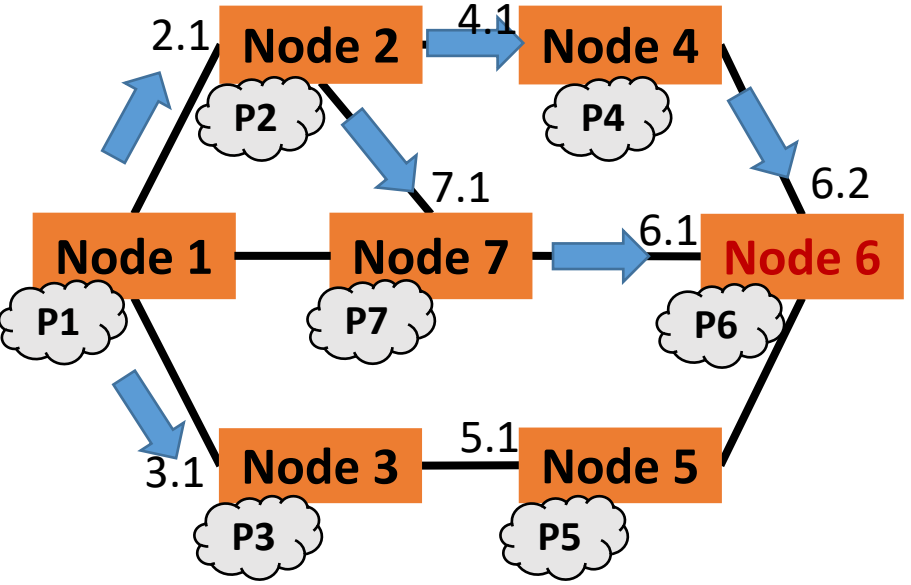
- Message from Node 2 to Node 7
  - Source prefix → P1
  - Propagation scope → P6, P7

- Node 7 generates the SAV rule for source prefix P1
  - <source prefix P1, incoming port 7.1>

- From Node 7's FIB, P6 takes Node 6 as the next hop, so Node 7 conducts message relaying and generates a relaying notification message to Node 6
  - Message from Node 7 to Node 6
    - Source prefix → P1
    - Propagation scope → P6

# An Example of DSAV Workflow (5)

| FIB for Node 4 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 2 |
| P2 | Node 2 |
| P3 | Node 2 |
| P5 | Node 6 |
| P6 | Node 6 |
| P7 | Node 2 |




The process of prefix notification for P1

When Node 6 receives the message from Node 4 at port 6.2 and the message from Node 7 at port 6.1

- Message from Node 4 to Node 6
  - Source prefix → P1
  - Propagation scope → P6
- Message from Node 7 to Node 6
  - Source prefix → P1
  - Propagation scope → P6

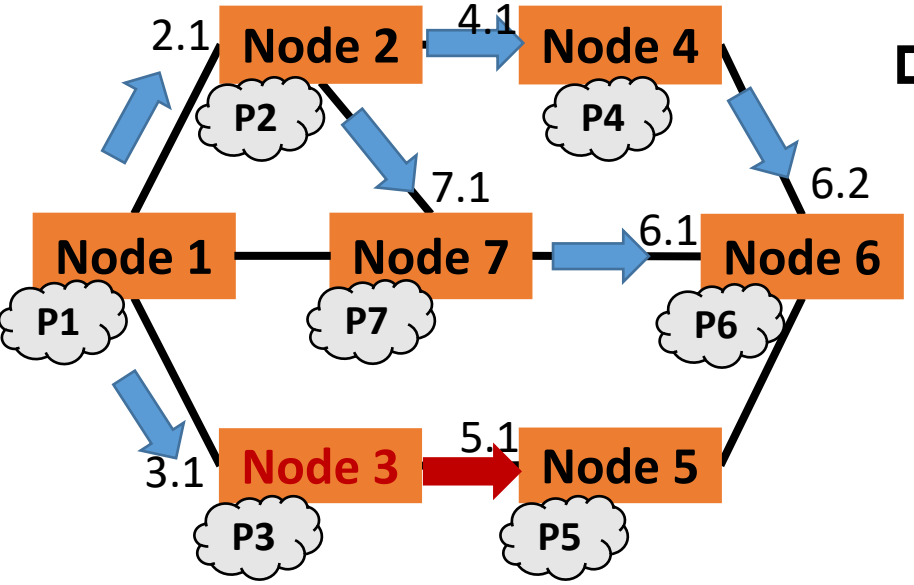
Node 6 generates the SAV rule for source prefix P1

- <source prefix P1, incoming port 6.1 and 6.2>

Node 6 conducts message termination because P6 is the source prefix of Node 6

# An Example of DSAV Workflow (6)

| FIB for Node 3 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 1 |
| P2 | Node 1 |
| P4 | Node 5 |
| P5 | Node 5 |
| P6 | Node 5 |
| P7 | Node 1 |




The process of prefix notification for P1

When Node 3 receives the message from Node 2 at port 3.1

- ◆Message from Node 2 to Node 3
  - ➢Source prefix → P1
  - ➢Propagation scope → P3, P5

