

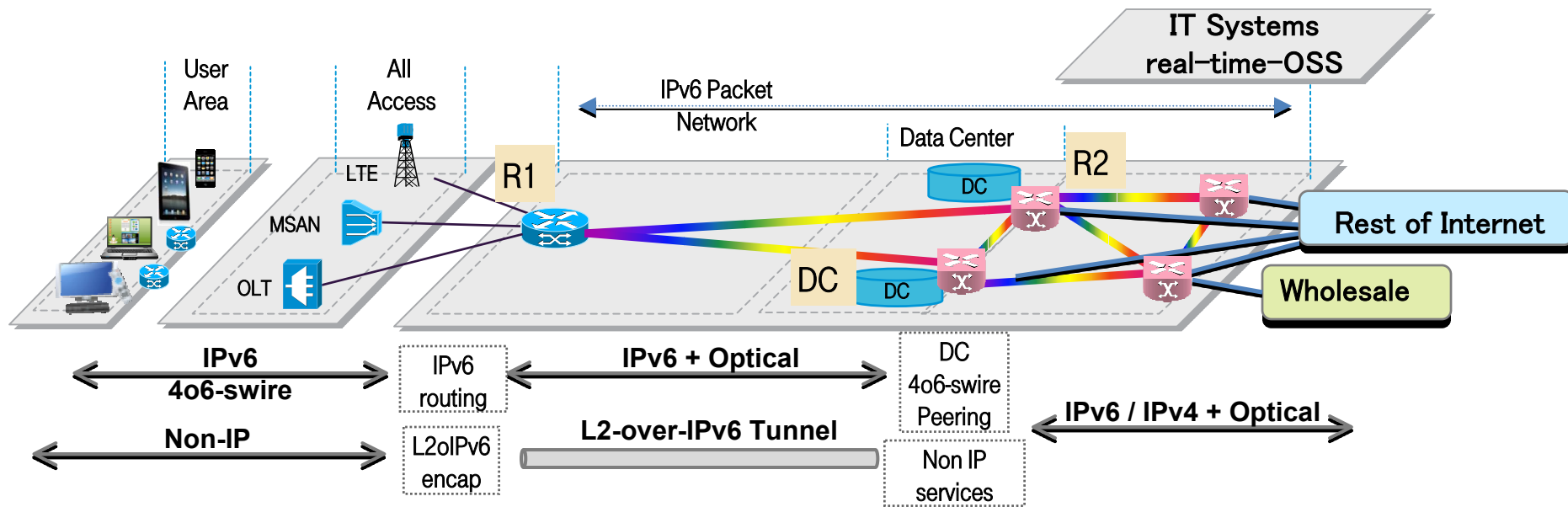
# TeraStream – IPv6

Peter Lothberg, Mikael Abrahamsson

Life is for sharing.



# TERASTREAM – DESIGN IN A NUTSHELL



## TeraStream key functional elements

### R1

- Terminate access interfaces
- Runs IPv6 routing only, integrates optical
- Access services
  - IPv6 - dealt with natively
  - IPv4 – IPv4 over IPv6 software between HGW / CPE and DC, R1 not involved
  - non-IP - L2-over-IPv6 encapsulation
- User configuration
  - using Netconf / Yang
  - Driven by real-time OSS i.e. self-service portal

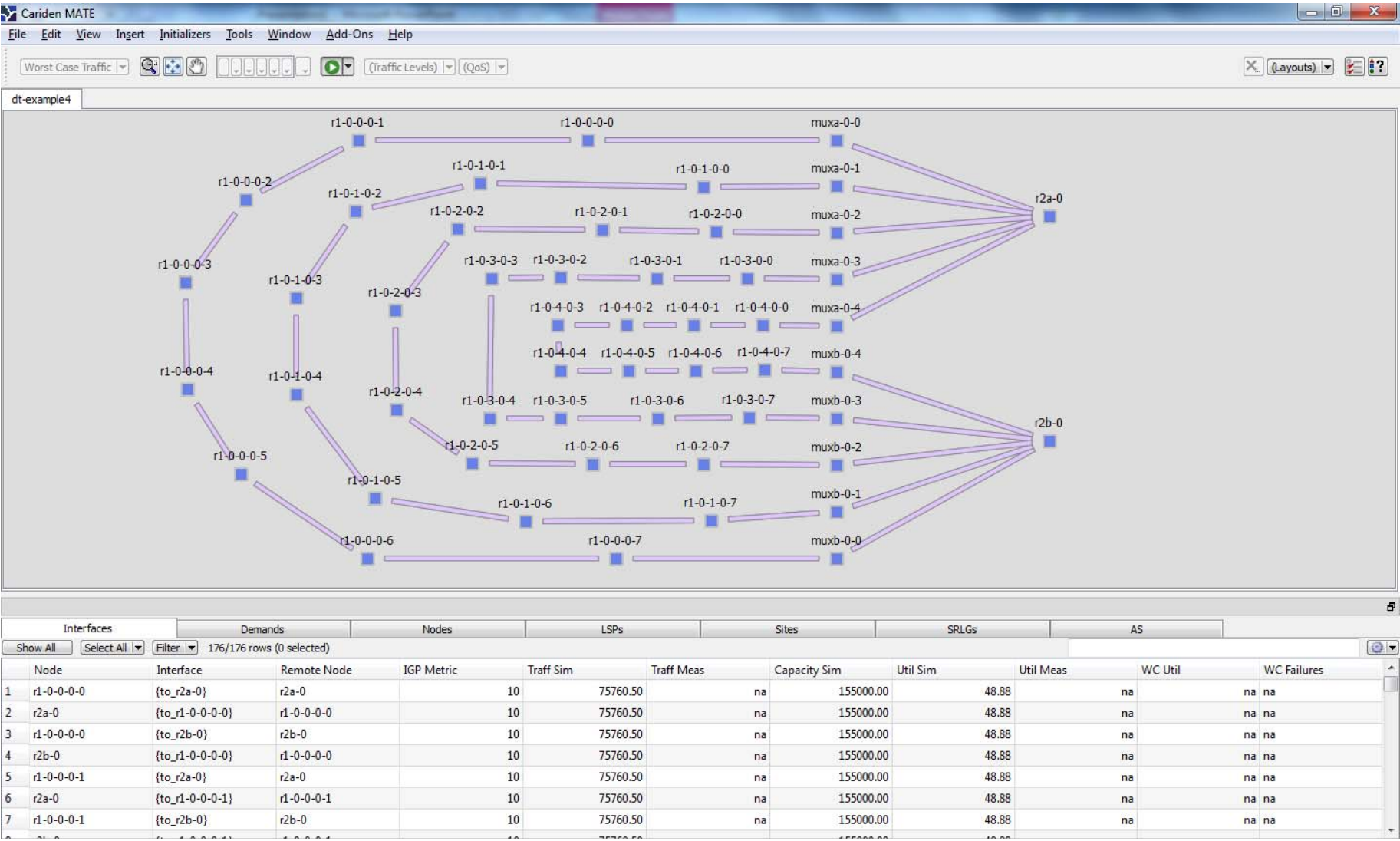
### R2

- Connects R1s, Data Centers and Internet peerings
- Runs IPv6 and IPv4 routing, integrates optical
- Closely integrated with Data Centers
  - Optimized handling of locally sourced services
- High scale IP bandwidth

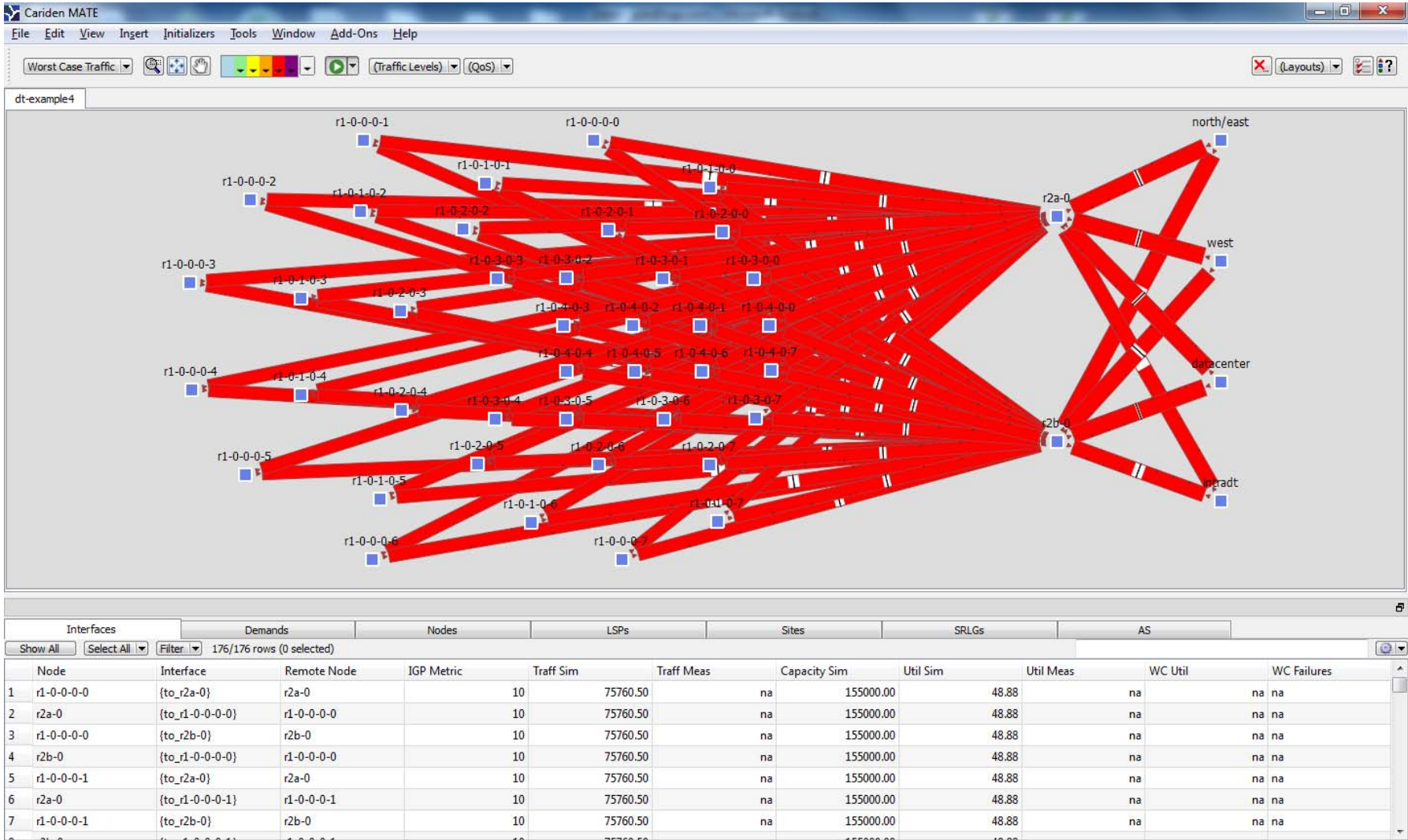
### Data Center / Services

- Distributed design
  - fully virtualized x86 compute and storage environment
- Network support functions - DNS, DHCP, NMS
- Real-time OSS incl. user self-service portal
- Cloud DC applications, XaaS services
- Complex network services e.g. high-touch subscriber handling

