APT: A Practical Transit-Mapping Service

Dan Jen, Michael Meisel, Dan Massey, Lan Wang,
Beichuan Zhang, and Lixia Zhang

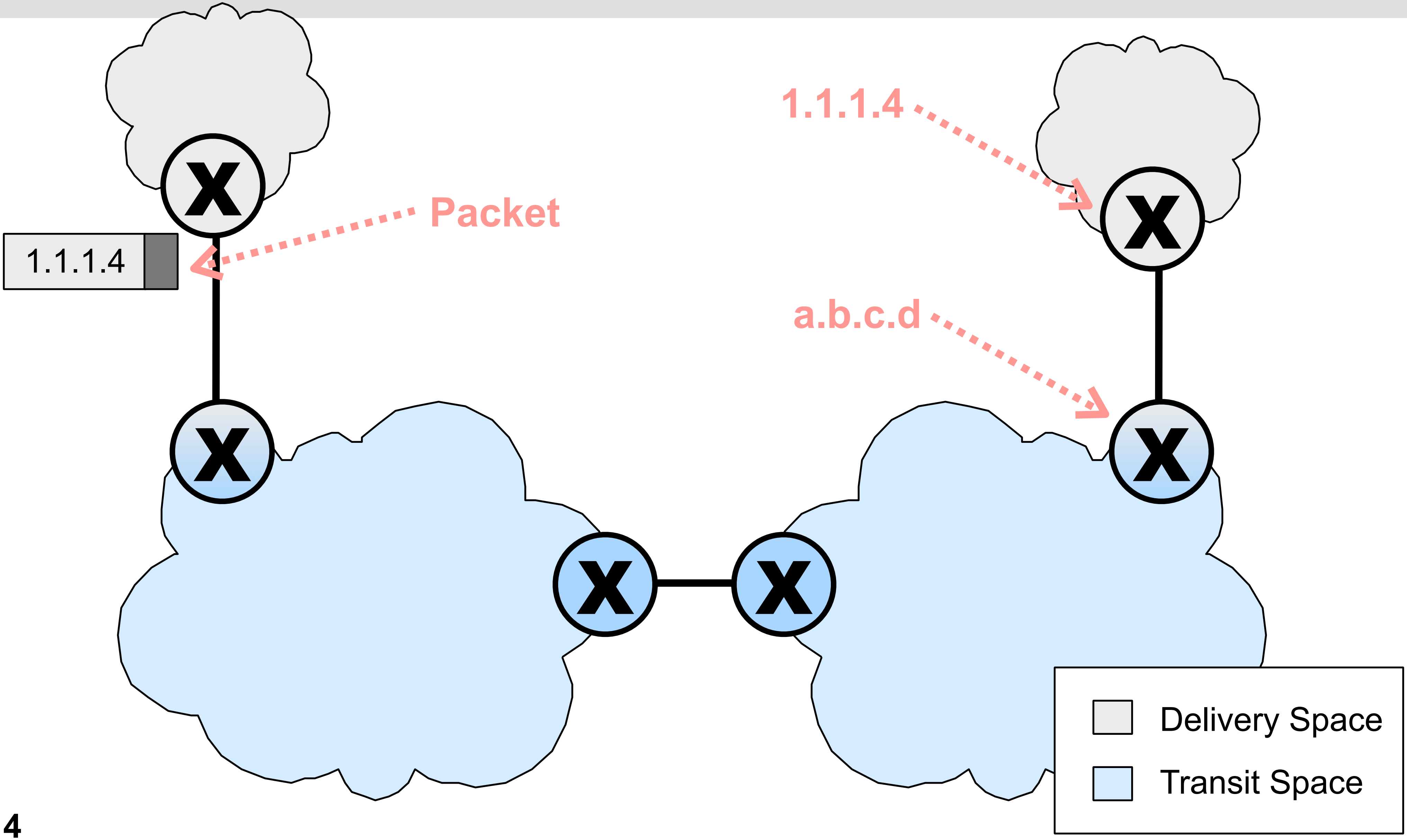
Motivation

- The current BGP routing system doesn’t scale
  - Router hardware may not be able to keep up
- The conflict between ISPs and their customers
  - ISPs want aggregatable addresses
  - Their customers want
    - Multihoming
    - Better traffic engineering
    - Provider-independent (PI) addresses

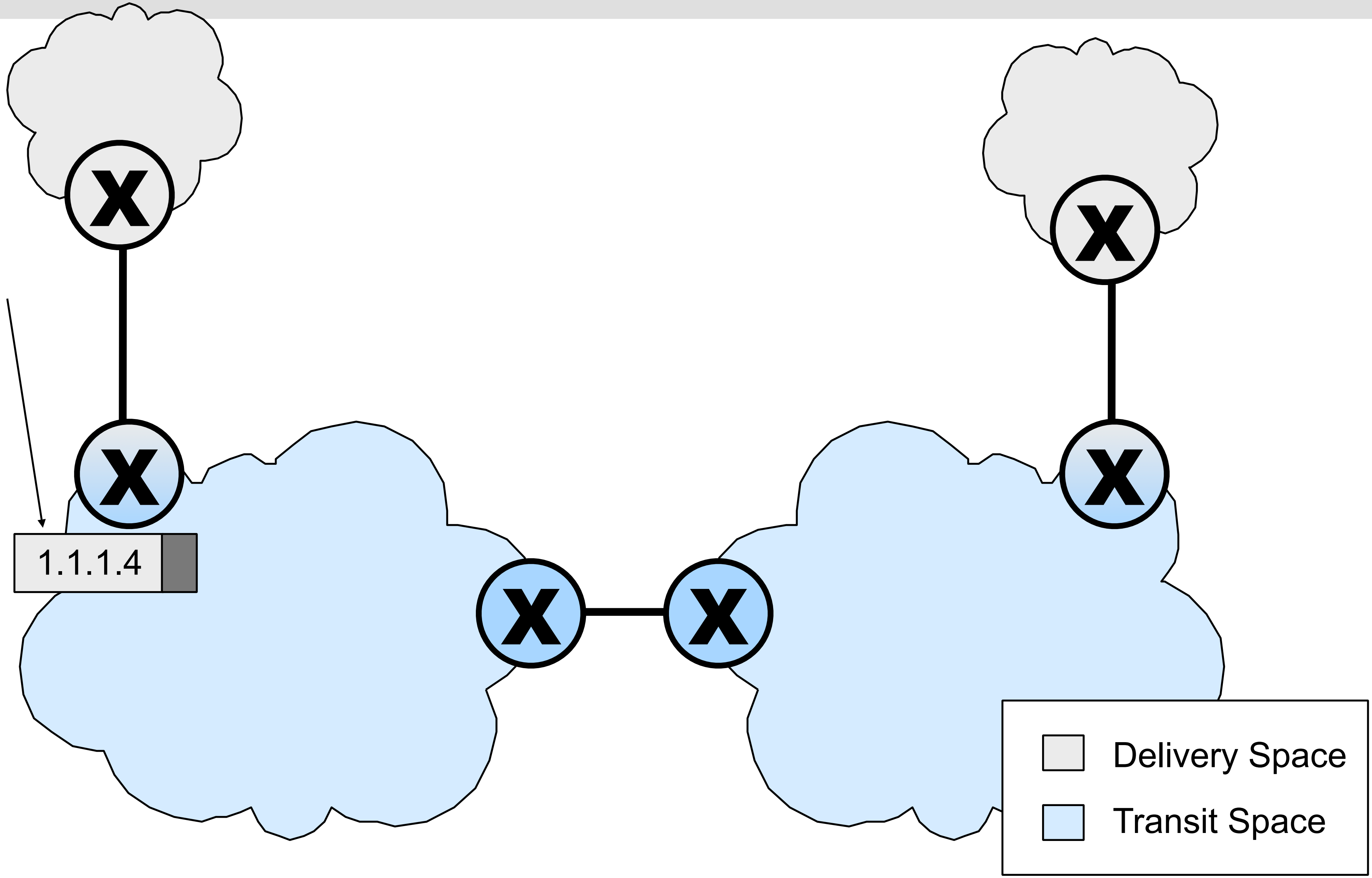
A General Solution

- Divide the Internet into two address spaces
  - Delivery space
  - Transit space
- Delivery space packets are UDP tunneled through transit space
- Transit addresses (Taddrs) appear in the global routing table, delivery addresses (Daddrs) do not
- LISP also falls into this category
  - LISP EIDs <=> APT Daddrs
  - LISP RLOCs <=> APT Taddrs