□Node 3 generates the SAV rule for source prefix P1

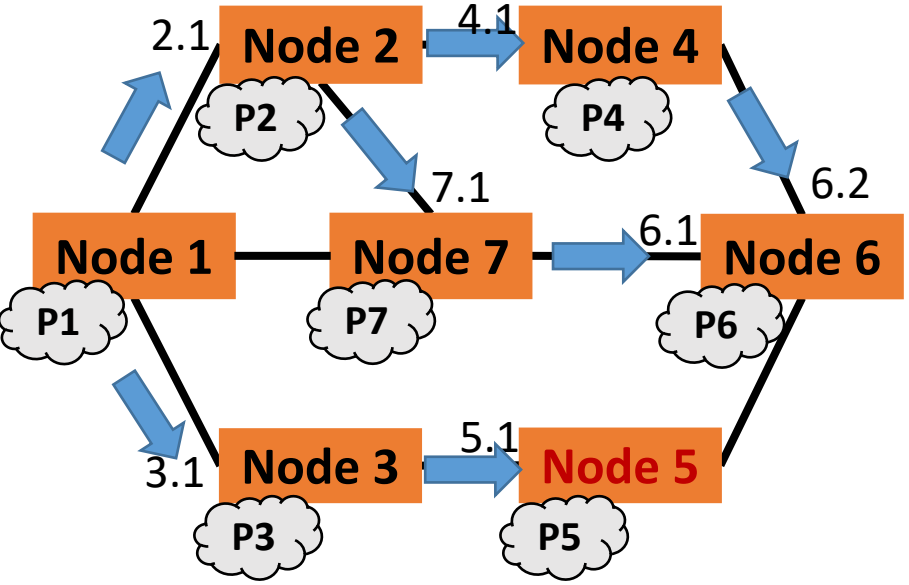
- ◆<source prefix P1, incoming port 3.1>

□From Node 3's FIB, P5 takes Node 5 as the next hop, so Node 3 conducts message relaying and generates a relaying notification message to Node 5

- ◆Message from Node 3 to Node 5
  - ➢Source prefix → P1
  - ➢Propagation scope → P5

# An Example of DSAV Workflow (7)

| FIB for Node 3 | |
|---|---|
| Dest Prefix | Next hop |
| P1 | Node 1 |
| P2 | Node 1 |
| P4 | Node 5 |
| P5 | Node 5 |
| P6 | Node 5 |
| P7 | Node 1 |




The process of prefix notification for P1

When Node 5 receives the message from Node 3 at port 5.1

- ◆Message from Node 3 to Node 5
  - ➢Source prefix → P1
  - ➢Propagation scope → P5

- ◘Node 5 generates the SAV rule for source prefix P1
  - ◆<source prefix P1, incoming port 5.1>
- ◘Node 5 conducts message termination because P5 is the source prefix of Node 5

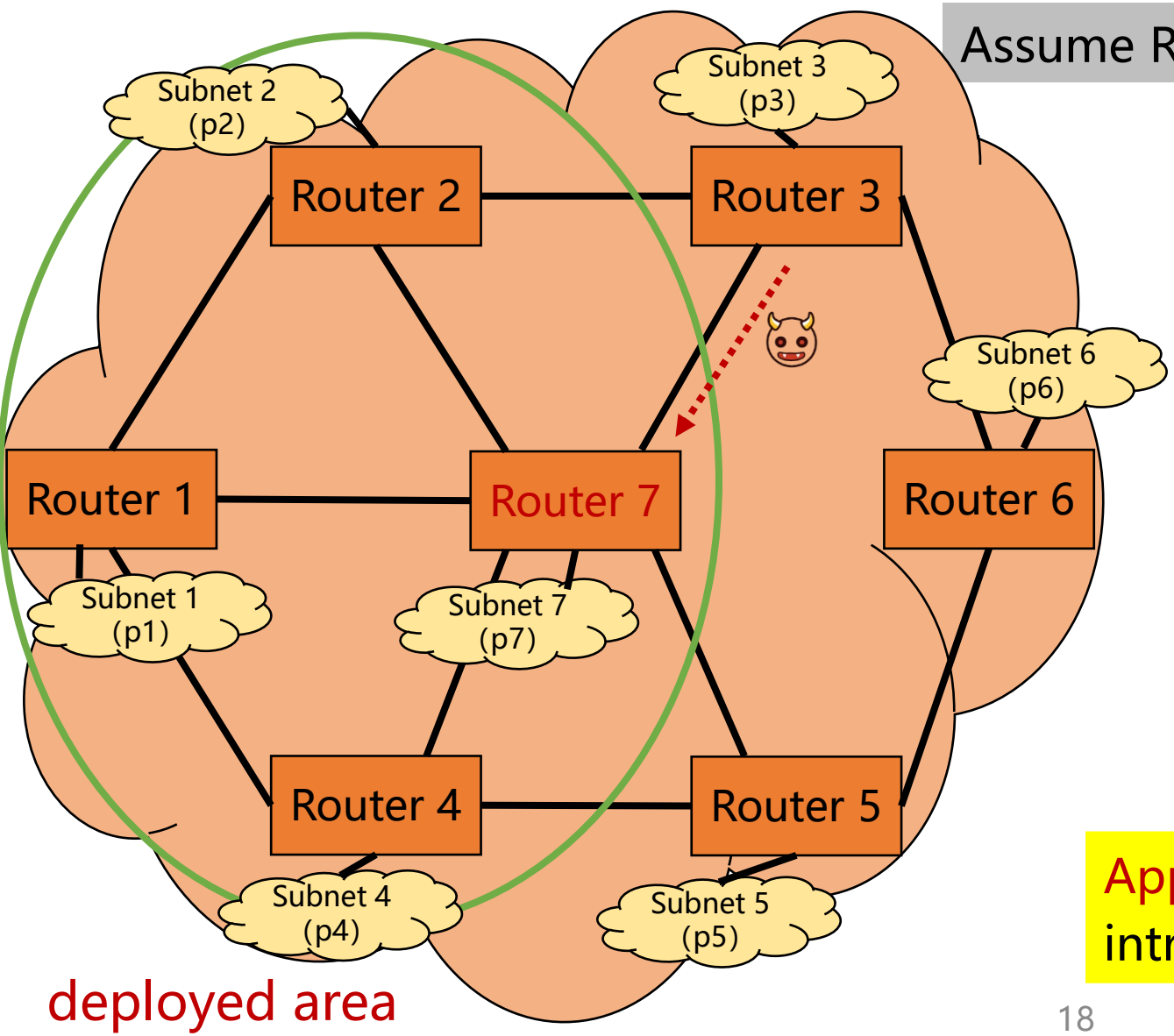
During the prefix notification, each node generates accurate SAV rules for P1 and receives only one message except for multi-path routing

