

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 29, 2017

Q. Sun
C. Xie
China Telecom
Y. Lee
Comcast
M. Chen
FreeBit
T. Li

Tsinghua University
January 25, 2017

Deployment Considerations for Lightweight 4over6
draft-ietf-softwire-lightweight-4over6-deployment-00

Abstract

Lightweight 4over6 is a mechanism which moves the translation function from tunnel lwAFTR (AFTR) to lwB4s (B4s), and hence reduces the mapping scale on the lwAFTR to per-customer level. This document discusses various deployment models of Lightweight 4over6. It also describes the deployment considerations and applicability of the Lightweight 4over6 architecture.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 29, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

Internet-Draft

lightweigh-4over6-deployment

January 2017

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Deployment Model	4
3.	Overall Deployment Considerations	6
3.1.	Addressing and Routing	6
3.2.	Port-set Management	6
3.3.	lwAFTR Discovery	7
3.4.	Impacts on Accouting	7
4.	lwAFTR Deployment Consideration	8
4.1.	Logging at the lwAFTR	8
4.2.	MTU and Fragmentation Considerations	8
4.3.	Reliability Considerations of lwAFTR	8
4.4.	Placement of AFTR	9
4.5.	Port set algorithm consideration	9
4.6.	Path Consistency Consideration	9
5.	lwB4 Deployment Consideration	11
5.1.	NAT traversal issue	11
5.2.	Static Port Forwarding Configuration	11
6.	DS-Lite Compatibility Consideration	12
6.1.	Case 1: Integrated Network Element with Lightweight 4over6 and DS-Lite AFTR Scenario	12
6.2.	Case 2: DS-Lite Coexistent scenario with Separated AFTR	13
7.	Acknowledgement	14
8.	References	15
	Appendix 1. China Telecom Experimental Result	18
1.1.	Experimental Environment	18
1.2.	Experimental Results	19
1.3.	Conclusions	20
	Appendix 2. Tsinghua University Experimental Result	21
2.1.	Experimental Environment	21
2.2.	Experimental Results	22
2.3.	Conclusions	22
	Authors' Addresses	23

Internet-Draft

lightweigh-4over6-deployment

January 2017

1. Introduction

Lightweight 4over6 [[I-D.ietf-softwire-lw4over6](#)] is an extension to DS-Lite which simplifies the AFTR module [[RFC6333](#)] with distributed NAT function among B4 elements. The lwB4 in Lightweight 4over6 is provisioned with an IPv6 address, an IPv4 address and a port-set. It performs NAPT on end user's packets with the provisioned IPv4 address and port-set. IPv4 packets are forwarded between the lwB4 and the lwAFTR over a Softwire using IPv4-in-IPv6 encapsulation. The lwAFTR maintains one mapping entry per subscriber with the IPv6 address, IPv4 address and port-set. Therefore, this extension removes the NAT44 module from the AFTR and replaces the session-based NAT table to a per-subscriber based mapping table. This should relax the requirement to create dynamic session-based log entries. This mechanism preserves the dynamic feature of IPv4/IPv6 address binding as in DS-Lite, so it has no coupling between IPv6 address and IPv4 address/port-set as any full stateless solution ([RFC6052](#) or [I-D.ietf-softwire-map](#)) requires. This document discusses deployment models of Lightweight 4over6. It also describes the deployment considerations and applicability of the Lightweight 4over6 architecture.

Terminology of this document follows the definitions and abbreviations of [[I-D.ietf-softwire-lw4over6](#)].

2. Deployment Model

Lightweight 4over6 is suitable for operators who would like to free any correlation of the IPv6 address with IPv4 address and port-set (or port-range). In comparison to full stateless solutions like MAP [[I-D.ietf-softwire-map](#)] and 4rd [[I-D.ietf-softwire-4rd](#)], Lightweight 4over6 frees address planning of IPv6 delegation for CPE from mapping rule administration and management in the network. Thus, IPv6 addressing is completely flexible to fit other deployment requirements, e.g., auto-configuration, service classification, user management, QoS support, etc. The philosophy here is that bits of IPv6 address should be left for IPv6 usage first.

Lightweight 4over6 can be deployed in a residential network (depicted in Figure1). In this scenario, a lwB4 would acquire an IPv4 address and a port-set after a successful user authentication process and IPv6 provisioning process. Then, it establishes an IPv4-in-IPv6 softwire using the IPv6 address to deliver IPv4 services to its connected host via the lwAFTR in the network. The lwB4 can act as a CPE, or software located in the host. The lwAFTR supports Lightweight 4over6 which keeps the mapping between lwB4's IPv6 address and its allocated IPv4 address + port set. The supporting system may keep the binding information as well for logging and user management.

```
+-----+
| Supporting |
| System    |
+-----+
```

