Tackling Spoofing Attacks in Broadband Access Networks

Bharat Joshi (bharat_joshi@infosys.com)
Pavan Kurapati (pavan_kurapati@infosys.com)
Ramakrishna Rao DTV (ramakrishnadtv@infosys.com)

Agenda

- Spoofing – What, Why and How
- Types of user connections in Broadband Access Concentrators
- Types of spoofing
- How to collect data to do Anti-spoofing in Access Network?
- Anti-spoofing
- How to recover anti-spoofing data after BAC crash/reboot?

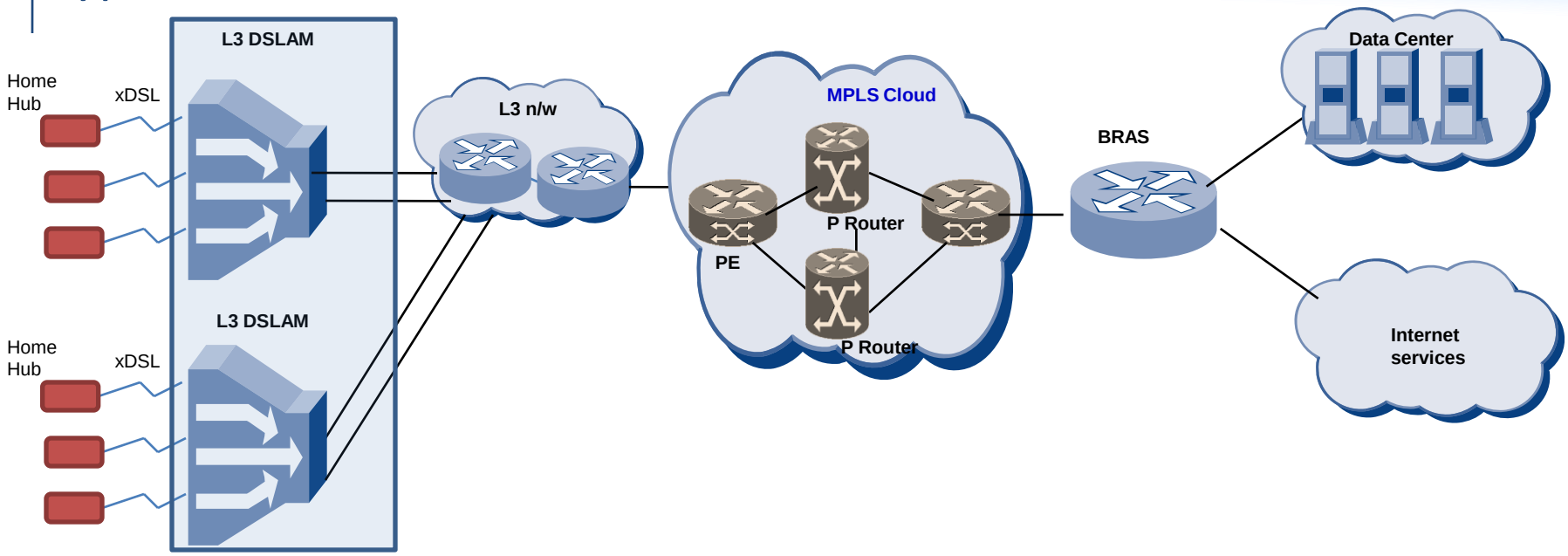
What is Spoofiing?

- Spoofing is a process whereby one entity masquerades as another entity
- Why is spoofing done
  - Spoofing A by B is done for various purposes
    - B seeks the privileges of A
    - B intends to hide its tracks
    - As an attack on A
- How is spoofing done?
  - We shall see in coming slides

The ultimate goal of spoofing

- **Unauthorized Service**
  - Get service on someone else's expense
- **Loss of Service on Target**
  - Make sure that the target does not get any service
- **Difficult to trace the attacker**
  - Make sure that people can not find who attacked them.
- **Unnecessary packets clogging the network**
  - Make sure that nobody gets a good service.
- **Secondary victim**
  - Primary target responds to spoof packet and overwhelm the source which becomes secondary victim.

Types of user connections for an IP based DSLAM



Bridged IP Routing

- RFC 2684 based bridged encapsulation between End User and DSLAM
- DSLAM in routing mode with routed VLANs configured on uplink i/f
- Dynamic IP allocation using **DHCP**

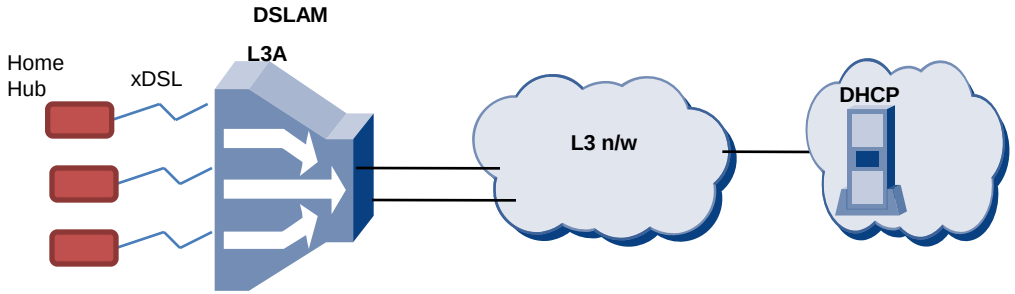
PPPoE/A

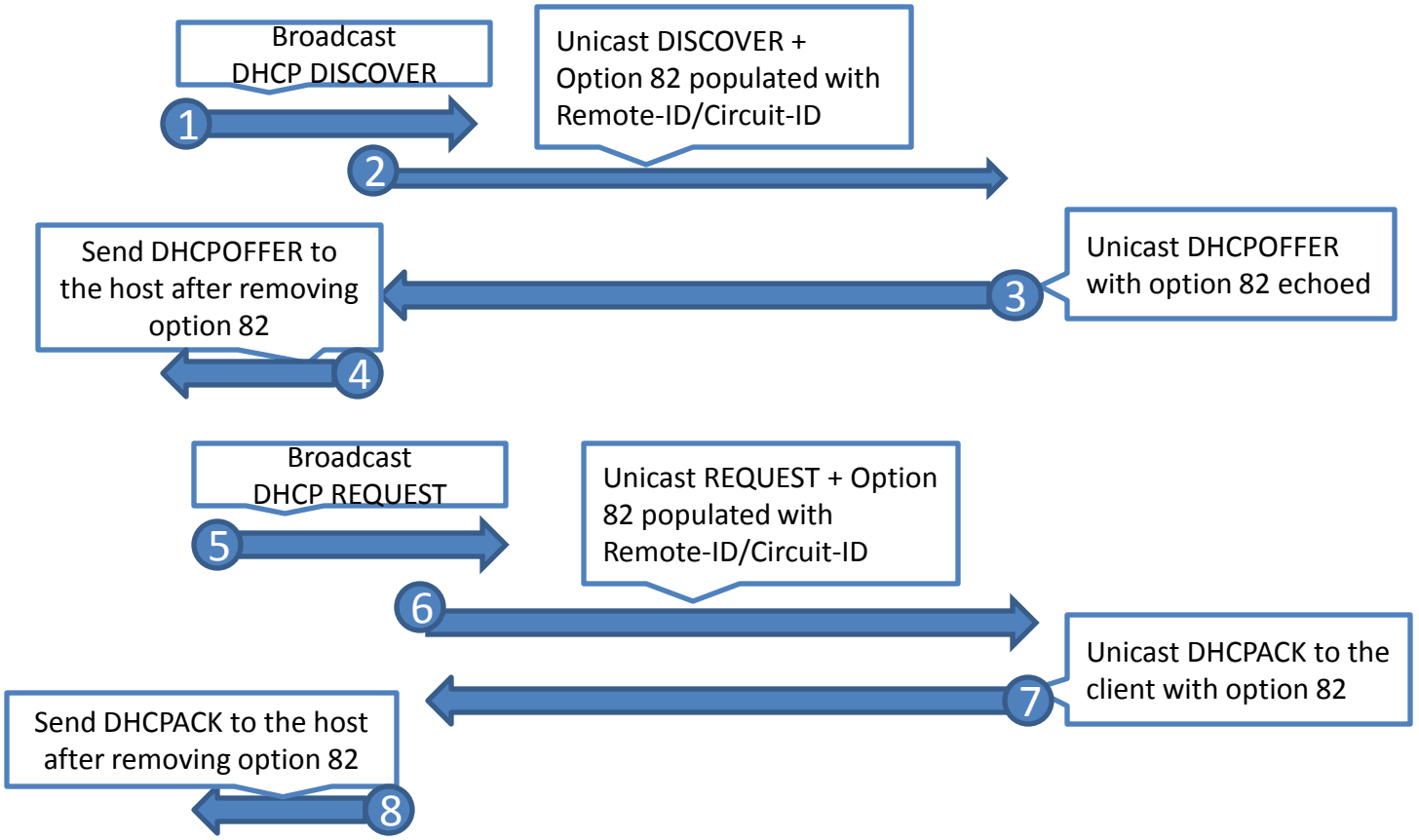
- PPP termination in DSLAM
- IP allocation from local pool
- DSLAM in routing mode with routed VLANs configured on uplink i/f

IPoA

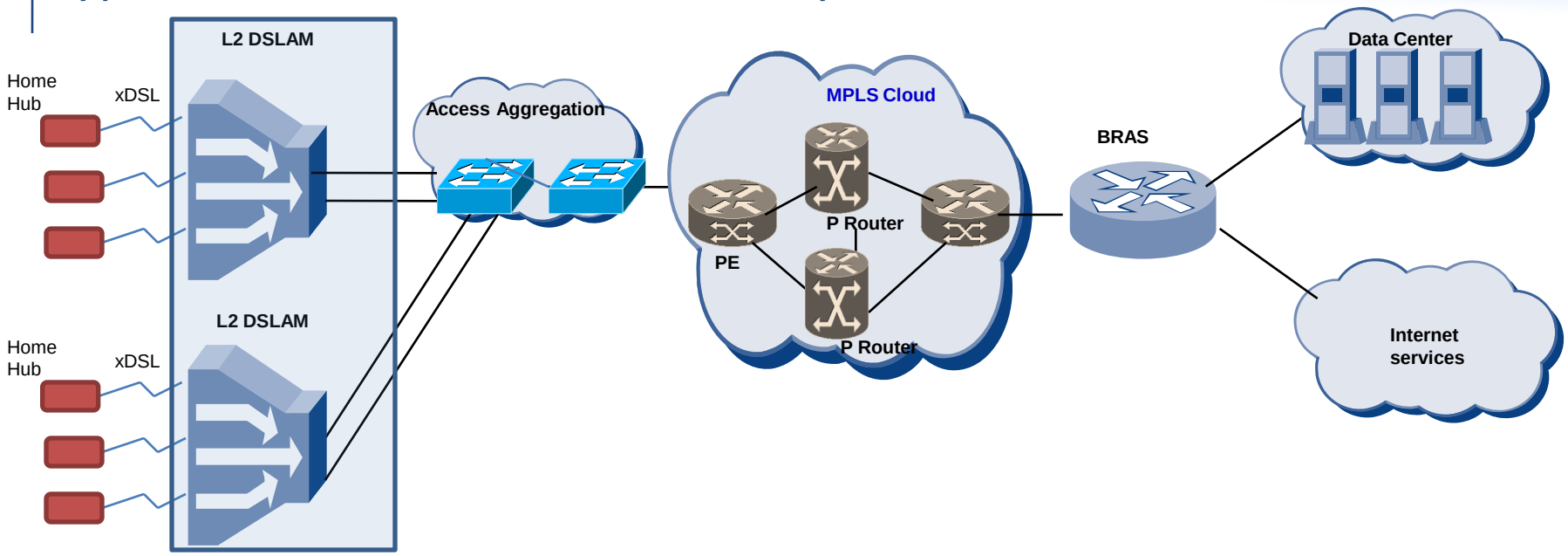
- RFC 2684 based routed encapsulation between End User and DSLAM
- DSLAM in routing mode with routed VLANs configured on uplink i/f
- Dynamic IP allocation using DHCP

Address allocation mechanisms for IP DSLAM – DHCP





Types of user connections for a Layer 2 DSLAM



1:1 VLANs

- Map every user connection to one unique 802.1q based VLAN
- No need of any MAC learning of individual hosts
- Downstream traffic mapping done based on VLANs

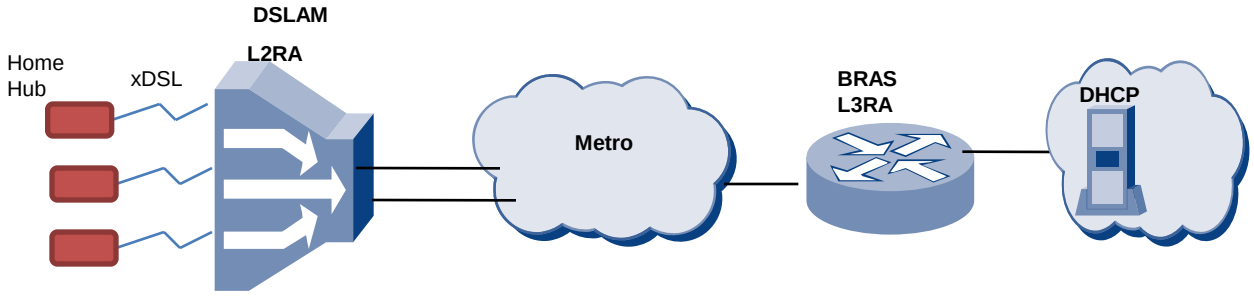
Q in Q or Stacked VLANs

- An outer Service VLAN identifying a specific service is added
- Downstream mapping done based on combination of CVLAN and SVLAN