# DSAV Update

- Periodic update
  - Each initial node generates protocol messages periodically
- Triggered update
  - When routing state changes, the initial node generates protocol messages to add updated SAV rules or delete outdated SAV rules for the affected nodes

We suggest intra-domain DSAV supports both periodic update and triggered update, while inter-domain DSAV only supports triggered update

# uRPF's Improper Permit in Intra-domain SAV

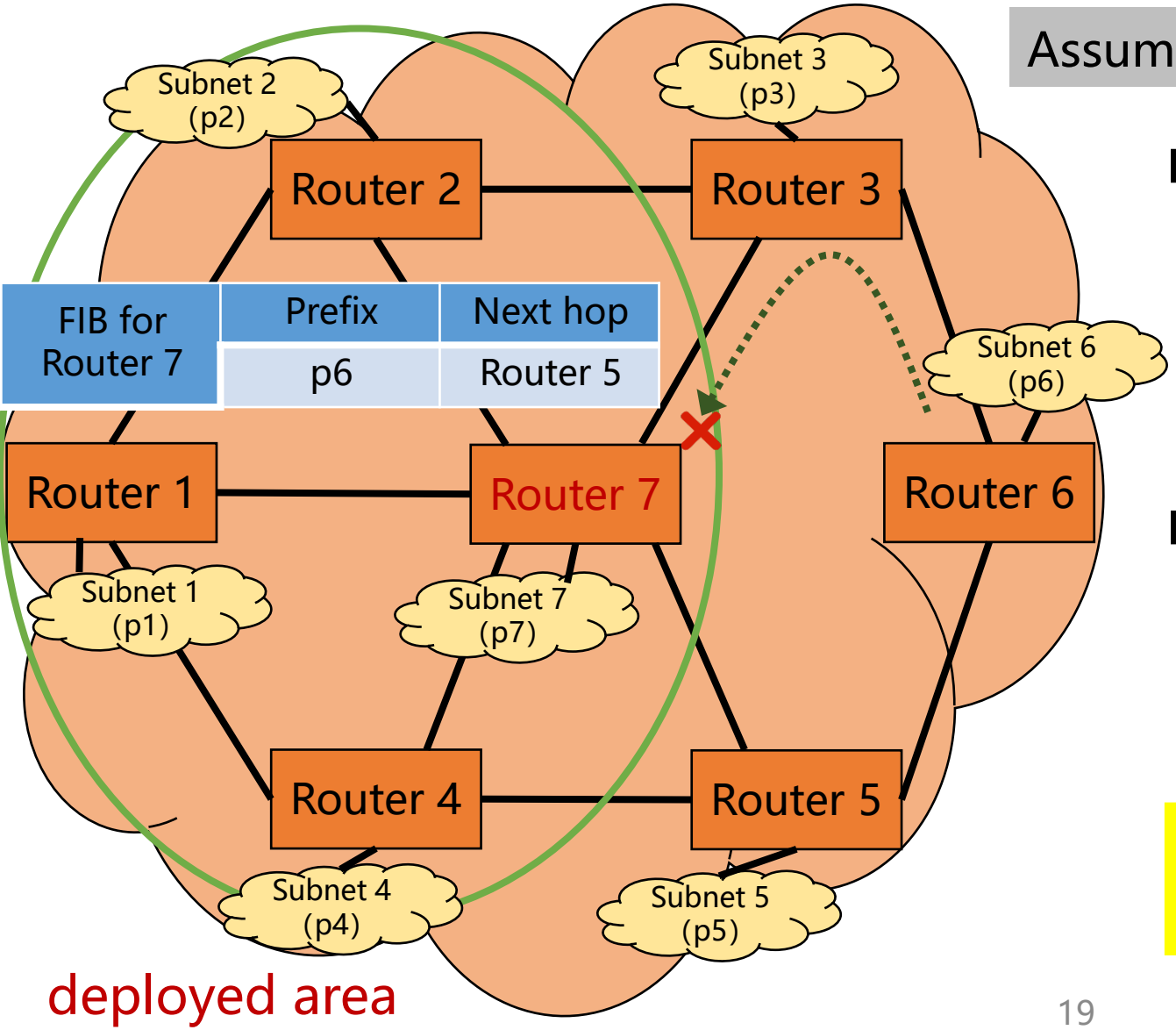



Assume Router 7 applies strict-uRPF only at subnet port

- If all the other routers make the same deployment
  - No problem
- If only Router 1,2,4,7 make the same deployment, there will be problem
  - When Router 3 sends packets to Router 7 by spoofing the source addresses of p1, p2, p4, Router 7 will improperly permit the packets
  - Subnets in the undeployed area can spoof the source addresses of the deployed area

Applying strict-uRPF only at subnet port in intra-domain SAV has improper permit problem.