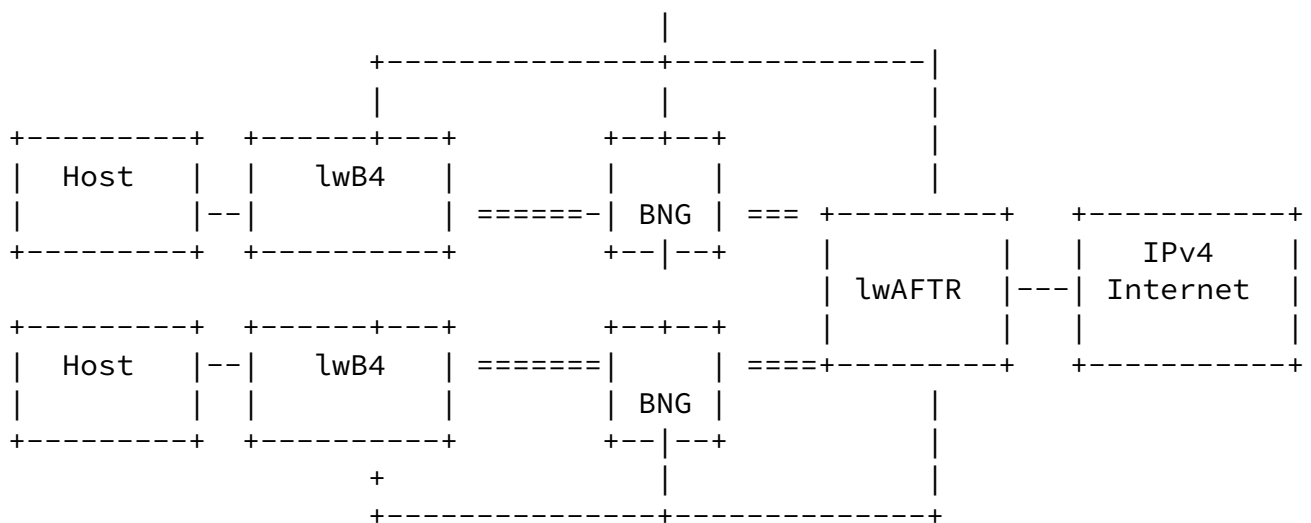


Figure 1 Deployment Model

There are two deployment models in practice: one is called bottom-up and the other is top-down. In bottom-up model, after port-restricted

IPv4 address is allocated to a given subscriber, the lwAFTR will report mapping records to the supporting system on creating a binding for traffic logging if necessary. Operators may use [\[I-D.ietf-behave-syslog-nat-logging\]](#) or [\[I-D.ietf-behave-ipfix-nat-logging\]](#) to report the port set allocated by lwAFTR. In this way, the lwAFTR can determine the binding by its own and there is little impact on existing network architecture. In top-down model, the Supporting system should firstly determine the binding information for each subscriber and then synchronize it with the lwAFTR. With this method, one binding record can be easily synchronized with multiple lwAFTRs and stateless failover can be achieved. However, new mechanism (e.g. [\[I-D.zhou-dime-4over6-provisioning\]](#)) needs to be introduced to notify each individual binding record between the Supporting system and the lwAFTR.

Lightweight 4over6 can also be deployed in a campus or enterprise network, (depicted in Figure2). In this scenario, a lwB4 acts as a wireless AP and is connected to a number of hosts. The lwB4 first acquire the IPv4 address and port-set information, then it establishes an IPv4-in-IPv6 softwire using the IPv6 address to deliver IPv4 services to its connected host via the lwAFTR in the

network. A network management system could be used to receive statistic information of the network equipments, such as the binding table, network load, and connected device. NETCONF[RFC6241] could be used for synchronising lwB4's IPv6 address and its allocated IPv4 address + port set with the lwAFTR. The network management system may keep the binding information as well for logging and user management.

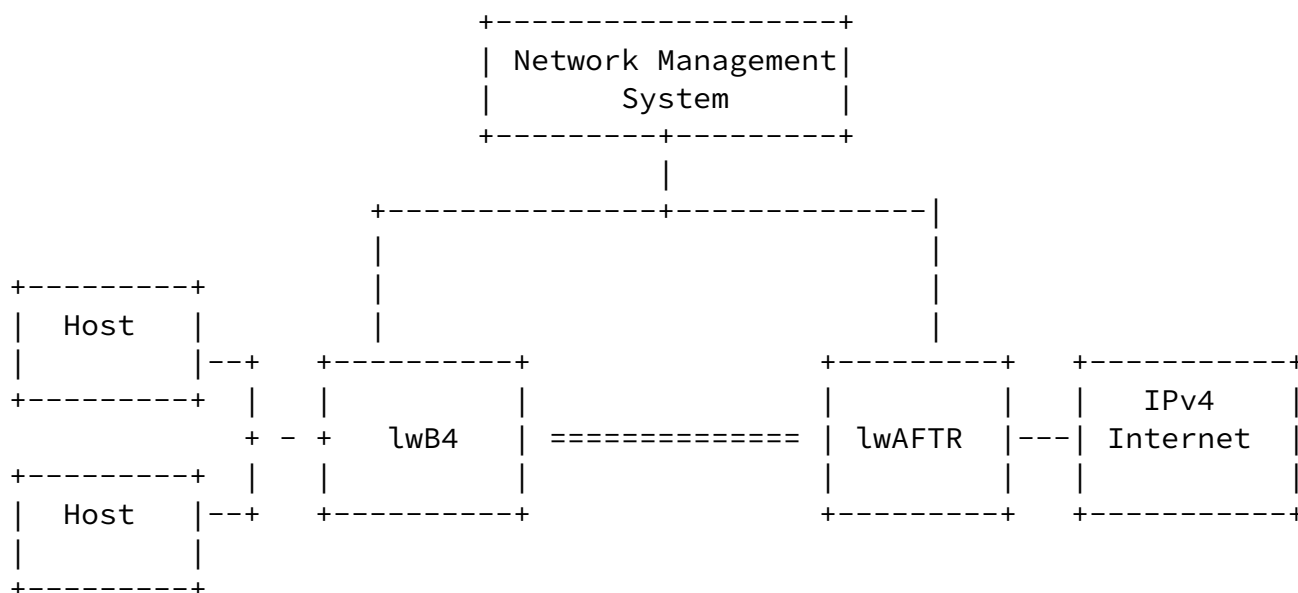


Figure 2 Deployment Model

[3. Overall Deployment Considerations](#)

[3.1. Addressing and Routing](#)

In Lightweight 4over6, there is no inter-dependency between IPv4 and IPv6 addressing schemes. IPv4 address pools are configured centralized in lwAFTR for IPv6 subscribers. These IPv4 prefix must advertise to IPv4 Internet accordingly.

For IPv6 addressing and routing, there are no additional addressing and routing requirements. The existing IPv6 address assignment and routing announcement should not be affected. For example, in PPPoE scenario, a CPE could obtain a prefix via prefix delegation procedure, and the hosts behind CPE would get its own IPv6 addresses within the prefix through SLAAC or DHCPv6 statefully. This IPv6

address assignment procedure has nothing to do with restricted IPv4 address allocation.