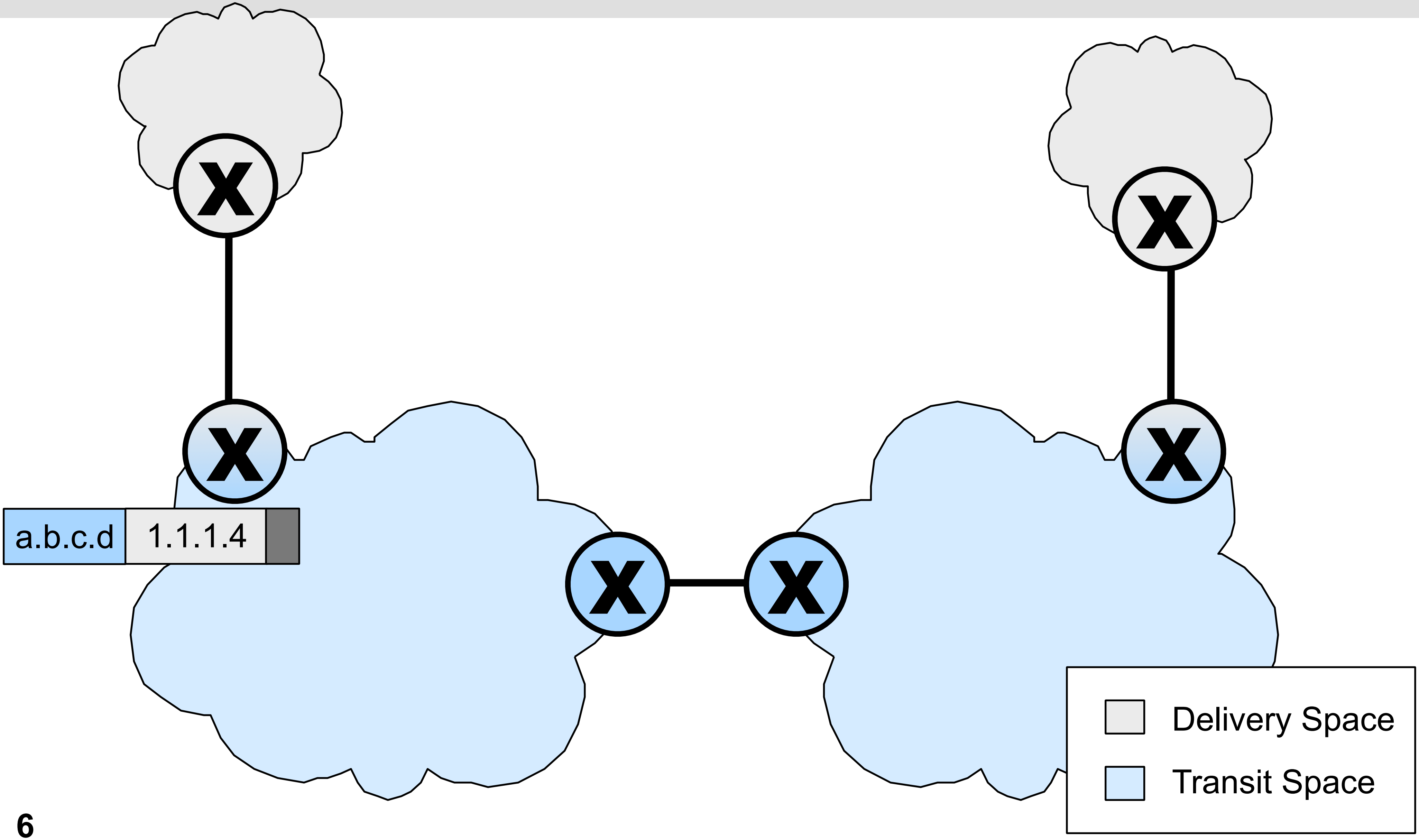
Tunneling Example



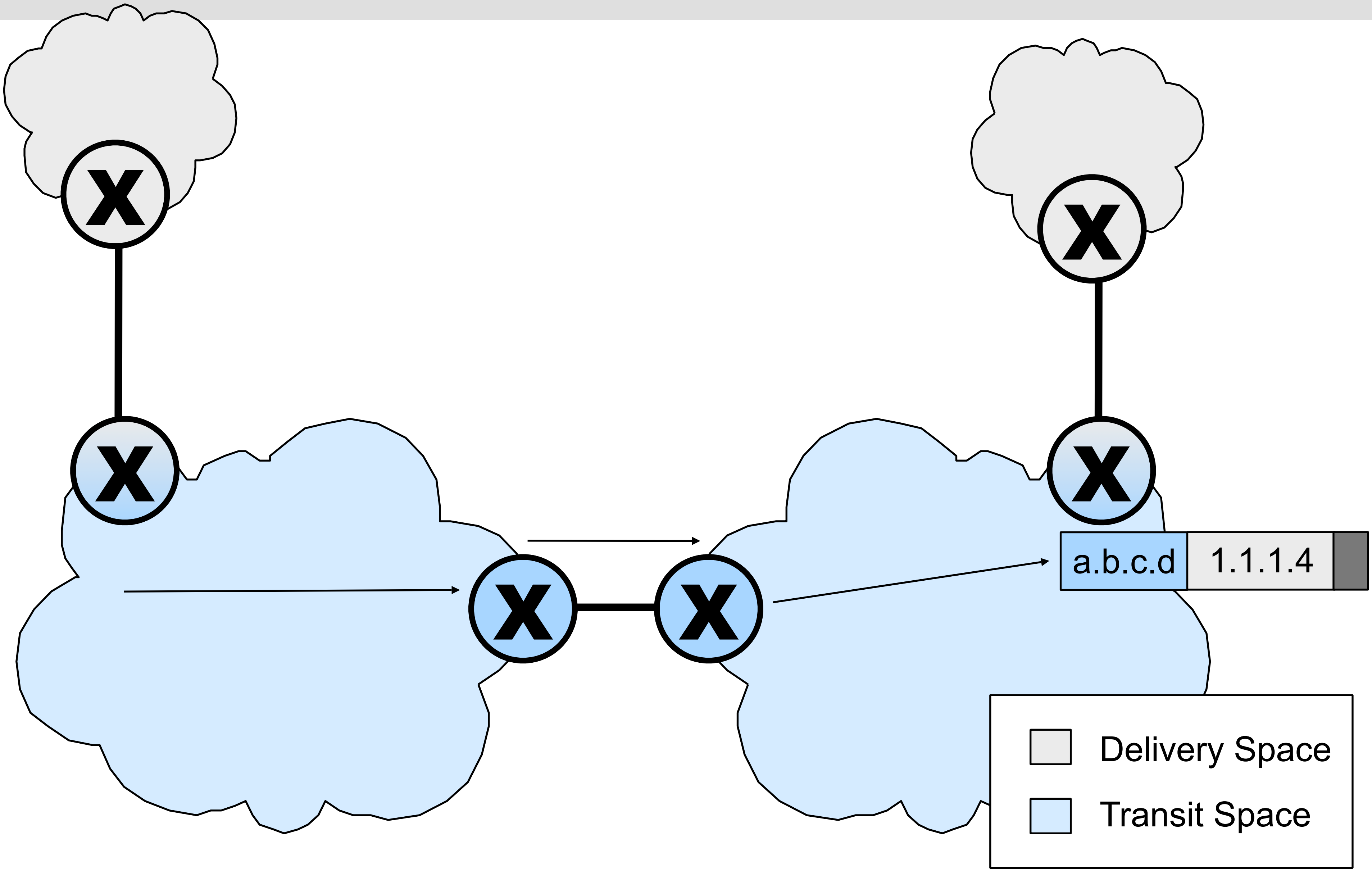
Packet Arrives at ISP



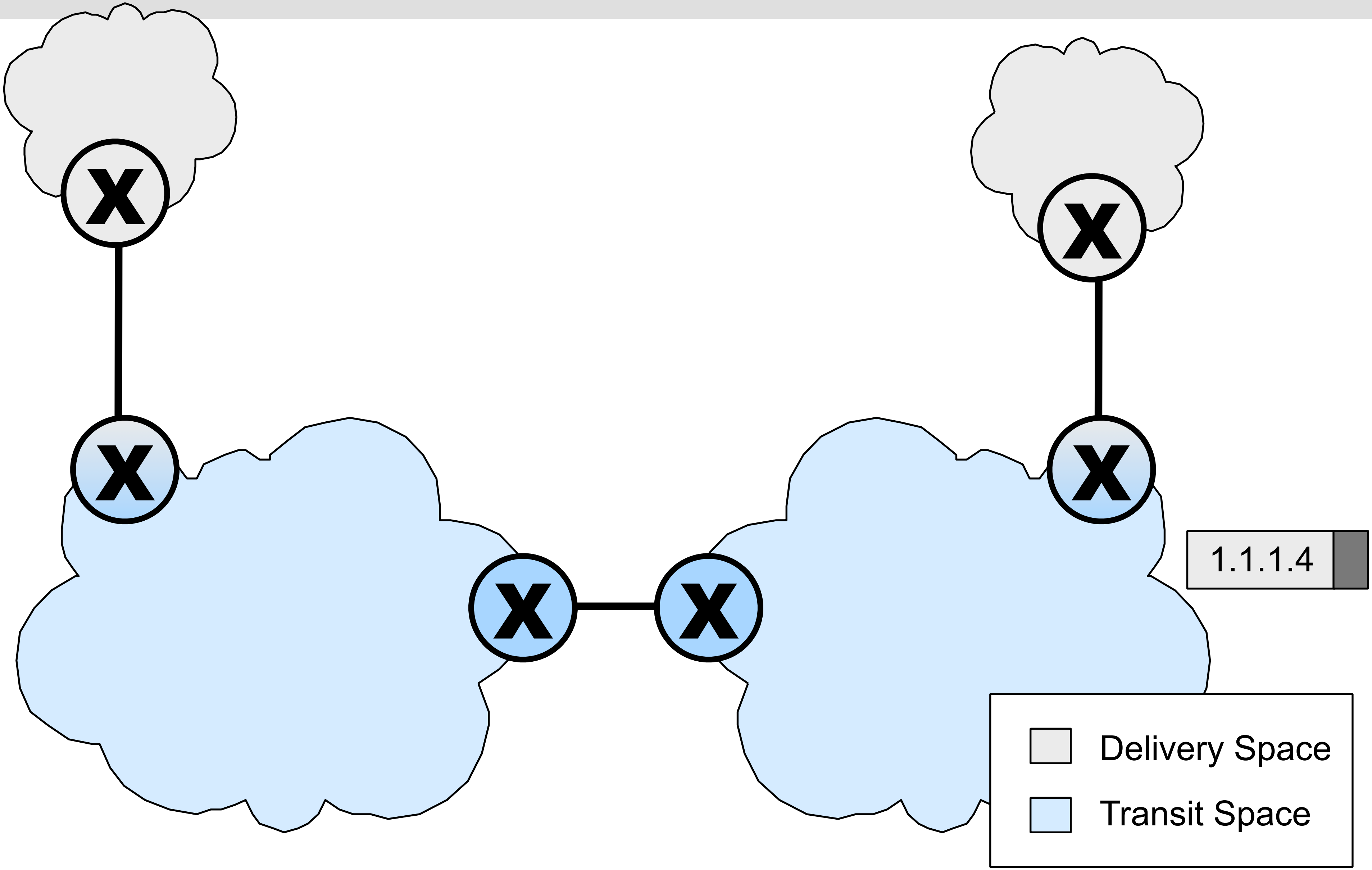
Packet Encapsulated in Transit Space Header



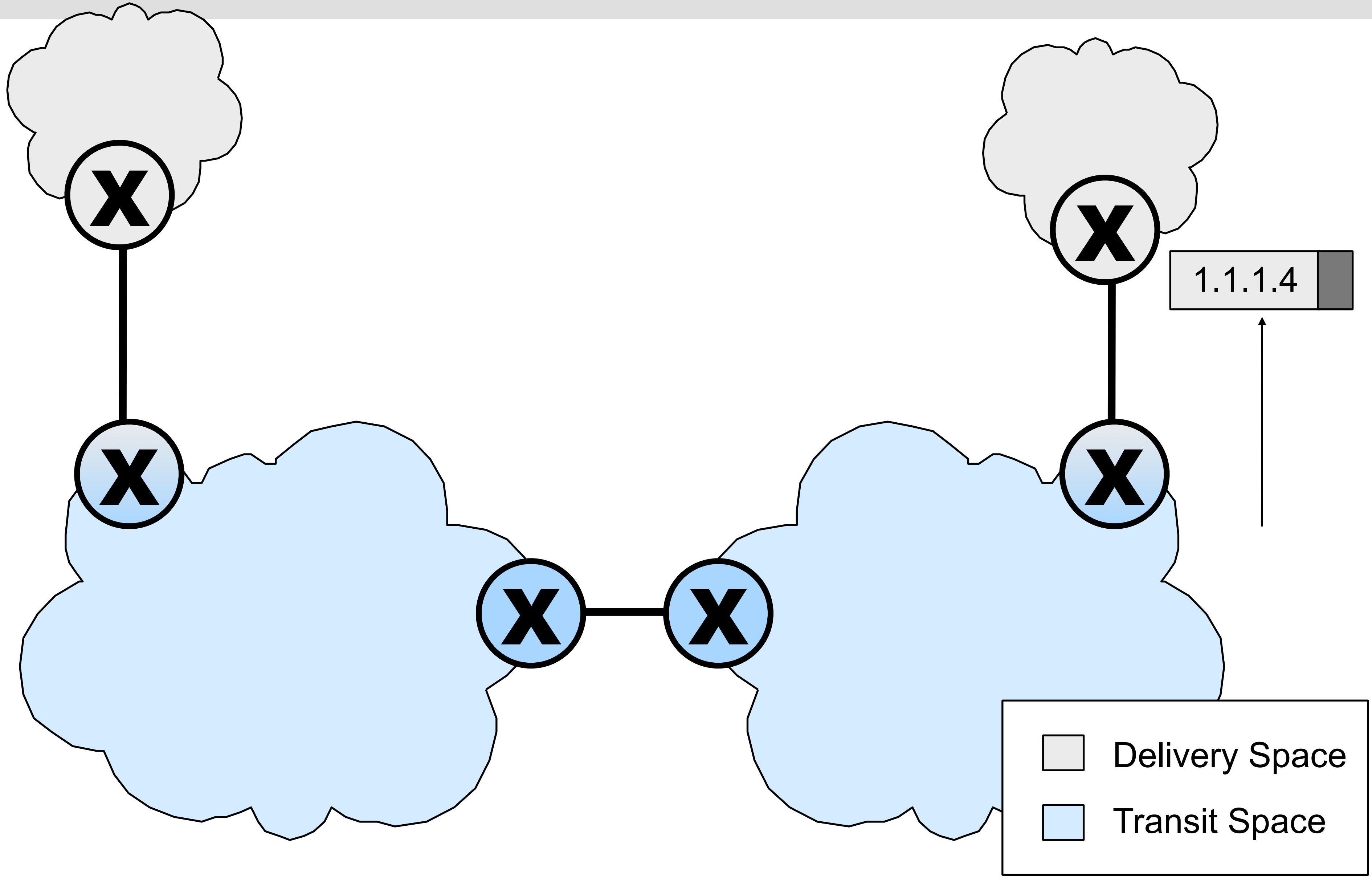
Packet Delivered across Transit Space



Packet Decapsulated



Packet Delivered



Connecting the Two Address Spaces

- The source transit address (Taddr) is the encapsulating router
- But what is the destination Taddr?
- We have to ask APT -- the mapping service.

New Device Types Required for APT

- Default mappers
  - An additional device in each transit network (TNs)
  - Q: Would it be practical to build them on a router platform?
- Tunnel routers (“TRs”)
  - Replace provider-edge (PE) routers
  - Q: Can currently deployed PE routers become TRs with only a software update?

