A+P implementations and experiments results

X.Deng, M. Boucadair
T.Zheng L.Wang
France Telecom
Y.Lee
Comcast

X.Huang Q.Zhao Yan.Ma
BUPT

IETF 82

Taipei

Outlines

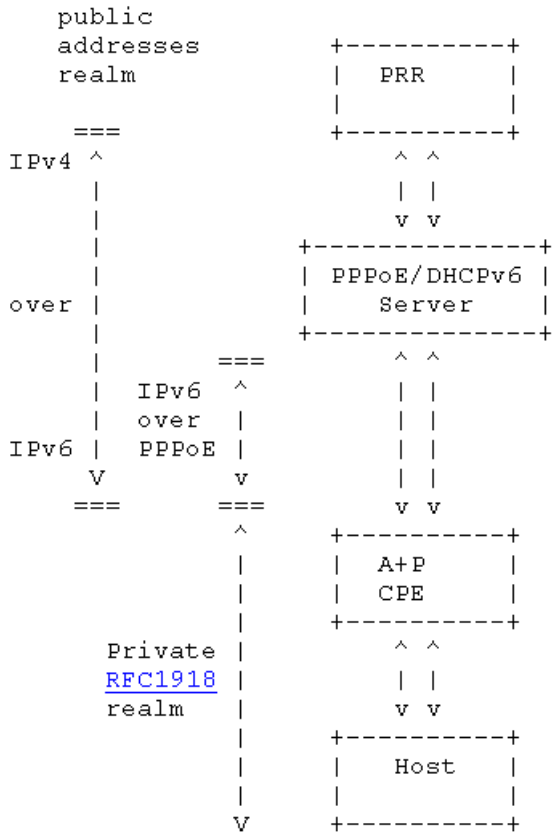
- A work since 2008…
- Why?
  - Impacts on customers, operational issues, and protocol design requirements/considerations..
- Items
  - Implementing and deploy A+P with operators network
  - Implementing Non-continuous port allocation flavor of A+P
  - UPnP 1.0 efforts with A+P alike approach
  - Experiments results of Port/session usages (applicable to general IPv4 sharing context)

Where are we?

- Implementing and deploy A+P with operators network
- Implementing Non-continuous port allocation flavor of A+P
- UPnP 1.0 efforts with A+P alike approach
- Experiments results of Port/session usages (applicable to general IPv4 sharing context)

Implementation Overview

- Network Topology



- CPE Parameters

Model	CPU Speed (MHz)	Flash (MB)	RAM (MB)	Wireless NIC	Wireless Standard	Wired Ports
Linksys WRT54GS	200	8	32	Broadcom (integrated)	11g	5

Two flavors of implementations

- Port Range A+P (Continuous port range)
  - Allocates a range of ports per customer

- Scattered Port Sets A+P (Non-Continuous port sets)
  - Validate feasibility of non-continuous ports with A+P approach;
    - one possible solution among others to offer non-continuous port provisioning.
  - Evaluate efforts and investigate possibility of making UPnP 1.0 applications still work with this approach

Where are we?

- Implementing and deploy A+P with operators network
- Implementing Non-continuous port allocation flavor of A+P
- UPnP 1.0 efforts with A+P alike approach
- Experiments results of Port/session usages (applicable to general IPv4 sharing context)

Scattered ports sets allocation Targets

- Not sacrifice port randomization compare to Continuous port range allocation

- Evaluate efforts and investigate possibility of making UPnP 1.0 applications still work with A+P approach

How to provision scattered ports?