# uRPF's Improper Block in Intra-domain SAV

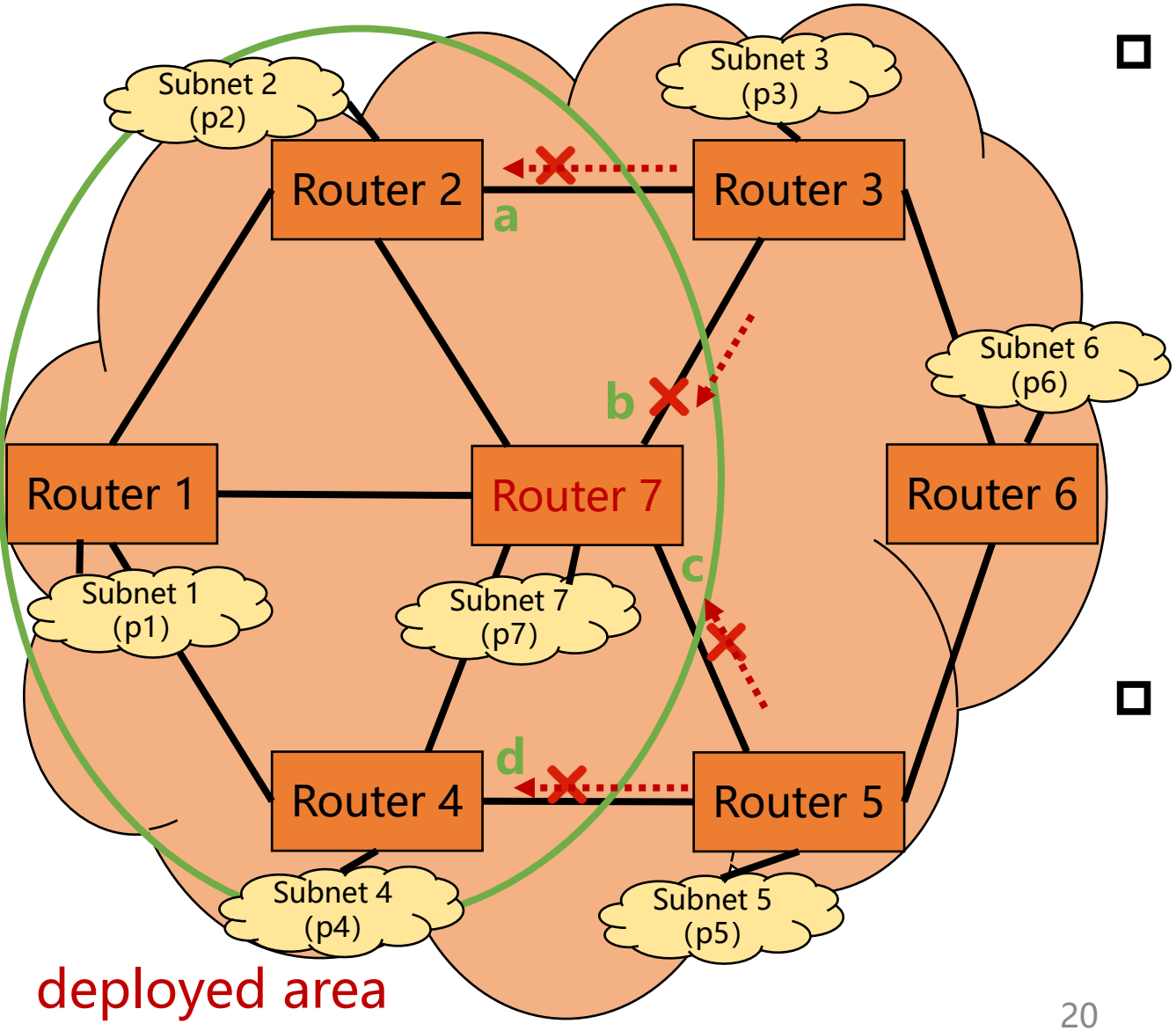



Assume Router 7 applies strict-uRPF at all ports

- If there is asymmetric routing
  - The routing path from Router 7 to Router 6 is Router 7 -> Router 5 -> Router 6
  - The routing path from Router 6 to Router 7 is Router 6 -> Router 3 -> Router 7
- The problem
  - When Router 6 sends valid packets to Router 7 through Router 3, Router 7 will improperly block the packets