Default Mappers

- Store *all* Daddr-prefix-to-Taddr mappings (MapSets)
  - Each Daddr prefix maps to a non-empty **set** of Taddrs
  - As many Taddrs per MapSet as providers per delivery network (DN)
  - Each Taddr has a priority for multihoming support
- At least one default mapper per transit network (TN)
  - Any default mapper can be reached using the same anycast address for reliability

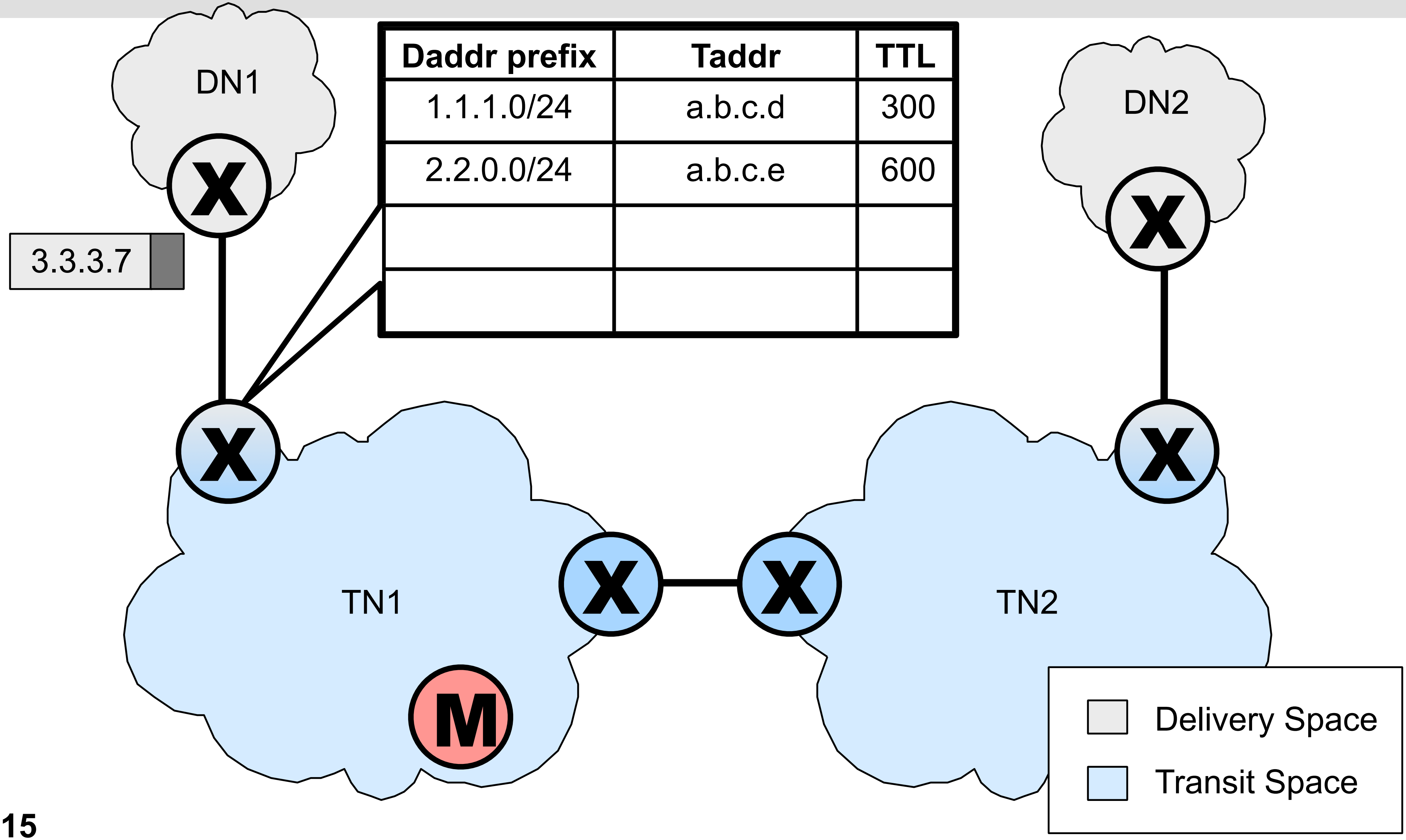
Tunnel Routers (TRs)

- Encapsulate outgoing packets (ITR mode)
- Decapsulate incoming packets (ETR mode)
- Cache only Daddr-to-single-Taddr mappings (MapRecs)
- Cache only MapRecs that are currently in use
  - Delete after the MapRec's time to live (TTL) expires
  - No MapRec? Tunnel the packet to a default mapper.
  - Default mapper re-tunnels the packet to an ETR for you and responds with a Cache Add Message containing a MapRec

Terminology Review

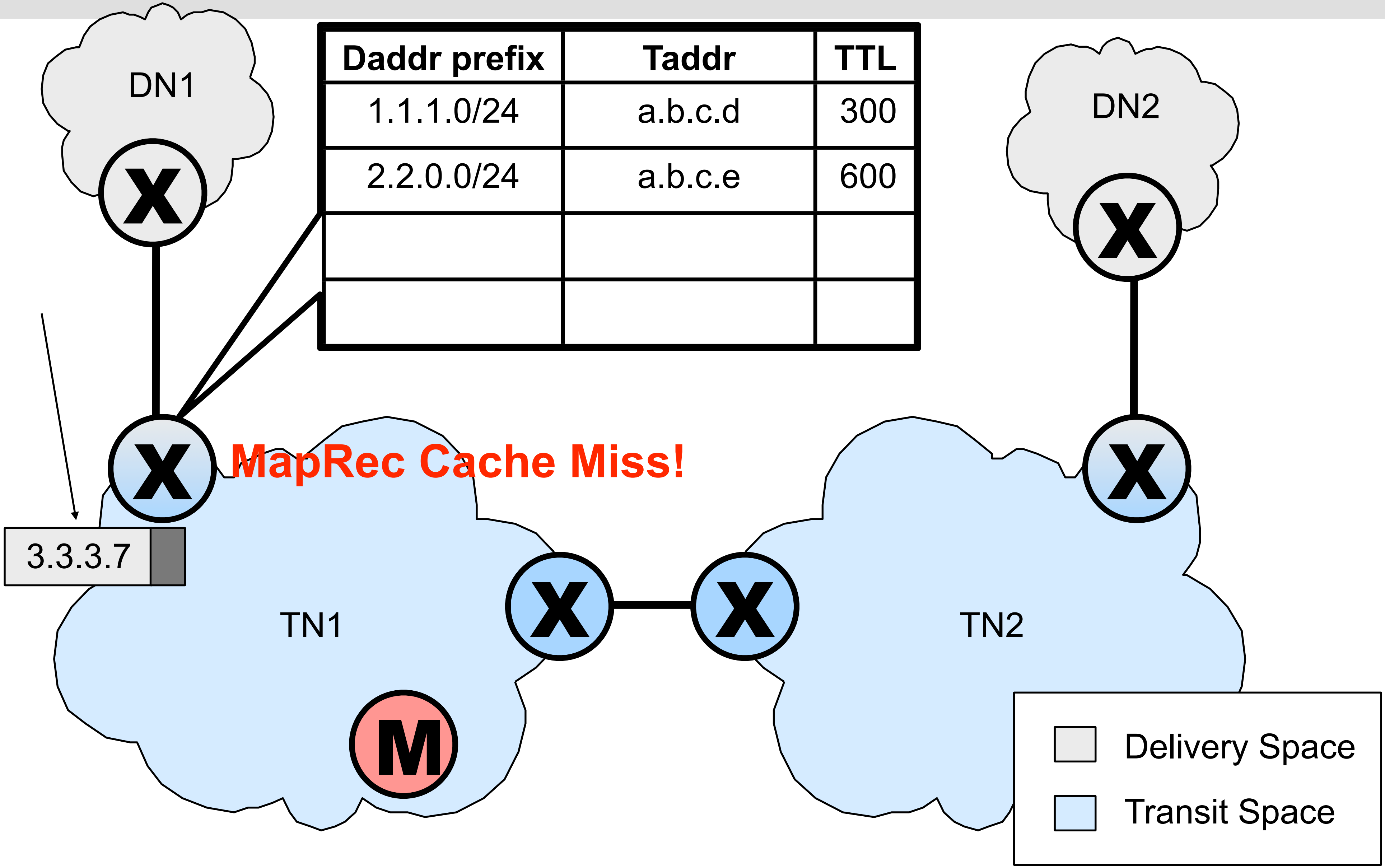
- Transit Network (TN)
  - An AS that provides packet transport services, but not endpoints
- Transit Address (Taddr)
  - An address in the address space used by TNs
- Delivery Network (DN)
  - A network that is a source or destination of IP packets
- Delivery Address (Daddr)
  - An address in the address space used by DNs
- MapSet
  - Maps a Daddr prefix to a non-empty SET of ETR Taddrs, used by default mappers
- MapRec
  - Maps a Daddr prefix to a single ETR Taddr, used by TRs

