Using PCP to trigger dynamic DNS updates

draft-deng-pcp-ddns-01

Xiaohong Deng, Mohamed Boucadair and Xu Wang

IETF 84, July 2012

Contents

- Focuses on problems encountered when using dynamic DNS in address sharing context (e.g., DS-Lite, NAT64, A+P)
- Particular focus on the DS-Lite case
- Issues, possible solutions
- Preliminary implementation results
- Validation of one of the solutions

Current use of DDNS

- Dynamic DNS (DDNS) is a widely deployed service to facilitate hosting servers at home premises
- Works in a client and server mode
- Update IP address changes to DDNS Server when the DDNS client detects any change of the assigned IP address

Challenges for DDNS in IP Address Sharing Context

- The DDNS service MUST be able to maintain an alternative port number instead of the default port number

- Appropriate means to instantiate port mapping in the address sharing device MUST be supported to allow for incoming sessions

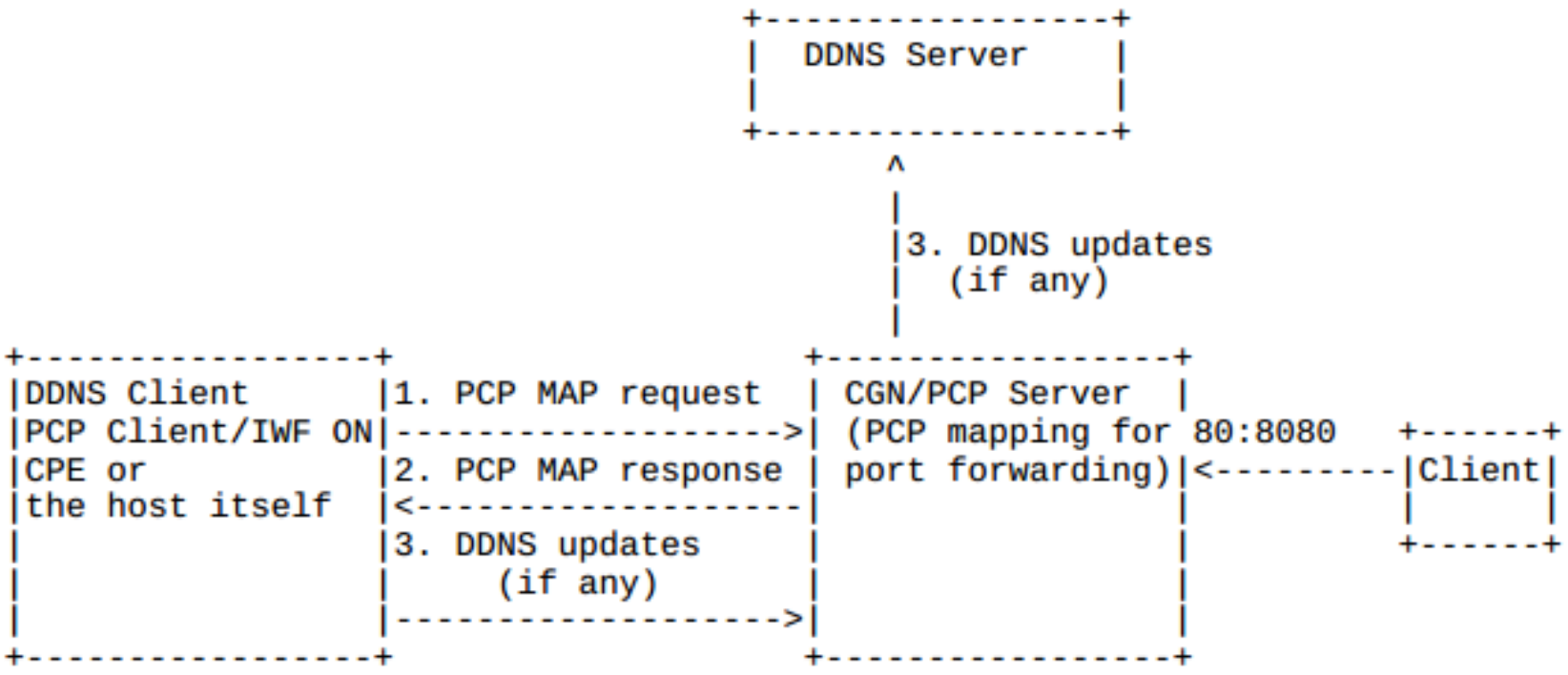
- DDNS client MUST be triggered by the change of the external IP address and the port number assigned by the address sharing device