# LAYER 1 TOPOLOGY



# LAYER 3 TOPOLOGY



# IPV6 ADDRESSING FORMAT, USERS

2.4  
20111006

```

      0               1               2               3               4               5               6
0 1 2 3 4 5 6 7!8 9 0 1 2 3 4 5!6 7 8 9 0 1 2 3!4 5 6 7 8 9 0 1!2 3 4 5 6 7 8 9!0 1 2 3 4 5 6 7!8 9 0 1 2 3 4 5!6 7 8 9 0 1 2 3
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      REGISTRY/IANA assigned      |P|I|E|S S S|R|a a a a a a a a a a a|p p p p p p p p p p p p|u u u u u u u u u|
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

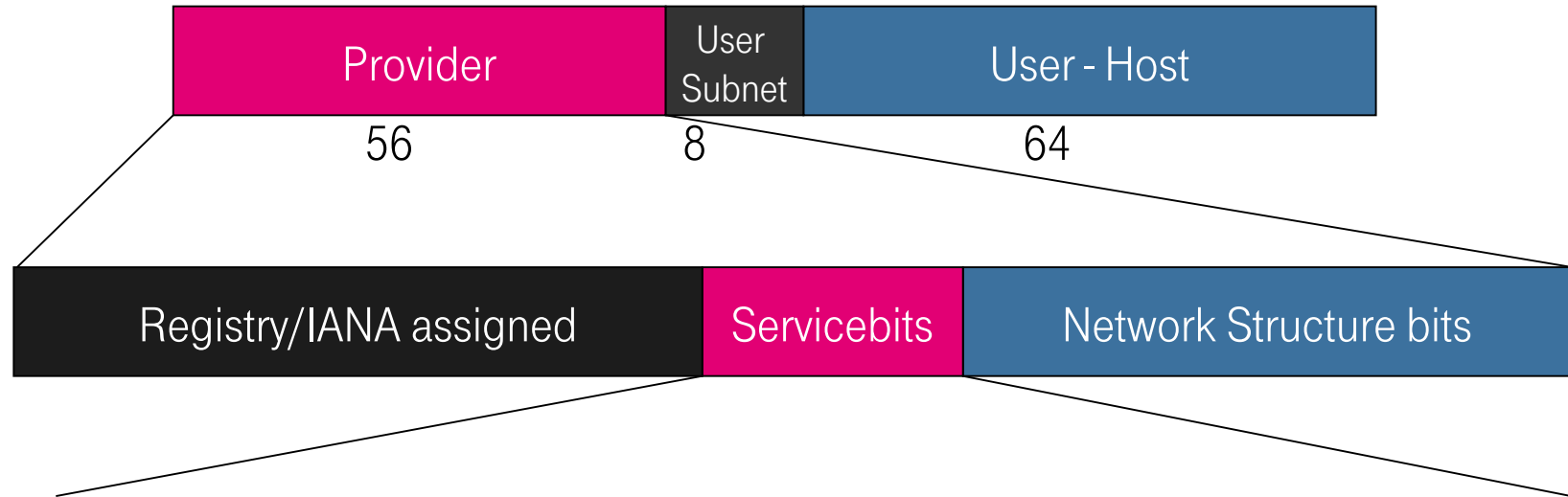
P Public	0=traffic internal to local SP
I Infrastructure	0=user traffic
E Endpoint/Service	0=network endpoint, 1=service
S Logical Network (Internal ISP#)	0=res, 1=res, 2=internet, 3=res, 4=video, 5=L2 service, 6=voice, 7=management
R Reserved	
a R1 Area 14 bit	Indicates what R1 that the address is delegated from, max 16,384 R1
p R1 User 13 bit	User identifier, max 8192 users
u User subnet	Delegated to user

Examples:	Source PIESSS	Destination PIESSS
-----		
User -> Voice	000110	011110
Voice -> User	011110	000110
User -> User (best effort)	X00001	X00001
User -> Internet (best effort)	100001	XXXXXX
Internet -> User (best effort)	XXXXXX	100001
Lan-Lan service	010101	010101



# SERVICE DIFFERENTIATION BASED ON ADDRESSES

## USING IPv6 ADDRESS SPACE AS LABELS