APT Example



Daddr prefix	Taddr	TTL
1.1.1.0/24	a.b.c.d	300
2.2.0.0/24	a.b.c.e	600

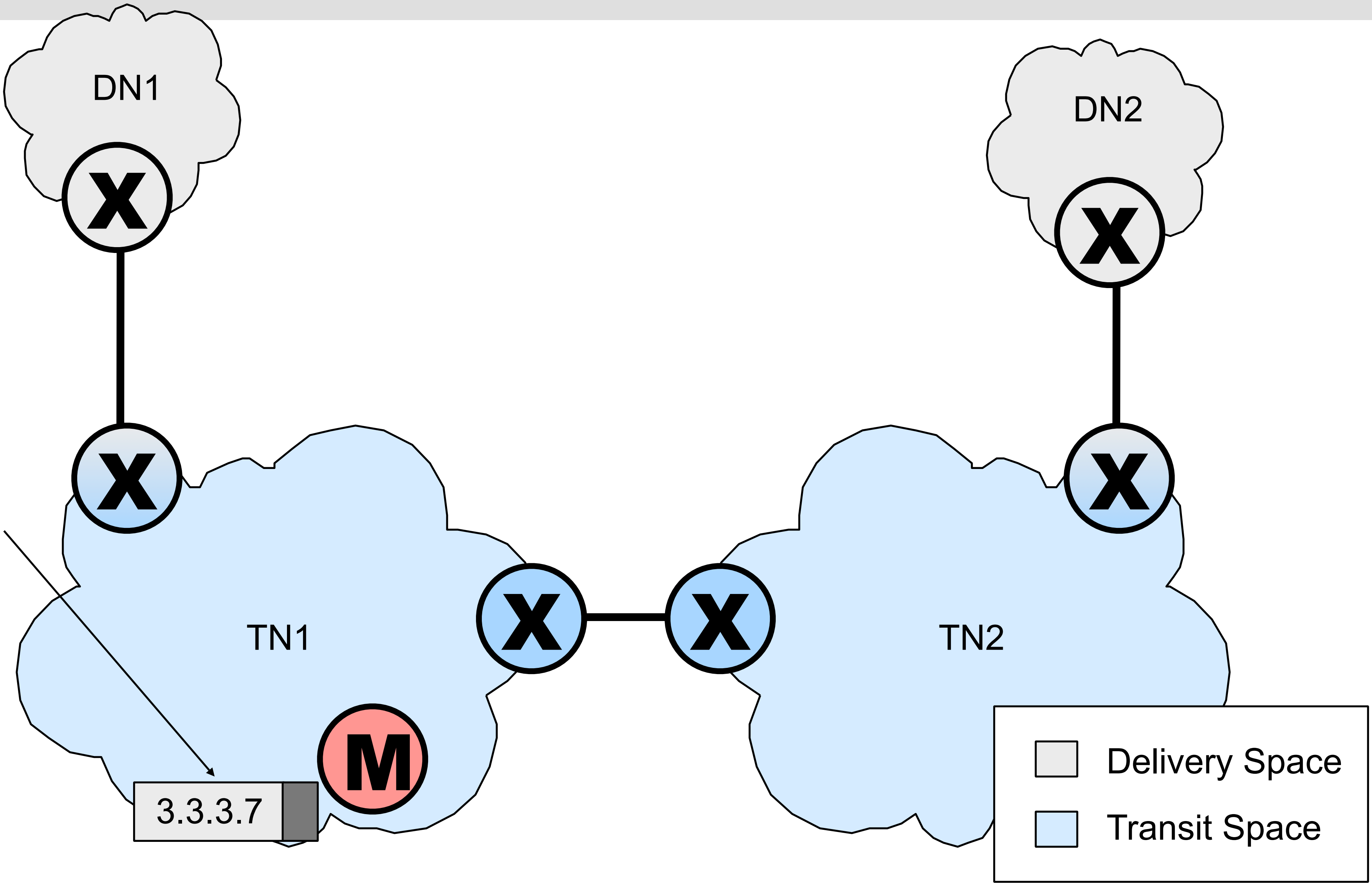
MapRec Not in Cache



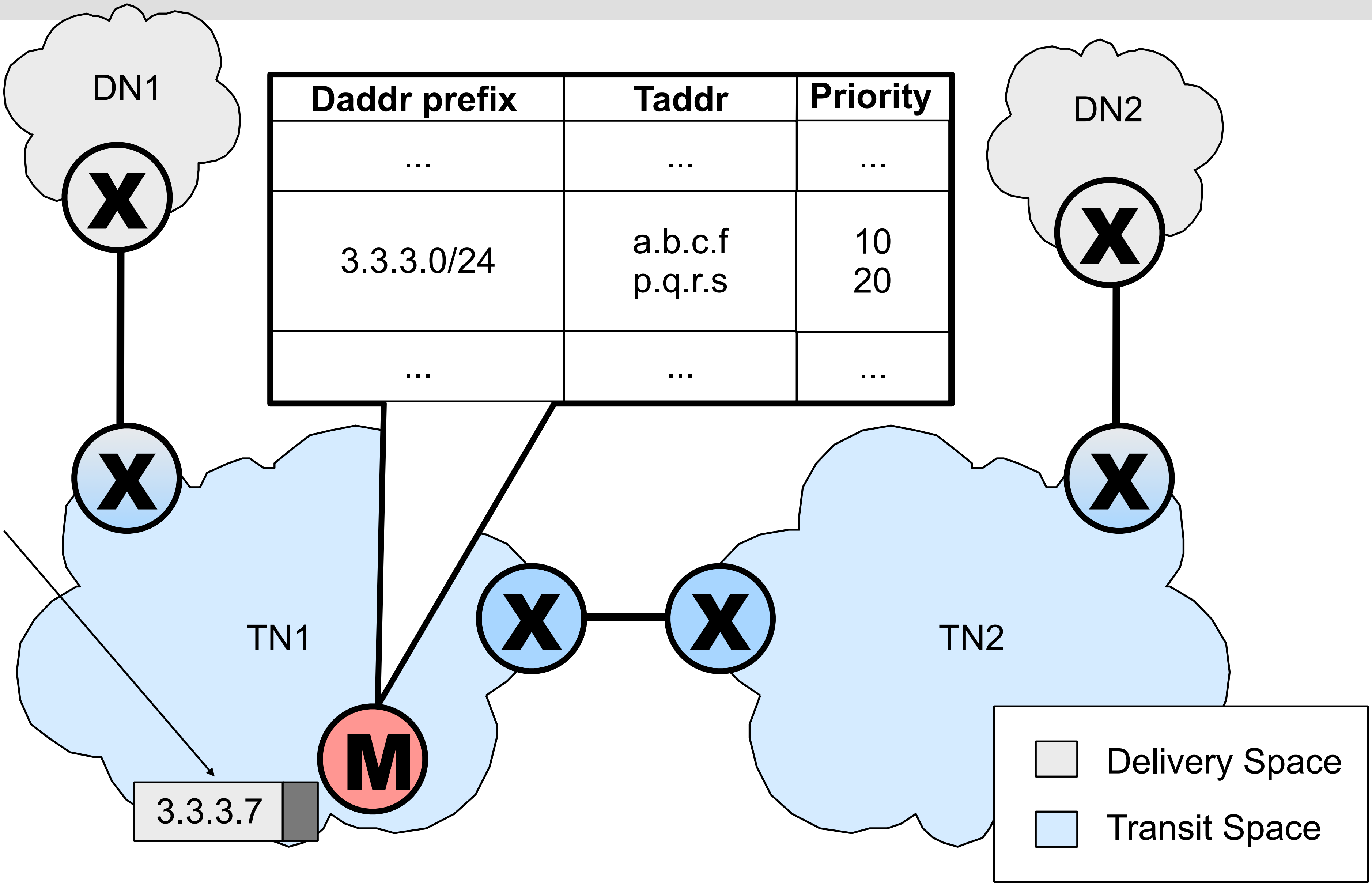
Daddr prefix	Taddr	TTL
1.1.1.0/24	a.b.c.d	300
2.2.0.0/24	a.b.c.e	600

MapRec Cache Miss!

Use the Default Mapper

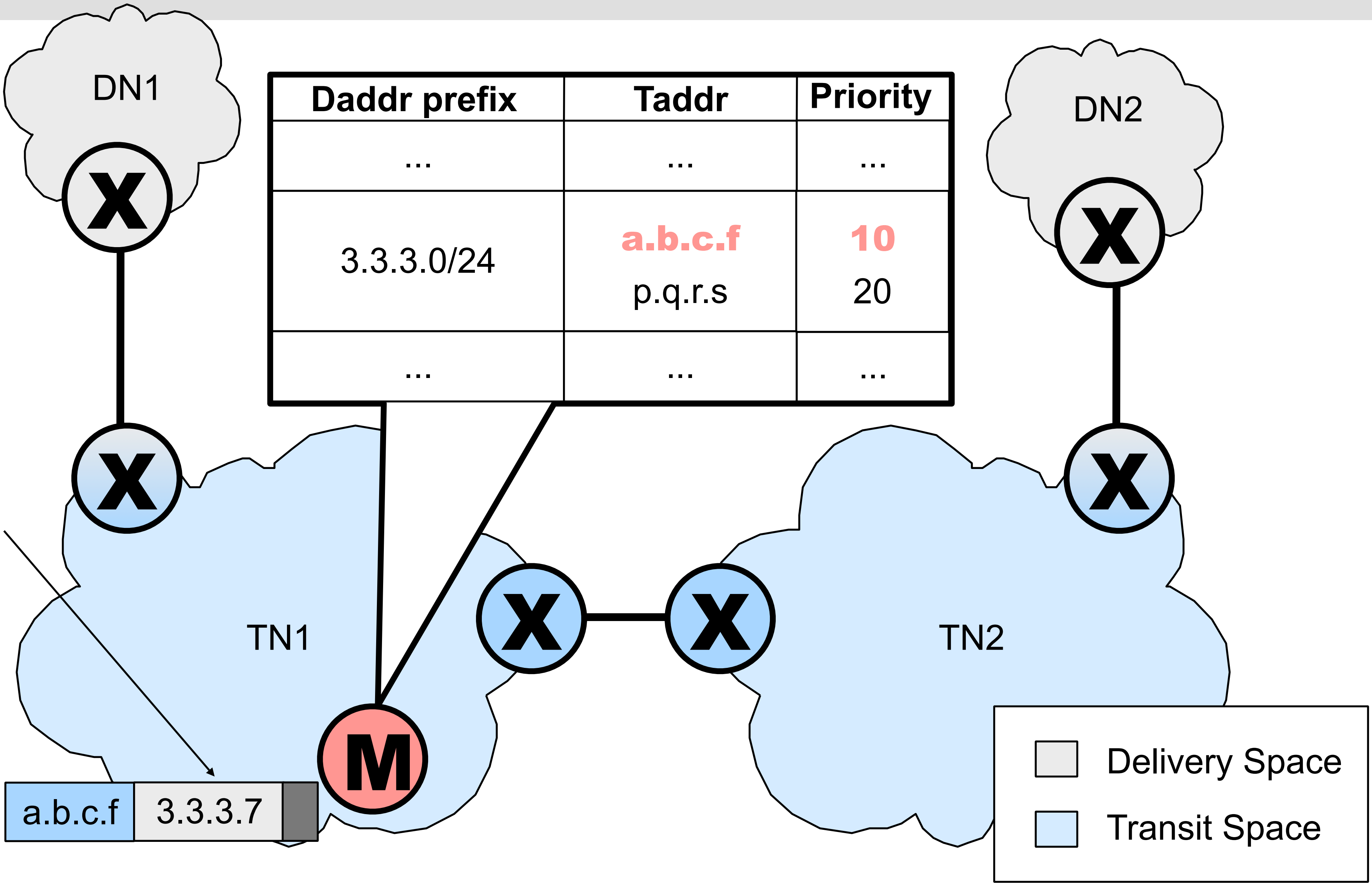


Daddr Prefix is Multihomed



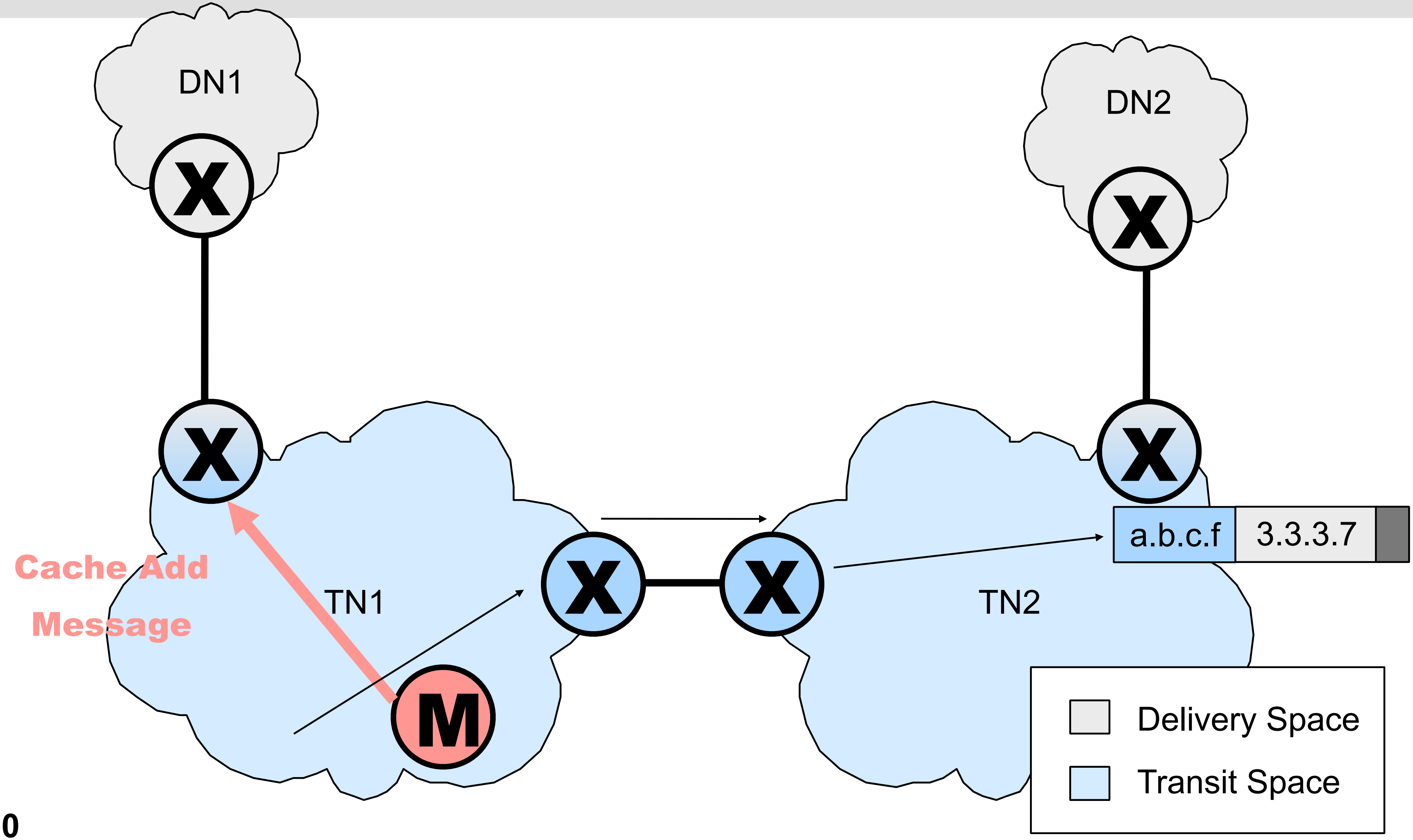
Daddr prefix	Taddr	Priority
...	...	...
3.3.3.0/24	a.b.c.f p.q.r.s	10 20
...	...	...