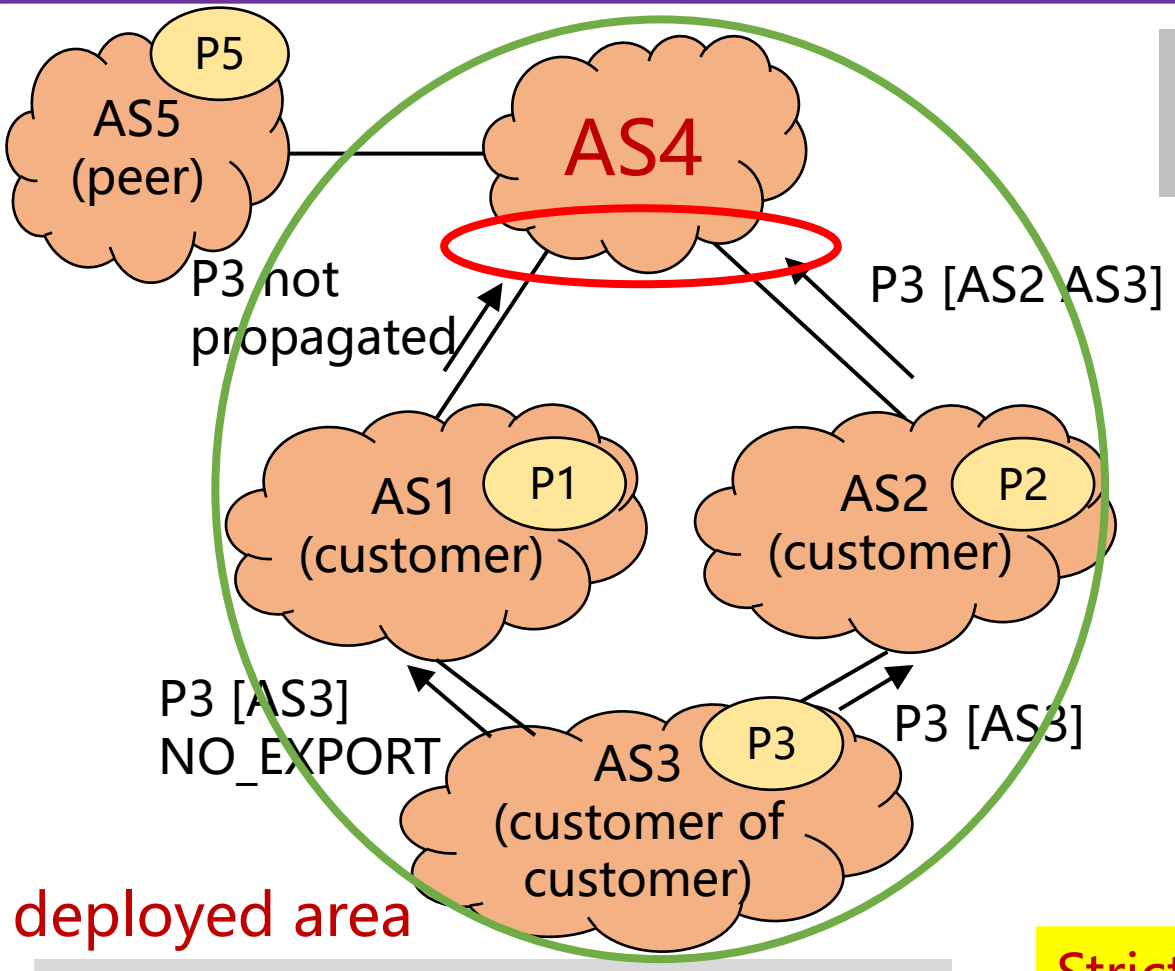
Applying strict-uRPF at all ports in intra-domain SAV has improper block problem.

# Benefit of Intra-domain DSAV Compared with uRPF




- Router 1, 2, 4, and 7 run DSAV
  - Subnet 1, 2, 4, 7 cannot spoof each other
  - Packets from subnet 3, 5, 6 with spoofed source addresses of p1, p2, p4, p7 will be accurately blocked at port a, b, c, d (while strict uRPF may have improper permit problem)
  - Legitimate packets from subnet 3, 5, 6 will be accurately permitted at port a, b, c, d (while strict uRPF may have improper block problem)
- Intra-domain DSAV
  - Subnets within the deployed area cannot spoof each other
  - Subnets in the undeployed area cannot spoof the source addresses of the deployed area