N:1 Transparent Bridged VLANs

- Multiple users mapped to a common VLAN
- Downstream mapping done based on VLAN and Dst MAC combination
- MAC learning is required for operation

Address allocation mechanisms for L2 DSLAM – DHCP



Broadcast DHCP DISCOVER

1

Broadcast DISCOVER + Option 82 populated with Remote-ID/Circuit-ID

2

Unicast DISCOVER with 'giaddr' populated

3

Unicast DHCPOFFER with option 82 echoed

4

Broadcast DHCPOFFER with option 82 and without 'giaddr'

5

Send DHCPOFFER to the host after removing option 82

6

Broadcast DHCP REQUEST

7

Broadcast REQUEST + Option 82 populated with Remote-ID/Circuit-ID

8

Unicast REQUEST with 'giaddr' populated

9

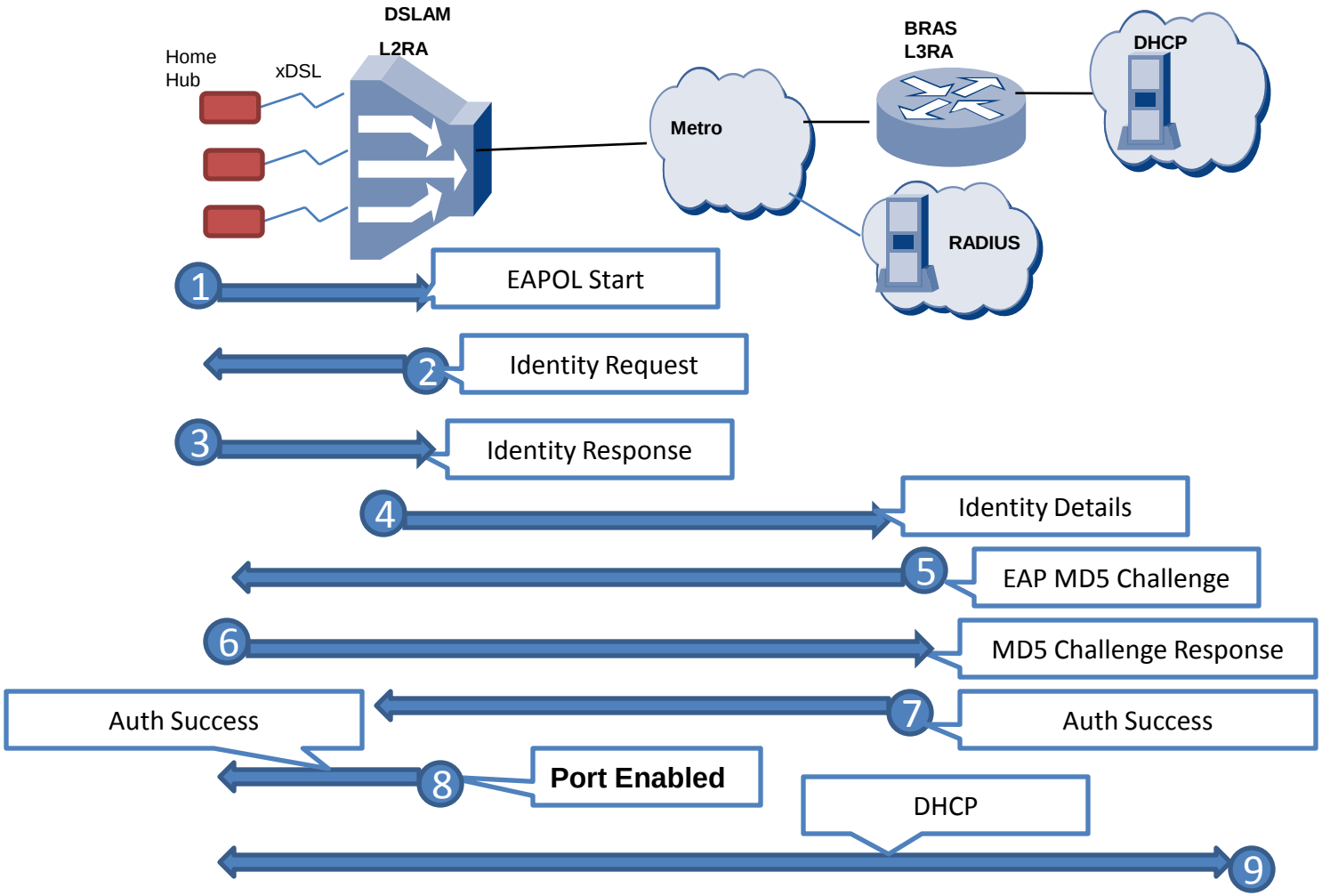
Unicast DHCPACK to the client with option 82

10

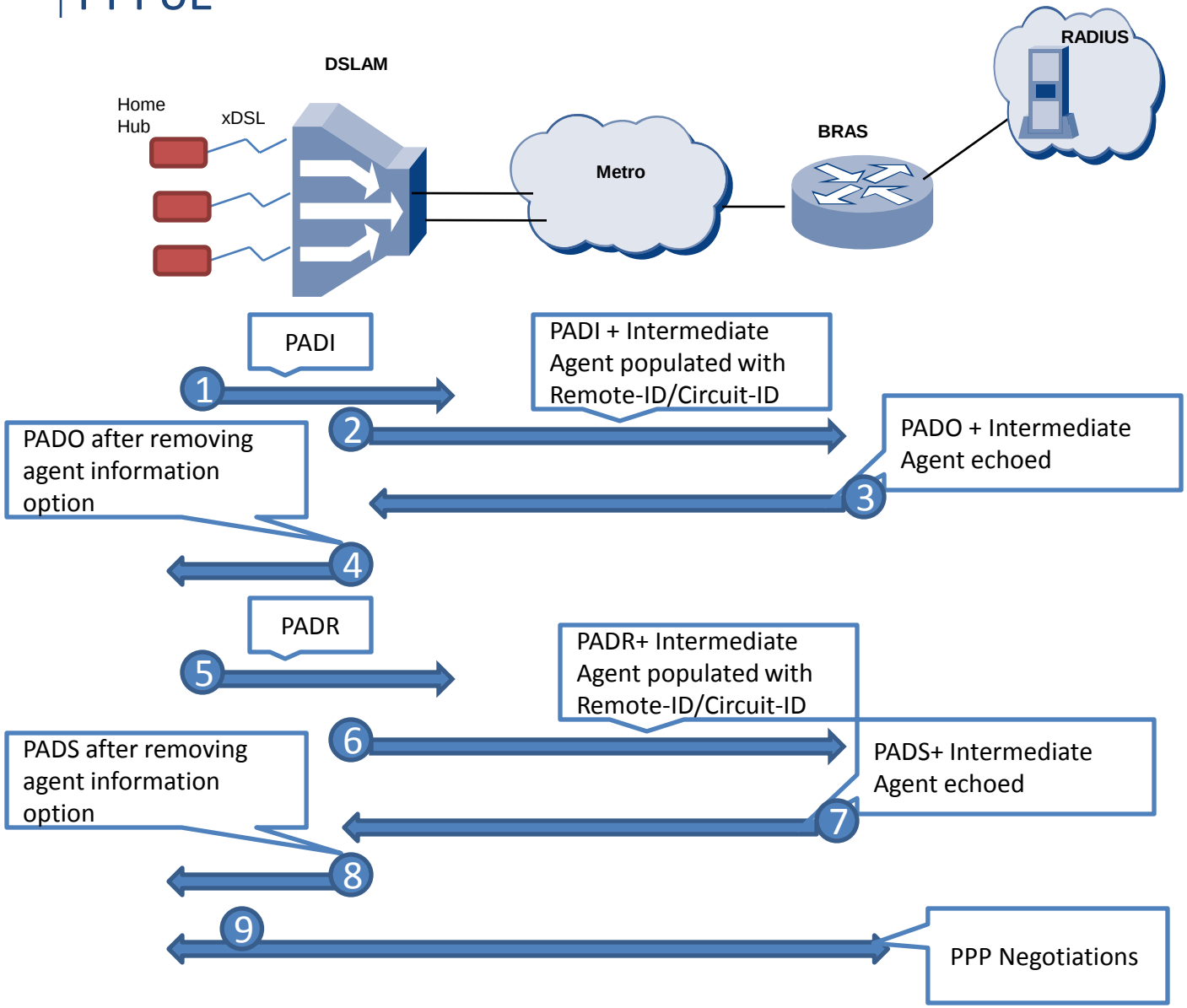
Send DHCPACK to the host after removing option 82

11

Authentication & Address allocation mechanisms for L2 DSLAM – DHCP + 802.1x



Authentication & Address allocation mechanisms for L2 DSLAM - PPPoE

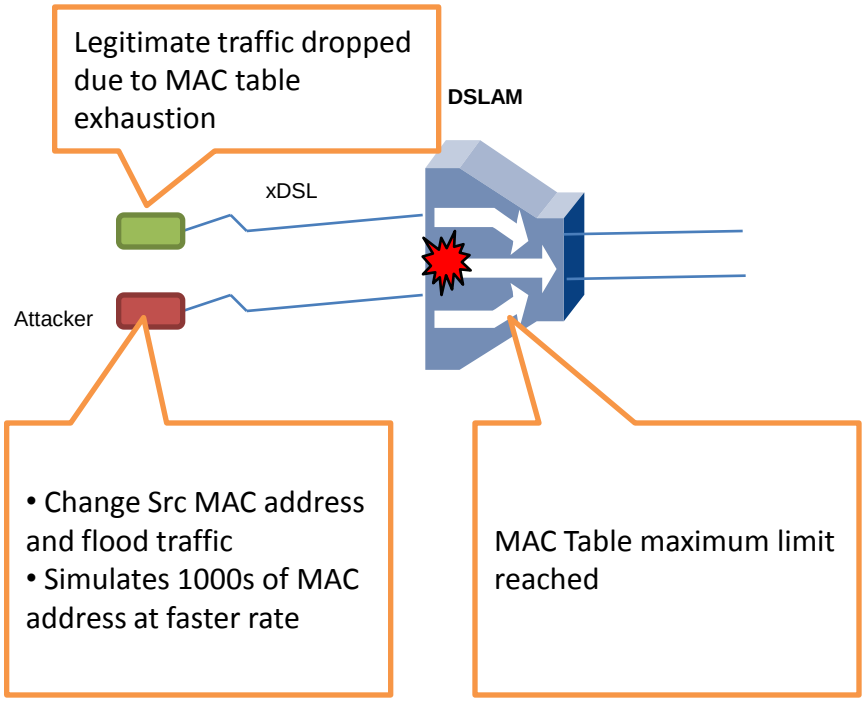


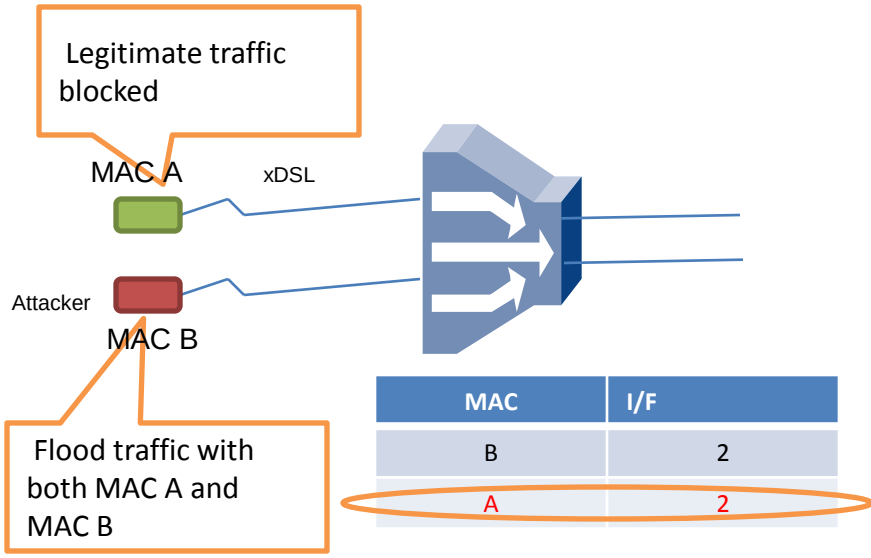
Type of Spoofing

- **MAC Spoofing**
- **IP Spoofing**
- **ARP Spoofing**
- **Control protocol internal header spoofing**
  - PPPoE session-id spoofing
  - DHCP chaddr, ciaddr, relay-agent-information option spoofing