Default Mapper Selects a MapRec

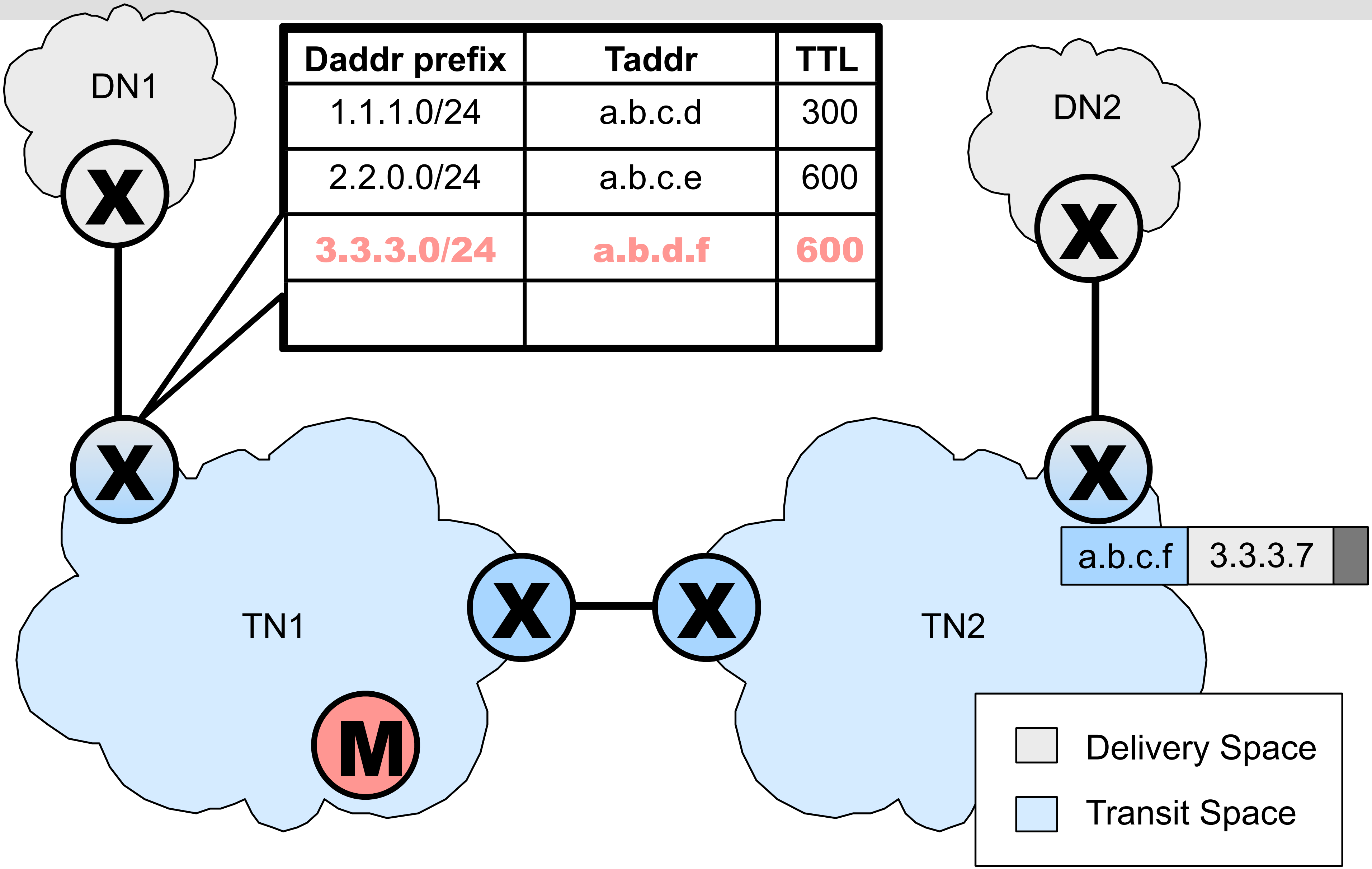


Daddr prefix	Taddr	Priority
...	...	...
3.3.3.0/24	**a.b.c.f** p.q.r.s	**10** 20
...	...	...

Default Mapper Responds with MapRec and Delivers Packet

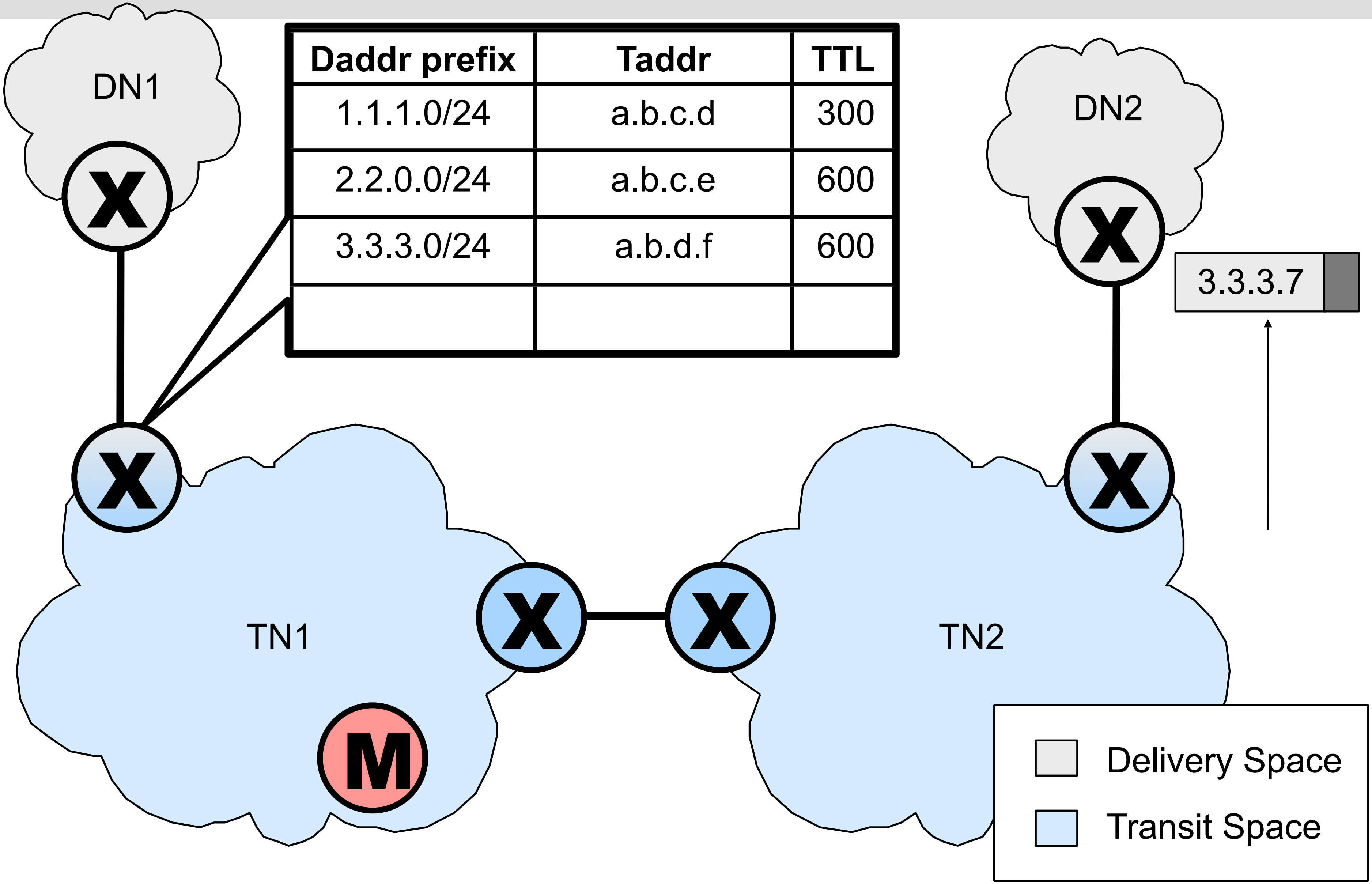


MapRec Added to Cache



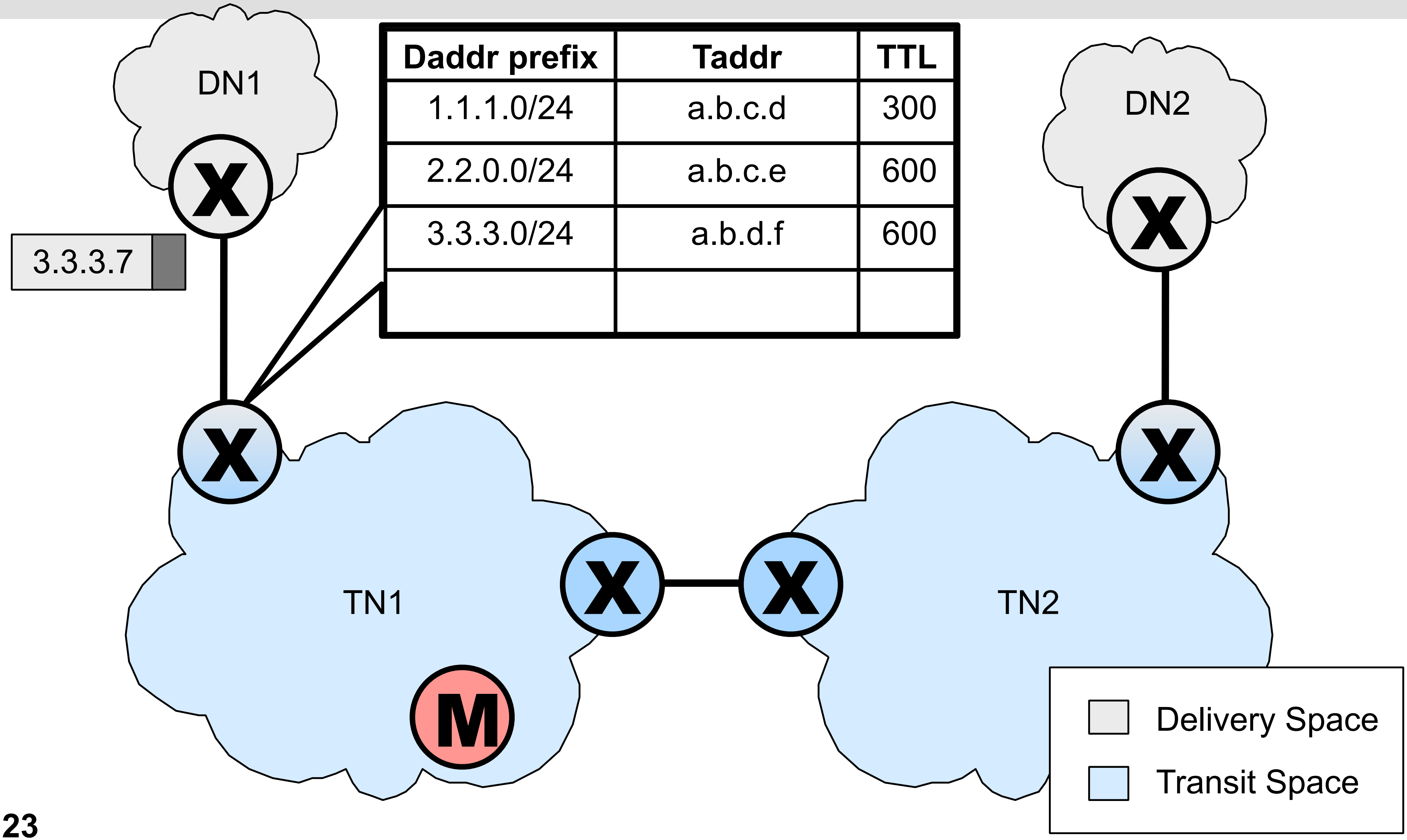
Daddr prefix	Taddr	TTL
1.1.1.0/24	a.b.c.d	300
2.2.0.0/24	a.b.c.e	600
3.3.3.0/24	**a.b.d.f**	**600**

Packet Decapsulated and Delivered



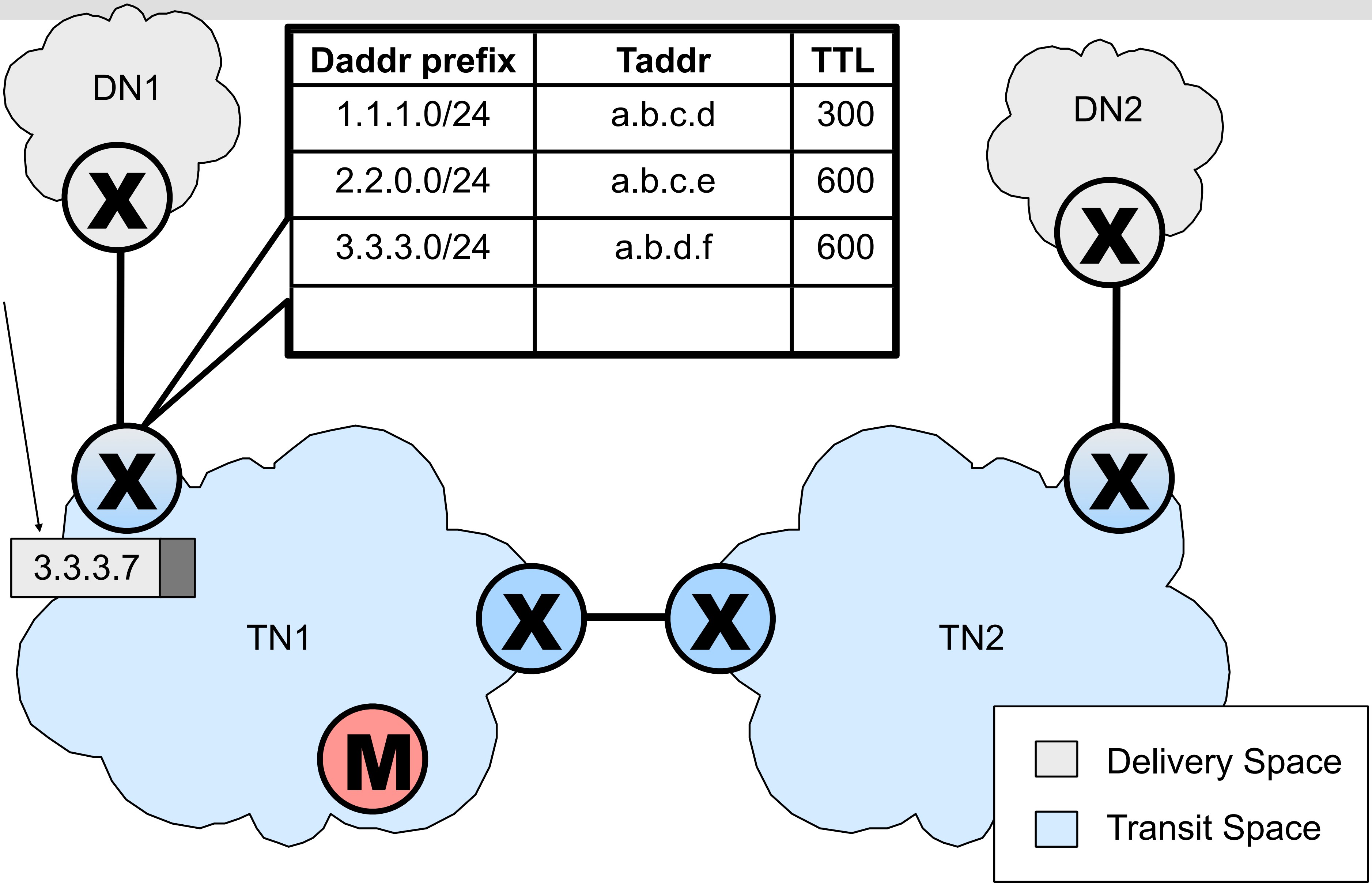
Daddr prefix	Taddr	TTL
1.1.1.0/24	a.b.c.d	300
2.2.0.0/24	a.b.c.e	600
3.3.3.0/24	a.b.d.f	600