- Only two parameters
- Subscribers ID pattern

1st	2nd	3rd	4th	5th	6th	7th	8th
0	s	0	0	s	0	s	0

9th	10th	11th	12th	13th	14th	15th	16th
s	0	s	0	0	0	0	0

- Subscribers ID value

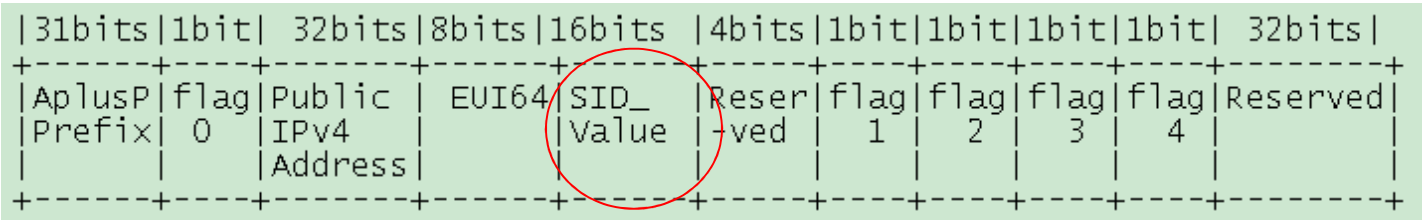
1st	2nd	3rd	4th	5th	6th	7th	8th
1	**0**	1	1	**0**	1	**0**	1

9th	10th	11th	12th	13th	14th	15th	16th
0	1	**1**	1	1	1	1	1

How to derive CPE IPv6 prefix in Scattered Port Sets context

Formed by stateless PRR:

- Subscribers ID value = Destination port & Subscribers ID pattern;
- Subscribers ID pattern could either be per domain or per address pool, depends on ports allocation policy.



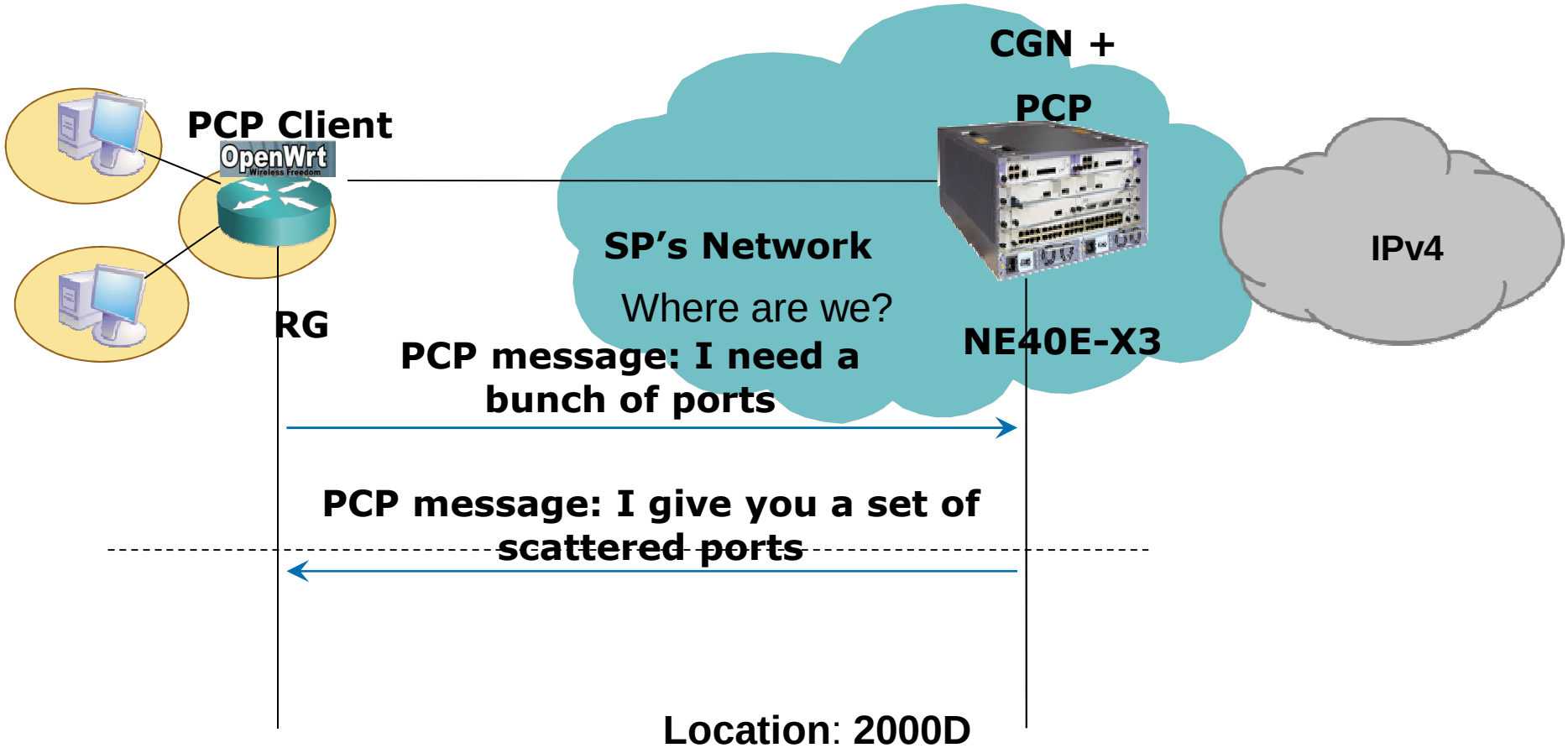
Random ephemeral port selection among restricted ports sets for Customer NAT