# uRPF's Improper Block in Inter-domain SAV



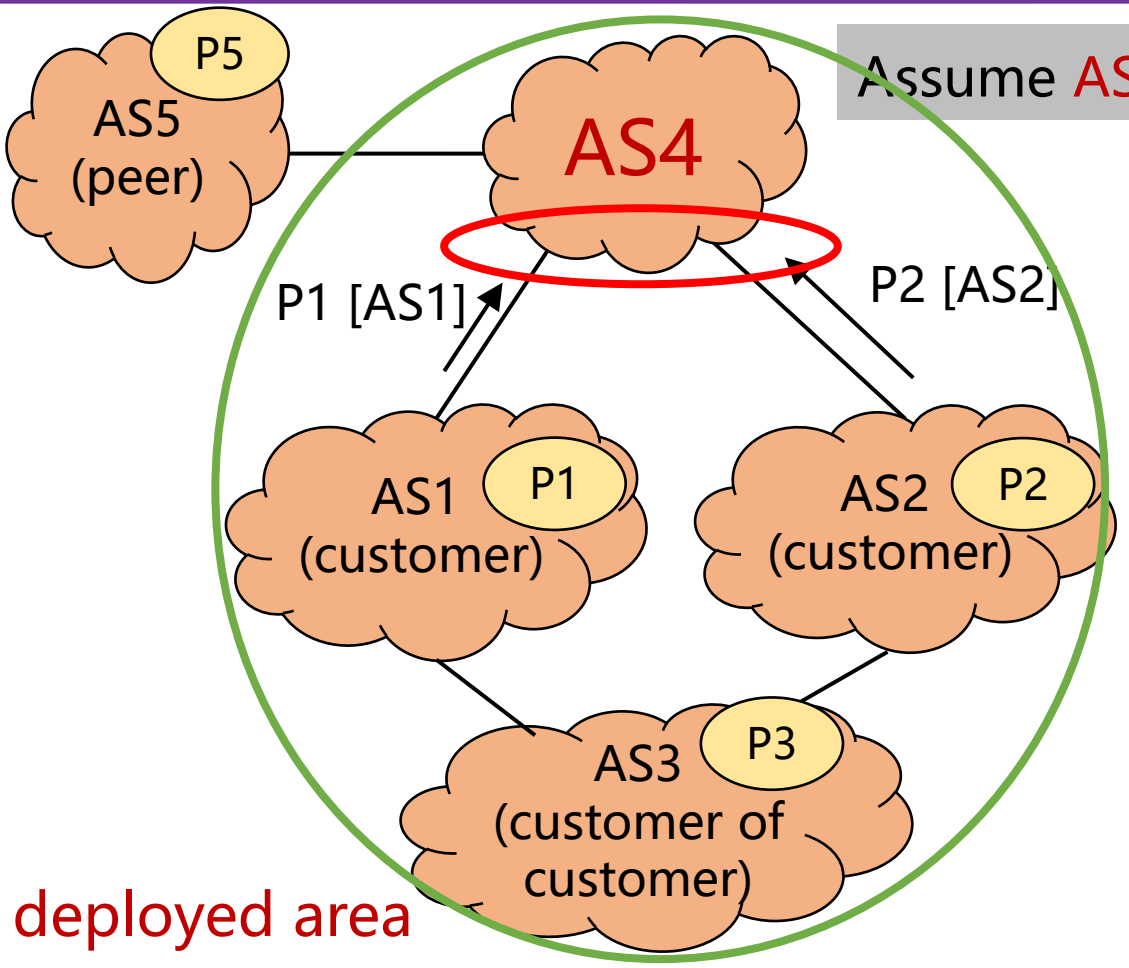

Due to the NO_EXPORT community, route for P3 is not propagated along the path of AS3->AS1->AS4.

Assume AS4 runs strict-uRPF / feasible-uRPF / EFP-uRPF (with Algorithm A) at customer ports

- The SAV rule at AS4's customer ports
  - Packets with source addresses of P3 can only arrive from AS2
- The problem
  - When AS3 sends packets with valid source addresses to AS4 through AS1, AS4 will improperly block these packets

Strict-uRPF / feasible-uRPF / EFP-uRPF (with Algorithm A) in inter-domain SAV has improper block problem.