```
P Public          0=SP-intern, 1=extern
I Infrastructure  0=end user, 1=infrastructure packet
E Endpoint/Service 0=endpoint, 1=service
SSS Logical Nw (int. ISP)# 0=res, 1=internet, 4=video, 5=L2, 6=voice, 7=mgmt
```

Examples:	Source PIESSS	Destination PIESSS
-----	-----	-----
User -> Voice	000110	011110
Voice -> User	011110	000110
User -> User (best effort)	X00001	X00001
User -> Internet (best effort)	100001	XXXXXX
Internet -> User (best effort)	XXXXXX	100001
Lan-Lan service	010101	010101



# ROUTING

## Static:

- 8000 R1's loopbacks (/128 in ISIS)
- 32,000 Internal links (R1-R2 /127 in ISIS)
- 64,000 IBGP routes from R1's (/43 )
- 3,800,000 IBGP routes that simulates the rest of the Internet

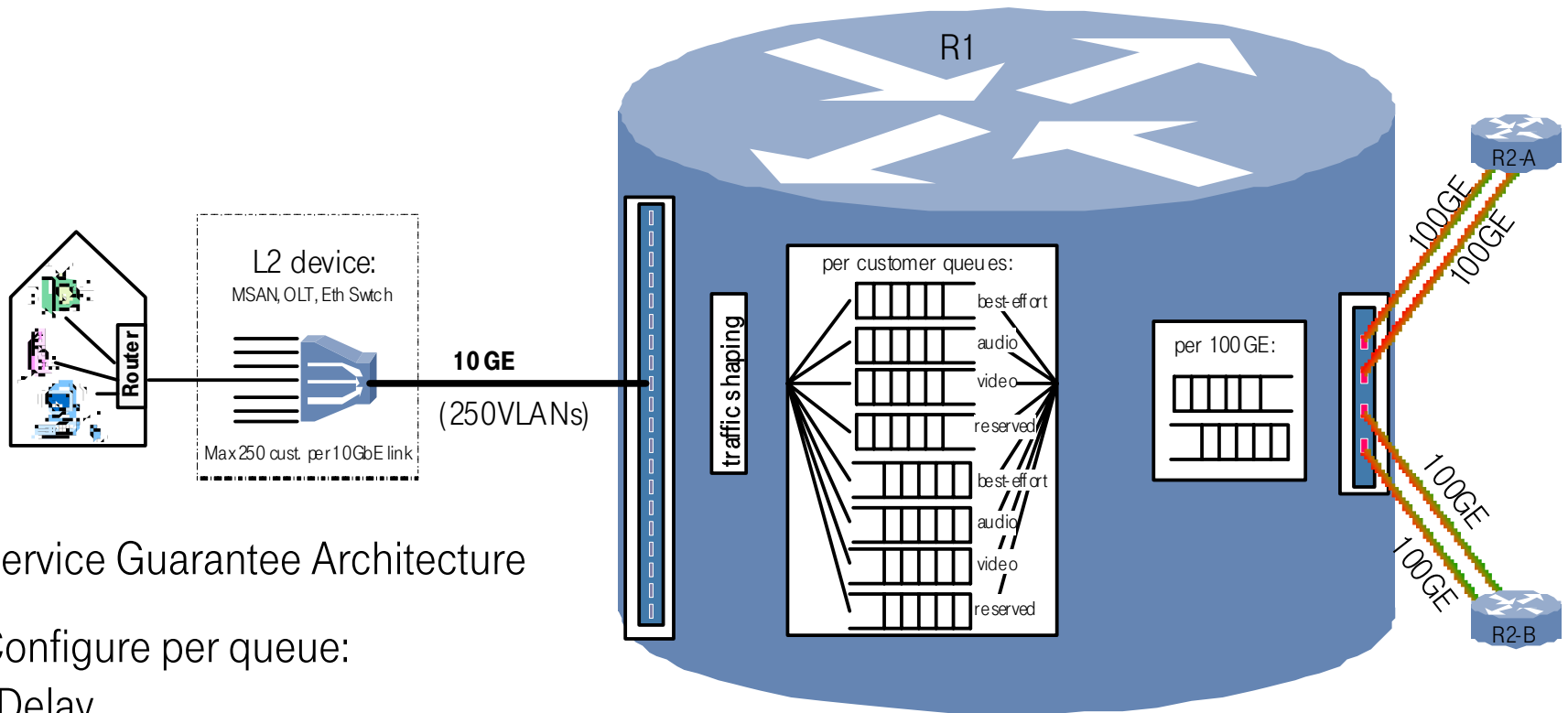
## Dynamic:

- 300 IBGP updates from the "outside" per second (/48 prefixes)  
emulating flapping (duty cycle 60 on 60 off)
- 64 ISIS updates/second representing random Terastream  
internal link failures

ISIS routes for R1's	8,000
Internal links	64,000
IPv6 IBGP routes from R1's	64,000
IPv4 routes from external AS to R2	800,000 (not propagated to R1)