MAC spoofing

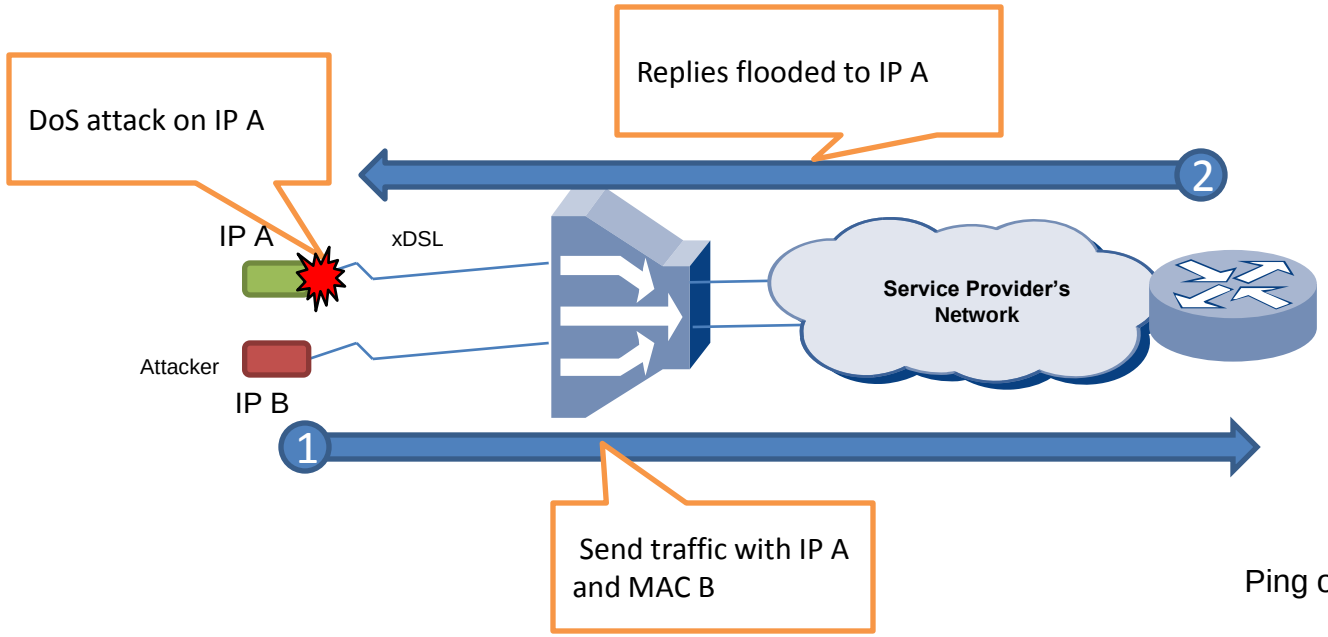
Changing Source MAC address to an illegitimate address





IP spoofing

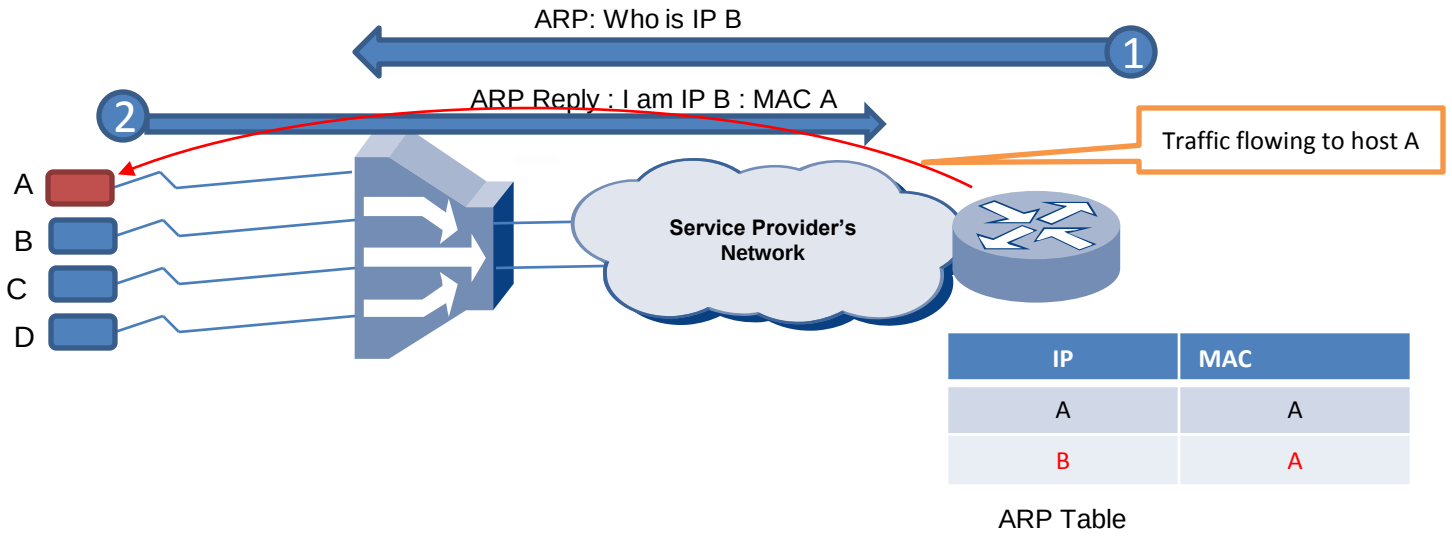
Changing Source IP address to an illegitimate address



Ping of Death

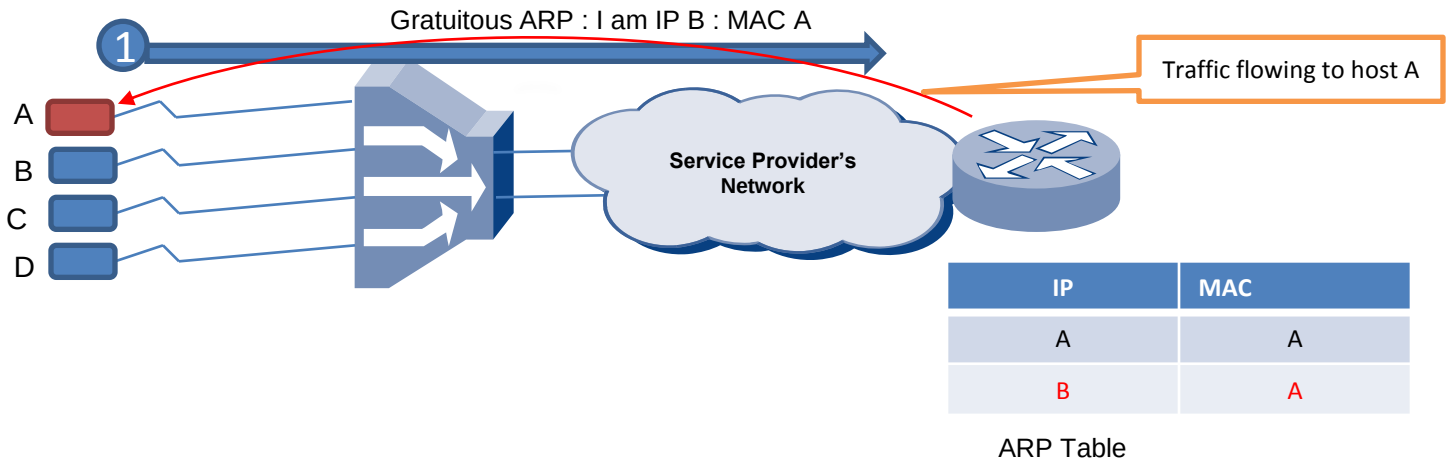
ARP spoofing

Respond/send ARP Response with illegitimate IP address



IP	MAC
A	A
B	A

ARP Table



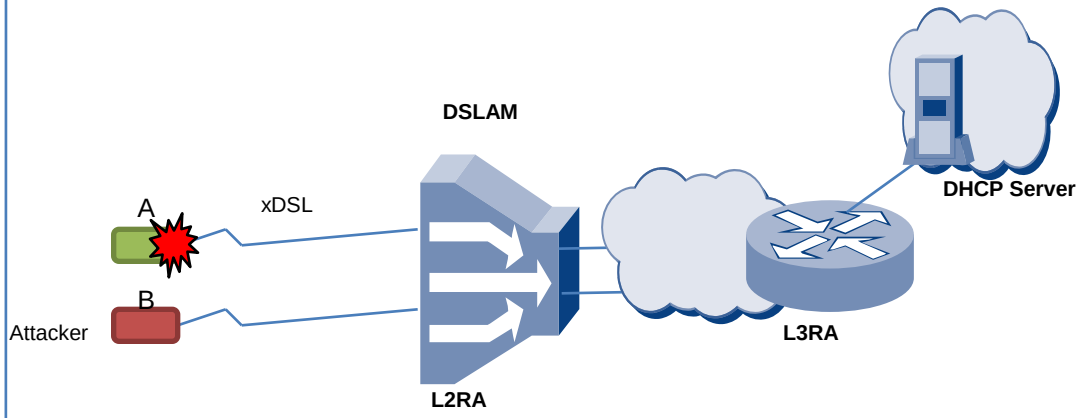
IP	MAC
A	A
B	A

ARP Table

DHCP Header spoofing

Changing Internal fields within DHCP header

DHCP Header



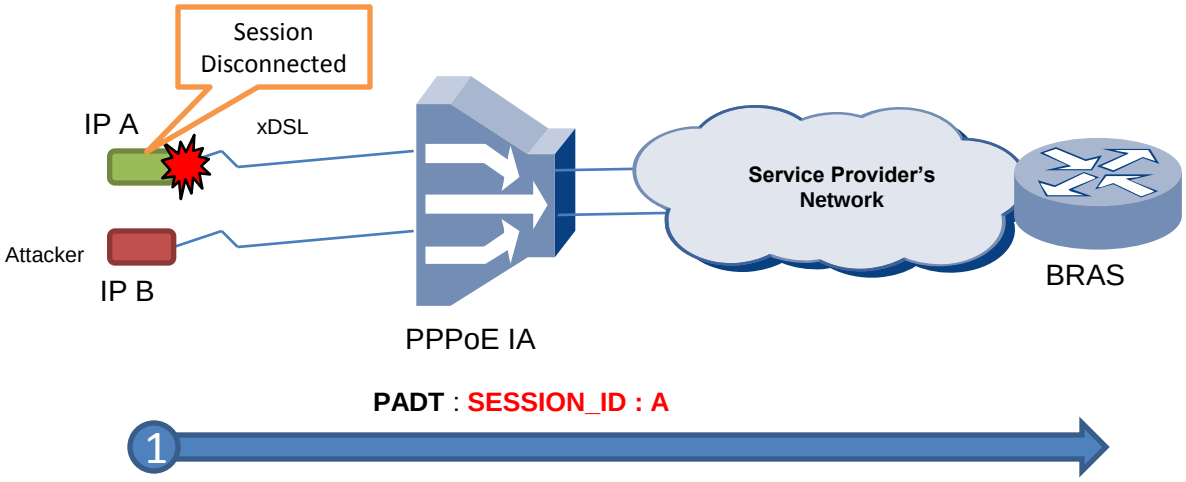
1. DHCP Release : MAC B: SrcIP: B **Ciaddr:A** — Spoof 'ciaddr' field
2. DHCP Release : MAC B: SrcIP: B **Chaddr:A** — Spoof 'chaddr' field
3. DHCP Release : MAC B: SrcIP: B **Client Id: A** — Spoof 'Client Identifier' field
4. DHCP Release : MAC B: SrcIP: B **Option82 Remote Id: A** — Spoof Option 82 field

PPPoE Header spoofing

Changing SESSION_ID field in PPPoE Header

- Similarly PPPoE Session-ID field identifies a unique session.
- Spoofing this can also cause service disruption

PPPoE Header



Anti-spoofing

- **What is anti-spoofing**
  - Mechanism to identify spoofing and stopping it.
- **How anti-spoofing is done**
  - By dropping the spoofed packets
- **How to identify the spoofed packets**
  - By verifying IP Address of the received packet.
  - By verifying MAC address of the received packet.
  - By verifying the combination of IP and MAC address for a given interface
  - By verifying the IP address, MAC address and other session based identification in the protocol header.

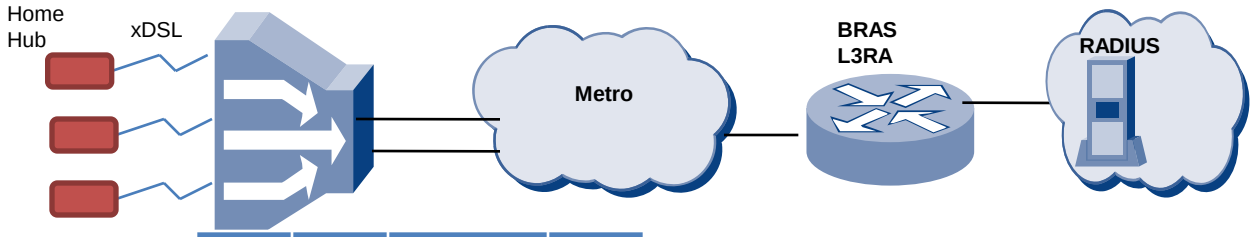
Data required for Anti-spoofing

- **For each user connection**
  - List of Valid IP addresses assigned
  - List of Valid MAC addresses and if possible the combination of MAC and IP addresses,
  - Time for which each IP address is valid.

Why anti-spoofing in Broadband Access Concentrator (BAC)?

- BAC is at the right place:
  - It knows all the required information to do anti-spoofing.
- Anti-spoofing becomes difficult and less effective if it is not done as near the source as possible.
- It is not only important to drop spoofed packets, it is important to drop them as early as possible.