# uRPF's Improper Permit in Inter-domain SAV



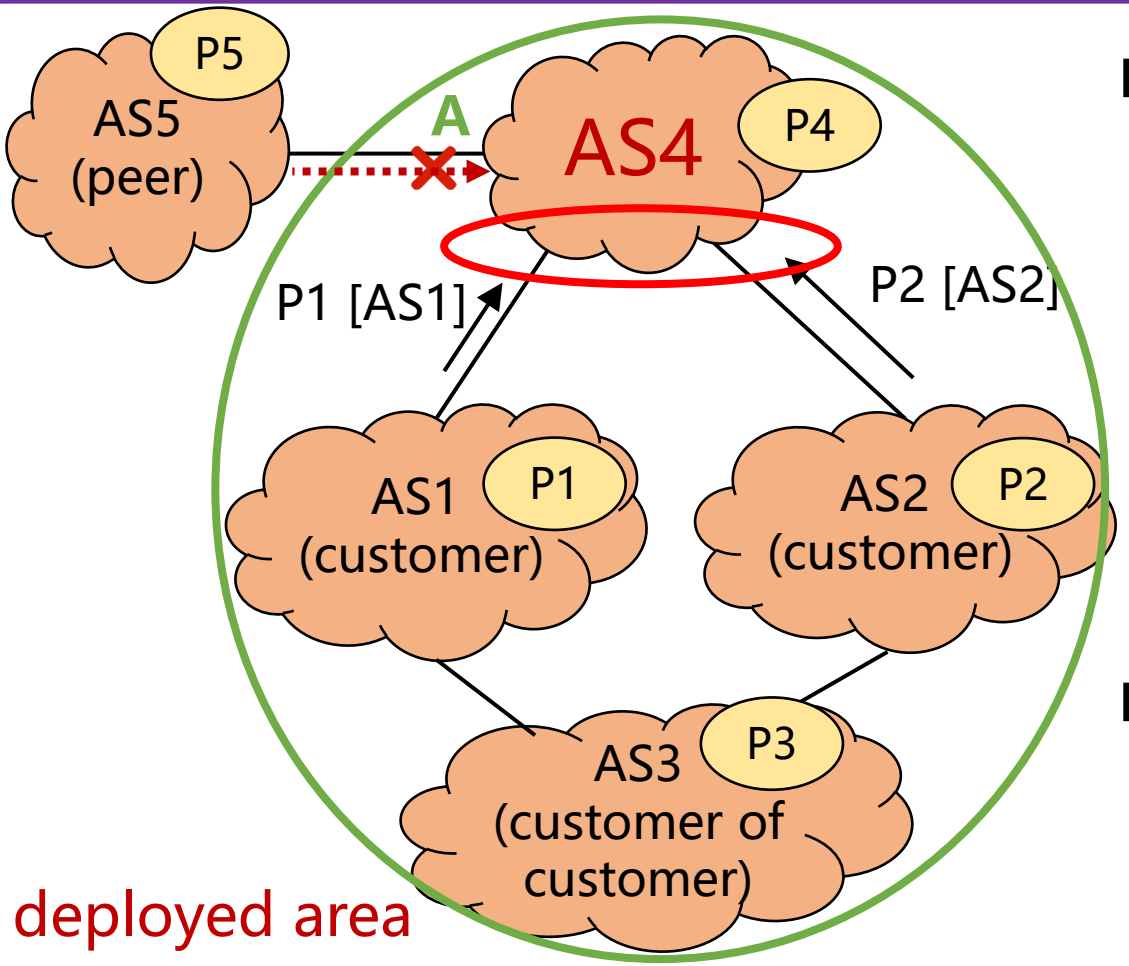

AS1 and AS2 advertise their routing information to AS4 through BGP

Assume AS4 runs EFP-uRPF (with Algorithm B) at customer ports

- The SAV rule at AS4's customer ports
  - AS4 generates an allowlist containing source prefixes of the customer cone, and applies the allowlist to all customer ports
  - Benefit: packets from AS4's customer cone cannot spoof the source addresses of outside ASes, which is finer-grained than using loose-uRPF
- Problem
  - When packets from AS1, AS2 and AS3 spoof the source addresses of each other, AS4 will improperly permit these packets

Loose-uRPF / EFP-uRPF (with Algorithm B) in inter-domain SAV has improper permit problem.

# Benefit of Inter-domain DSAV Compared with uRPF




AS1 and AS2 advertise their routing information to AS4 through BGP

- AS1, AS2, AS3 and AS4 run DSAV and generate SAV rules for P1, P2, P3, P4
  - AS1, AS2, AS3 and AS4 cannot spoof each other (while EFP-uRPF may have improper permit problem)
  - Packets from AS5 with spoofed source addresses of P1, P2, P3, P4 will be blocked at port A (while Loose-uRPF may have improper permit problem)
- Inter-domain DSAV
  - ASes within the deployed area cannot spoof each other
  - ASes in the undeployed area cannot spoof the source addresses of the deployed area

# Open Questions: Accuracy

- Question 1: The key of DSAV is to discover real data-plane forwarding path.

  Any factor that affects forwarding should be considered

  - Policy-based routing like static routing or ACL redirection may change the data-plane forwarding path of FIB table. How to handle?

# Open Questions: Scalability

- Question 2: Containing a long list of IP addresses in source prefix field and propagation scope field is costly. Can we compress DSAV protocol messages?

# Open Questions: Convergency

- Question 3: When updating, there may be a gap between the change of FIB table and the update of SAV table. How to avoid improper block?
- Question 4: In fast rerouting, when a link fails, the router can select a backup forwarding path immediately. How to handle resultant improper block since the SAV tables of downstream routers do not learn the backup forwarding path?

# Open Questions: Incremental deployment

- Question 5: How about multiple disconnected deployed areas??

# Open Questions: Security

- Question 6: What's the threat model of DSAV protocol messages and how to address?

# Open Questions: Privacy

- Question 7: In inter-domain DSAV, an AS will tell its local routing policy information to other ASes. Is it a leak of privacy?

# Thanks

# Backup Slides

# Considerations: Accuracy (Cont.)

- Question 1: Policy-based routing like static routing or ACL redirection may change the data-plane forwarding path of FIB table. How to handle?
  - DSAV can use the control-plane routing information to generate notification messages along policy-based forwarding path.

# Considerations: Scalability

- Question 2: Containing a long list of IP addresses in source prefix field and propagation scope field is costly. Can we compress DSAV protocol messages?
  - A node can be represented by a node ID (e.g., the router-ID for a router or the ASN for an AS). For each initial node, its source prefixes together with its node ID can be advertised to other nodes through broadcast or existing routing protocols.
  - In this way, other nodes will know the mapping from a node ID to a list of source prefixes. Therefore, the source prefix field of DSAV messages can be replaced with just one source node ID. Destination prefixes of propagation scope field can also be replaced with destination node IDs.