IPv6 routes received from external AS	3,064,000 (propagated to R1)



# TERASTREAM USER FACING ROUTER R1



## Service Guarantee Architecture

Configure per queue:

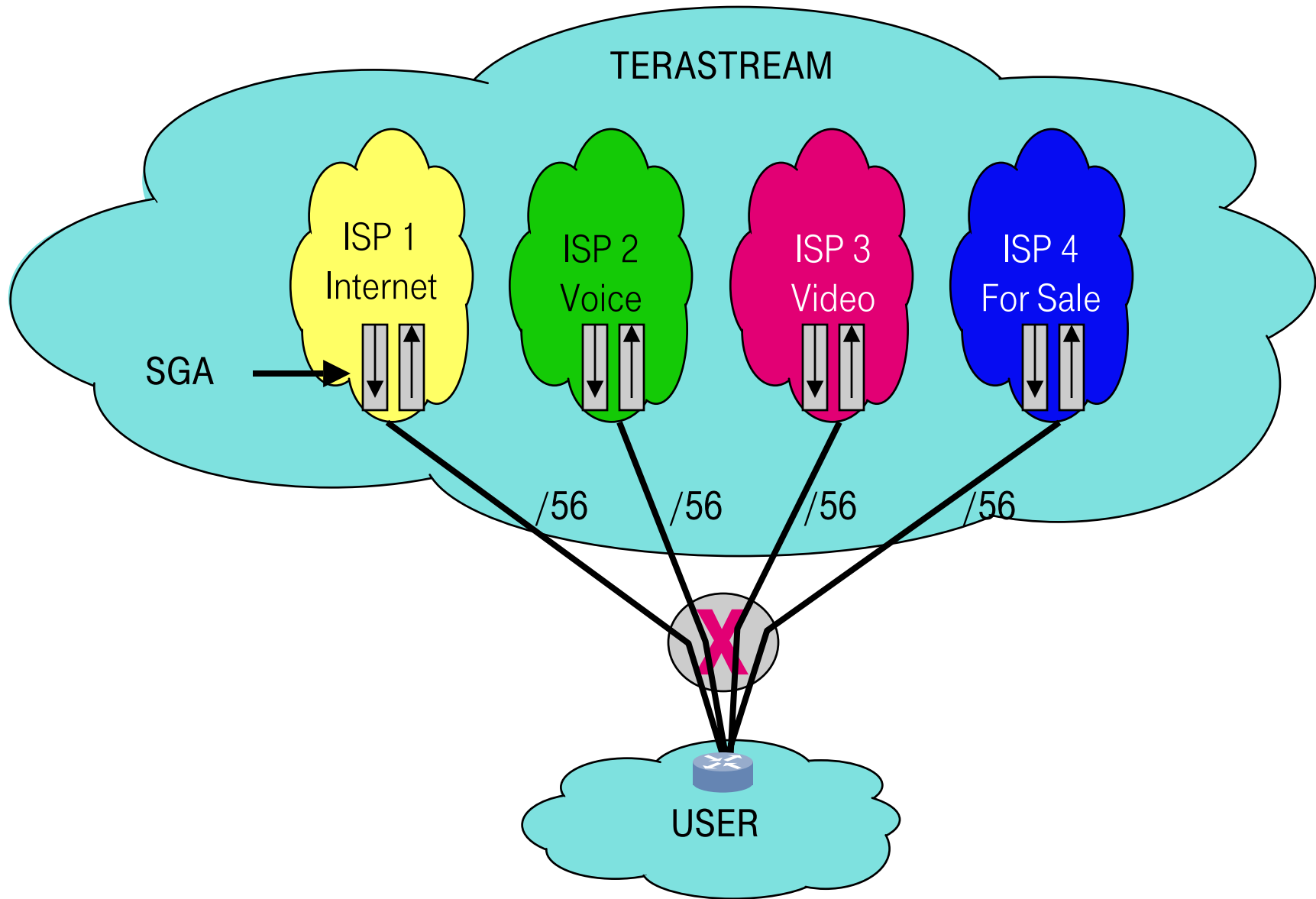
- Delay
- Drop
- Bandwidth
- Reorder
- Etc...

- IP traffic shaped to capabilities of L2 device
- 5000 customers connections per R1
- 20 \* 10GE port for L2 device
- 4 \* 100GE for R2 link

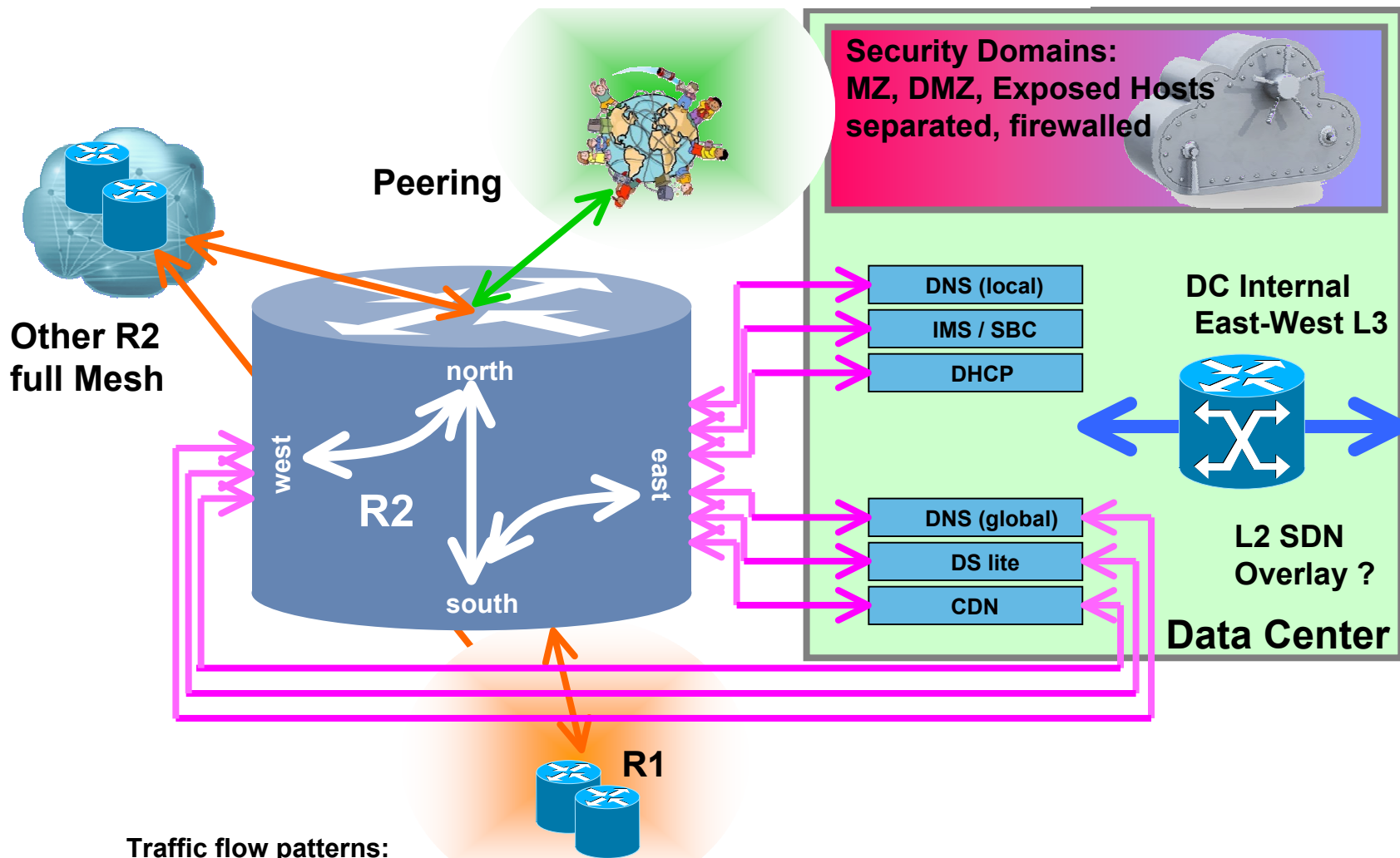




# CARRY POLICY AND BETWEEN ISP AND USER



# R2 ROUTER AND TRAFFIC PATTERNS



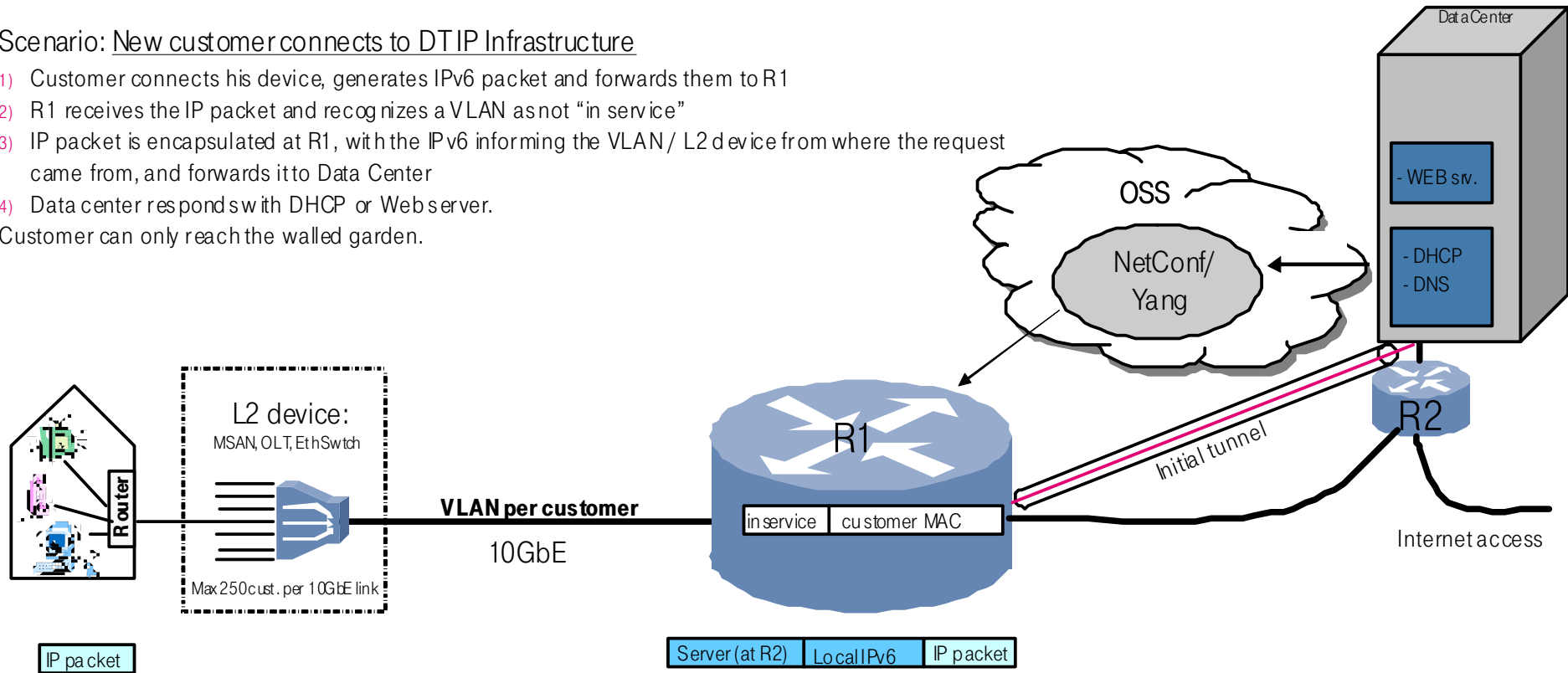
Traffic flow patterns:

- R1 ⇌ Peers and Other R2 going north ⇌ south (example: IPv6 Internet traffic)
- R1 ⇌ Data Center services going south ⇌ east (example: DHCP)
- R1 ⇌ Data Center ⇌ Peers going south ⇌ east ⇌ west ⇌ north (example: IPv4 Internet traffic)

# IF NOT IPV6, USE THE NETWORK AS A PTP ETHERNET

Scenario: New customer connects to DTIP Infrastructure

- 1) Customer connects his device, generates IPv6 packet and forwards them to R1
- 2) R1 receives the IP packet and recognizes a VLAN as not "in service"