- Why preserve randomness as much as possible ?
  - RFC 6056 "Recommendations for Transport-Protocol Port Randomization"
  - Should be a **preserved feature/requirement** for other Port sets allocation algorithms as well?

```
do{
    restricted_next_ephemeral = (random()|subscriber_ID_pattern)
                                & subscriber_ID_value;
    if(five-tuple is unique)
    return restricted_next_ephemeral;
```

Only one line code needs to be changed on legacy customer NAT!

An Implementation of Scattered Port Sets (Demoed in DS-Lite use case)



Check out website for this demo: http://130.129.48.23:35328/

This website worked based on the live demo during IETF 81, and has been moved to :
http://opensourcev6transtechnologies.weebly.com/ietf-81-pcp-demo-site.html

Where are we?

- Implementing and deploy A+P with operators network
- Implementing Non-continuous port allocation flavor of A+P
- UPnP 1.0 efforts with A+P alike approach
- Experiments results of Port/session usages (applicable to general IPv4 sharing context)

UPnP 1.0 extensions experiments

- UPnP 1.0 Actions/parameters extension
  - GetPortRangeLow(), and GetPortRangeHigh ()
  - NewExternal_IPAddress, NewPortRange_Low and NewPortRange High
  - Implemented IGD: Linux IGD 1.0
  - Implemented application / Emule 0.50a
- UPnP 1.0 friendly port allocation
  - UPnP 1.0 applications behaviors of asking for an external port
  - Do applications work with UPnP 1.0 friendly port sets allocations method?
- UPnP 2.0 standardized
  - But no/few applications upgraded

UPnP 1.0 applications behaviors of asking for an external port

Application	Behaviors
Microtorrent v2.2 (also known as uTorrent)	call GetSpecificPortMapping by incremental by 1 each time, until find an external port available, and then call AddPortMapping,or return error after five failures
Emule v0.50a	call AddPortMapping, after finding the external port not available return error
Azureus v4.6.0.2	call AddPortMapping, after finding the external port not available, try the same port 5 more times by call AddPortMapping, then return error
Shareazav2.2.5.7	call GetSpecificPortMapping, after finding the external port not available, return error without issuing AddPortMapping

Does it work with UPnP 1.0 friendly port provisioning method?

- If instance a scattered port allocation with port sets interval less than 5
  - Have to design Subscriber ID Pattern 0x02 and,
  - Sharing Ratio: 2 (Assumed to be *not* a practical/efficient sharing ratio in most use cases)
- Only one application among others would be made working
- Only the chances of success have been increased with other applications

Application	Does it work with UPnP 1.0 friendly port provisioning method?
Microtorrent v2.2	Yes
Emule v0.50a	1/5 chance of working
Azureus v4.6.0.2	1/5 chance of working
Shareazav2.2.5.7	1/5 chance of working

Where are we?