[3.2.](#) Port-set Management

In Lightweight 4over6, each lwB4 will get its restricted IPv4 address and a port-set after successful user authentication process and IPv6 provisioning process. This port-set assignment can be achieved by DHCPv4-over-DHCPv6 [I-D.ietf-dhc-dhcpv4-over-dhcpv6] and PCP [I-D.ietf-pcp-port-set].

Operator may use DHCPv4 to provision IPv4 address to the lwB4. In a typical deployment, the DHCP server is a centralized DHCP server and lwAFTR is the DHCP relay agent to relay the dhcp messages to the server over unicast. Rarely DHCP server will collocate with the lwAFTR to provision IPv4 resources to the lwB4.

Operator may also use PCP Port-set Option to provision IPv4 address and port-set to the lwB4. In a typical deployment, PCP server will collocate with lwAFTR, and the subscriber's binding can be determined by lwAFTR. The PCP request should be sent to the lwAFTR's tunnel end-point address. It is not common that PCP server will be centralized deployed in which the lwAFTR is the PCP proxy to relay PCP requests.

It is also possible that subscriber's binding is determined in AAA server. In this case, the BNGs will embed with a DHCPv4-over-DHCPv6 server function which allows them to locally handle any DHCPv4-over-DHCPv6 requests initiated by hosts. The AAA server will pass the subscriber's binding to a BNG using the AAA attribute in [I-D.sun-software-lw4over6-radext] and in turn populates the mapping of the lwB4.

Some operators may offer different service level agreements (SLA) to users that some users may require more ports than others. In this deployment scenario, the operator can implement differentiated policies in provisioning system specified to a user's lwB4 or a group of lwB4s to allocate a certain range of port-set. The lwAFTR may also run multiple instances with different port-set sizes to build the mapping table.

[3.3.](#) lwAFTR Discovery

A Lightweight 4over6 lwB4 must discover the lwAFTR's IPv6 address before offering any IPv4 services. This IPv6 address can be learned through an out-of-band channel, static configuration, or dynamic configuration. In practice, Lightweight 4over6 lwB4 can use the same DHCPv6 option [[RFC6334](#)] to discover the FQDN of the lwAFTR.

When Lightweight 4over6 is deployed in the same place with DS-Lite, either different FQDNs can be configured for Lightweight 4over6 and DS-Lite separately or different DHCPv6 options can be used for Lightweight 4over6 [[I-D.sun-softwire-lw4over6-dhcpv6](#)] and DS-Lite. More detailed considerations on DS-Lite compatibility will be discussed in Section 6.

[3.4.](#) Impacts on Accounting

In Lightweight 4over6, the accounting impact due to the tunneling protocol is the same with DS-Lite (see [section 6.2 of \[RFC6908\]](#)). However, since in Lightweight 4over6, the IPv4 service is only available after port-set allocation, if operators will regard IPv4 service as a on-demand value-added service, e.g. IPv6 connectivity is offered by default, while IPv4 connectivity will be offered until a subscriber requires, etc., IPv4 service accounting should start after port-set allocation has completely.

[4.](#) lwAFTR Deployment Consideration

As Lightweight 4over6 is an extension to DS-Lite, both technologies share similar deployment considerations. For example: Interface consideration, Lawful Intercept Considerations, Blacklisting a shared IPv4 Address, AFTR's Policies, AFTR Impacts on Accounting Process, etc., in [[RFC6908](#)] can also be applied here. This document only discusses new considerations specific to Lightweight 4over6.

[4.1.](#) Logging at the lwAFTR

In Lightweight 4over6, operators only log one entry per subscriber. The log should include subscriber's IPv6 address used for the software, the public IPv4 address and the port-set. The port set algorithm implemented in Lightweight 4over6 lwAFTR should be synchronized with the one implemented in logging system. For example, if contiguous port set algorithm is adopted in the lwAFTR, the same algorithm should also be applied to the logging system.

Since the mapping in lwAFTR does not contain destination-specific information, operator should be aware that they will not be able to have destination-specific log.

[4.2.](#) MTU and Fragmentation Considerations

As Lightweight 4over6 is also a tunneling protocol, the same consideration regarding to the fragmentation and reassembly in DS-Lite [[RFC6908](#)] can also be applied. The only difference is that NAT functionality has been removed to lwB4 from lwAFTR in Lightweight 4over6. Therefore, on receiving an IPv4 fragmented packet after decapsulation in the lwB4, the lwB4 should further re-assemble the packets before doing NAT since the transport protocol information is only available in the first fragment.

[4.3.](#) Reliability Considerations of lwAFTR

Operators may deploy multiple lwAFTRs for robustness, reliability, and load balancing. In Lightweight 4over6, subscriber to IPv4 and port-set mapping must be pre-provisioned in the lwAFTR before providing IPv4 services. For redundancy, the backup lwAFTR must either have the subscriber mapping already provisioned or notify the lwB4 to create a new mapping in the backup lwAFTR. The first option can be considered as Hot Standby mode, which requires state synchronization between multiple lwAFTRs. In Hot Standby mode, the bindings are replicated on-the-fly from the Primary lwAFTR to the Backup lwAFTR. When the Primary lwAFTR fails, the Backup lwAFTR will take over all the existing established sessions. In this mode, the internal hosts are not required to re-initiate the bindings with the

external hosts. In Lightweight 4over6, since the number of mapping states has been greatly reduced compared to DS-Lite, it is reasonable to adopt Hot Standby mode when there are only two lwAFTRs (one for Primary lwAFTR and one for Backup lwAFTR). However, if the number of lwAFTRs is larger than two, it is not scalable to deploy Hot Standby mode since each two of the lwAFTRs should to synchronize the binding states.