Data collection for Anti spoofing in BAC - PPPoE



IP	MAC	Session-ID	I/F
A	A	101	1
B	B	102	2

1 PADI

2 PADO

3 PADR

PADS

4 BAC
Obtain Session-ID from PADS

5 PPP LCP

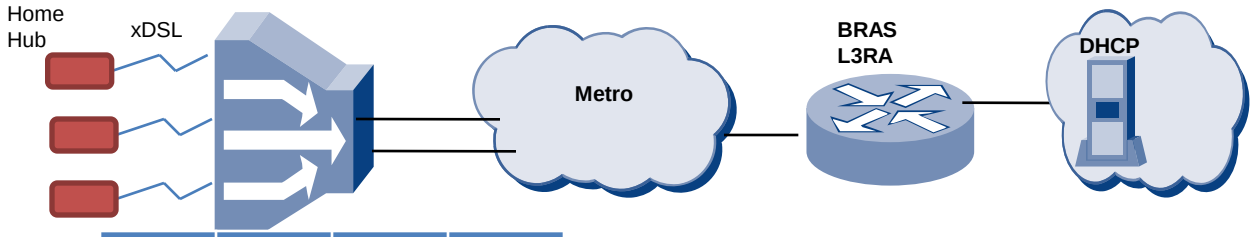
PPP IPCP

6 BAC
Obtain IP information from IPCP

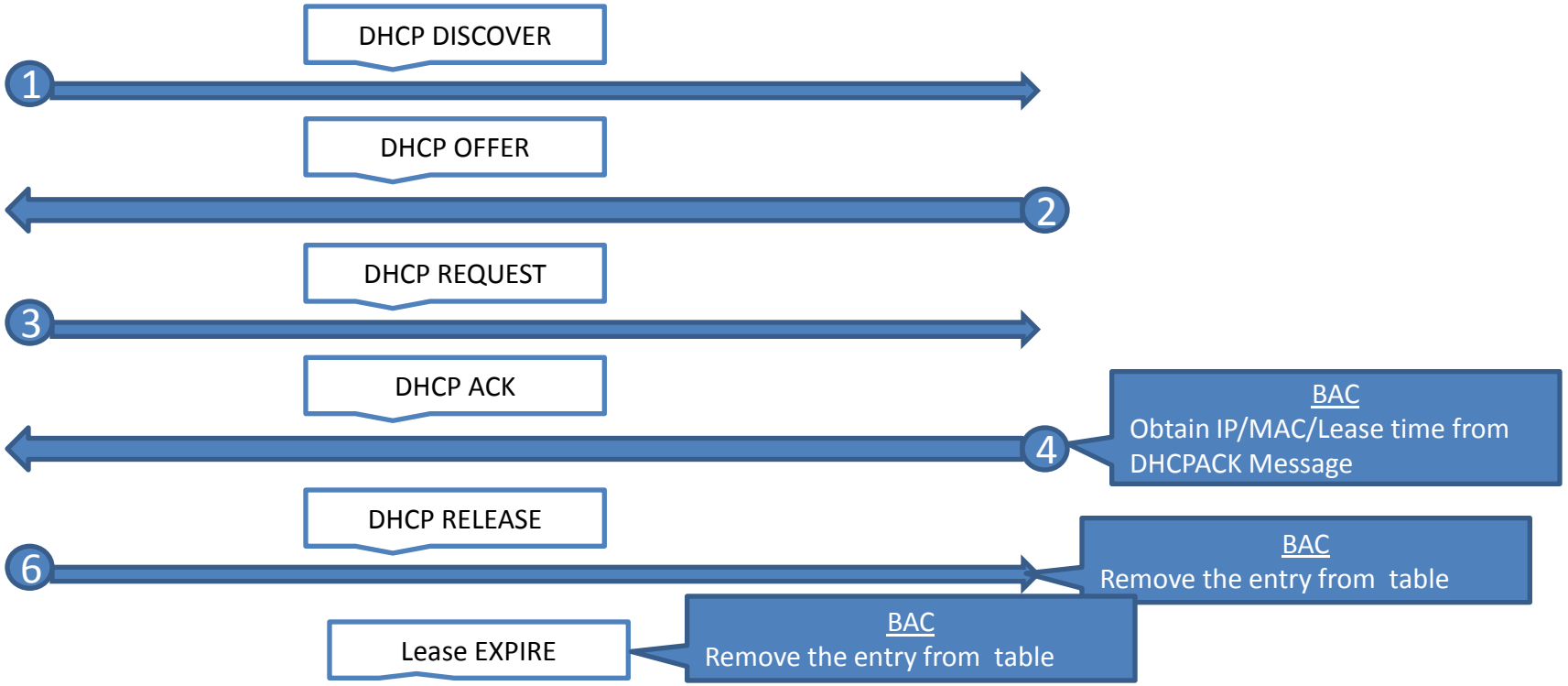
PADT

7 BAC
Delete the spoofing entry

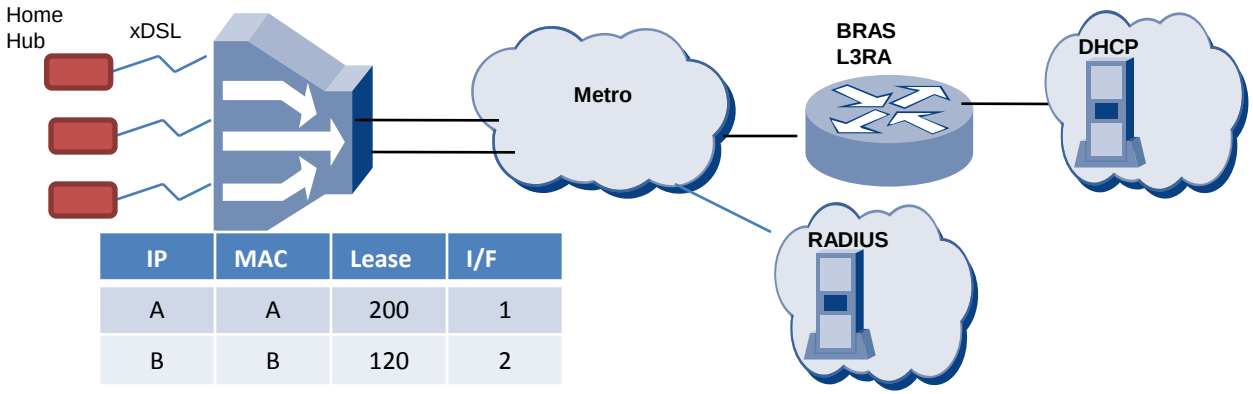
Data collection for Anti spoofing in BAC - DHCP



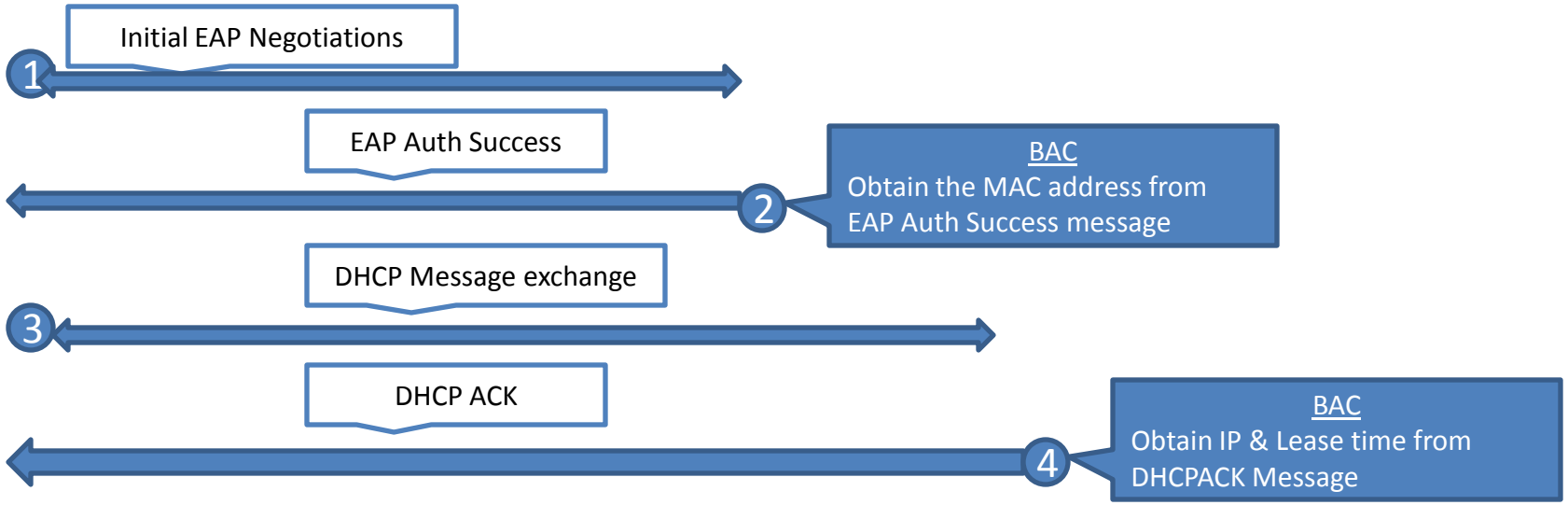
IP	MAC	Lease	I/F
A	A	200	1
B	B	120	2



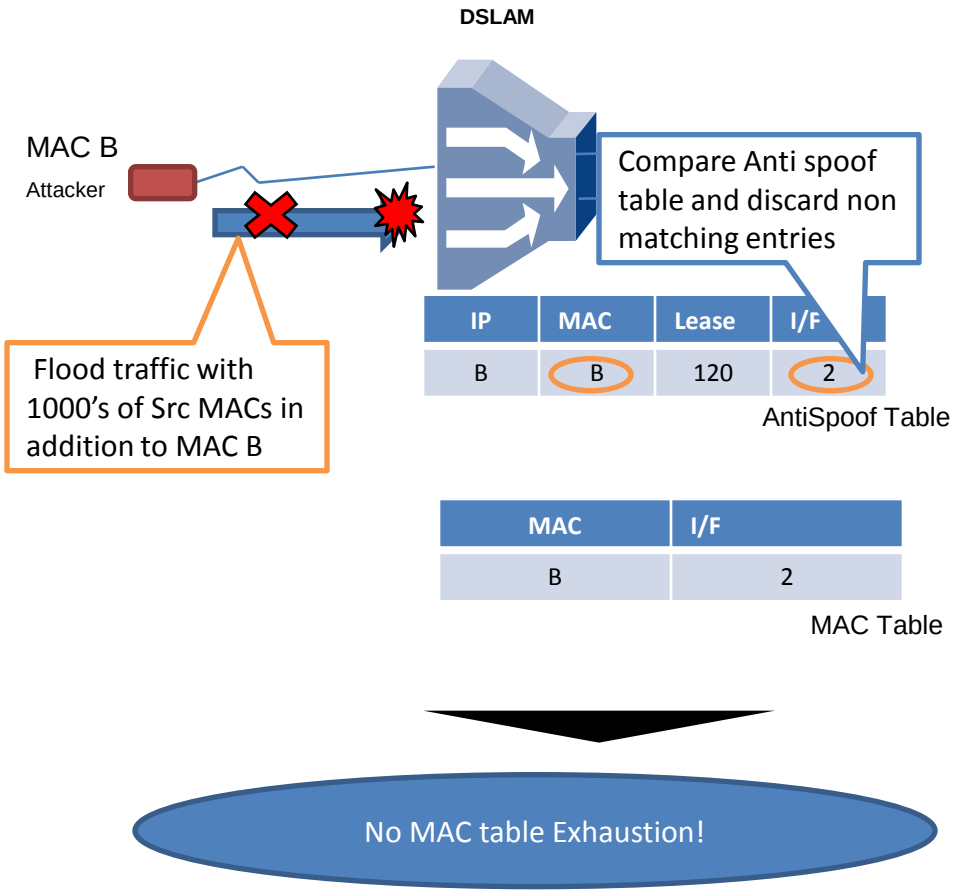
Data collection for Anti spoofing in BAC – 802.1x + DHCP