- Implementing and deploy A+P with operators network
- Implementing Non-continuous port allocation flavor of A+P
- UPnP 1.0 efforts with A+P alike approach
- Experiments results of Port/session usages (applicable to general IPv4 sharing context)

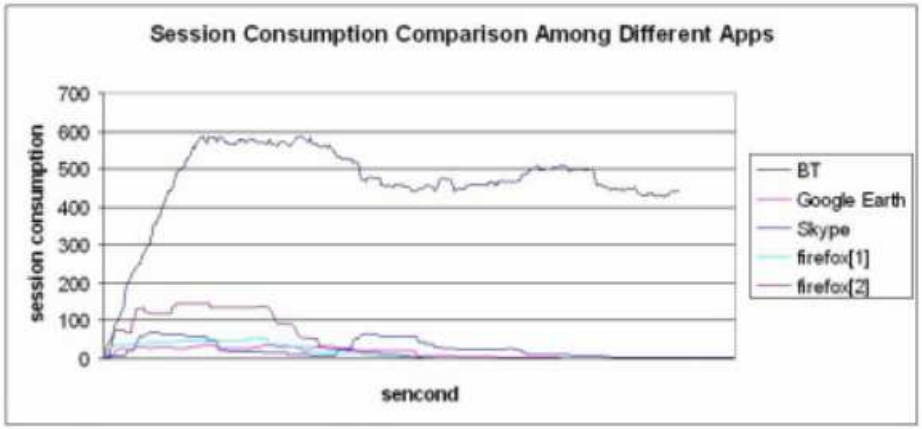
Port usages of applications

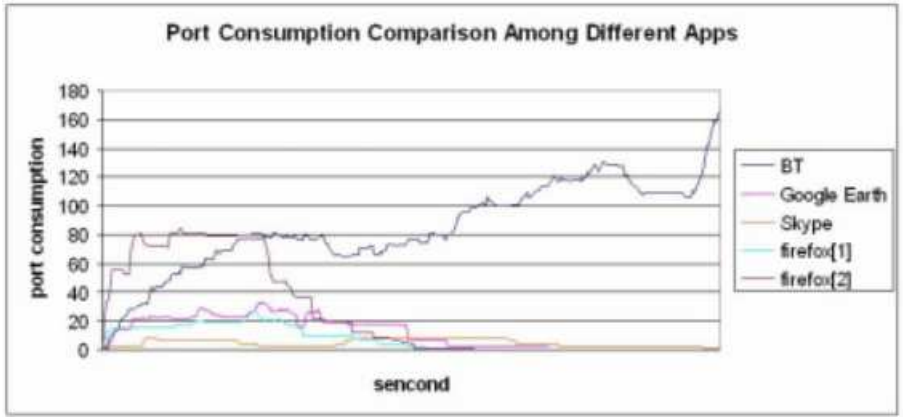
Application	Test case	Maximum port usage	Duration (seconds)
IE	browsing a news website	20-25	200
	browsing a video website	40-50	337
Firefox	browsing a news website	25-30	240
	browsing a video website	80-90	230
Chrome	browsing a news website	50-60	340
	browsing a video website	80-90	360
Android Chrome	browsing a news website	40-50	300
	browsing a video website	under 10	160
Google Earth	locating a place	30-35	240
Android Google Earth	locating a place	10-15	240
Skype	make a call	under 10	N/A
BitTorrent	downloading a file	200	N/A

Linux NAT: Kernel 2.6.32 (non-EIM)

Multiplexing sessions on the same source port?

- Common assumption of EIM NAT doesn't multiplex sessions on the same source port
- Test results shows otherwise, for UDP
  - For UDP it might not matter if the NAT is EIM or non-EIM, since hosts (Utorrent,skypes, etc.,) tend to reuse the same internal IP for different remote peers
  - Thanks to Simon Perreault, with whom discussion/conclusion achieved with offline based on the experiments results.





Summary
-What have been learnt?

- A+P is implementable and deployment with operators network

- Non-continuous port allocation is feasible for A+P alike approaches, Besides continuous port allocation.

- Making UPnP 1.0 work with A+P ?
  - Efforts VS. Results
  - Upgrading to UPnP 2.0 sounds a more simpler and reasonable approach

- Port/session usages, applicable to general IPv4 sharing context
  - Typical port usages of applications, thereby offering data for sharing ratio designing
  - UDP applications multiplexes sessions on the same port, which results in that the amount of sessions more than amount of ports no matter EIM NAT or non-EIM NAT
  - A need to document it in more detail?

Current Status and Next step

- Presented in v6ops IETF 81, Quebec
- Presented in Softwire interim meeting, Beijing
- Feedbacks from above presentations
  - people see a value of publishing it as informational
- Presented in Softwire IETF 82,Tapei
  - Feedbacks: WG adoption?

THANKS!