The second option is to use Cold Standby mode which does not require a Backup Standby lwAFTR to synchronize binding states. In failover, the lwAFTR has to notify the lwB4 to create a new binding, or fetch the binding by itself. [[I-D.lee-software-lw4over6-failover](#)] describes these two approaches for simple Cold Standby mode. For most deployment scenarios, we believe that Cold Standby mode should be sufficient enough and is thus recommended.

[4.4.](#) Placement of AFTR

The lwAFTR can be deployed in a "centralized model" or a "distributed model".

In the "centralized model", the lwAFTR could be located at the higher place, e.g. at the exit of MAN, etc. Since the lwAFTR has good scalability and can handle numerous concurrent sessions, we recommend to adopt the "centralized model" for Lightweight 4over6 as it is cost-effective and easy to manage.

In the "distributed model", lwAFTR is usually integrated with the BRAS/SR. Since newly emerging customers might be distributed in the whole Metro area, we have to deploy lwAFTR on all BRAS/SRs. This will cost a lot in the initial phase of the IPv6 transition period.

[4.5.](#) Port set algorithm consideration

If each lwB4 is given a set of ports, port randomization algorithm can only select port in the given port-set. This may introduce security risk because hackers can make a more predictable guess of what port a subscriber may use. Therefore, non-continuous port set algorithms (e.g. as defined in [[I-D.ietf-software-map](#)]) can be used to improve security.

[4.6.](#) Path Consistency Consideration

In Lightweight 4over6, if the binding state is not synchronized among multiple lwAFTRs, the lwAFTR in which the subscriber's binding state is stored should be exactly the one to service the subscriber.

Otherwise, there will be no match in lwAFTR. This requires the provisionsion packets (either using DHCPv4-over-DHCPv6 or PCP Port-set)

should arrive at the same lwAFTR as the subsequent IP-in-IP traffic. If multiple lwAFTRs are using the same Tunnel End Point address and there are intermediate routers between lwB4 and lwAFTR, there might be a problem when intermediate routers perform ECMP based on L4 hash for the plain provisionsion packets while doing L3 hash for subsequent IP-in-IP traffic. In this case, it is recommended that the privioning packet is sent over IPv6 tunnel so that intermediate routers can only process ECMP using L3 hash.

[5.](#) lwB4 Deployment Consideration

For lwB4 consideration, the DNS Deployment Considerations and B4 Remote Management in [[RFC6908](#)] can also be applied here. In this section, we only describe the considerations sepcific to Lightweight 4over6.

[5.1.](#) NAT traversal issue

In Lightweight 4over6, since the subscriber's source port will be restricted to the port-set allocated from the provisioning system, this will have impact on some NAT traversal mechanisms. For example, in UPnP 1.0, the external port number which can be used by remote peer is selected by UPnP client in end host. If the client randomly selects a port number which is not in that valid port-set, the UPnP process will fail. This is likely to happen because end-host does not know the port-set in lwB4. More detailed experimental results can be found in [[I-D.deng-aplusp-experiment-results](#)]. This problem will not exist in UPnP 2.0 because the UPnP client in the end-host will negotiate the external port number with the server. Another way is to implement a mechanism (e.g. [[I-D.ietf-pcp-port-set](#)], etc.) in end host to fetch the port-set from lwB4. The UPnP client can then select the port number within the port-set.

[5.2.](#) Static Port Forwarding Configuration

Currently, some external initiated applications rely on manual port configuration to reserve a port in the CPE. The restricted port-set in lwB4 will also have impacts on manual port forwarding configuration. It is recommended that the port-set allocated from the provioning system should be shown explicitly in the lwB4, which can be used as a hint for subscribers to add port forwarding mapping.

6. DS-Lite Compatibility Consideration

Lightweight 4over6 can be either deployed all alone, or combined with DS-Lite [[RFC6333](#)]. Since Lightweight 4over6 does not any have extra requirement on IPv6 addressing, it can use use the same addressing scheme with DS-Lite, together with routing policy, user management policy, etc. Besides, the bottom-up model has quite similar requirement and workflow on the supporting system with DS-Lite. Therefore, it is suitable for operators to deploy incrementally in existing DS-Lite network

6.1. Case 1: Integrated Network Element with Lightweight 4over6 and DS-Lite AFTR Scenario

In this case, DS-Lite has been deployed in the network. Later in the deployment schedule, the operator decided to implement Lightweight 4over6 lwAFTR function in the same network element(depicted in Figure3). Therefore, the same network element needs to support both transition mechanisms.

There are two options to distinguish the traffic from two transition mechanisms.

The first one is to distinguish using the client's source IPv4 address. The IPv4 address from Lightweight 4over6 is public address

as NAT has been done in the lwB4, and IPv4 address for DS-lite is private address as NAT will be done on AFTR. When the network element receives an encapsulated packet, it would de-capsulate packet and apply the transition mechanism based on the IPv4 source address in the packet. This requires the network element to examine every packet and may introduce significant extra load to the network element. However, both the B4 element and Lightweight 4over6 lwB4 can use the same DHCPv6 option [[RFC6334](#)] with the same FQDN of the AFTR and lwAFTR.

The second one is to distinguish using the destination's tunnel IPv6 address. One network element can run separated instances for Lightweight 4over6 and DS-Lite with different tunnel addresses. Then B4 element and Lightweight 4over6 lwB4 can use the same DHCPv6 option [[RFC6334](#)] with different FQDNs pointing to corresponding tunnel addresses. This requires the supporting system should distinguish different types of users when assigning the FQDNs in DHCPv6 process. Another option is to use a new DHCPv6 option [[I-D.sun-softwire-lw4over6-dhcpv6](#)] to discover lwAFTR's FQDN.