# Considerations: Convergency

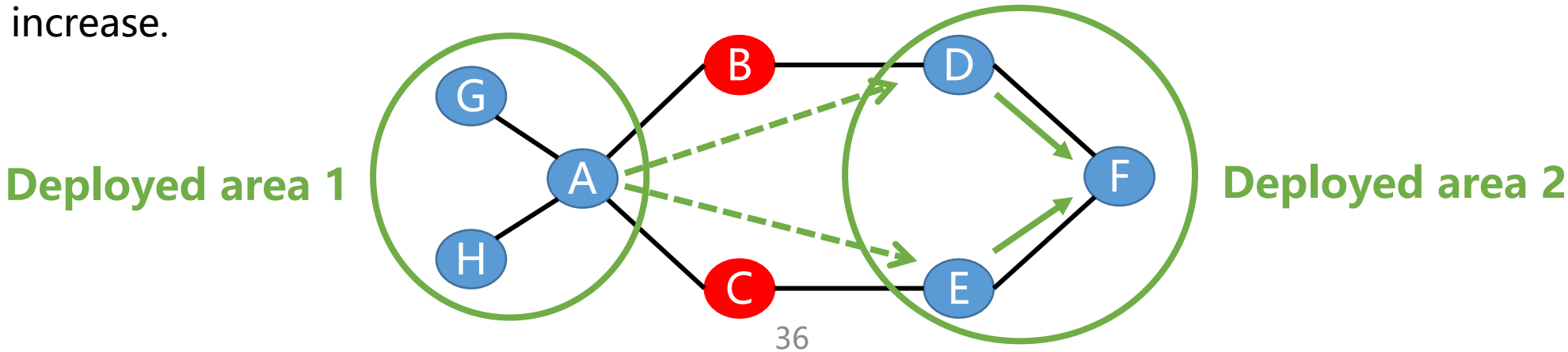
- Question 3: When updating, there may be a gap between the change of FIB table and the update of SAV table. How to avoid improper block?
  - Improper block caused by inconsistency can never be completely avoided. Actually it is the same for routing protocols. Although every routing protocol tries to minimize inconsistency, routing loop can never be completely avoided during the converging period and packet may be dropped due to TTL degradation from loop.
  - But we can further reduce improper block by various ways. It is a tradeoff between network security and network availability. If a network operator can tolerate rare improper block of normal traffic to increase network security, it can also directly drop packets if it does not match the SAV table. But if a network operator cares more about network availability, an alert mechanism will be a reasonable choice.
  - There are other ways for setting the tradeoff. For instance, if a flow comes from an invalid incoming port, a "tolerance window" can be used to pass the first few packets of the flow within the window. However, if the SAV table is not updated after the window, subsequent packets of the flow will be dropped. Some other network protocols take the similar "tolerance" idea to solve the inconsistency problem.

# Considerations: Convergency

- Question 4: In fast rerouting, when a link fails, the router can select a backup forwarding path immediately. How to handle resultant improper block since the SAV tables of downstream routers do not learn the backup forwarding path?
  - DSAV can also discover the backup forwarding path to build a backup SAV table in routers. We have two possible methods to use the backup SAV table, but not sure which one is better.
  - 1) All the traffic matching SAV table and backup SAV table are permitted. The downside is that the filtering will be very loose.
  - 2) When a link fails, the failure information is advertised to each node. Then, each node can quickly switch to the backup SAV table. The downside is that improper block may occur, since there will be a time gap.

# Considerations: Incremental deployment

- Question 5: How about multiple disconnected deployed areas?
  - If a node in a deployed area can discover the location of other deployed areas, it can bypass the undeployed area and generate notification messages to other deployed areas.
  - For inter-domain DSAV, we recommends incremental deployment by customer cones.
    - With the merger of different customer cones where DSAV is deployed, the deployed area will gradually expand, and the defense capability against source address spoofing will gradually increase.



# Considerations: Security

- Question 6: What's the threat model of DSAV protocol messages and how to address?
  - The adversary model includes: 1) Message alteration: A malicious AS alters any part of a DSAV message, such as source prefix field or propagation scope field.
  - 2) Message injection: A malicious AS injects a "valid" DSAV message and sends it to the corresponding next-hop AS, such as replay attacks.
  - 3) Path deviation: A malicious AS sends a DSAV message to a wrong next-hop AS which conflicts with the propagation scope field.
  - 4) Combination attack: A malicious AS alters a DSAV message to prevent path deviation from causing obvious conflicts, which is the most challenging attack.
  - We are facing the same problem as route origin hijacking in BGP routing. So RPKI-like approaches can be taken to prevent nodes from advertising the false prefixes. Existing symmetric cryptography mechanisms for origin and path validation can be used solve these adversary models. Other security problems and mechanisms are to be explored.

# Considerations: Privacy

- Question 7: In inter-domain DSAV, an AS will tell its local routing policy information to other ASes. Is it a leak of privacy?
  - This is actually a tradeoff between security and privacy. An AS can better protect its IP addresses from being forged, by making certain sacrifice on the path selection privacy. Moreover, the path selection policy of an AS can also be learnt by neighboring ASes by detecting the real traffic direction, even if the path selection policy is not explicitly advertised.

# Thanks

# Considerations: Incremental deployment

- Question 5: How does DSAV support incremental deployment?
  - Suppose router C does not support DSAV and there are multiple data-plane forwarding paths from A to F: A→C→E→F and A→B→D→F. Since F cannot learn SAV information for P_A from the first path, F will improperly block legitimate traffic with source addresses of P_A from E.
  - Each router discovers the topology of all deployed router. A router can send protocol messages directly to a "logical neighbor" over undeployed routers. Given that we have no way to know the forwarding tables of the undeployed routers, all the traffic coming from the undeployed area should not be filtered (to prevent improper block).