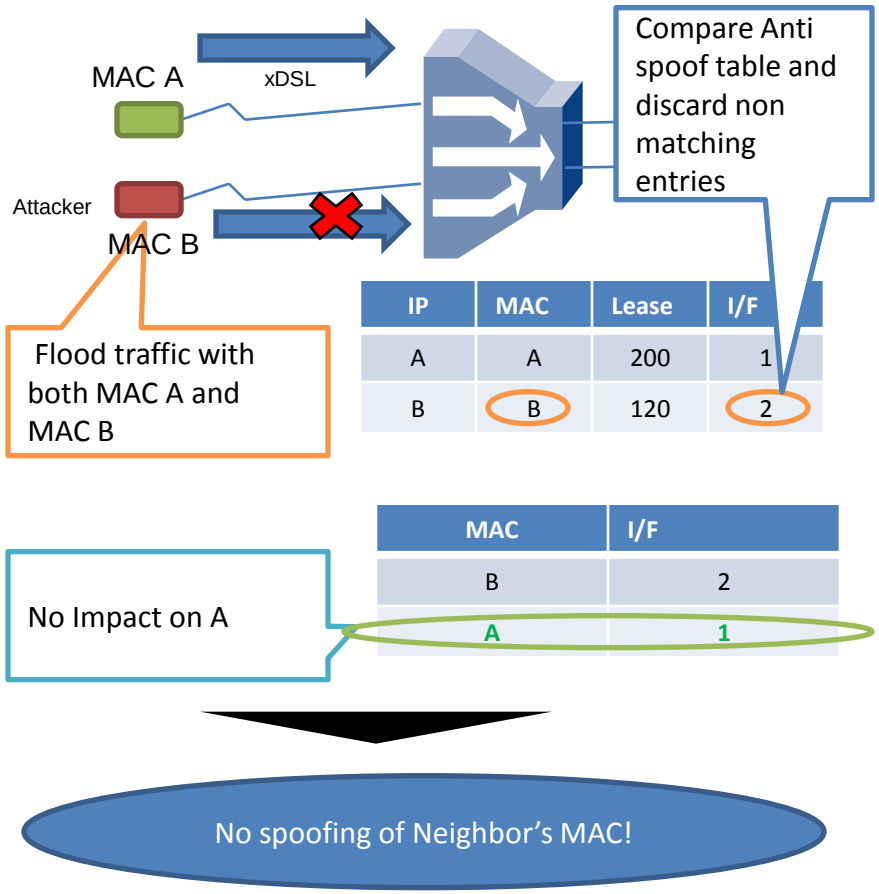


IP	MAC	Lease	I/F
A	A	200	1
B	B	120	2

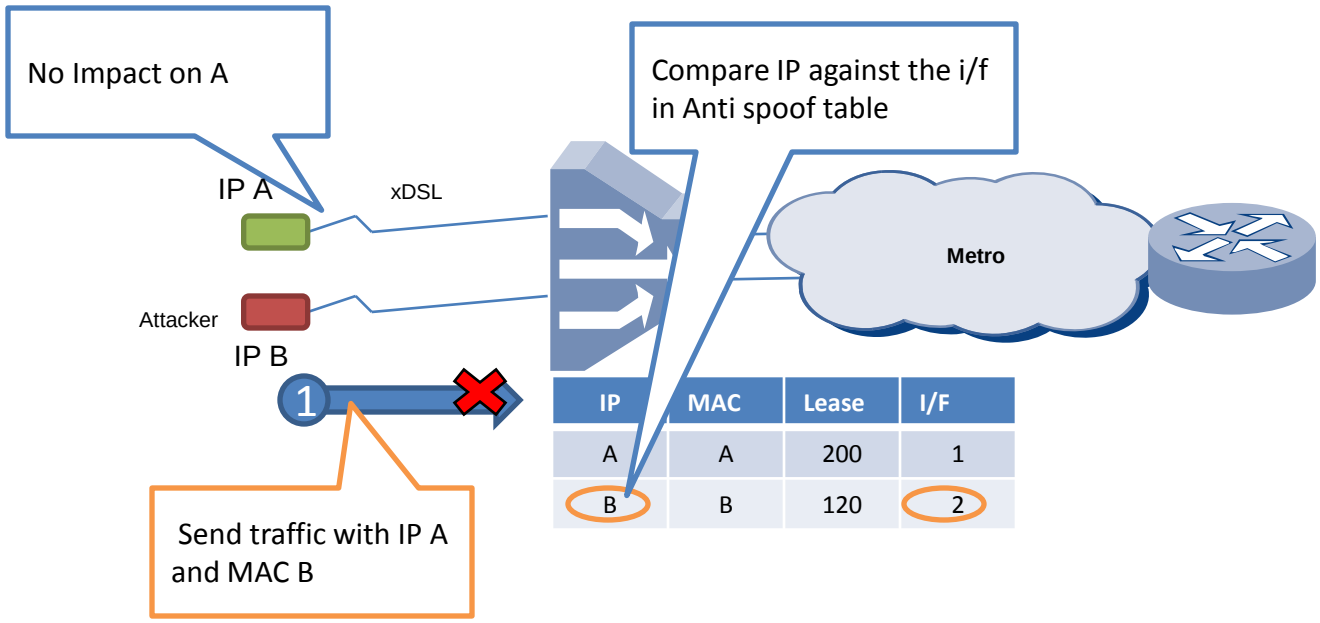


MAC Anti spoofing



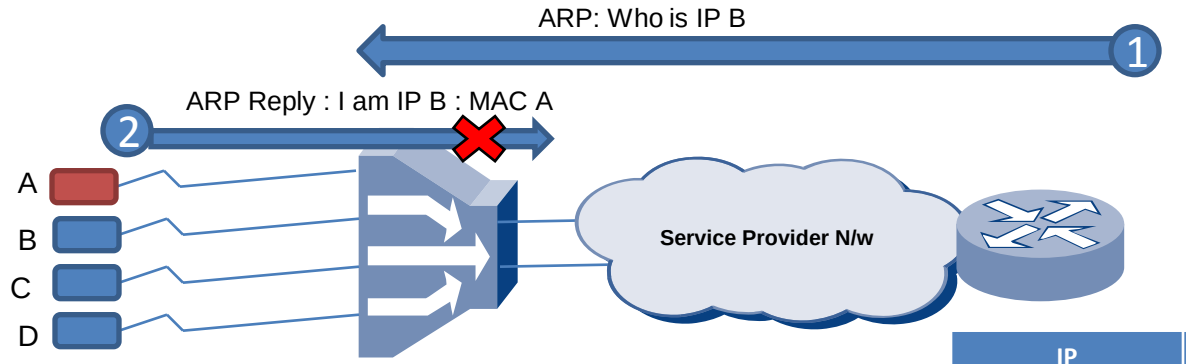


IP Anti spoofing



IP	MAC	Lease	I/F
A	A	200	1
B	B	120	2

ARP Anti spoofing

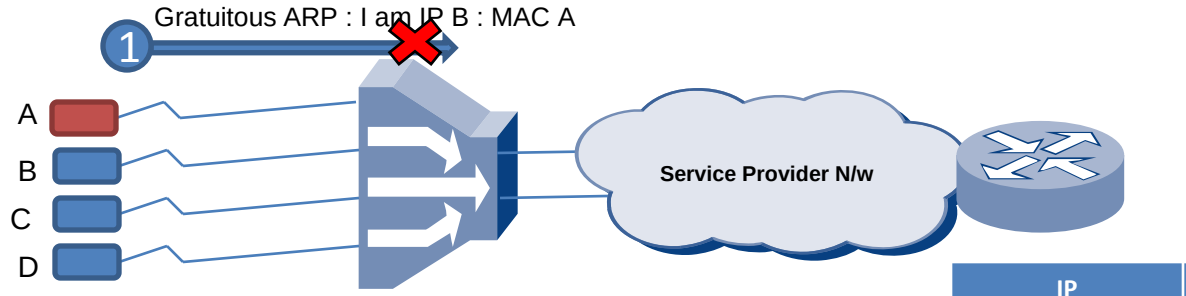


Compare IP/MAC against the i/f in Anti spoof table

IP	MAC	Lease	I/F
A	A	200	1
B	B	120	2

IP	MAC
A	A
B	B

ARP Table

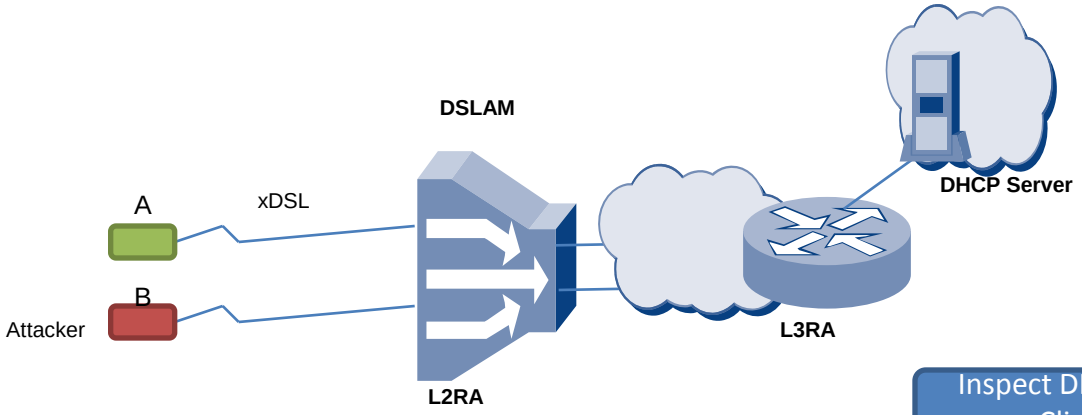


IP	MAC	Lease	I/F
A	A	200	1
B	B	120	2

IP	MAC
A	A
B	B

ARP Table

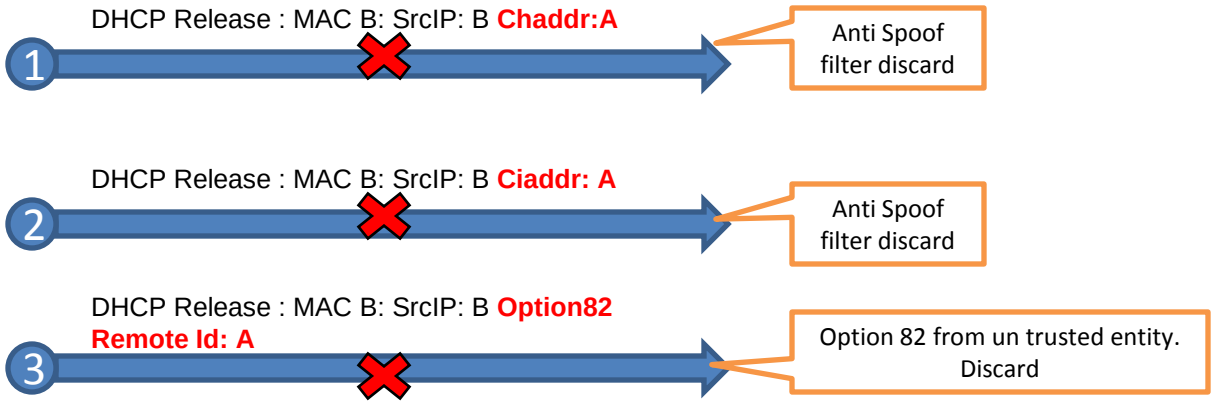
DHCP Header Anti spoofing



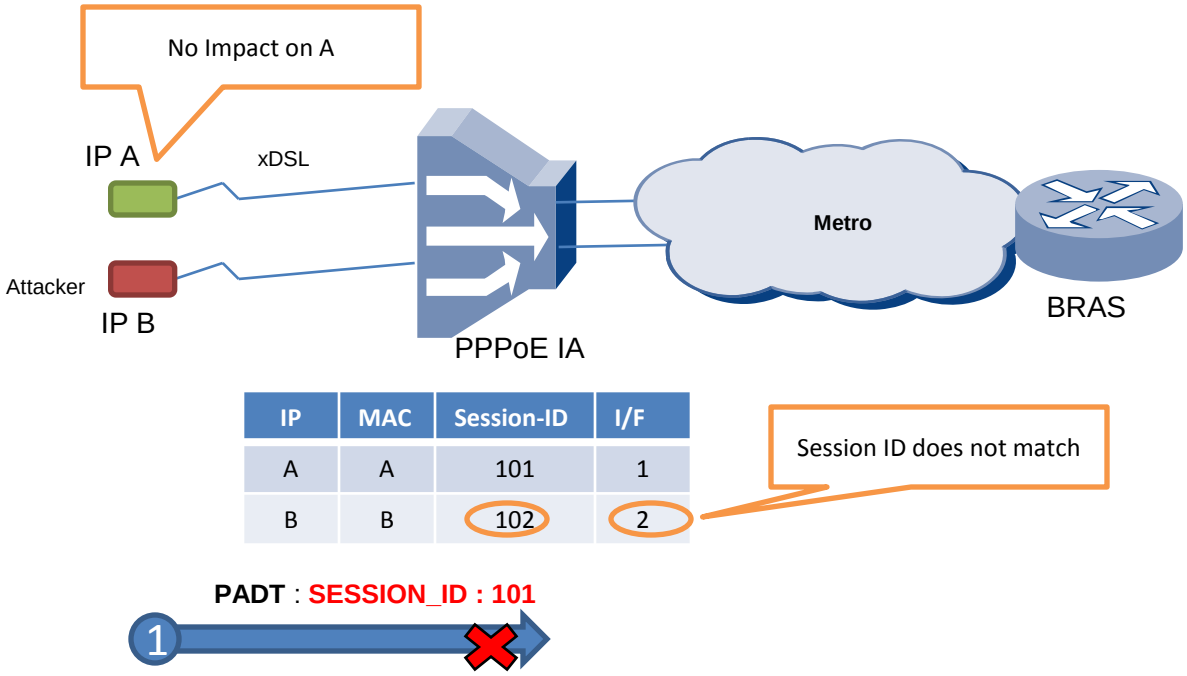
IP	MAC	Lease	I/F
A	A	200	1
B	B	120	2

Inspect DHCP Header & compare chaddr & Client ID with anti spoof table

Accept DHCP with option 82 only if it is coming from **trusted** entity



PPPoE Header Anti spoofing



Losing data collected for anti-spoofing

- **Data used in Anti spoofing can be lost due to various reasons**
  - Planned reboot
  - Software crash
  - Power failure
  - Replacement of system
  - Software upgrade

How to recover lost data?

- **Static configuration**
  - Required Data is available in the configuration.
- **PPPoE**
  - For PPPoE, the keep-alive timers are configured and the session is re-initiated if there are no replies to the keep-alive messages
- **DHCP**
  - DHCP does not have keepalive mechanism in place. DHCP has a ‘leasetime’ which is usually in order of ‘days’.
  - ***How to recover from this situation?***

Recovering Lease information for DHCP

- **Stable Storage:**
  - Not very useful as not many BACs support stable storage.
  - Limited erase cycles is also a bottleneck in this approach

- **Broadcast ARPs:**
  - Need to wait for downstream traffic to arrive and initiate ARP requests. Will increase the delay.
  - Can not get the complete information in one request.
  - Prone to spoofing attacks if a malicious user replies to the ARP request.