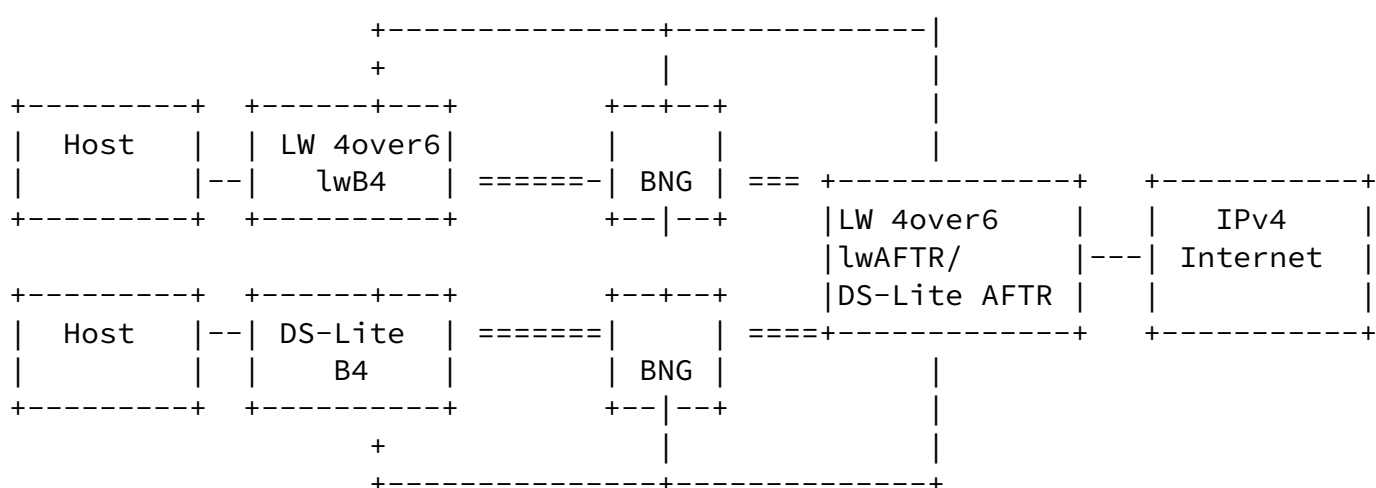


Figure 3 DS-Lite Coexistence scenario with Integrated AFTR

6.2. Case 2: DS-Lite Coexistent scenario with Separated AFTR

This is similar to Case 1. The difference is the lwAFTR and AFTR functions won't be co-located in the same network element (depicted in Figure4). This use case decouples the functions to allow more flexible deployment. For example, an operator may deploy AFTR closer to the edge and lwAFTR closer to the core. Moreover, it does not require the network element to pre-configure with the CPE's IPv6 addresses. An operator can deploy more AFTR and lwAFTR at needed. However, this requires the B4 and lwB4 to discover the corresponding network element. In this case, B4 element and Lightweight 4over6 lwB4 can still use [\[RFC6334\]](#) with different FQDNs pointing to corresponding tunnel end-point addresses, and the supporting system should distinguish different types of users.

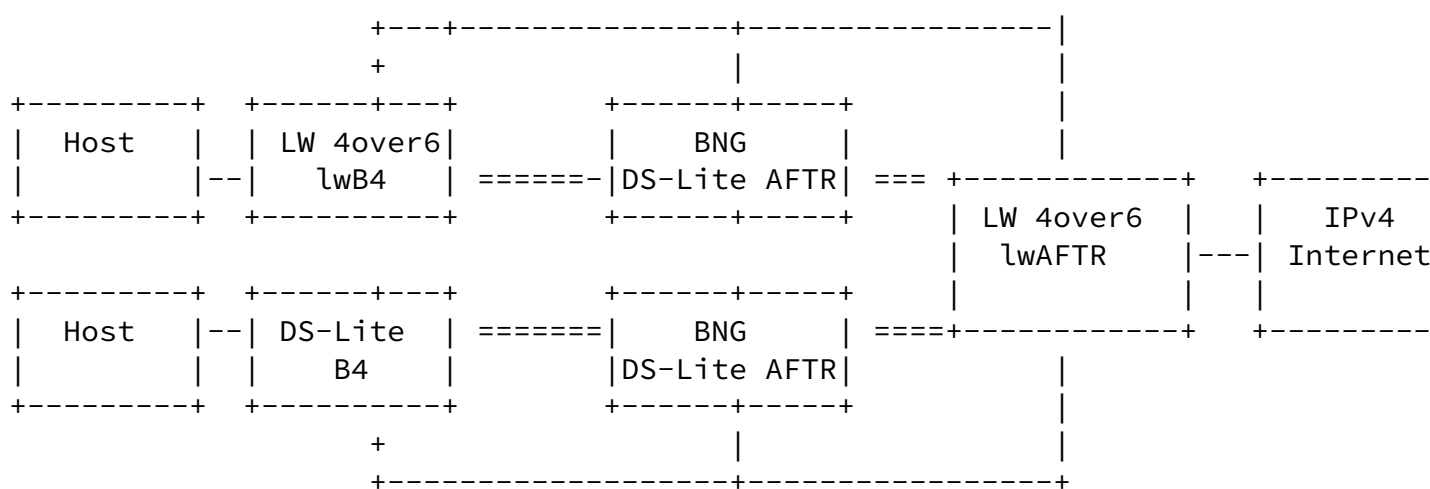


Figure 4 DS-Lite Coexistence scenario with Separated AFTR

7. Acknowledgement

TBD

[8.](#) References

[I-D.bajko-pripaddrassign]

Bajko, G., Savolainen, T., Boucadair, M., and P. Levis,

"Port Restricted IP Address Assignment",
[draft-bajko-pripaddrassign-04](#) (work in progress),
April 2012.

[I-D.cui-software-b4-translated-ds-lite]

Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", [draft-cui-software-b4-translated-ds-lite-11](#) (work in progress), February 2013.