Next Packet



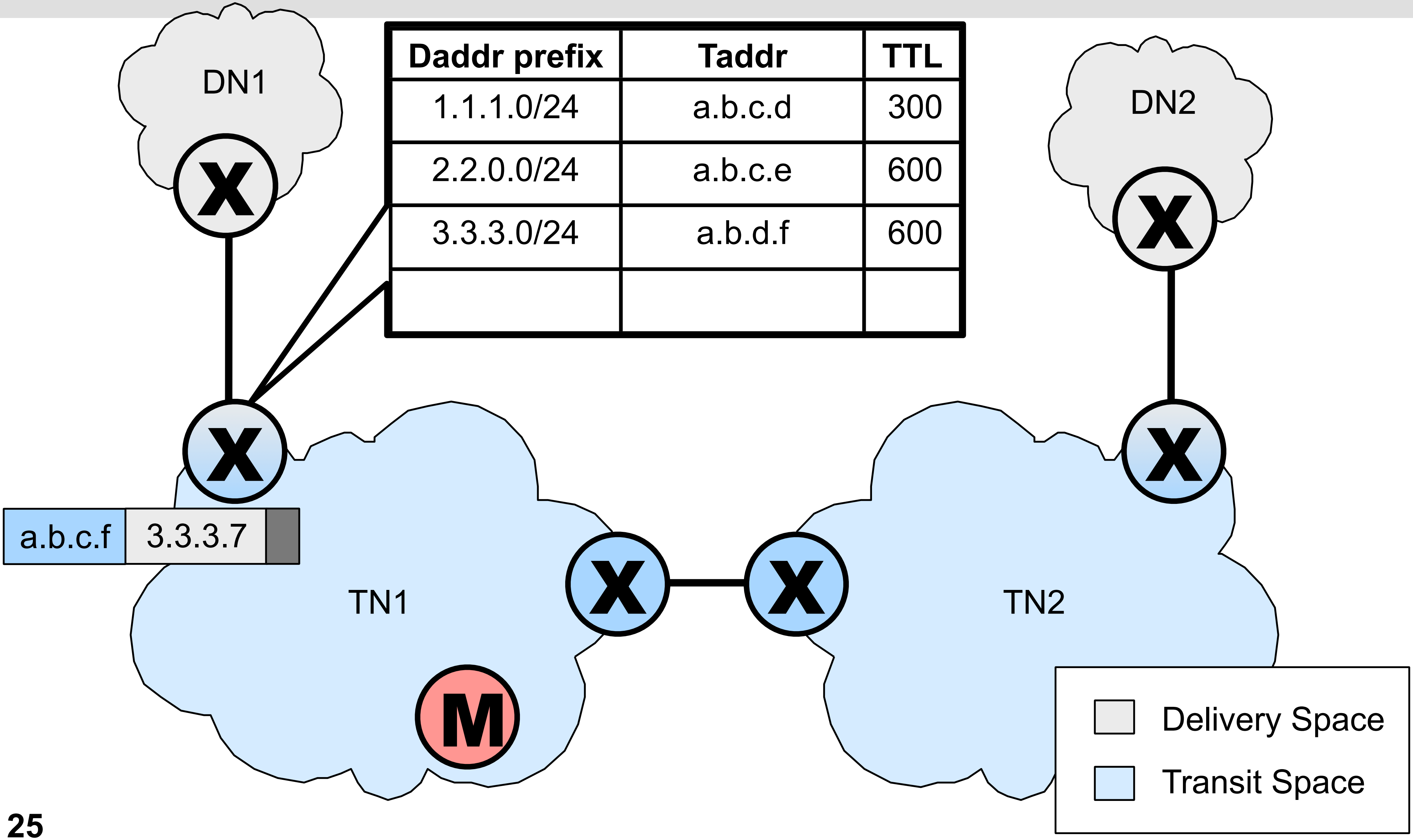
Daddr prefix	Taddr	TTL
1.1.1.0/24	a.b.c.d	300
2.2.0.0/24	a.b.c.e	600
3.3.3.0/24	a.b.d.f	600

MapRec Already in Cache



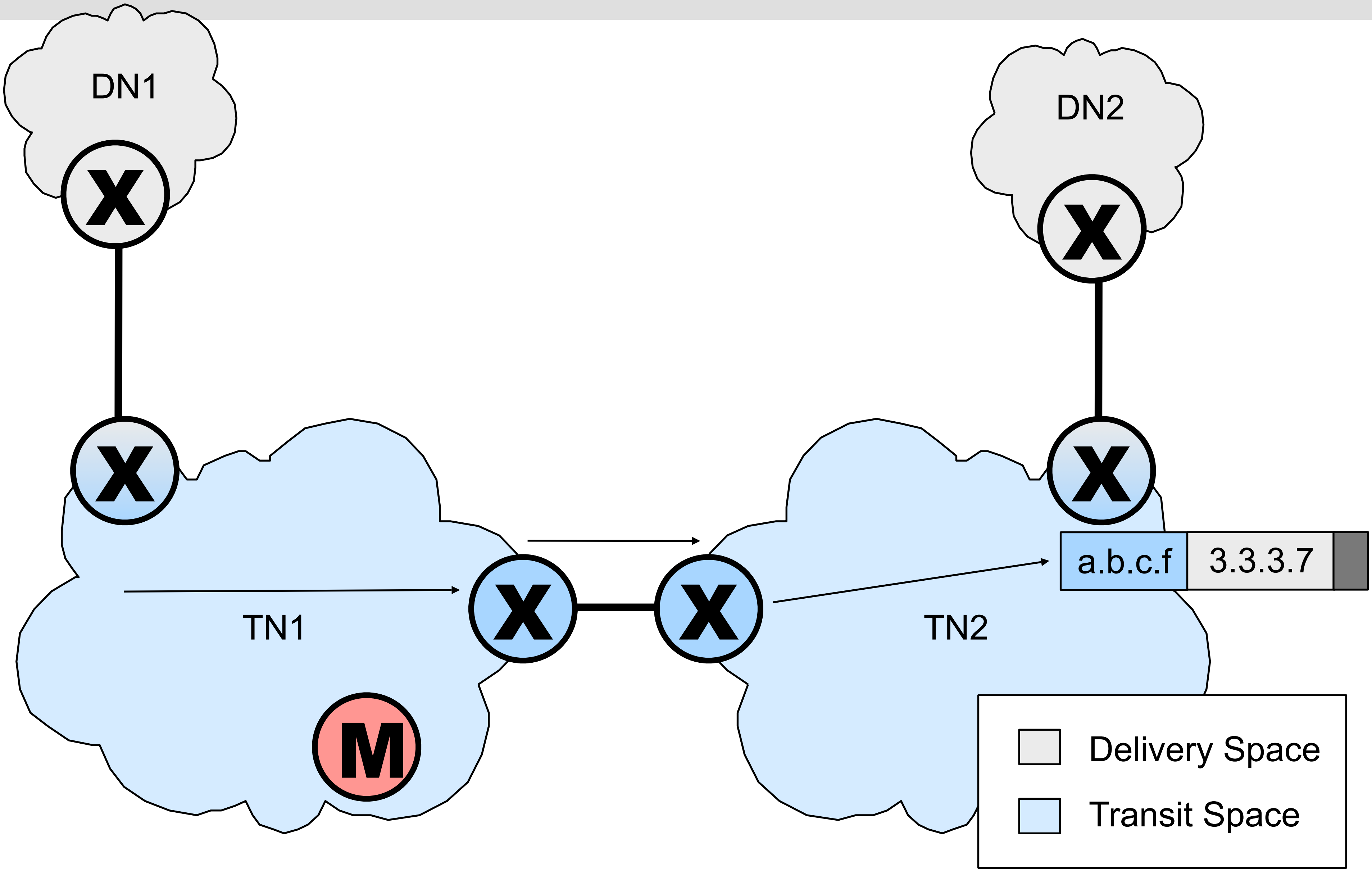
Daddr prefix	Taddr	TTL
1.1.1.0/24	a.b.c.d	300
2.2.0.0/24	a.b.c.e	600
3.3.3.0/24	a.b.d.f	600

Packet Encapsulated

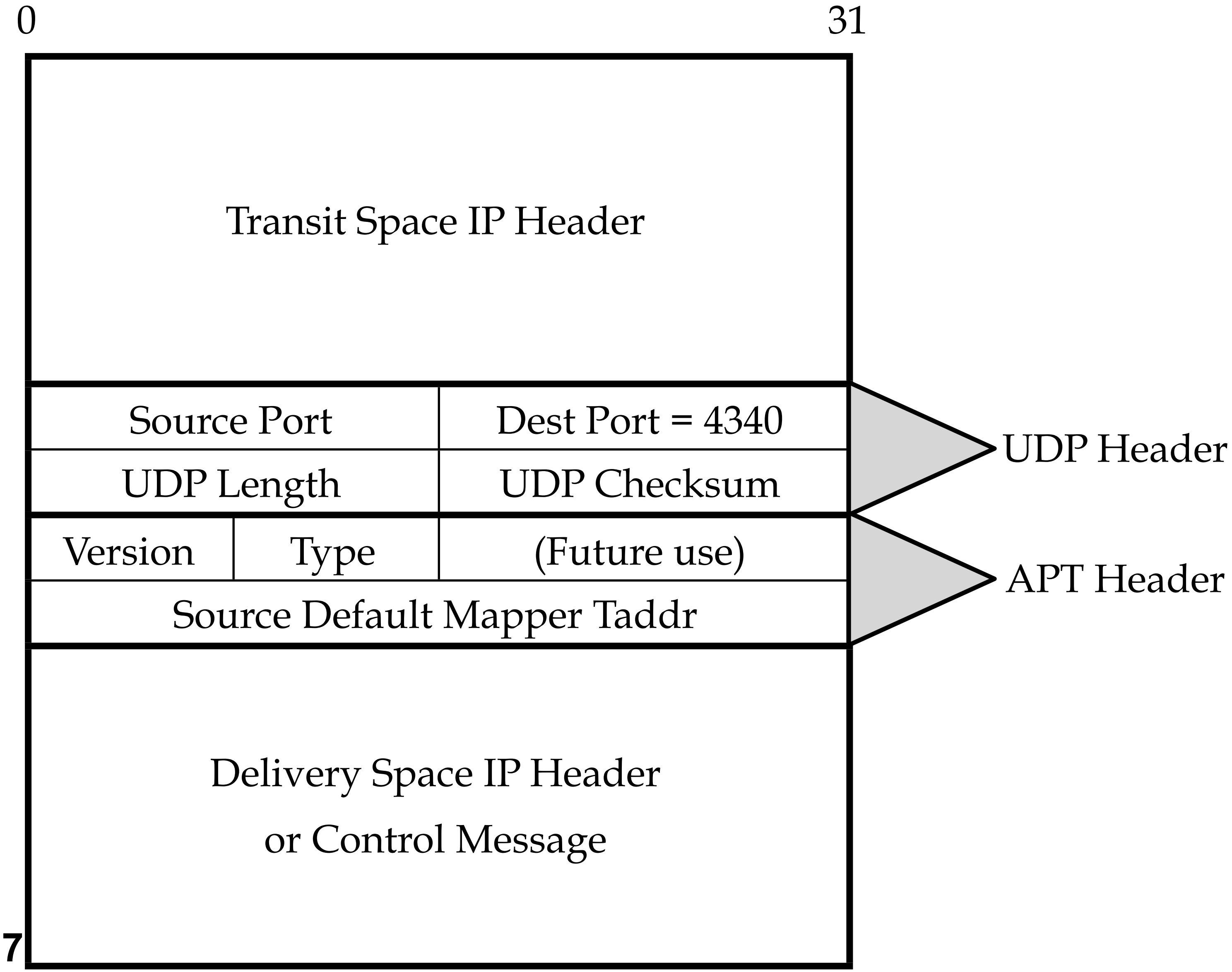


Daddr prefix	Taddr	TTL
1.1.1.0/24	a.b.c.d	300
2.2.0.0/24	a.b.c.e	600
3.3.3.0/24	a.b.d.f	600