Locate a Service Port

- Use service URIs (e.g., FTP, SIP, HTTP) which embed an explicit port number
  - Uniform Resource Identifier (URI) defined in [RFC3986] allows to carry port number in the syntax (e.g., mydomain.example:15687)

- Use SRV records
  - The majority of browsers however do not support this record type

Detect the changes



Means of detecting the change

- Two alternatives have been considered:
  - **If PCP mapping repair is not supported**, the DDNS client issues periodically PCP requests to check whether any change has occurred
  - **If PCP mapping repair is supported**, the PCP Server will notify the client when there is any change. The DDNS client does only check a local mapping table

Means of detecting the change (1)

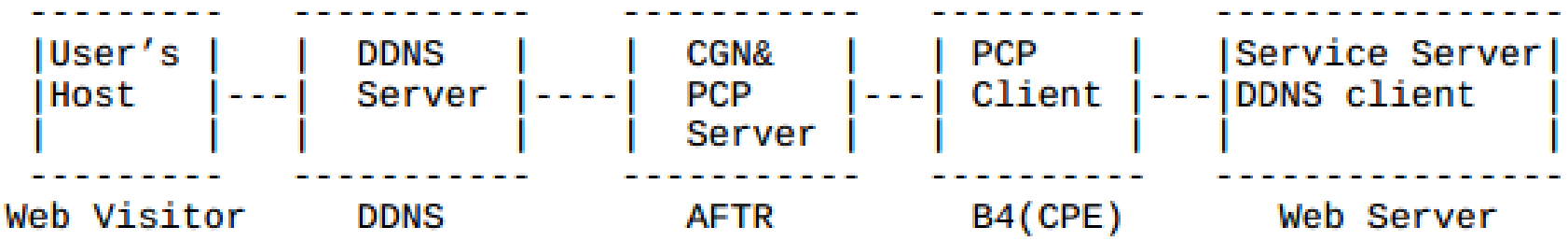
- The host issues periodically (e.g., 1h) PCP MAP requests with a lifetime (e.g., 30mn) for the purpose of enquiring external IP address
  - Companion mappings are needed to reach the internal servers. The short-lived mapping will trigger refreshing these mappings
  - The server will assign the minimum lifetime it is configured
- Upon change of the external IP address, the DDNS client updates the records
  - The mappings are needed to be re-installed

Means of detecting the change (2)