- **Redundant controllers**
  - BAC can have redundant controllers and upon one controller crash, the other controller can take over with pre-synched lease data.
  - Not suitable for power failure scenarios or for upgrades.
  - Having redundant controllers also add to hardware costs

Recovering Lease information for DHCP

- **Query through SNMP/LDAP**
  - Currently no standard MIBs are available for DHCP lease information.
  - BACs typically do not support SNMP client interfaces

- **Query lease information from DHCP server**
  - ***Solves most of the problems stated above***

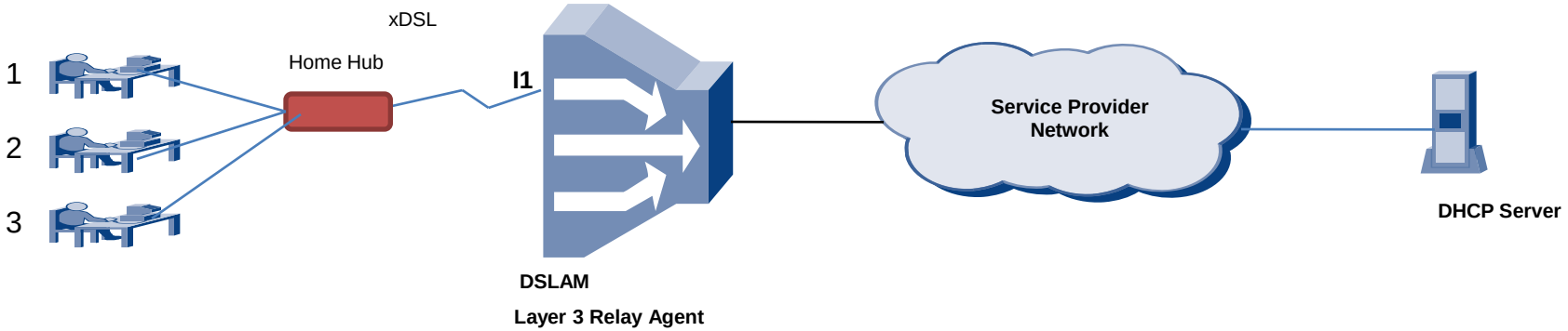
Lease query for DHCP (RFC 4388)

- RFC 4388 introduced a new DHCP request Leasequery which a BAC can use to query DHCP server to obtain lease information.
- Three types of queries are supported
  - Query by IP address
    - Only IP address is populated in the query message.
  - Query by MAC address
    - Only MAC address is populated in the query message. If more than one lease is available, then corresponding IP addresses are returned in associated-ip option.
    - BAC then gets additional data by generating query by IP address.
  - Query by Client identifier
    - Only client identifier option is populated in the query message. If more than one lease is available, then corresponding IP addresses are returned in associated-ip option.
    - BAC then gets the additional data by querying by IP address.

Lease query for DHCP (RFC 4388)

- Three types of reply message types are introduced:
  - DHCPLEASEACTIVE
    - When DHCP server knows about the query identifier.
  - DHCPLEASEUNKNOWN :
    - When DHCP server does not know about the query identifier.
    - An Access Concentrator cache this information so that this can be used to avoid generating Lease Query for the query identifier. This is known as **Negative Caching.**
  - DHCPLEASEUNASSIGNED:
    - When DHCP server does manage the query identifier but no lease is yet assigned.
    - **Negative Caching** is done for this response as well.

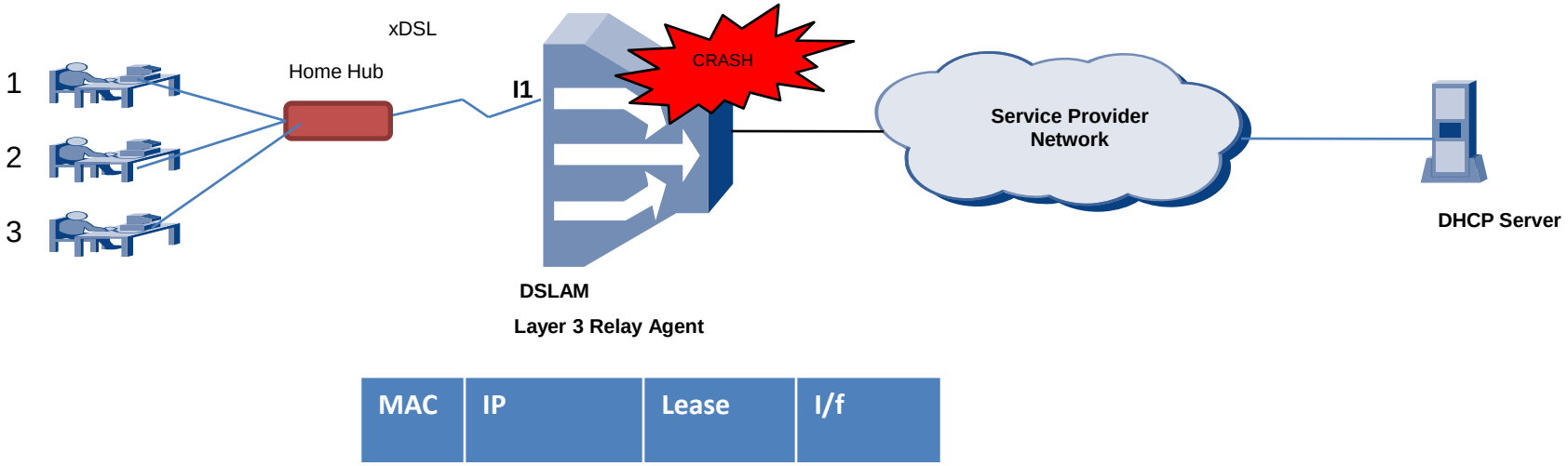
RFC 4388 based lease query – Data Driven



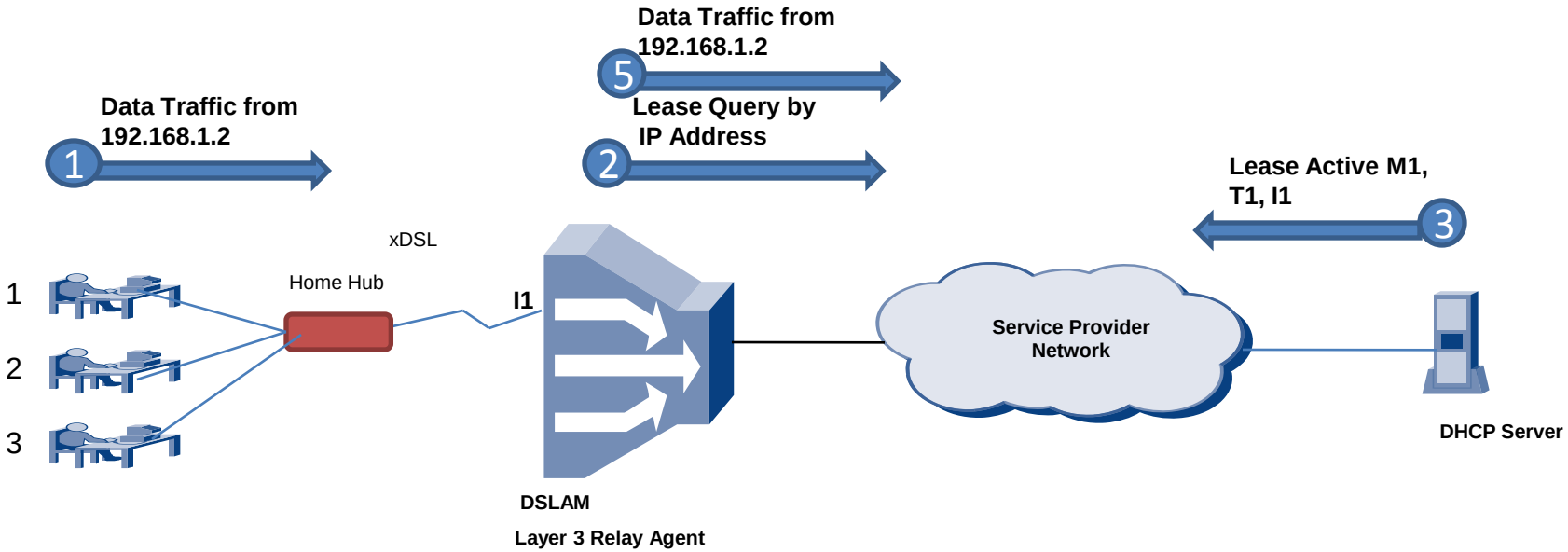
MAC	IP	Lease	I/f
M1	**192.168.1.2**	**T1**	**I1**
M2	192.168.1.8	T2	I1