[I-D.deng-aplusp-experiment-results]

Deng, X., Boucadair, M., and F. Telecom, "Implementing A+P in the provider's IPv6-only network", [draft-deng-aplusp-experiment-results-00](#) (work in progress), March 2011.

[I-D.ietf-behave-ipfix-nat-logging]

Sivakumar, S. and R. Penno, "IPFIX Information Elements for logging NAT Events", [draft-ietf-behave-ipfix-nat-logging-13](#) (work in progress), January 2017.

[I-D.ietf-behave-syslog-nat-logging]

Chen, Z., Zhou, C., Tsou, T., and T. Taylor, "Syslog Format for NAT Logging", [draft-ietf-behave-syslog-nat-logging-06](#) (work in progress), January 2014.

[I-D.ietf-dhc-dhcpv4-over-ipv6]

Cui, Y., Wu, P., Wu, J., Lemon, T., and Q. Sun, "DHCPv4 over IPv6 Transport", [draft-ietf-dhc-dhcpv4-over-ipv6-09](#) (work in progress), April 2014.

[I-D.ietf-pcp-base]

Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", [draft-ietf-pcp-base-29](#) (work in progress), November 2012.

[I-D.ietf-pcp-port-set]

Qiong, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port Set Allocation", [draft-ietf-pcp-port-set-13](#) (work in progress), October 2015.

[I-D.ietf-softwire-4rd]

Despres, R., Jiang, S., Penno, R., Lee, Y., Chen, G., and M. Chen, "IPv4 Residual Deployment via IPv6 - a Stateless Solution (4rd)", [draft-ietf-softwire-4rd-10](#) (work in progress), December 2014.

[I-D.ietf-softwire-dslite-deployment]

Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", [draft-ietf-softwire-dslite-deployment-08](#) (work in progress), January 2013.

[I-D.ietf-softwire-lw4over6]

Cui, Y., Qiong, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the DS-Lite Architecture", [draft-ietf-softwire-lw4over6-13](#) (work in progress), November 2014.

[I-D.ietf-softwire-map]

Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, "Mapping of Address and Port with Encapsulation (MAP)", [draft-ietf-softwire-map-13](#) (work in progress), March 2015.

[I-D.lee-softwire-lw4over6-failover]

Lee, Y., Qiong, Q., and C. Liu, "Simple Failover Mechanism for Lightweight 4over6", [draft-lee-softwire-lw4over6-failover-01](#) (work in progress), July 2013.

[I-D.sun-softwire-lw4over6-dhcpv6]

Xie, C., Qiong, Q., Lee, Y., Tsou, T., and P. Wu, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Lightweight 4over6", [draft-sun-softwire-lw4over6-dhcpv6-00](#) (work in progress), July 2013.

[I-D.zhou-dime-4over6-provisioning]

Zhou, C., Taylor, T., Qiong, Q., and M. Boucadair, "Attribute-Value Pairs For Provisioning Customer Equipment Supporting IPv4-Over-IPv6 Transitional Solutions", [draft-zhou-dime-4over6-provisioning-05](#) (work in progress), September 2014.

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

Internet-Draft

lightweigh-4over6-deployment

January 2017

-
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", [RFC 6052](#), DOI 10.17487/RFC6052, October 2010, <<http://www.rfc-editor.org/info/rfc6052>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", [RFC 6333](#), DOI 10.17487/RFC6333, August 2011, <<http://www.rfc-editor.org/info/rfc6333>>.
- [RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", [RFC 6334](#), DOI 10.17487/RFC6334, August 2011, <<http://www.rfc-editor.org/info/rfc6334>>.
- [RFC6431] Boucadair, M., Levis, P., Bajko, G., Savolainen, T., and T. Tsou, "Huawei Port Range Configuration Options for PPP IP Control Protocol (IPCP)", [RFC 6431](#), DOI 10.17487/[RFC6431](#), November 2011, <<http://www.rfc-editor.org/info/rfc6431>>.
- [RFC6908] Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", [RFC 6908](#), DOI 10.17487/RFC6908, March 2013, <<http://www.rfc-editor.org/info/rfc6908>>.
- [RFC7341] Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4-over-DHCPv6 (DHCP 4o6) Transport", [RFC 7341](#), DOI 10.17487/RFC7341, August 2014, <<http://www.rfc-editor.org/info/rfc7341>>.
- [RFC7618] Cui, Y., Sun, Q., Farrer, I., Lee, Y., Sun, Q., and M. Boucadair, "Dynamic Allocation of Shared IPv4 Addresses", [RFC 7618](#), DOI 10.17487/RFC7618, August 2015, <<http://www.rfc-editor.org/info/rfc7618>>.

Internet-Draft

lightweigh-4over6-deployment

January 2017

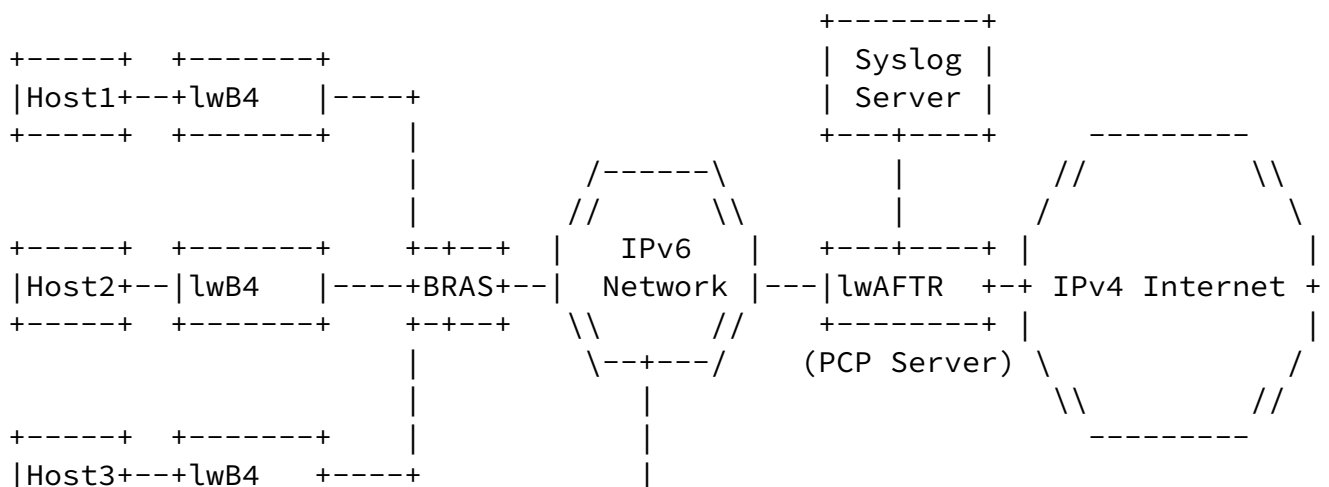
[1.](#) China Telecom Experimental Result