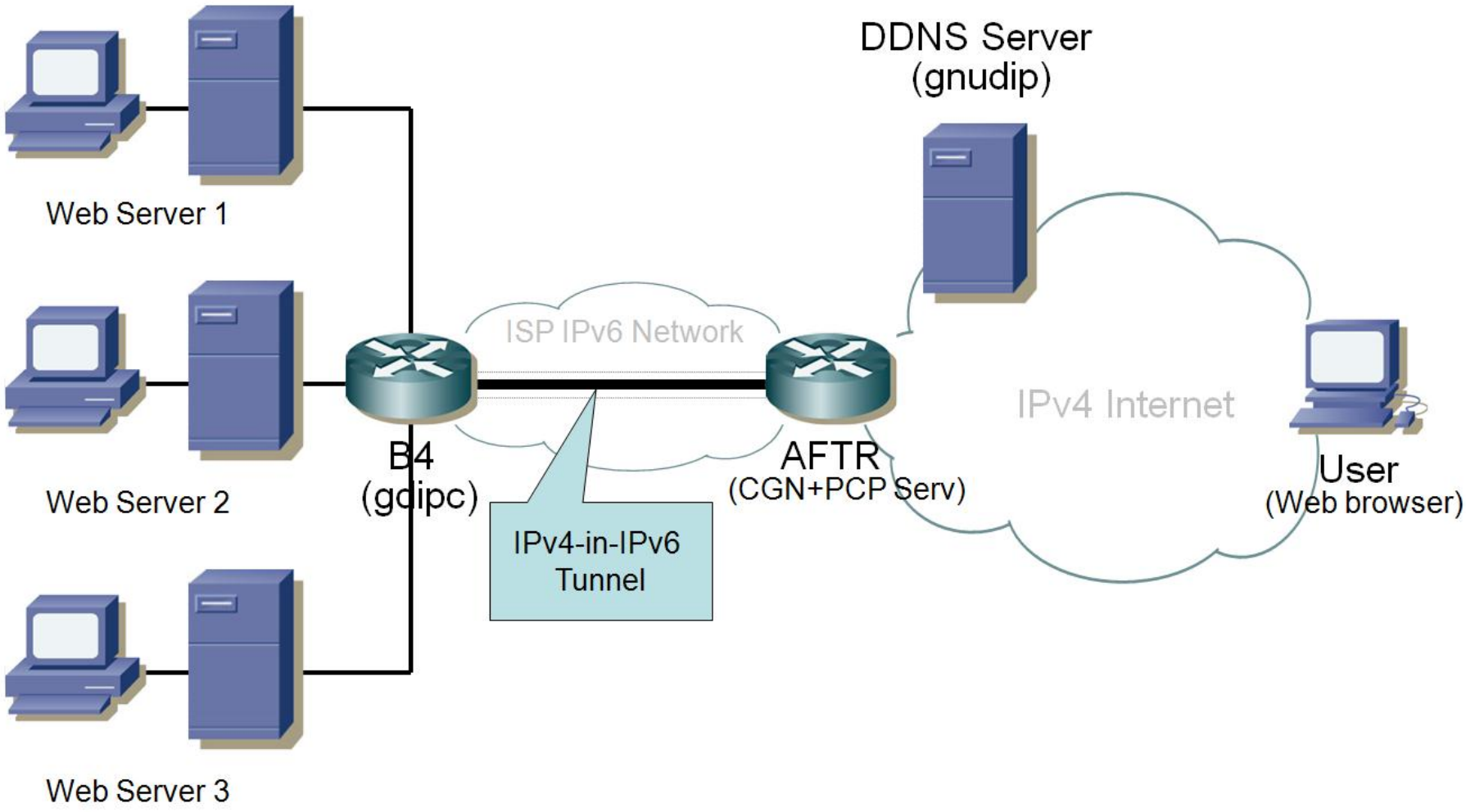
- It repeatedly checks the local mapping table periodically (e.g., 5mn, 30mn, 1h) if there is any change of the state
- Upon change of a mapping at the server side, mapping repair mechanism is executed
- Upon change of the external IP address, the DDNS client updates the records in the DDNS server

An implementation : Topology

- Web Visitor
- DDNS
- AFTR
- B4 (CPE)
- Web Server



An implementation :Typology



Implementation overview

- Implemented nodes: DDNS Server (gnudip), DDNS Client (gdipc)
- the new function introduced: PCP support (gdipc could fetch External IP address and port number from PCP Server)
- Program language: Perl
- Platform (OS): Openwrt
- Open source ddns system
  - name : GnuDIP
  - version: 2.3

An implementation : functions

- DDNS Client
  - PCP support (could get PCP records)
  - Update DDNS records dynamically
    - Now the application has supported IP address dynamically update (not for port number so far)
    - Some commercial DDNS server supports port redirect
  - Two communication methods (HTTP and TCP)
    - HTTP used to update external IP change

An implementation : Configuration and running

- gdipc configuration:

```
# ./gdipc.pl -c
Using Update Configuration Mode
Configuration file name: /root/.GnuDIP2
Username: gesaint
Domain: dyn.borgmann.tv
Connect by direct TCP (d) or web server (w) [d]: w
GnuDIP Server - host[:port]: 170.56.59.197
Server URL [/gnudip/cgi-bin/gdipupdt.cgi]: /cgi-bin/gdipupdt.cgi
Password: gesaint
Cache File [/root/.GnuDIP2.cache.gesaint.dyn.borgmann.tv]:
Minimum Seconds Between Updates [0]: 10
Maximum Seconds Between Updates [2073600]: 30
```

- gdipc -p option: PCP support

```
[root@localhost bin]# ./gdipc.pl -p
====  gdipc.pl running:  Fri Apr 27 18:17:46 2012  ====
Configuration file name: /root/.GnuDIP2
Cache file name: /root/.GnuDIP2.cache.gesaint.dyn.borgmann.tv
resultcode:0
PCP External IP Address:198.18.200.1
Cache IP:123.127.237.248
Attempting update at dyn.borgmann.tv ...
Update to address 198.18.200.1 from 123.127.237.248 successful for gesaint.d
yn.borgmann.tv
```

Implementation: running code & packets on wires (1)