Anti Spoof Table

RFC 4388 based lease query – Data Driven



RFC 4388 based lease query – Data Driven

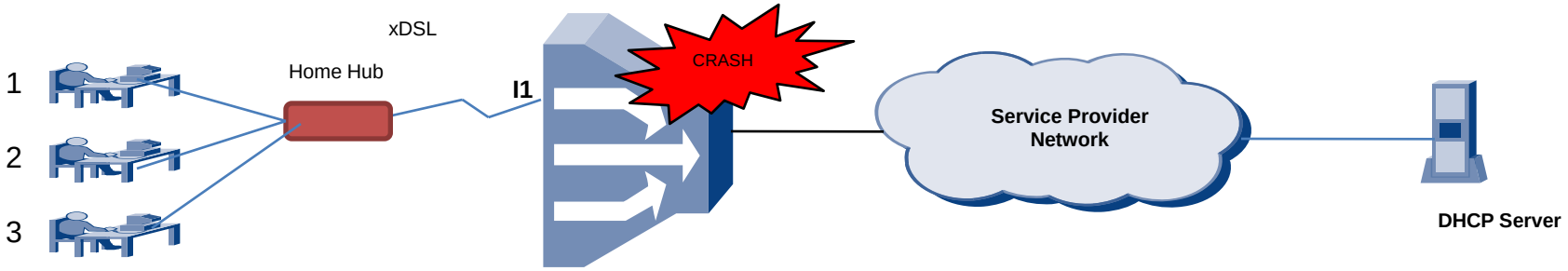


MAC	IP	Lease	I/f
M1	192.168.1.2	T1	I1

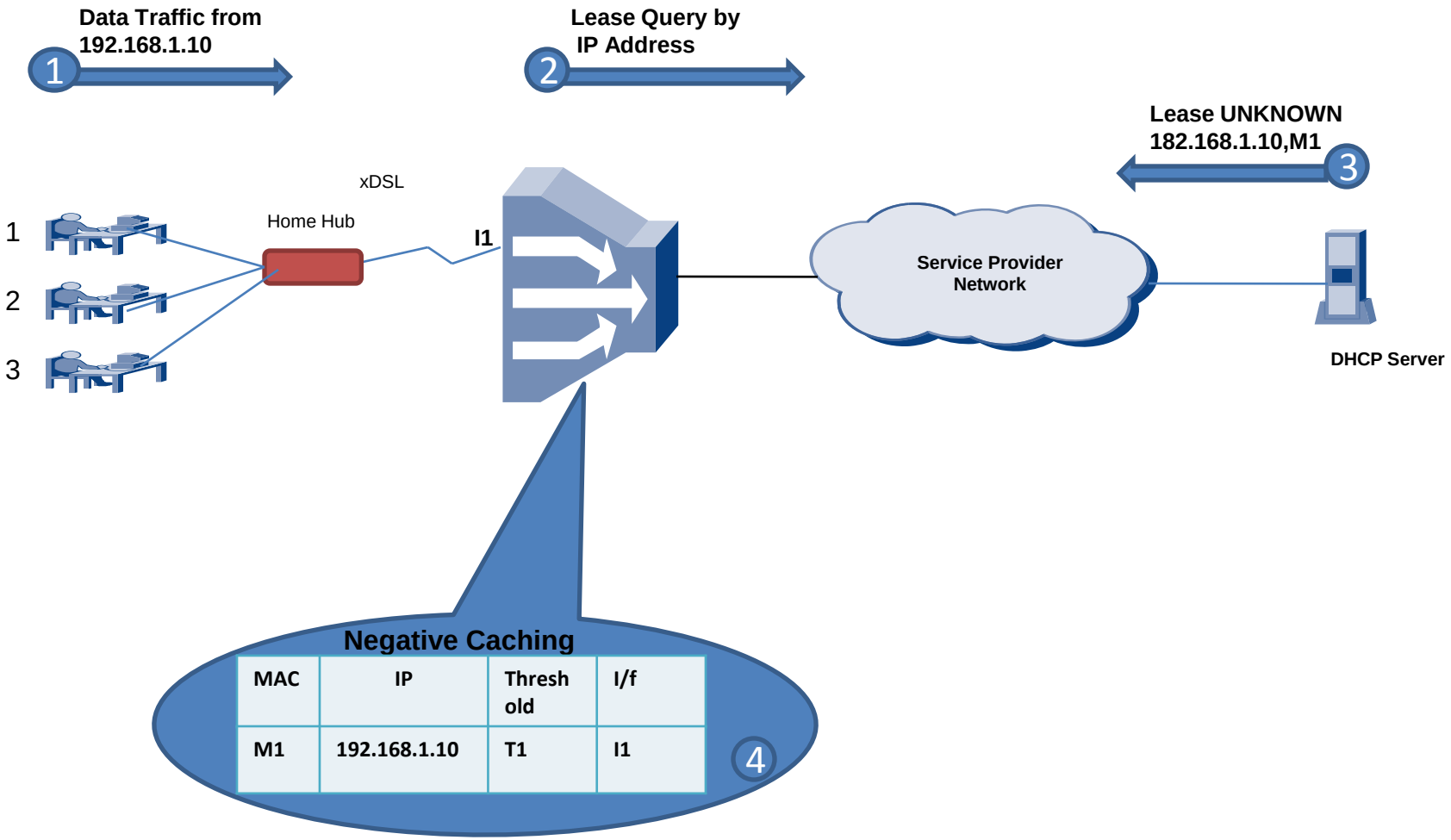
4

Anti Spoof Table

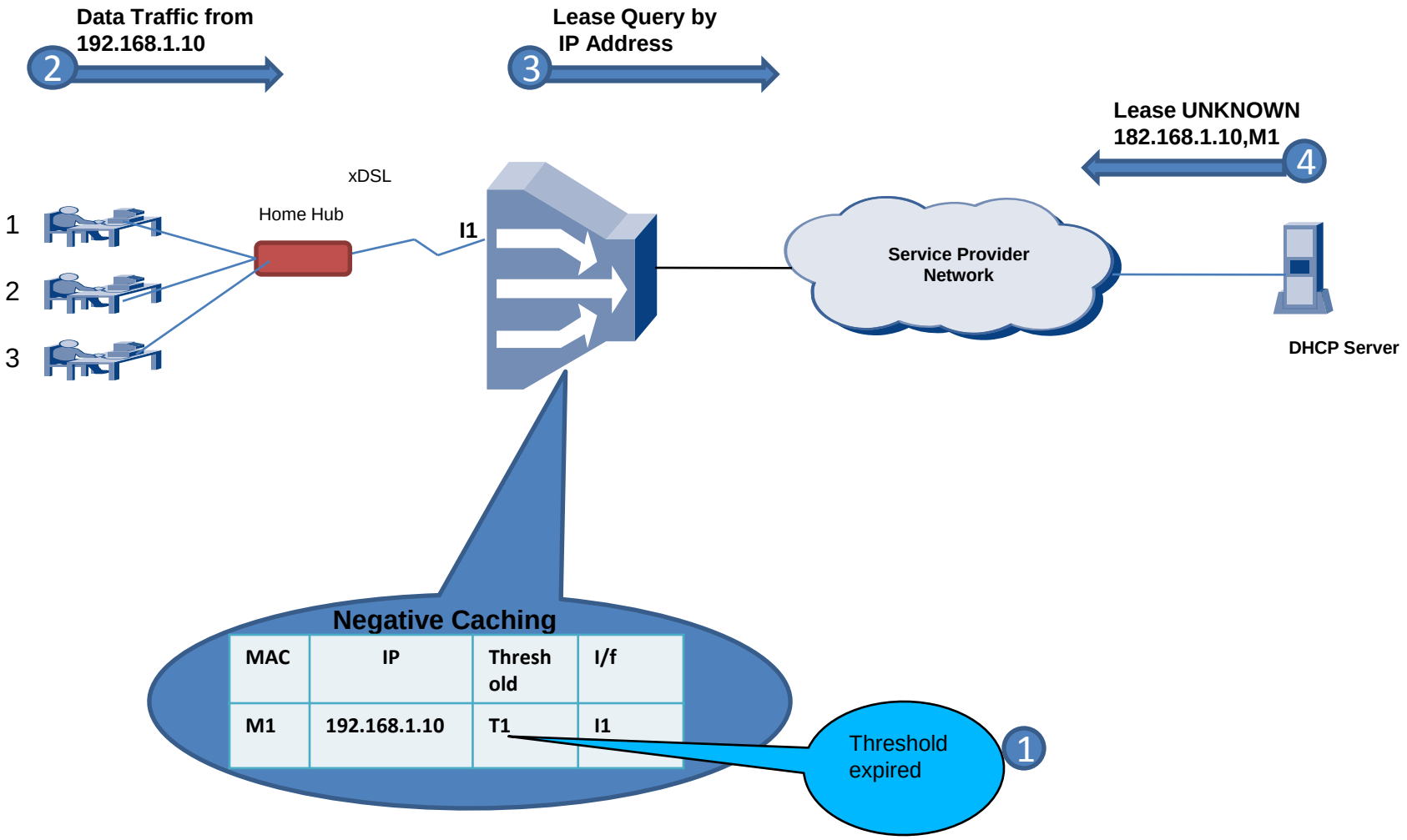
RFC 4388 based lease query – Negative Caching



RFC 4388 based lease query – Negative Caching



RFC 4388 based lease query – Negative Caching



Issues with RFC 4388 based lease query

- Existing Leasequery mechanism is data driven:
  - Leasequery is initiated only when Access Concentrators receives data
  - Existing method suggests the use of negative caching. Negative Caching consumes lot of resources under spoof attacks.
  - Results in increased outage time for the clients.

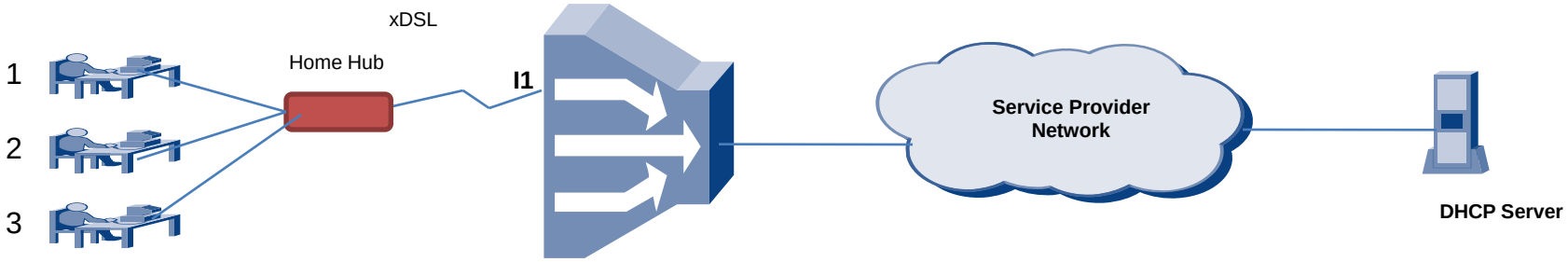
Issues with RFC 4388 based lease query (contd ..)

- Getting consolidated lease information per connection is not possible:
  - Existing mechanism doesn't have any methods to get consolidated lease information for all the clients belonging to a connection/circuit
  - Multiple clients can reside for a given connection/circuit.
  - If Access concentrator has lease information of all the clients for a given connection/circuit, anti-spoofing can be done in data plane (fast path)

Query by remote-id

- Remote-ID sub option identifyies a connection/circuit uniquely. This is globally unique identifier
- Remote-ID can be trusted as they are created by Relay Agent.
- Access Concentrator need not wait for the traffic to arrive and can generate LeaseQuery as soon as it comes up after a reboot.
- DHCP Server can provide consolidated Lease Information for a specific connection/circuit.
- Once all the lease information for a given connection/circuit is obtained, anti-spoofing can be done in data plane (fast path).
- No need for Negative Caching.

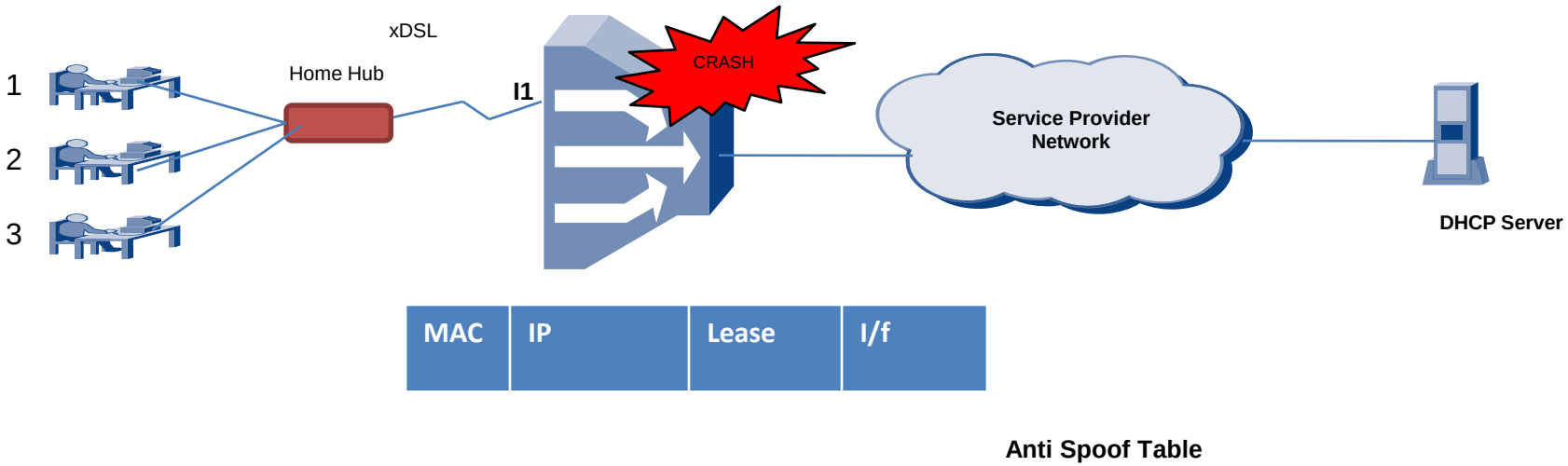
Lease Query by Remote Id



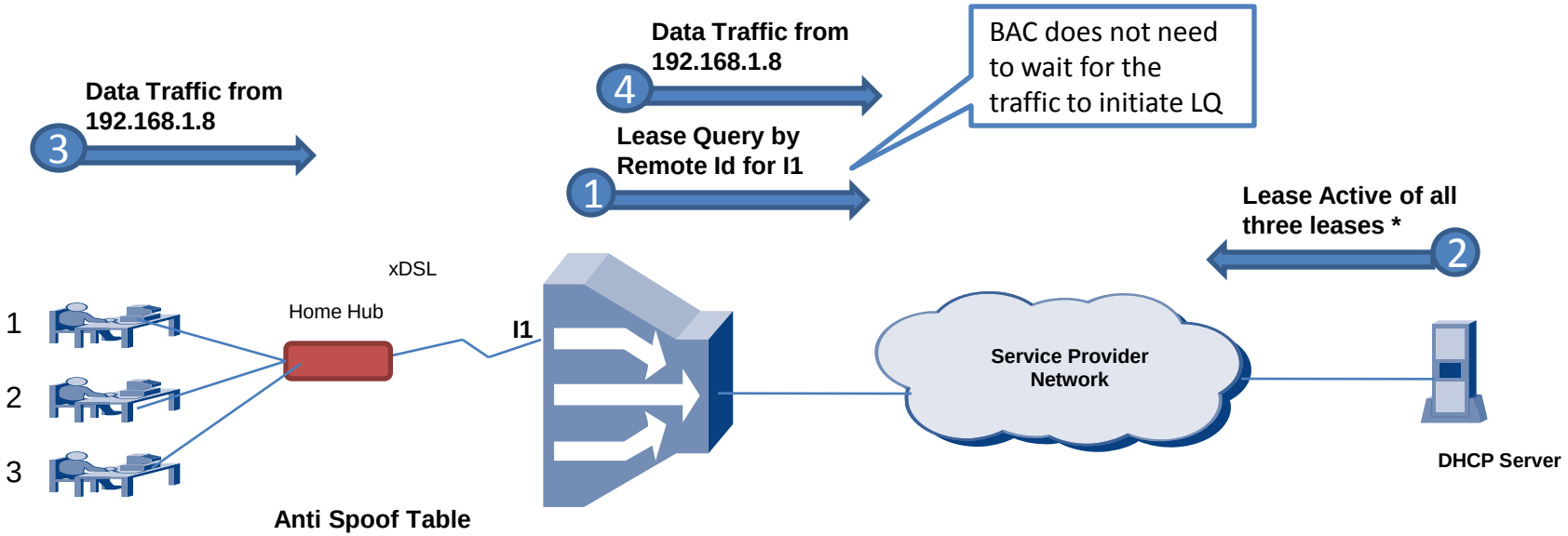
MAC	IP	Lease	I/f
M1	**192.168.1.2**	**T1**	**I1**
M2	192.168.1.8	T2	I1
M3	192.168.1.10	T3	I1

Anti Spoof Table

Lease Query by Remote Id



Lease Query by Remote Id



MAC	IP	Lease	I/f
M1	**192.168.1.2**	**T1**	**I1**
M2	192.168.1.8	T2	I1
M3	192.168.1.10	T3	I1

Leasequery by remote-id results in obtaining **complete information** on a given interface. No need of initiating subsequent queries

* Lease active for one lease is returned followed by associated-IP option. This results in subsequent query by IP for remaining leases

Protocol Details:

- Server identifies a Leasequery by remote-id when the leasequery message has:
  - Chaddr,siaddr,Ciaddr, htype, hlen and chaddr is zero and
  - Client identifier option is not present and
  - Option 82 with only Remote-Id sub-option is present.
- Sends a LEASEACTIVE populating the ciaddr with the IP address that was most recently accessed by the client. All other IP addresses are returned in Associated-IP option.
- Relay agent then sends a Leasequery with “Query by IP Address” for all the additional IP addresses returned in Associated-ip option.