- 3) IP packet is encapsulated at R1, with the IPv6 informing the VLAN / L2 device from where the request came from, and forwards it to Data Center
- 4) Data center responds with DHCP or Web server.  
Customer can only reach the walled garden.



Scenario: customer registers

- 1) Web server at Data Center generates a request to OSS to configure a new customer via NetConf / Yang at router R1, Line ID.
- 2) The OSS via NetConf configures the R1 as "in service" for a customer located at a specific interface (IPv6 address).
- 3) From now on, the customer is outside the walled garden and can reach other Internet addresses.



# IPV4 DECOMMISSIONING STRATEGY

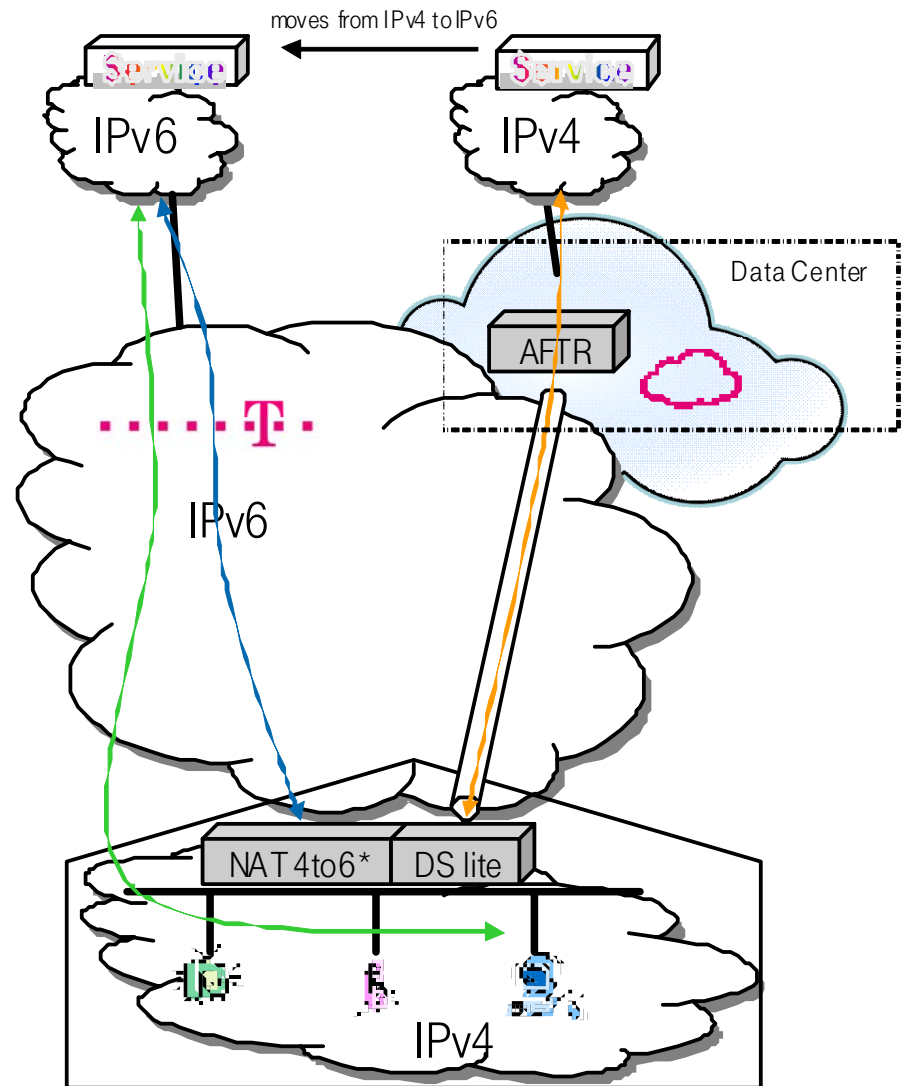
The Internal IP network of DT is IPv6. All IPv4 traffic to and from the customer will be translated to IPv6 at the borders of the network. 2 alternatives are seen as viable:

- 1) Customer IPv4 traffic is encapsulated on IPv6 via DS-lite to a AFTR element located at the Data Center. RFC 6333.
- 2) Customer IPv4 traffic is translated to IPv6 at the customer's device (NAT 4to6). (Standard not defined)

In the long term, the expectation is that most customers will be IPv6 capable and that the services will move to IPv6.