We have deployed Lightweight 4over6 in our operational network of HuNan province, China. It is designed for broadband access network, and different versions of lwB4 have been implemented including a linksys box, a software client for windows XP, vista and Windows 7. It can be integrated with existing dial-up mechanisms such as PPPoE, etc. The major objectives listed below aimed to verify the functionality and performance of Lightweight 4over6:

- o Verify how to deploy Lightweight 4over6 in a practical network.
- o Verify the impact of applications with Lightweight 4over6.
- o Verify the performance of Lightweight 4over6.

[1.1.](#) Experimental Environment

The network topology for this experiment is depicted in Figure 5.



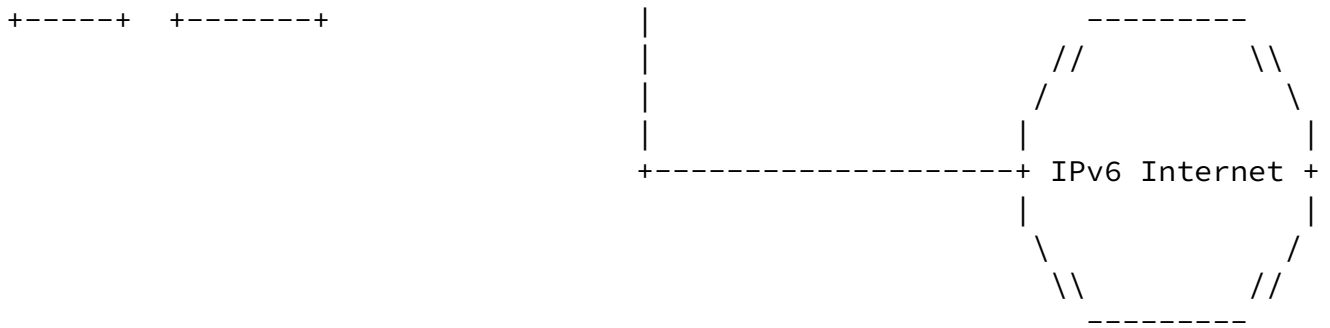


Figure 5 China Telecom Lightweight 4over6 experiment topology

In this deployment model, lwAFTR is co-located with a extended PCP server to assign restricted IPv4 address and port set for lwB4. It also triggers subscriber-based logging event to a centrilized syslog

server. IPv6 address pools for subscribers have been distributed to BRASs for configuration, while the public available IPv4 address pools are configured by the centralized lwAFTR with a default address sharing ratio. It is rather flexible for IPv6 addressing and routing, and there is little impact on existing IPv6 architecture.

In our experiment, lwB4 will firstly get its IPv6 address and delegated prefix through PPPoE, and then initiate a PCP-extended request to get public IPv4 address and its valid port set. The lwAFTR will thus create a subscriber-based state accordingly, and notify syslog server with {IPv6 address, IPv4 address, port set, timestamp}.

1.2. Experimental Results

In our trial, we mainly focused on application test and performance test. The applications have widely include web, email, Instant Message, ftp, telnet, SSH, video, Video Camera, P2P, online game, voip and so on. For performance test, we have measured the parameters of concurrent session numbers and throughput performance.

The experimental results are listed as follows:

Application Type	Test Result	Port Number Occupation
------------------	-------------	------------------------

Web	ok IE, Firefox, Chrome	normal websites: 10~20 Ajax Flash webs: 30~40
Video	ok, web based or client based	30~40
Instant Message	ok QQ, MSN, gtalk, skype	8~20
P2P	ok utorrent,emule,xunlei	lower speed: 20~600 (per seed) higher speed: 150~300
FTP	need ALG for active mode, flashxp	2
SSH, TELNET	ok	1 for SSH, 3 for telnet
online game	ok for QQ, flash game	20~40

Figure 6 China Telecom 4over6 experimental result

The performance test for lwAFTR is taken on a normal PC. Due to limitations of the PC hardware, the overall throughput is limited to around 800 Mbps. However, it can still support more than one hundred million concurrent sessions.

[1.3.](#) Conclusions

From the experiment, we can have the following conclusions:

- o Lightweight 4over6 has good scalability. As it is a lightweight solution which only maintains per-subscription state information, it can easily support a large amount of concurrent subscribers.
- o Lightweight 4over6 can be deployed rapidly. There is no modification to existing addressing and routing system in our operational network. And it is simple to achieve traffic logging.
- o Lightweight 4over6 can support a majority of current IPv4 applications.

[2.](#) Tsinghua University Experimental Result

Lightweight 4over6 has also been deployed in the campus of Tsinghua University, China. We use DHCPv4 over DHCPv6 [[RFC7341](#)] for the dynamic provisioning of lwB4's IPv4 address and port set [[RFC7618](#)]. We deployed wireless APs for Lightweight 4over6, covering a large portion of the campus, allowing mobile devices to connect to the lwB4. We also deployed lwB4 gateway in some of our buildings so PCs could connect directly to the lwB4. Users could access the IPv4 Internet through the CNGI IPv6 Network.