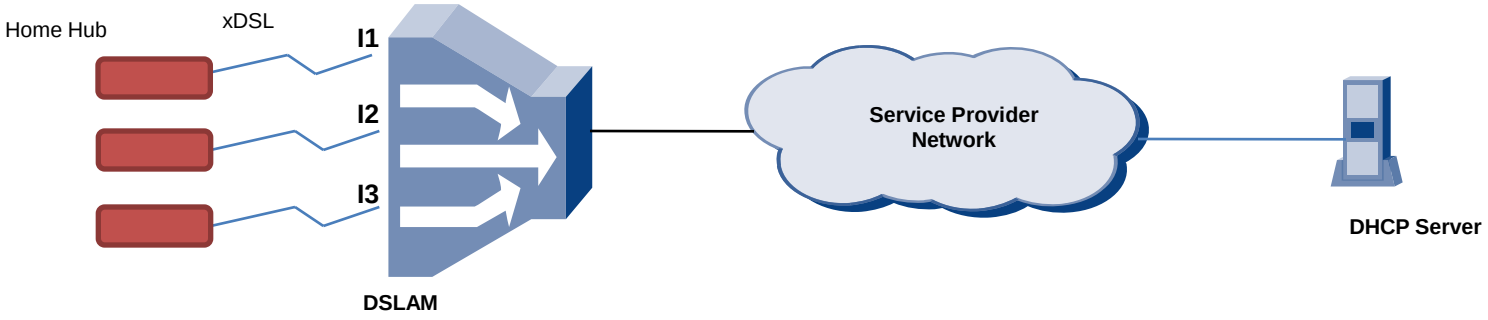
Protocol Details:

- Server may return a LEASEUNASSIGNED if it knows it manages the lease for the connection identified by Remote-Id sub-option but no lease is assigned yet.
- Server may return LEASEUNKNOWN if it does not know the corresponding Remote-id sub-option.

Why Bulk Leasequery?

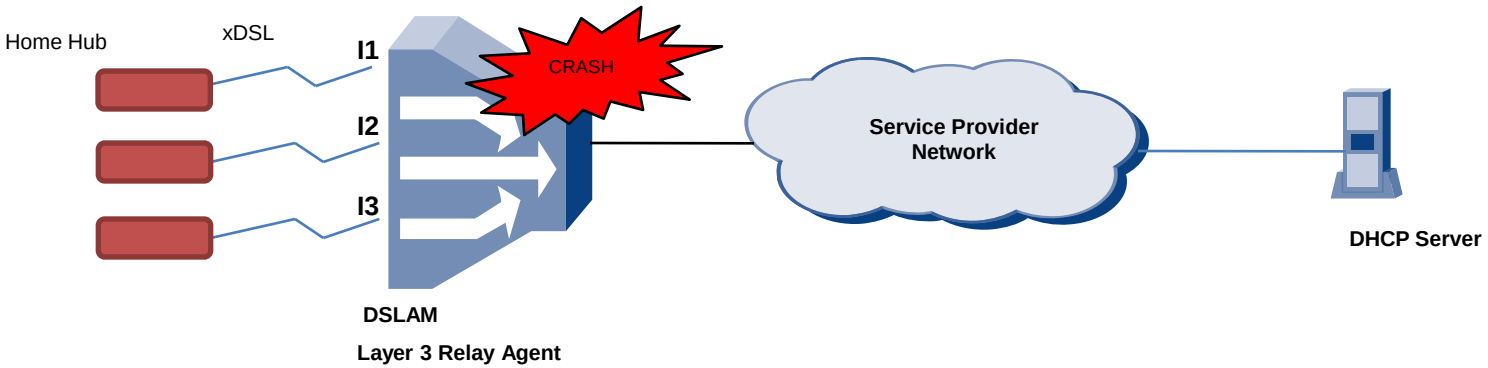
- Traditional leasequery (Both 4388) and leasequery by remote-id works on the principle of retrieving one lease at a time
- While query by remote-id solves all the problems associated with RFC 4388 based leasequery mechanism, it still involves generating huge number of leasequeries to get all the possible data
- Bulk leasequery works on the principle of establishing TCP connection between RA and Server and retrieving information in bulk

Bulk Lease Query



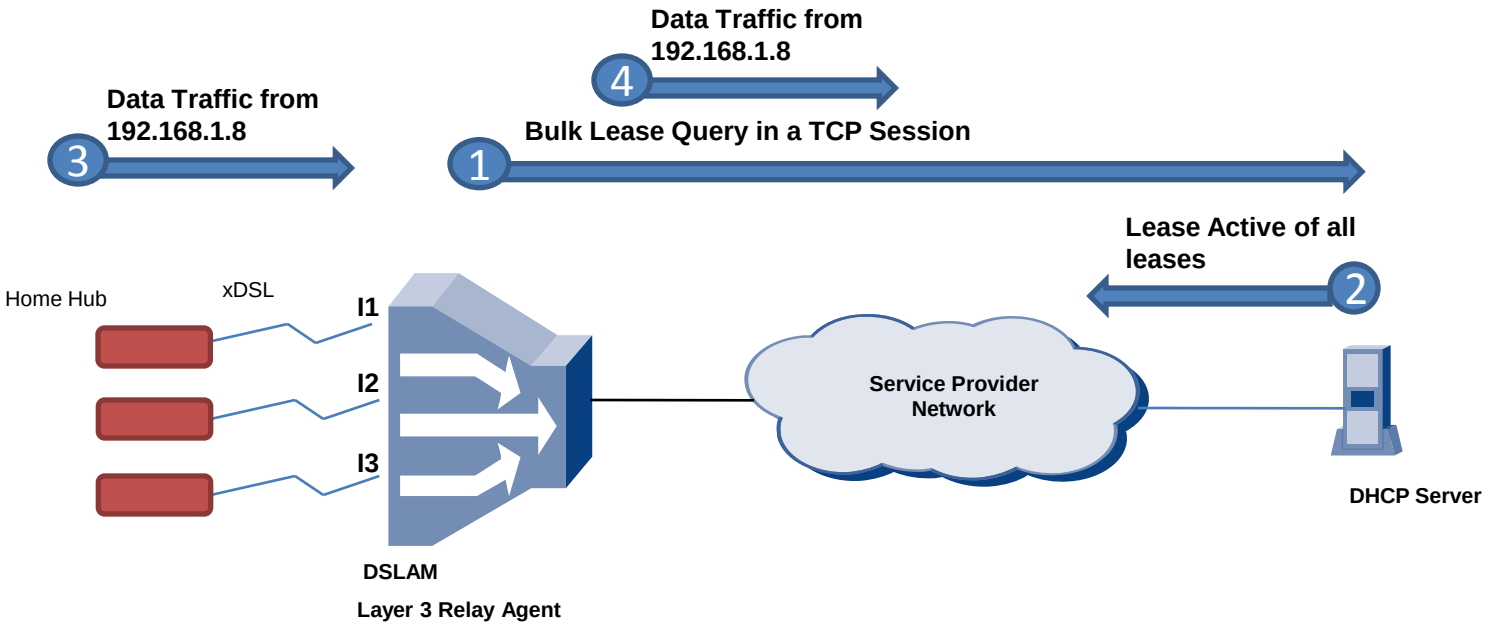
MAC	*IP Address*	*Lease Time*	*Interface*
M1	192.168.1.2	T1	I1
M2	192.168.1.8	T2	I2
M3	192.168.1.3	T2	I3

Bulk Lease Query



MAC	IP	Lease	I/f

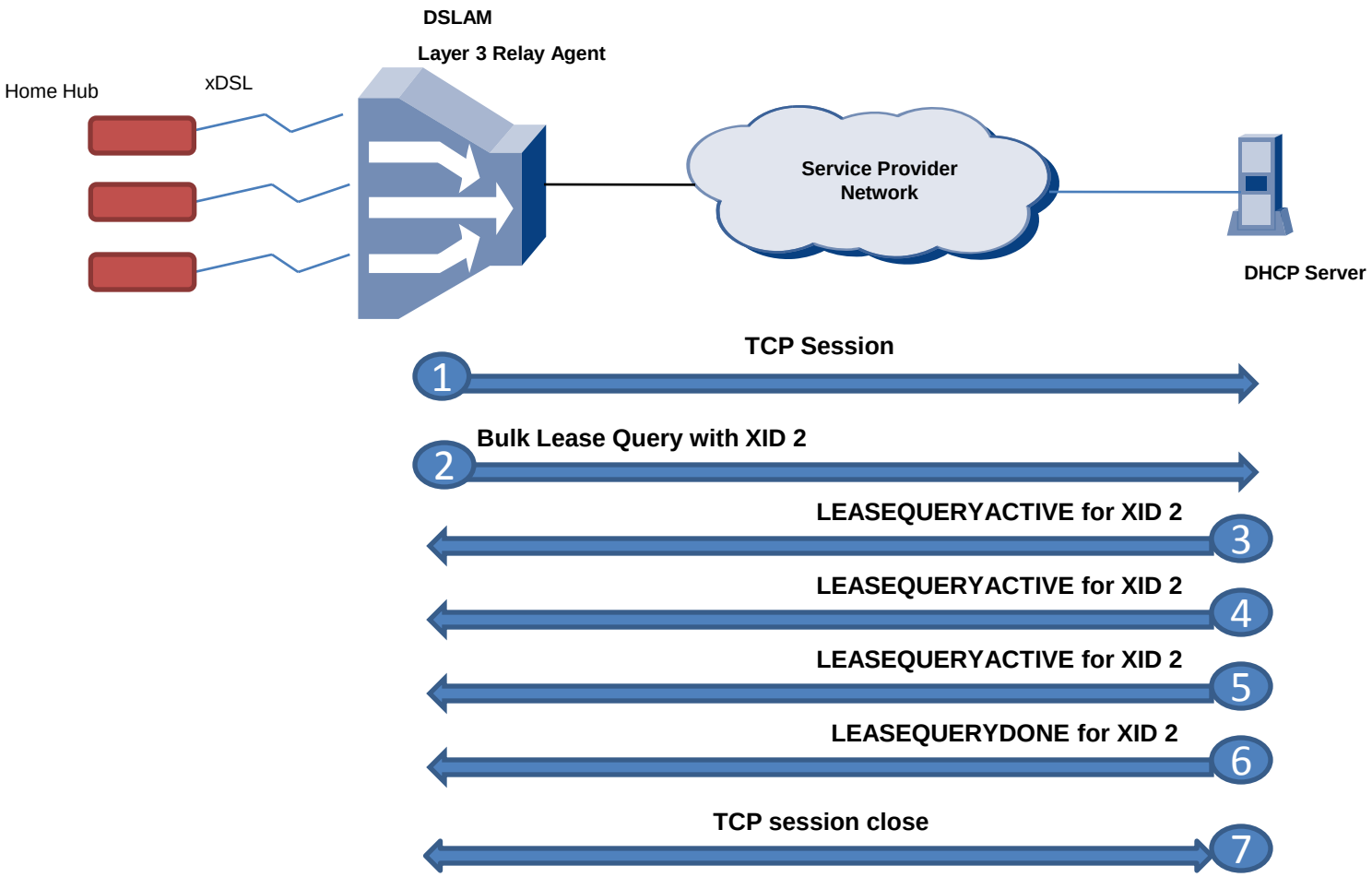
Bulk Lease Query



MAC	*IP Address*	*Lease Time*	*Interface*
M1	192.168.1.2	T1	I1
M2	192.168.1.8	T2	I2
M3	192.168.1.3	T2	I3

Lease information of all interfaces obtained in in one query

Protocol Details:



Protocol Details:

- A Querier (Typically a Relay Agent) establishes a TCP connection with the server on port 67.
- Two new query types are added
  - “Query by Relay-ID” where relay-id is a unique Relay agent Identifier. All leases allocated through a specific Relay Agent.
  - “Query for all configured IPs” where all IP address held by DHCP Server irrespective of state is returned. In this case, unassigned IP addresses are returned with UNASSIGNED state.
- New filters are added:
  - Start and End time filter can be passed to retrieve leases for which state has changed within the specified time.
- Other query types (Query by IP Address, MAC address, Client-ID and remote-id) are also supported.

Protocol Details:

- Upon receiving a BULKLEASEQUERY, DHCP server generates a stream of LEASEACTIVE for each lease that fulfils the query.
- End of lease for a given query is indicated by the LEASEQUERYDONE message.
- Multiple Bulk Leasequery can be initiated over a single TCP connection. Transaction id (XID) is used to distinguish between the replies for multiple queries.

Standardization and Implementation efforts

- **Standardization efforts:**
  - Query by remote-id and Bulk Lease Query draft is being standardized in DHC working group of IETF.

- **Implementation efforts:**
  - We have created a Proof-Of-Concept implementation of ‘Query by Remote-Id’ and ‘Bulk Lease Query’ by enhancing ISC DHCP server.

References:

- S. Bellovin, “Security problems in the TCP/IP protocol suite,” SIGCOMM Computer Communication Review, vol. 19, no. 2, pp. 32–48, 1989.
- R. Beverly and S. Bauer, “The spoofer project: inferring the extent of source address filtering on the internet,” in SRUTI’05: Proc. of the Steps to Reducing Unwanted Traffic on the Internet, 2005.
- IETF Standards:
  - RFC 2131, Dynamic Host Configuration Protocol
  - Layer 2 Relay Agent
    - http://www.ietf.org/id/draft-ietf-dhc-l2ra-04.txt
    - http://www.ietf.org/id/draft-ietf-dhc-l2ra-extensions-01.txt
  - Query by remote-id
    - http://www.ietf.org/id/draft-ietf-dhc-leasequery-by-remote-id-02.txt
  - Bulk lease query
    - http://www.ietf.org/id/draft-ietf-dhc-dhcpv4-bulk-leasequery-00.txt
- TR-101 from Broadband Forum
  - http://www.broadband-forum.org/technical/download/TR-101.pdf

Infosys®

Thank You