In the transition time DS lite should provide the mechanism to connect IPv4 devices to other networks.

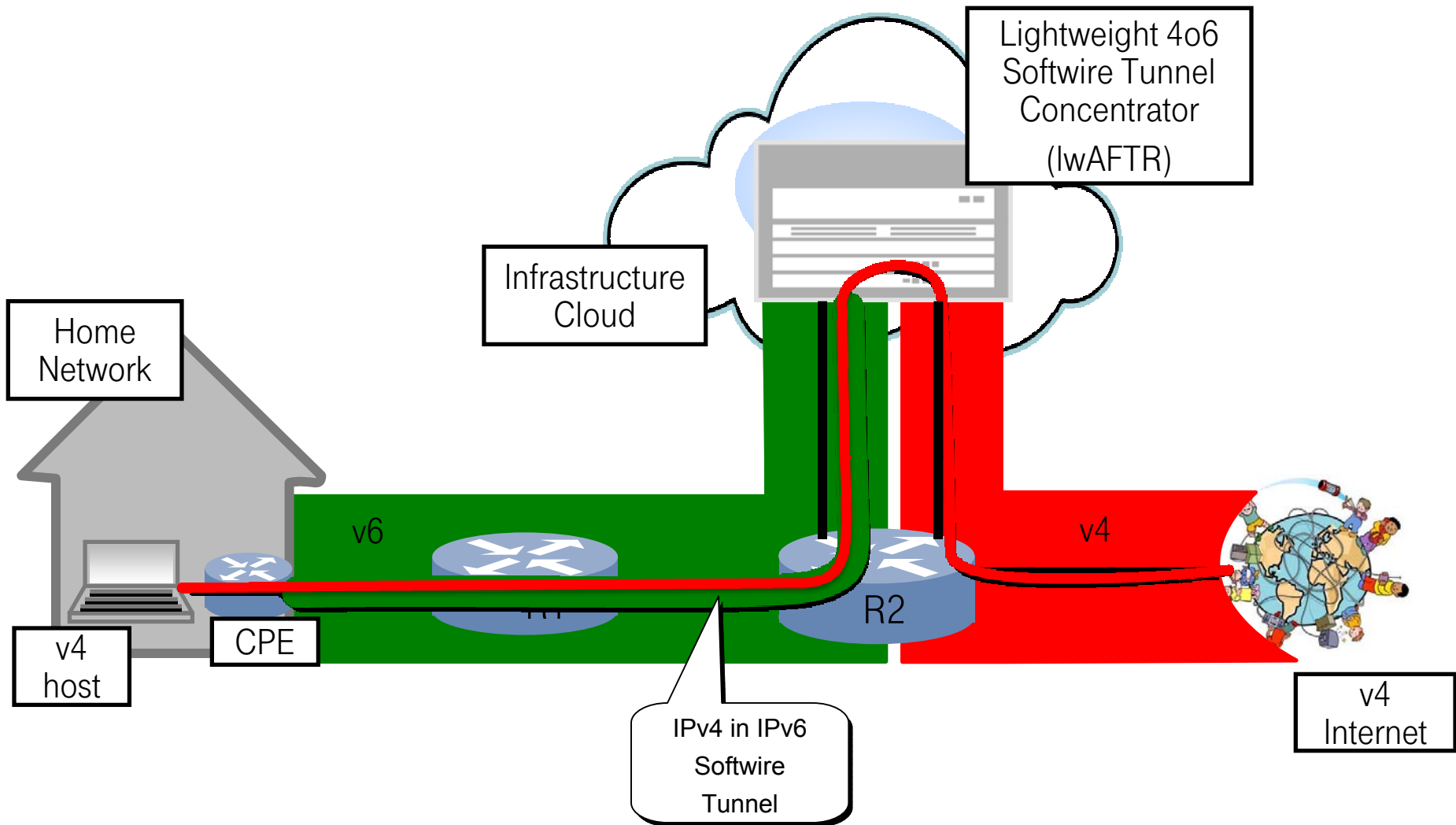
There is no standard describing NAT 4to6, i.e. translating IPv4 packets to IPv6. This standard remains for further work.



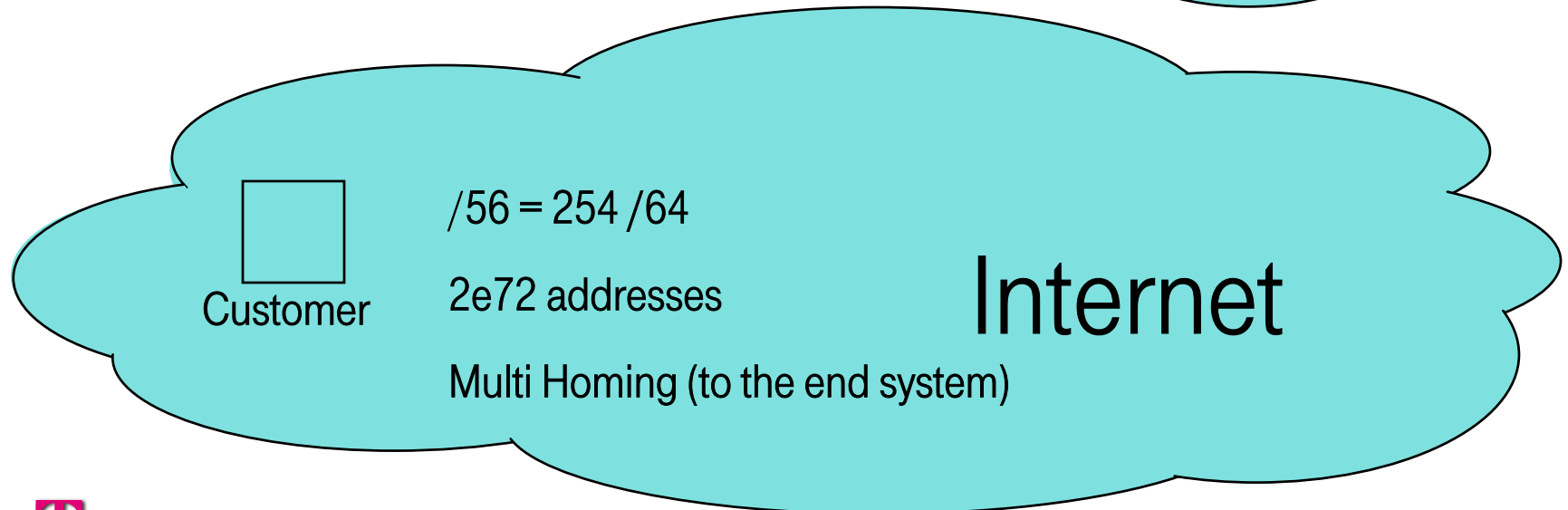
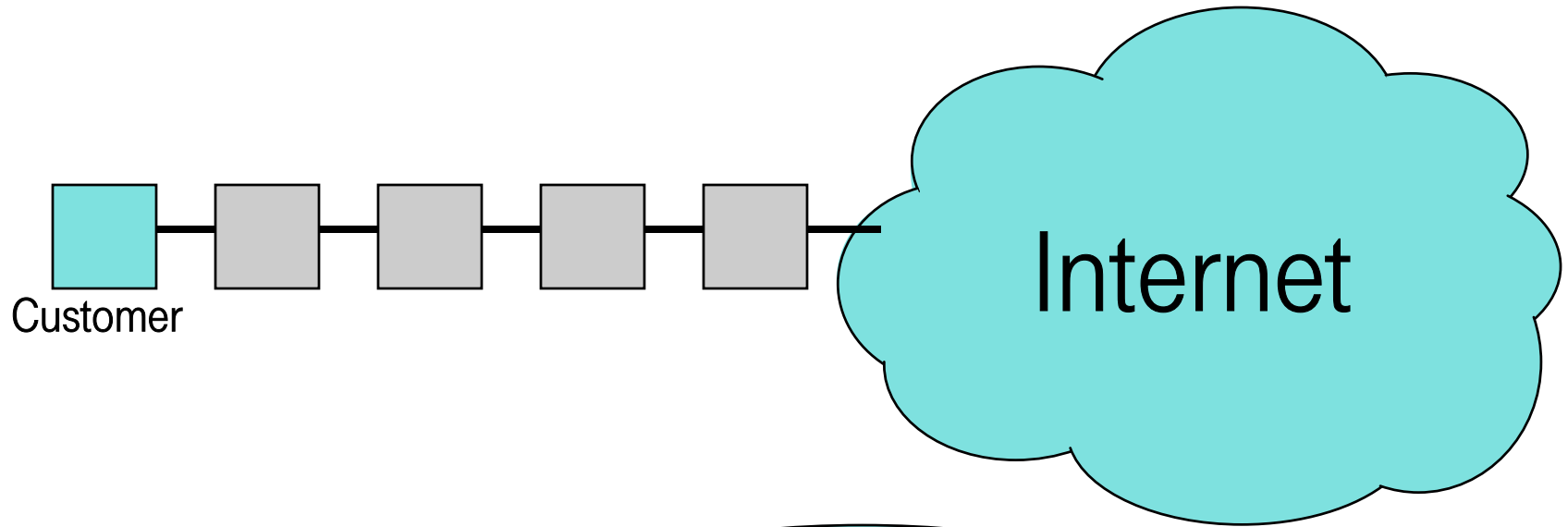
\* NAT 4to6 – Standard not defined



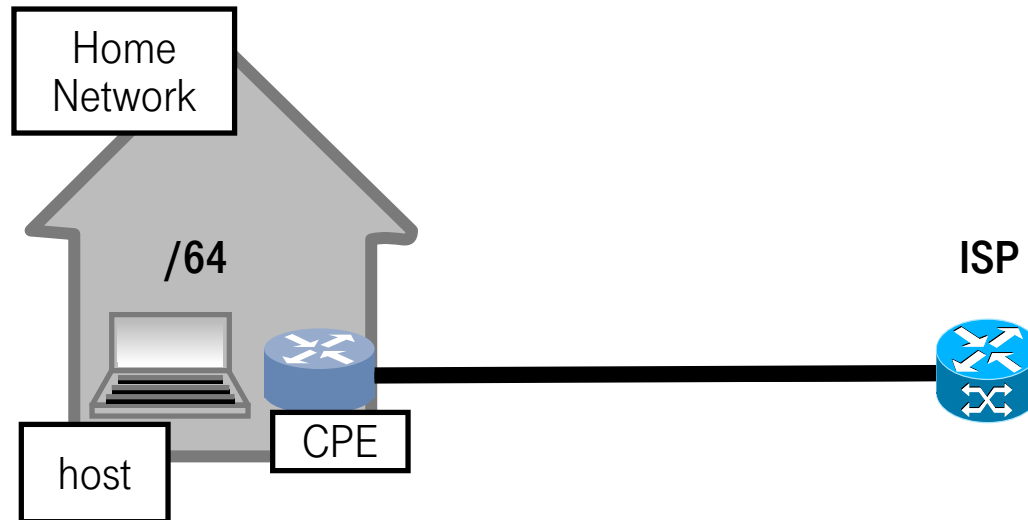
# IPv4 AS A SERVICE - LIGHTWEIGHT 4o6 SOFTWARES



# CHANGING THE BROADBAND PARADIGM

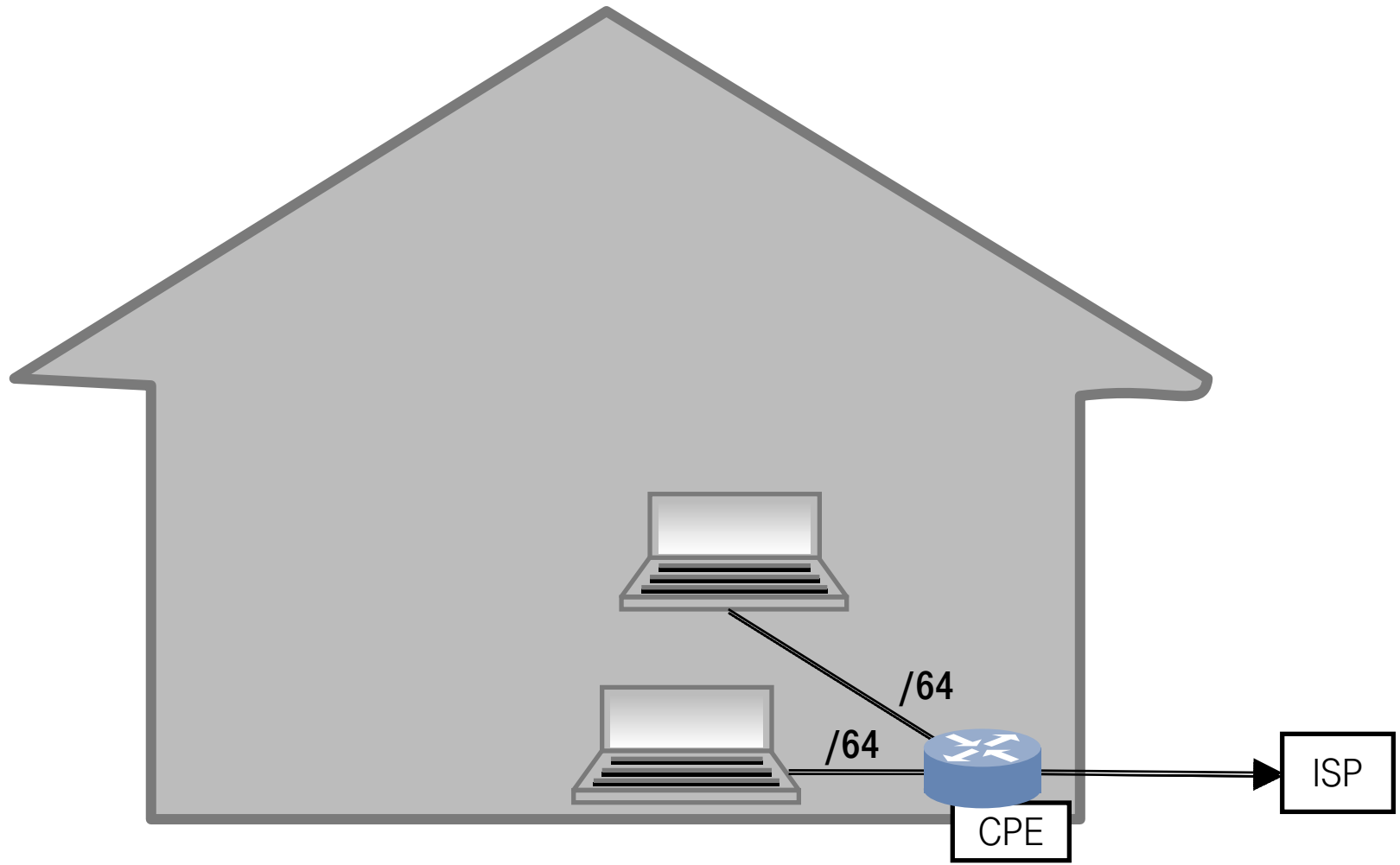


# USUAL HOME NETWORK DRAWINGS

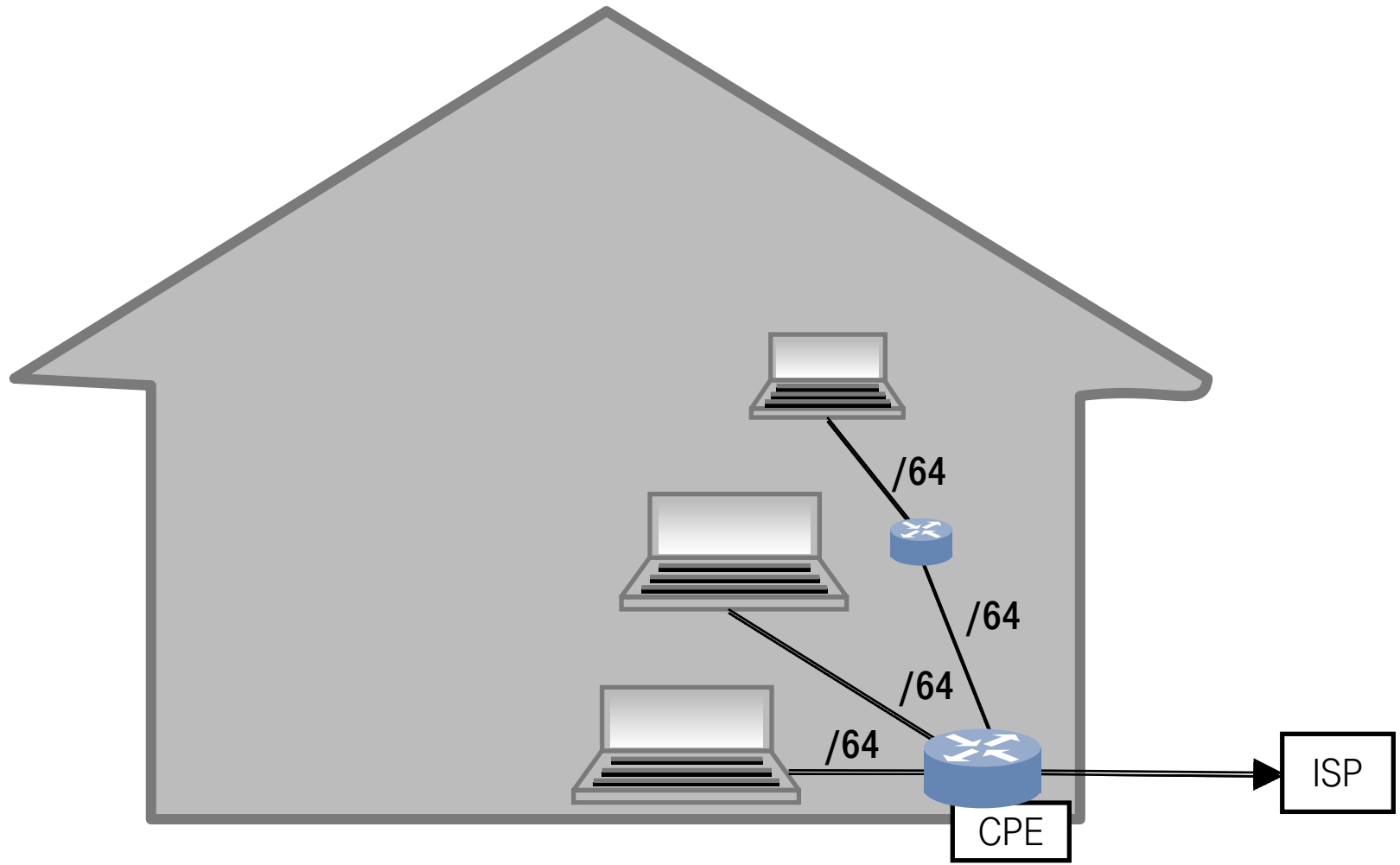




# SOMETIMES LIKE THIS



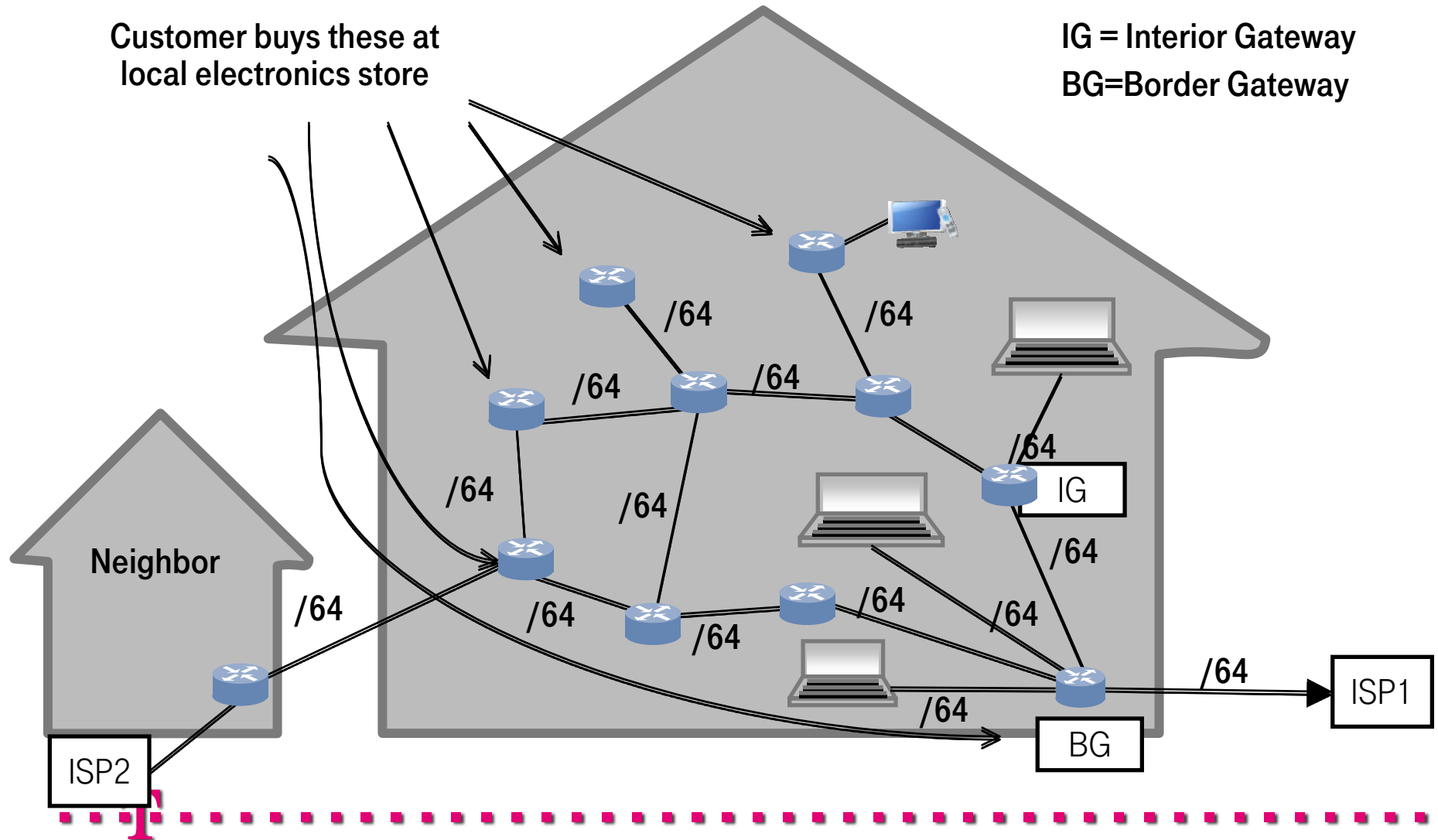
# PERHAPS EVEN PREFIX DELEGATION!



# WE'RE AIMING TO HANDLE THIS

Customer buys these at local electronics store

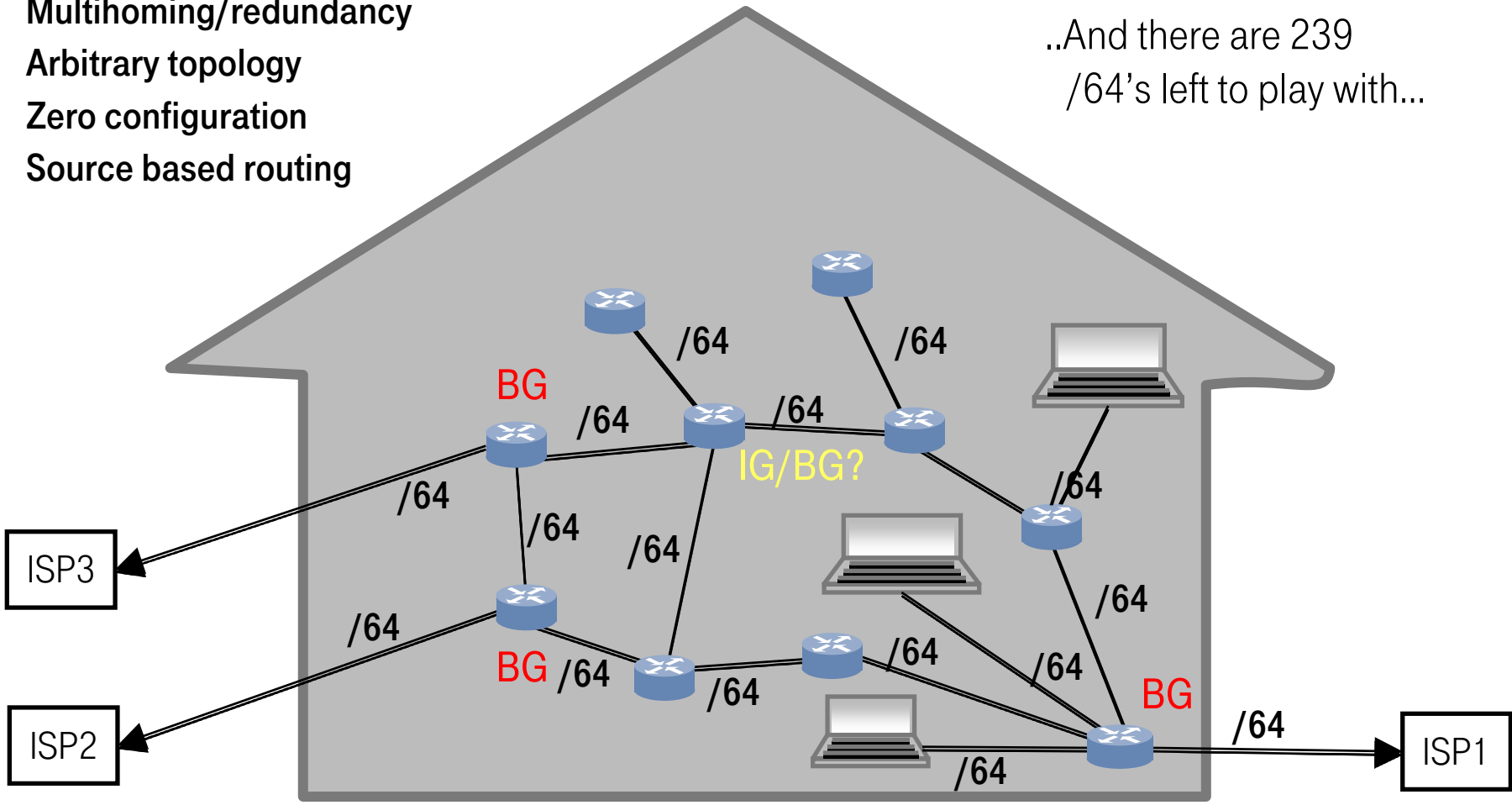
IG = Interior Gateway  
BG=Border Gateway



# ... AND THIS

- Multihoming/redundancy
- Arbitrary topology
- Zero configuration
- Source based routing

..And there are 239  
/64's left to play with...



# PREFIX COLOURING

Multiple prefixes on the same wire