Connection data and socket initialization:

```
if($opt_p){
  my $pcpserveripaddr = "2001:da8:202:108::1";
  my $pcpserverport = 5351;
  my $sinpcpserveraddr = sockaddr_in6($pcpserverport, $pcpserveripaddr);

  socket(SERVER, PF_INET6, SOCK_STREAM, getprotobyname('udp'));
```

Issuing packet:

```
$pcpmapreq = pack('b8', $version);
 ......
$pcpmapreq .= inet_pton(AF_INET6, $sugipaddr);
```

Communicate with PCP Server

```
send(SERVER, $pcpmapreq, 0, $sinpcpserveraddr);
recv(SERVER, $pcpmapres, 256, 0);
```

Parse result code and PCP external IP address

```
my $resultcode = substr $pcpmapres, 3, 1;
my $pcpextipaddr = substr $pcpmapres, (32 + 12), 4;
```

Update process

```
if ($opt_p) {
  $ip = $pcpextipaddr;
}
```

Implementation: running code & packets on wires (2)

1.	1 0.000000	2001:ffff:ffff:300::1	2001:ffff:ffff:300::2	PCP	132 Source port: 34453 Destination port: nat-pmp
2.	2 0.000703	2001:ffff:ffff:300::2	2001:ffff:ffff:300::1	PCP	132 Source port: nat-pmp Destination port: 34453
	3 2.310392	192.168.1.1	170.56.59.197	TCP	116 59440 > http [SYN] Seq=0 Win=5680 Len=0 MSS=14
	4 2.770631	170.56.59.197	192.168.1.1	TCP	76 http > 59440 [SYN, ACK] Seq=0 Ack=1 Win=5792 L
	5 2.770890	192.168.1.1	170.56.59.197	TCP	108 59440 > http [ACK] Seq=1 Ack=1 Win=5760 Len=0
3.	6 2.770895	192.168.1.1	170.56.59.197	HTTP	383 GET /cgi-bin/qdipupdt.cgi?salt=KLb27Pq9hV&tim
	7 3.855456	170.56.59.197	192.168.1.1	HTTP	767 HTTP/1.1 200 OK (text/html)
	8 3.886411	170.56.59.197	192.168.1.1	TCP	68 http > 59440 [FIN, ACK] Seq=700 Ack=276 Win=69
	9 3.886673	192.168.1.1	170.56.59.197	TCP	108 59440 > http [FIN, ACK] Seq=276 Ack=701 Win=71
	10 4.368846	170.56.59.197	192.168.1.1	TCP	68 http > 59440 [ACK] Seq=701 Ack=277 Win=6912 Le

1. PCP request from DDNS Client 2. PCP response from PCP server

```
Port Control Protocol
  Version: 1
  R bit: Request (0)
  Opcode: MAP (0x01)
  Reserved: 0
  Requested Lifetime: 3600 sec
  PCP Client's IP Address: 2001:ffff:ffff:300::1 (2001:ffff:ffff:300::1)
MAP  Request
    Protocol: UDP (17)
    Reserved: 0
    Internal Port: 80
    Suggested External Port: 80
    Suggested External IP Address: ::ffff:0.0.0.0 (::ffff:0.0.0.0)
Third Party Option
    Option Code: third party (0x01)
    Reserved: 0
    Option Length: 16 bytes
    Data: 00000000000000000000ffffc0a80201
```

```
Port Control Protocol
  Version: 1
  R bit: Response (1)
  Opcode: MAP (0x01)
  Reserved: 0
  Result Code: 0
  Lifetime: 3600
  Epoch Time: 47
  Reserved: 000000000000000000000000
MAP  Response
    Protocol: UDP (17)
    Reserved: 0
    Internal Port: 80
    Assigned External Port: 64001
    Assigned External IP Address: ::ffff:198.18.200.1 (::ffff:198.18.200.1)
Third Party Option
    Option Code: third party (0x01)
    Reserved: 0
    Option Length: 16 bytes
    Data: 00000000000000000000ffffc0a80201
```

3. External IP Update from DDNS Client to DDNS Server:

Capture packets file:

capture_packets.pcap

Next Step

- For Mapping domain name to IP address + port number
- DDNS Client
  - support maintaining port number from client (now client has supported IP updating)
- DDNS Server
  - Mappings between domain names and IP address + port number maintaining

Q & A

Thanks!