Packet Delivered



Header Layout



0

31

Transit Space IP Header

Source Port | Dest Port = 4340

UDP Length | UDP Checksum

Version | Type | (Future use)

Source Default Mapper Taddr

Delivery Space IP Header
or Control Message

UDP Header

APT Header

Major Issues for Any Mapping Service

- Disseminating mapping information
- Securing mapping dissemination
- Handling ETR failures
- Incremental deployment
  - We aren’t going to talk about this today
  - Ask us if you want to hear our ideas

Disseminating MapSets Between TNs

- Default mappers need to learn other TNs' mapping information
- Mapping information is exchanged via DM-BGP
  - A separate instance of BGP running on a different TCP port
  - Only default mappers peer
  - Mapping information is carried in a new attribute
  - DM-BGP is only used to disseminate mapping information, not to store it
  - DM-BGP is not used for routing

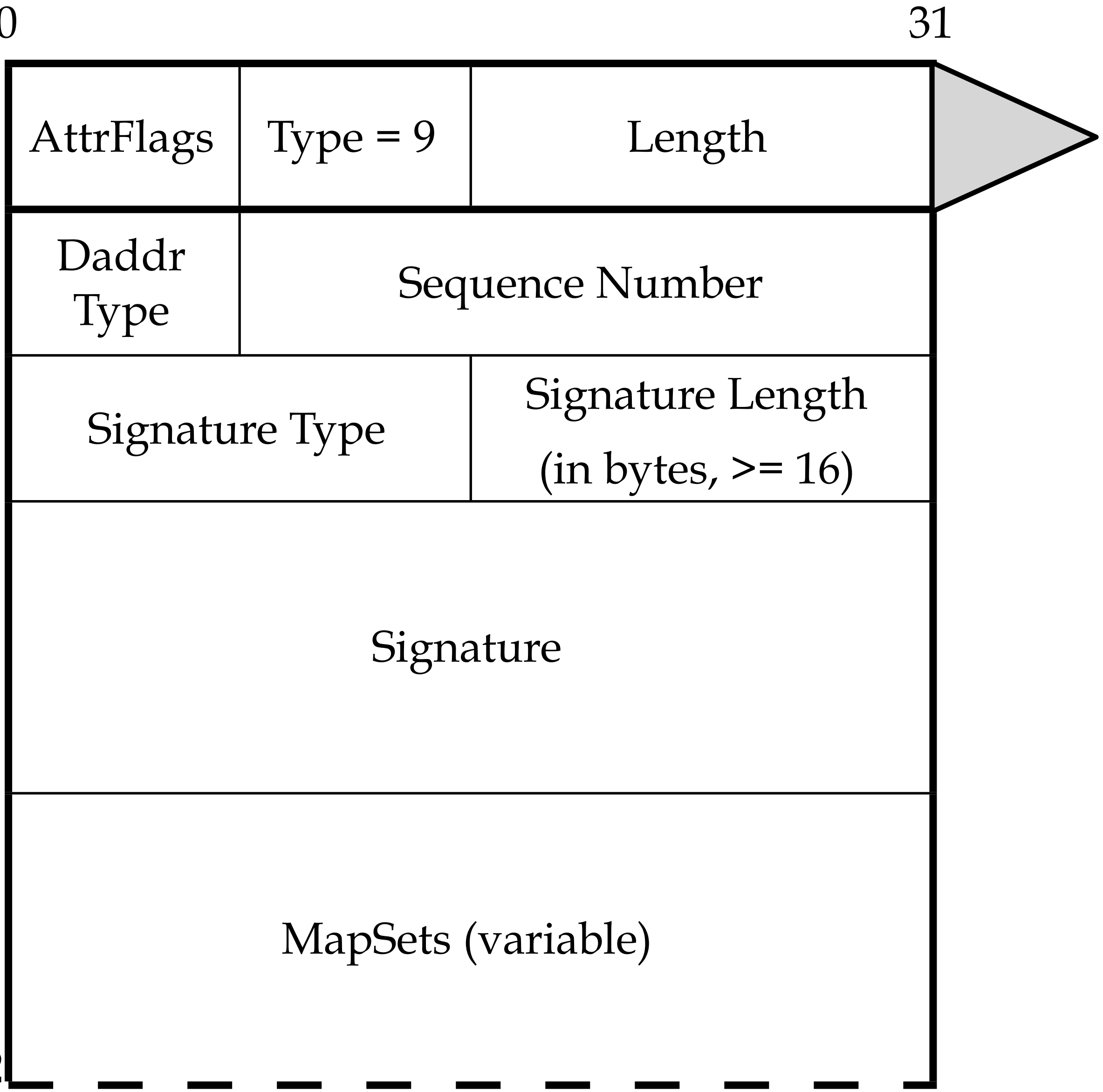
Security for Mapping Announcements

- Authentication of mapping information is critical
  - False MapSets could cause major problems
    - Network-wide traffic hijacking
    - DDoS attacks
- Default mappers have public/private key pairs
  - Default mappers in the same TN use the same key pair
- Mapping announcements must be cryptographically signed by the originator
  - The signature must be verified at each DM-BGP hop, but not changed
  - Prevents spoofing, corruption, and modification of mapping information

Default Mapper Requirements for Mapping Announcement Security

- Store a public key table
  - One entry per transit network (TN)
  - We didn't mention our public-key distribution method
    - We are working on a separate paper describing this method
    - Ask us if you want to know the details
- Lookup the key and verify all incoming announcements
- Sign all originated announcements

Mapping Announcement Attribute



0 31

AttrFlags | Type = 9 | Length

Daddr Type | Sequence Number

Signature Type | Signature Length (in bytes, >= 16)

Signature

MapSets (variable)

DM-BGP Path Attribute Header

Handling ETR Failures

- Failures break down into three situations
  1. The Taddr prefix containing the ETR address is unreachable
  2. The ETR itself is unreachable
  3. The link between the ETR and its DN is down
- In all three situations, APT can avoid dropping any packets
- Situations 2 and 3 require control messages, which can be secured
- Ask if you want to know the details

Feedback?

- Q: Would it be practical to build default mappers on a router platform?
- Q: Can currently deployed PE routers become TRs with only a software update?
- To review...

Default Mapper Review

- Encapsulate and decapsulate IP-in-UDP packets
- Store and retrieve *all* MapSets in a table
  - Lookup Daddr prefixes in the table and pick an ETR
- Send Cache Add and Cache Drop Messages to TRs
- Run DM-BGP
- Store a public key table, one entry per TN
- Create/verify mapping announcement signatures
- **Q: Would it be practical to build default mappers on a router platform?**

TR Review

- Encapsulate outgoing packets (ITR mode)
- Decapsulate incoming packets (ETR mode)
- Cache only Daddr-to-single-Taddr mappings (MapRecs)
- Cache only MapRecs that are currently in use
  - Delete after the MapRec's time to live (TTL) expires
  - No MapRec? Tunnel the packet to your default mapper.
  - Default mapper re-tunnels the packet to an ETR for you and responds with a Cache Add Message containing a MapRec
- **Q: Can currently deployed PE routers become TRs with only a software update?**

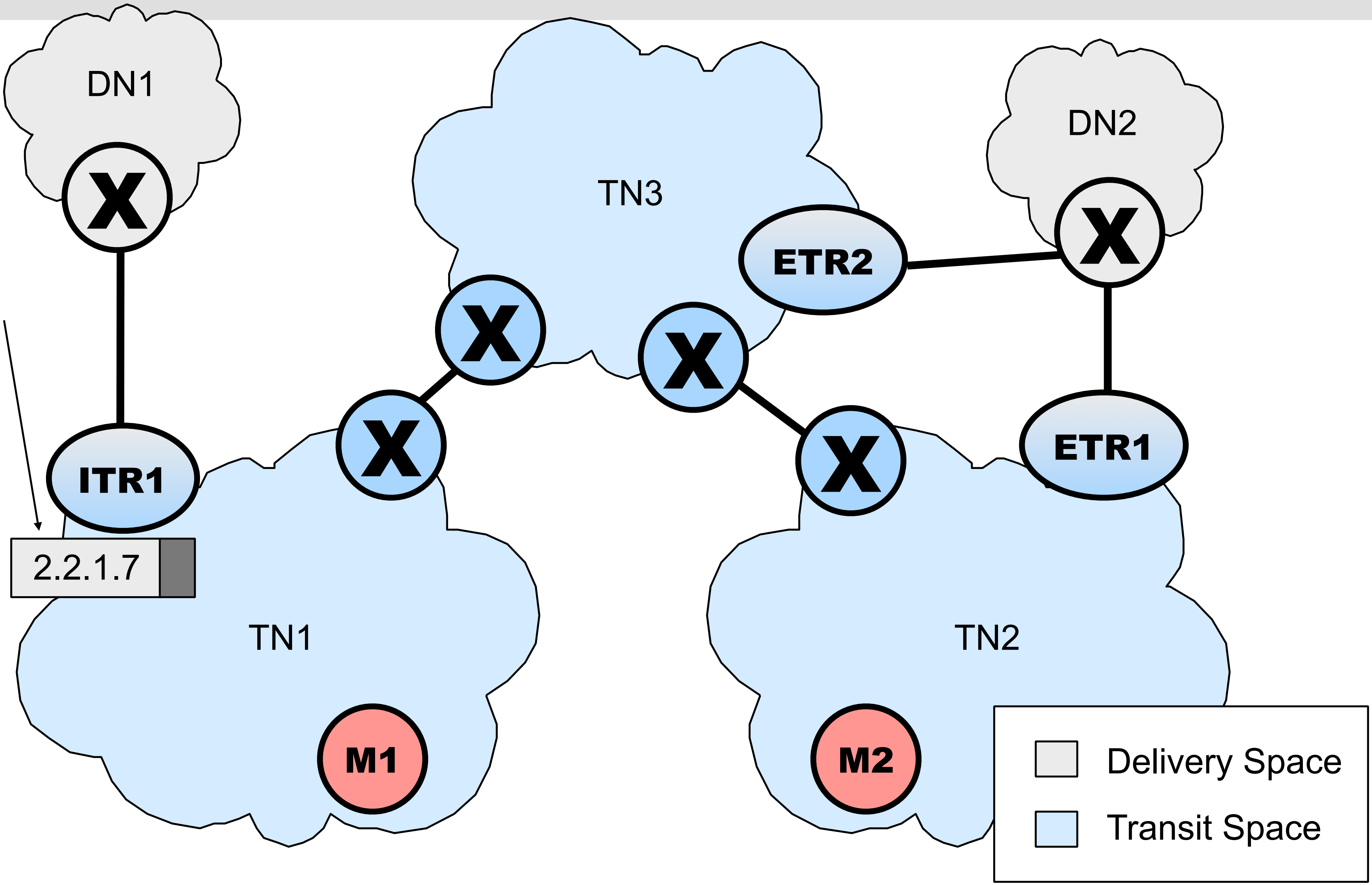
Thank You!

- Questions?
- Comments?