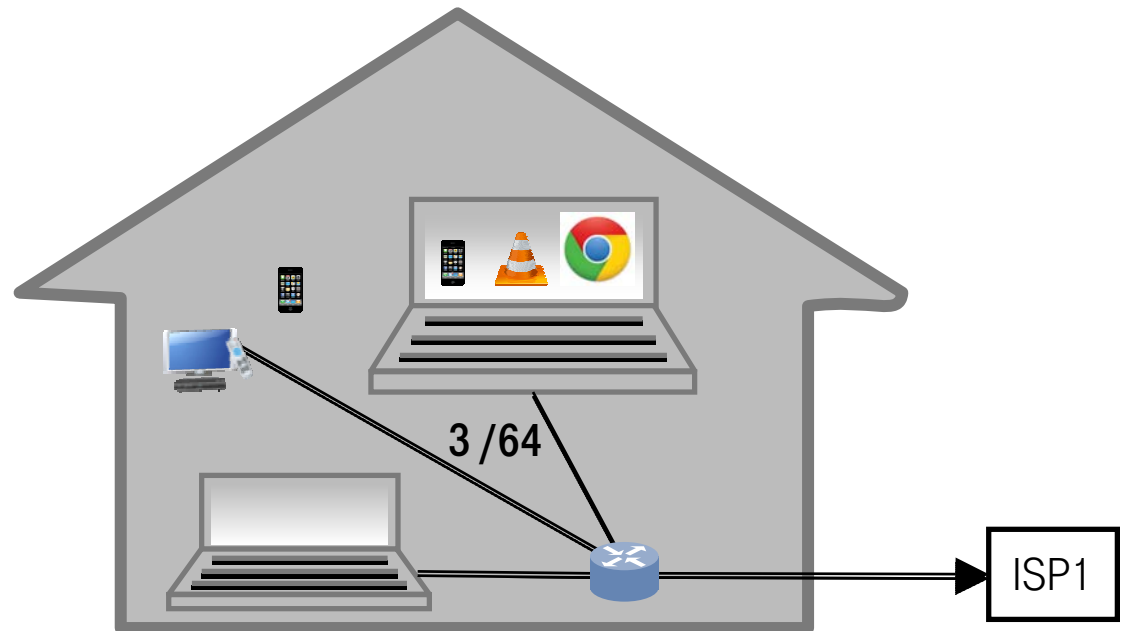
Each prefix has meaning = “colour”

TV application should use TV prefix

Phone application should use phone prefix

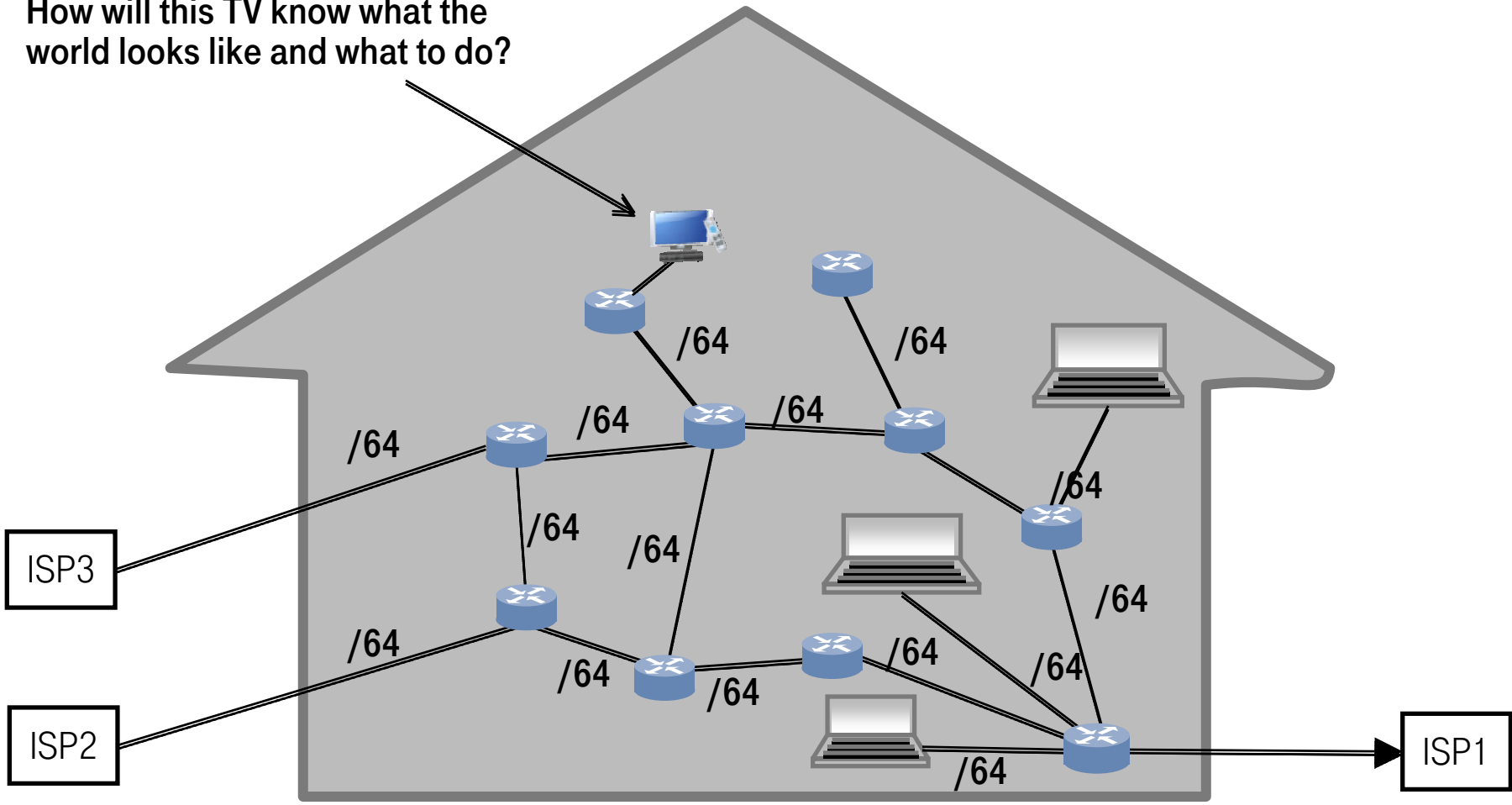
Need API for applications to understand this

Work being done in IETF MIF WG



# HOW TO SPREAD INFORMATION

How will this TV know what the world looks like and what to do?



# LET'S START FROM THE BEGINNING

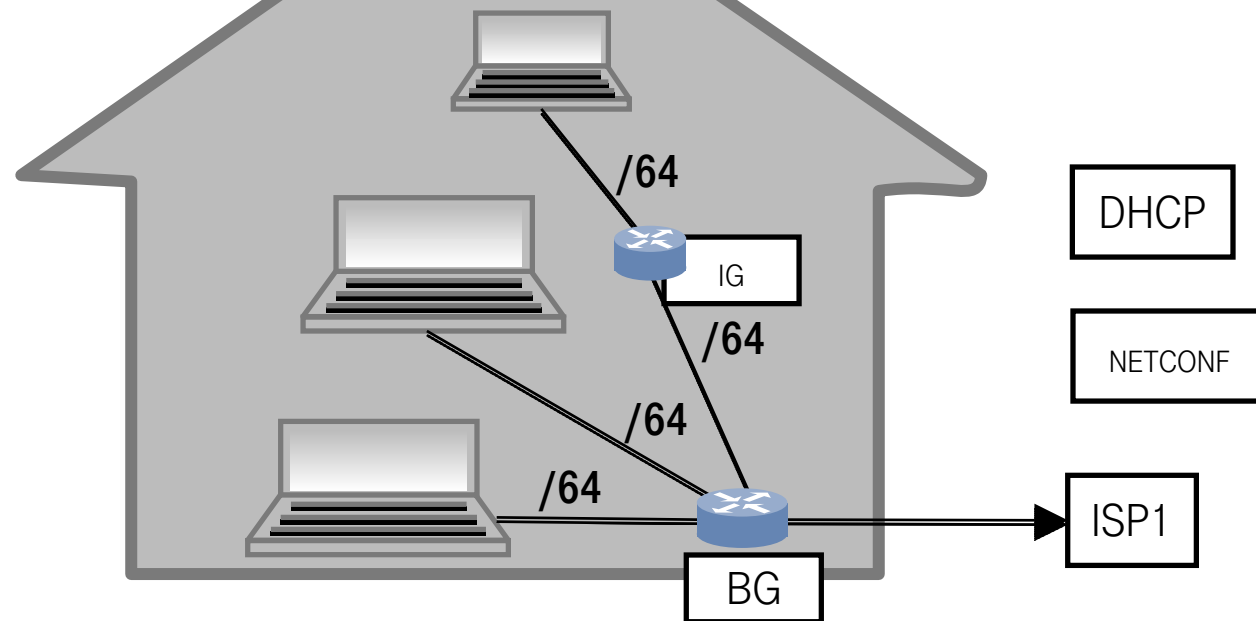
BG starts as host using RA+DHCP

From DHCP it gets NETCONF server address and cryptographic keys

Asks NETCONF server to connect to it (or starts Netconf session on initial TCP session), then tells it capabilities and desired services

NETCONF server sends service information plus other information to BG

Now BG has enough information to become router





# NEXT STEP, SAME THING AGAIN

IG starts as host using RA+DHCP

From DHCP it gets NETCONF server address and cryptographic keys

NETCONF server sends service information plus other information to BG

Now BG has enough information to become router

Things configured include:

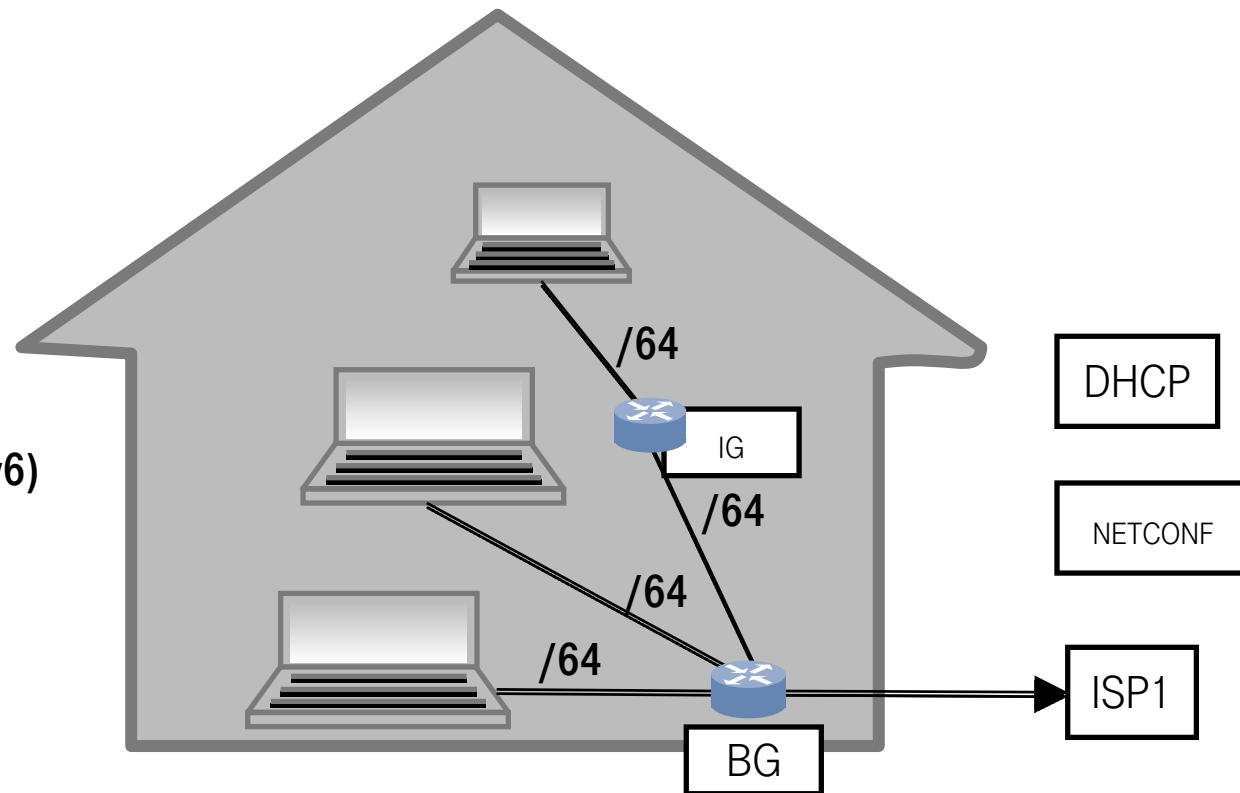
Prefix and Color information

DNS resolver

... etc

... Basically anything in DHCP(v6)

... And more



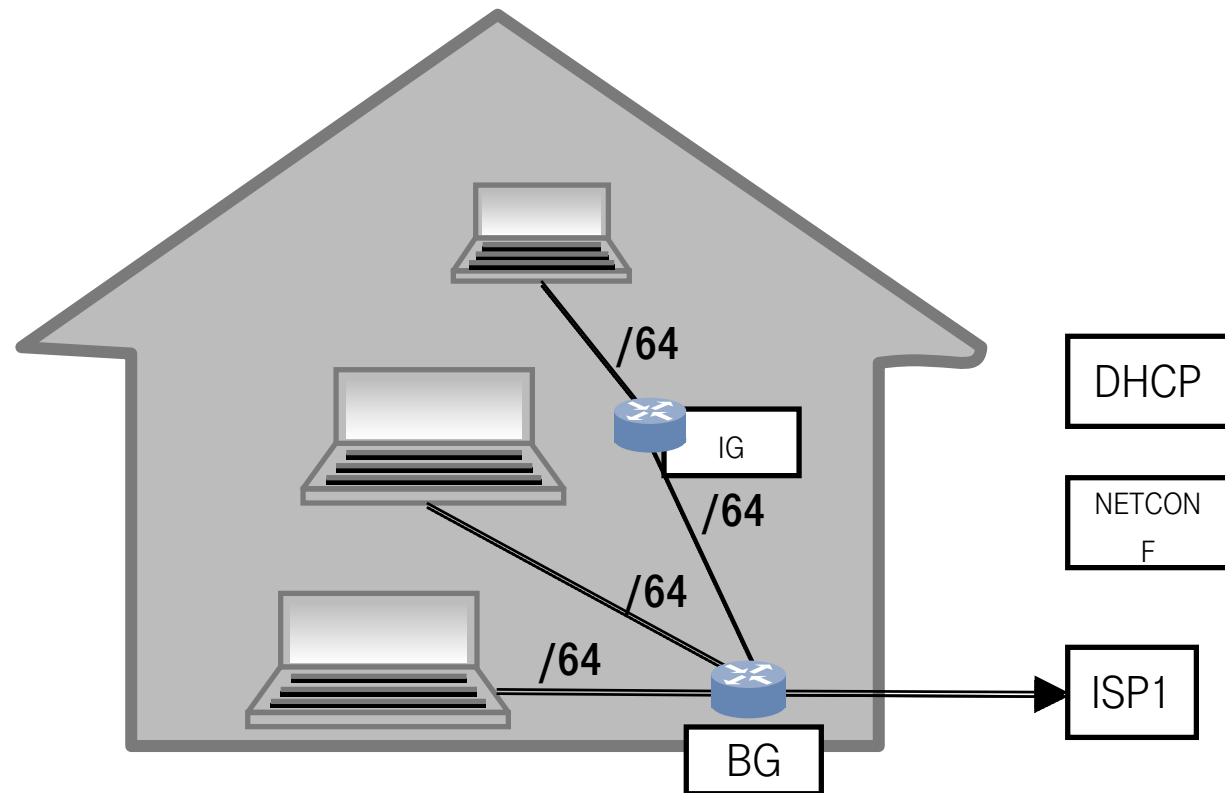
# Hosts, same thing again

HOST starts as host using RA+DHCP

From DHCP it gets NETCONF server address and cryptographic keys

NETCONF server sends service information plus other information to HOST

Now BG can start using services



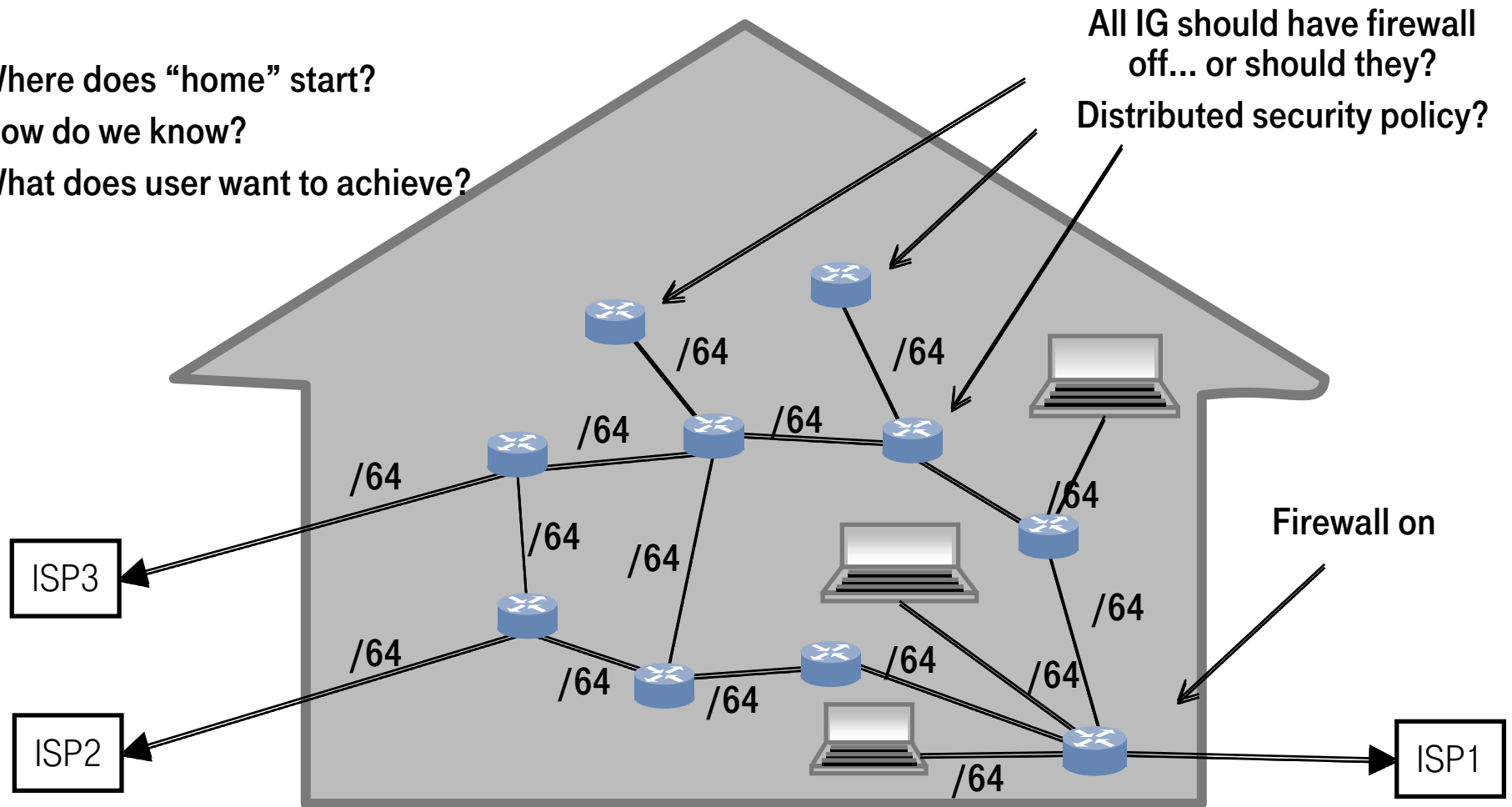
# SECURITY

We want security policy, but how to do that?

Where does “home” start?

How do we know?

What does user want to achieve?

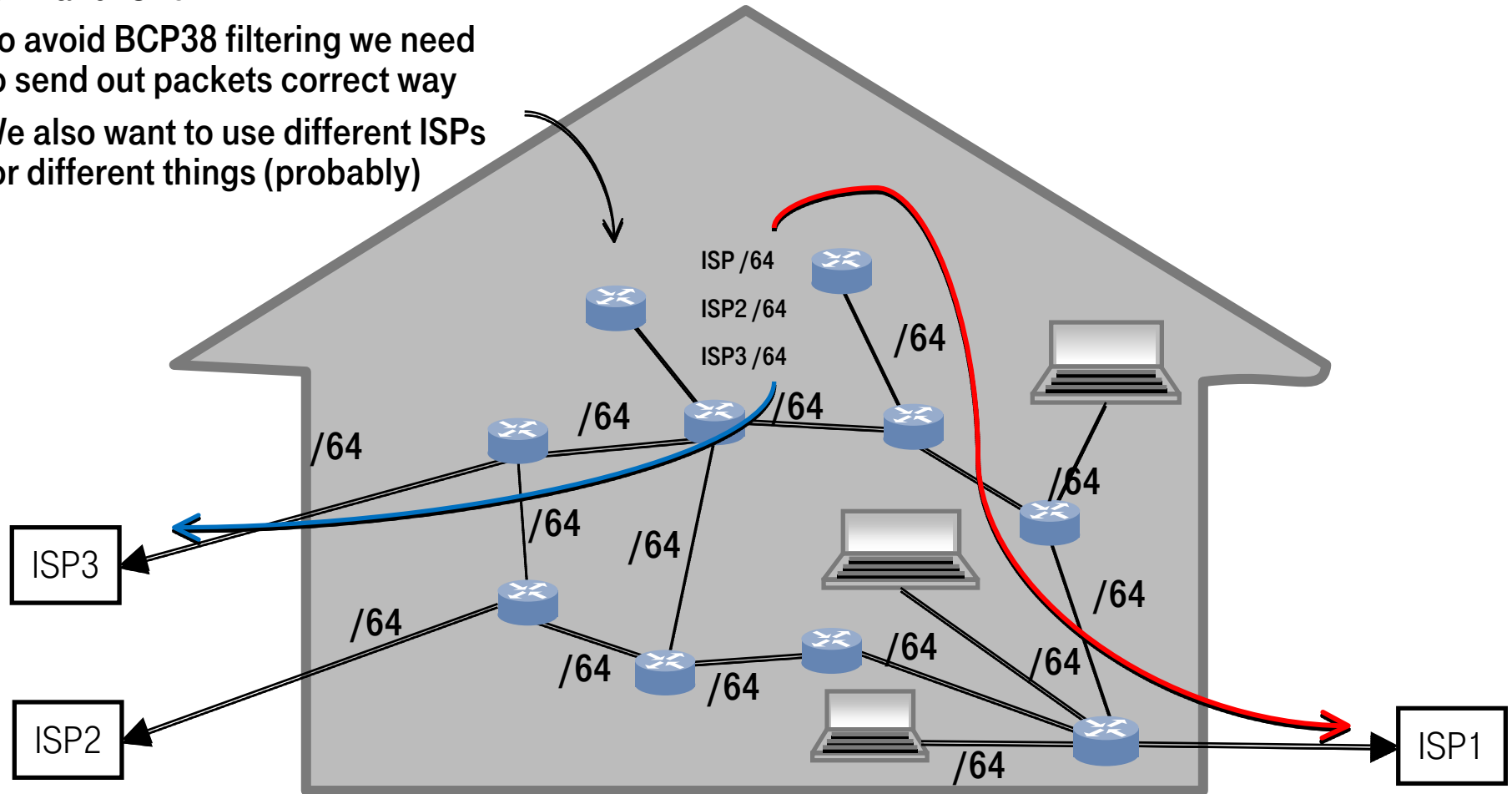


# SOURCE BASED ROUTING

Link carry prefix(es) from ISP,  
ISP2 and ISP3

To avoid BCP38 filtering we need  
to send out packets correct way

We also want to use different ISPs  
for different things (probably)

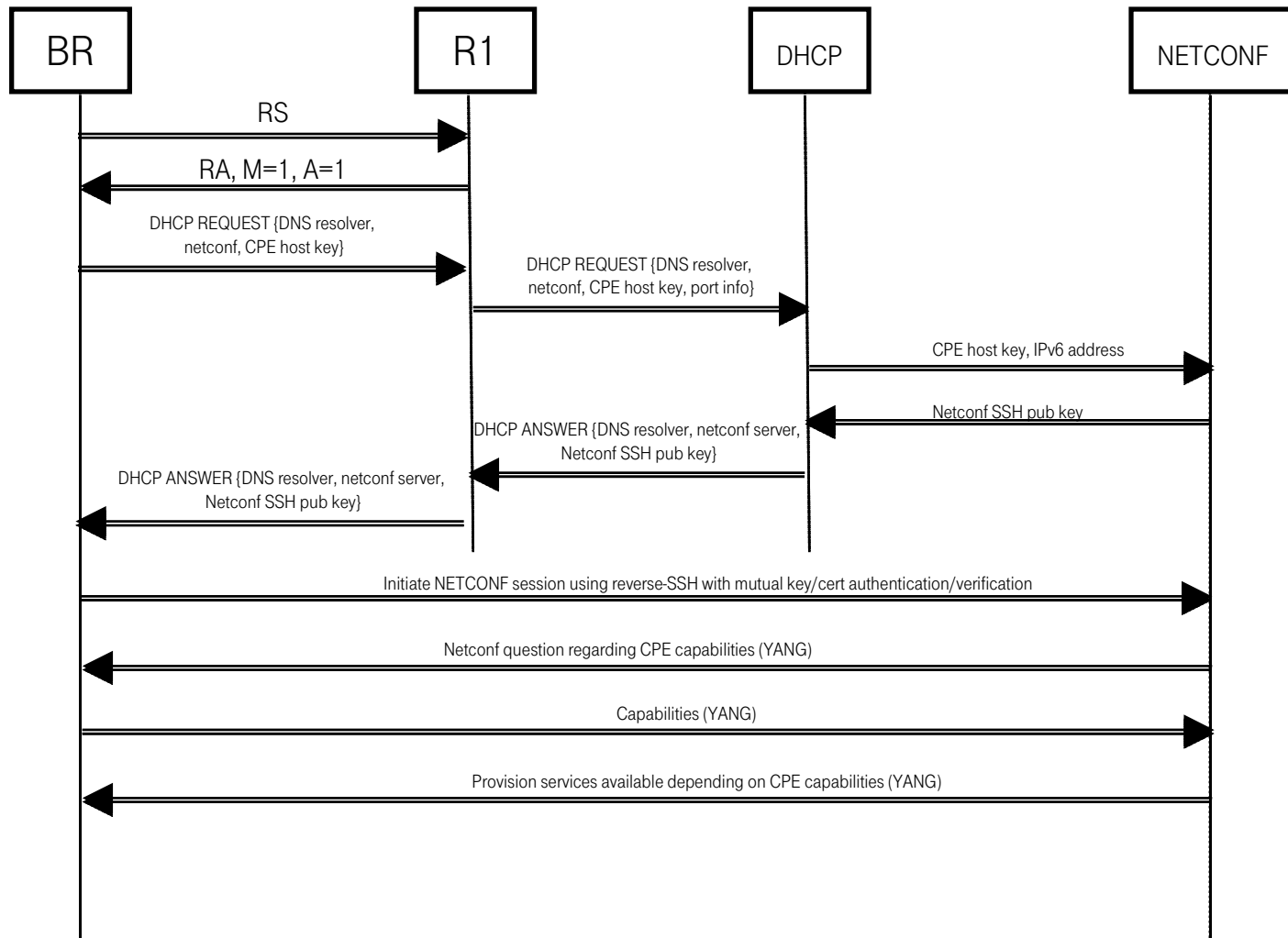


**ISIS can make everybody aware of the topology of the network.**

## Address family for capabilities and device type?



# BOOTSTRAP PROCESS BR



# QUESTIONS?

Now you can bring out your tar and feathers and start throwing things at me..

# THANKS!