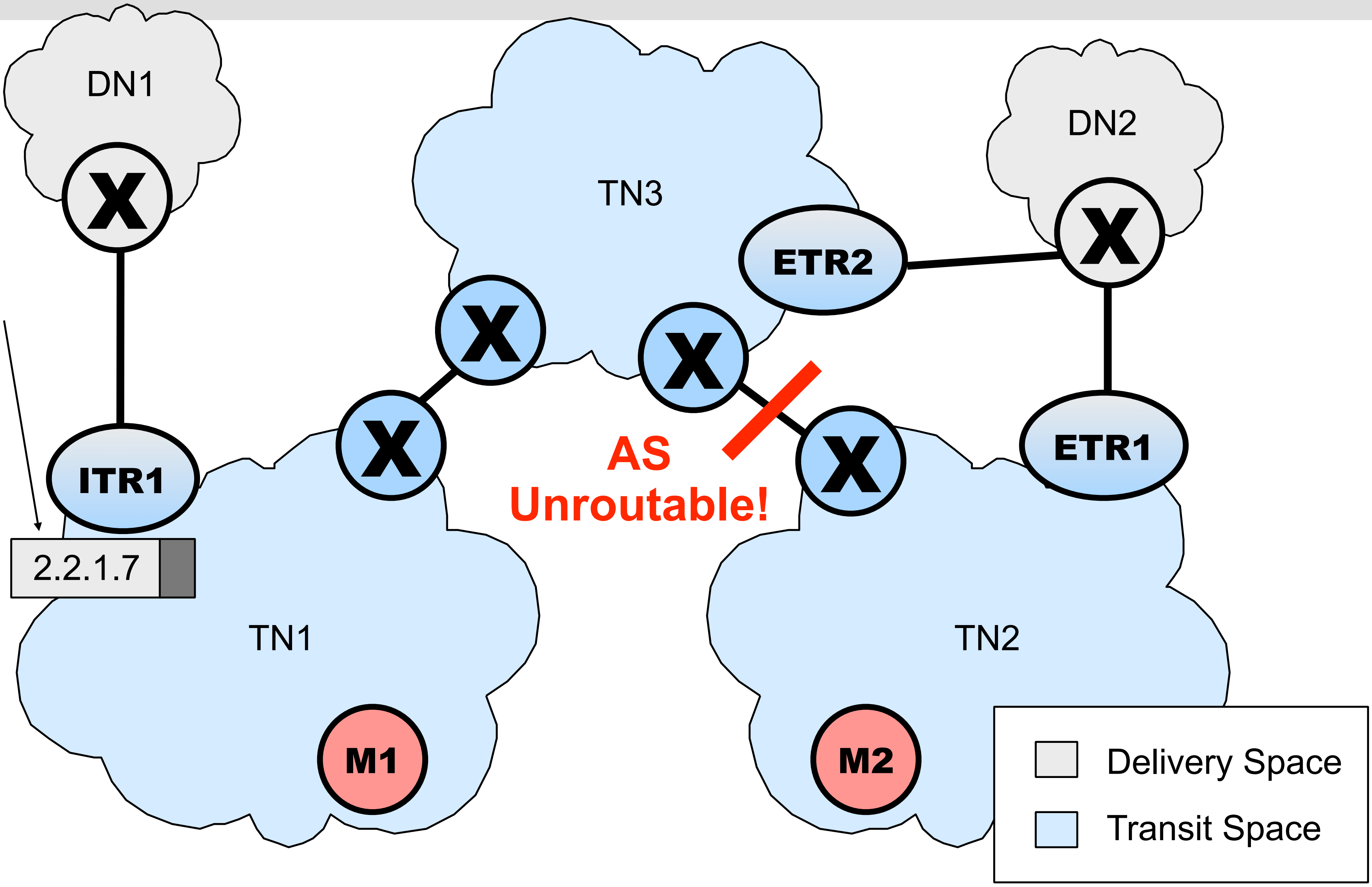
Handling Failures

- Three situations require failover to alternate ETRs
  1. A Taddr prefix is unroutable via BGP
  2. The ETR itself becomes unreachable
  3. The link between an ETR and delivery space fails
- Additions to default mappers
  - List of TRs using the default mapper
  - Time Before Retry (TBR) for each MapRec

Example


DN1
TN3
DN2
ETR2
ITR1
ETR1
2.2.1.7
TN1
TN2
M1
M2
Delivery Space
Transit Space

Situation 1 Example

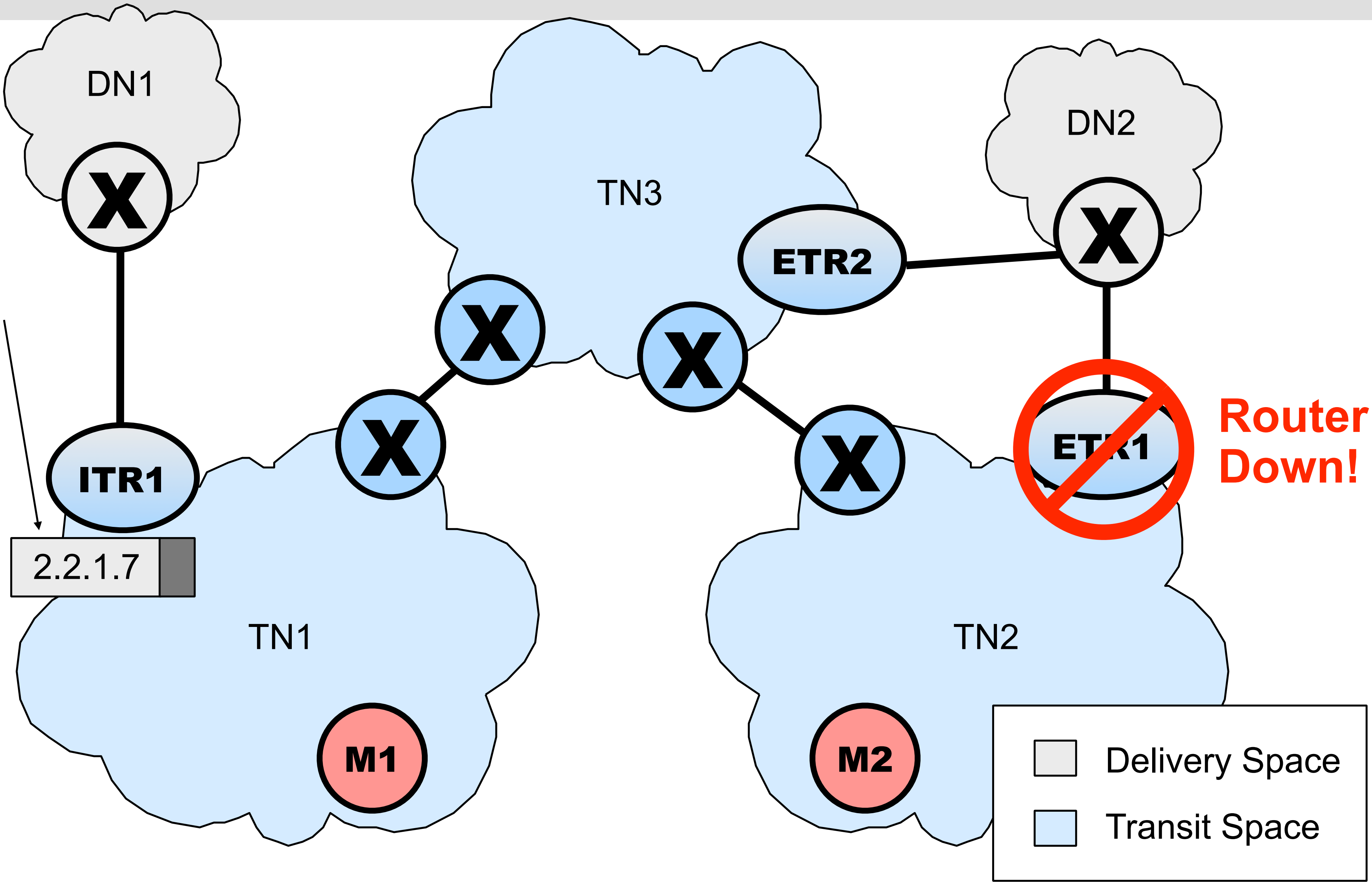


⬜	Delivery Space
🟦	Transit Space

Situation 1: Taddr Prefix Unroutable

- ITRs use their default mapper as their default route
- Regardless of whether ETR1′s Taddr is in ITR1′s cache
  - ITR1 forwards these packets to its default mapper (M1)
  - M1 uses DN2′s MapSet to find ETR2′s Taddr
  - M1 sends the packet to ETR2
  - M1 sends a MapRec containing ETR2′s Taddr and a short TTL to ITR1

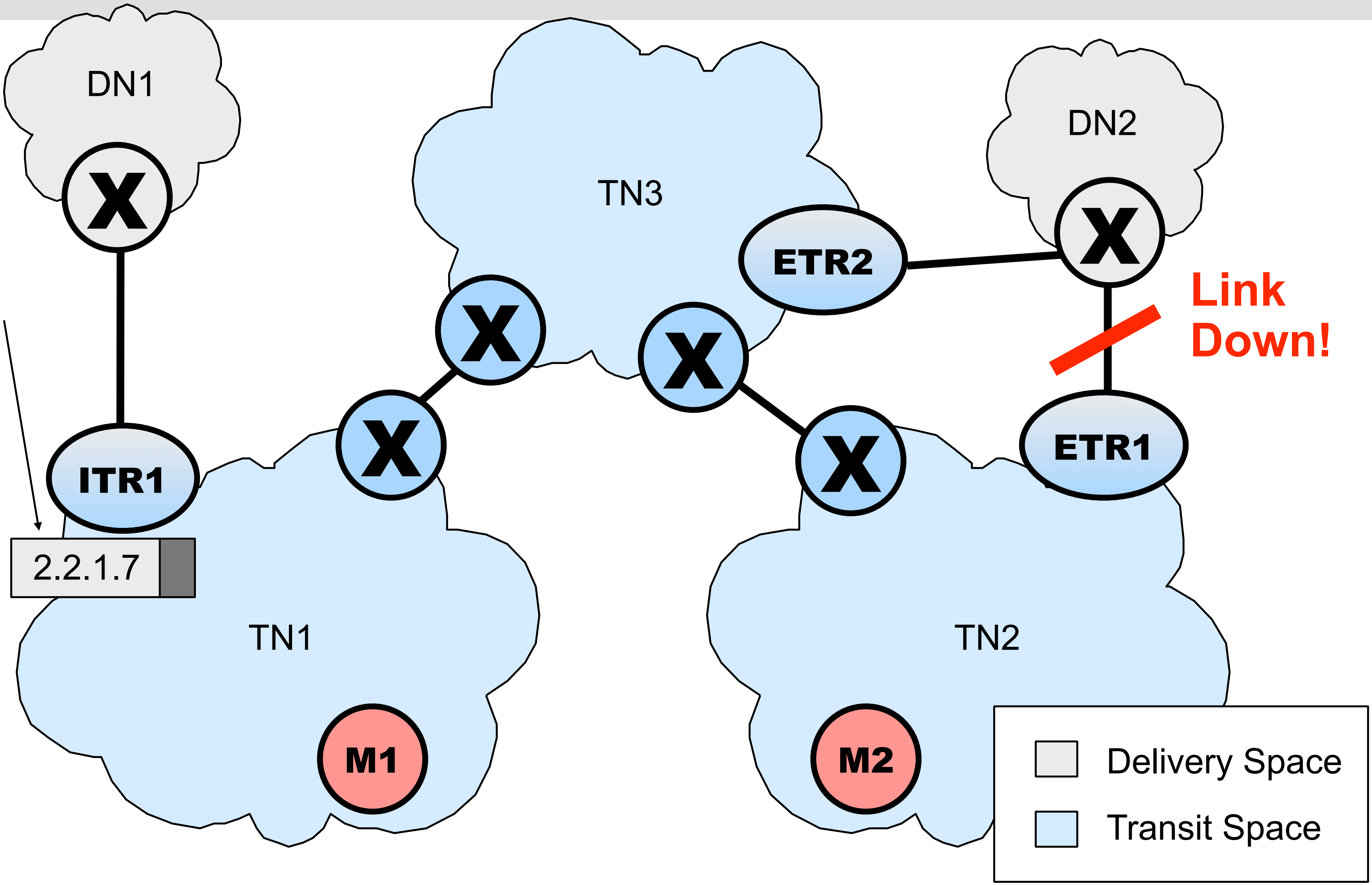
Situation 2 Example



Situation 2: Single ETR Failure

- ETR1′s default mapper (M2)
  - Uses TN2′s IGP to intercept packets destined for ETR1
  - Finds ETR1′s Taddr in its TR list
  - Sets the time before retry (TBR) for ETR1 in DN1′s MapSet
  - Sends the packet to an alternate ETR (ETR2)
  - Sends an ETR Unreachable Message to ITR1′s default mapper (M1)
- ITR1′s default mapper (M1)
  - Sets the same TBR
  - Sends a Cache Drop Message to its TRs

Situation 3 Example



Situation 3: TR-to-DN Link Failure

- ETR1
  - Detects that the link has failed
  - Forwards the packet to M2 with type set to TR-to-DN Link Failure
- ETR1’s default mapper (M2)
  - Almost the same procedure as in Situation 2
    - Sets the TBR for ETR1 in DN1’s MapSet
    - Sets the time before retry (TBR) for ETR1 in DN1’s MapSet
    - Sends the packet to an alternate ETR (ETR2)
    - Sends a **DN Unreachable Message** to ITR1’s default mapper (M1)
- ITR1’s default mapper (M1)
  - Exactly the same procedure as in Situation 2