[2.1.](#) Experimental Environment

The network topology for this experiment is depicted in Figure 7.

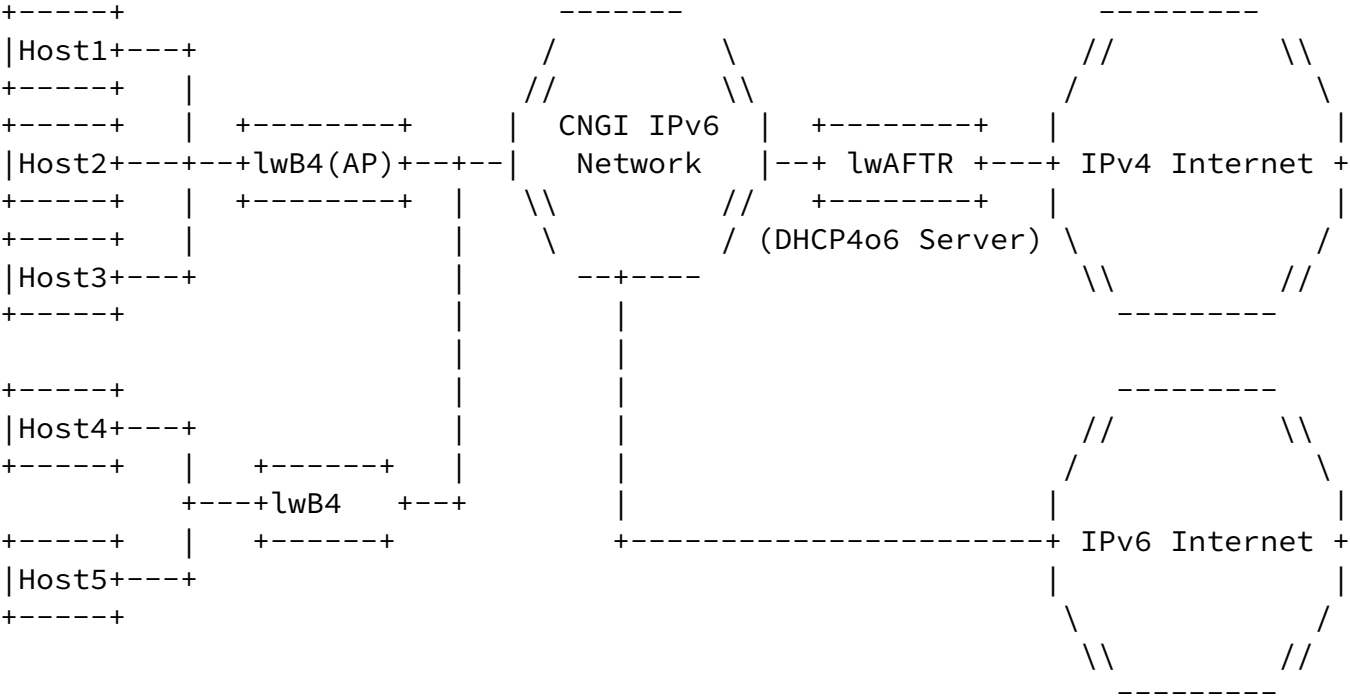


Figure 7 Tsinghua University Lightweight 4over6 experiment topology

In this deployment model, lwAFTR is co-located with a DHCP4o6 server to assign restricted IPv4 address and port set for lwB4. lwAFTR listens to all DHCPv4 over DHCPv6 messages generated or received by the DHCP4o6 server and updates its binding table through valid messages.

In our experiment, the lwB4 gets its IPv6 address through static configuration or dynamic configuration. It will then send a request to the lwAFTR device to get the public IPv4 address and its valid port set. The lwAFTR will add the IPv6 address, IPv4 address, and

port set information of the lwB4 into its binding table.

2.2. Experimental Results

In our experiment, we tested the performance of various applications,

including web, video, p2p, ping, tracer, telnet, SSH, email. online storage, instant message, online gaming, online payment and so on. We also tested different terminal devices including PC, laptop computer, and cell phone. These include devices using different operating systems, including Windows 7, MacOS, Android, and IOS.

The experimental results are listed as follows:

Application Type	Test Applications	Test Subjects	Result
Web	IE, Chrome, Sougou	Browse websites, download files	OK
Video	Youku, pptv, qqlive (Web based,client based)	VOD, live video	OK
P2P	Bittorrent, xunlei	Download files	OK
Ping/tracer	Command line	Ping/tracer URL	OK
TELNET/SSH	Putty, secureCRT	Telnet/SSH login	OK
Email	126, QQ, hotmail (Web based,client based)	Send/receive email	OK
Cloud storage	Baidu Cloud	Upload/download files	OK
Instant messaging	Skype, QQ	Send/receive messages	OK
Online gaming	QQ game	Enter game	OK
Online payment	JD, Taobao	Complete payment	OK

Figure 8 Tsinghua University Lightweight 4over6 experimental result

2.3. Conclusions

Lightweight 4over6 supports the majority of current IPv4 applications and services. The user experience of using Lightweight 4over6 is no different from using the native IPv4 network. It can satisfy the IPv4 network service demands of IPv6 network users.

Authors' Addresses

Qiong Sun
China Telecom
Room 708, No.118, Xizhimennei Street
Beijing 100035
P.R.China

Phone: +86-10-58552936>
Email: sunqiong@ctbri.com.cn

Chongfeng Xie
China Telecom
Room 708, No.118, Xizhimennei Street
Beijing 100035
P.R.China

Phone: +86-10-58552116>
Email: xiechf@ctbri.com.cn

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: yiul_lee@cable.comcast.com

Maoke Chen
FreeBit Co., Ltd.
13F E-space Tower, Maruyama-cho 3-6
Shibuya-ku, Tokyo 150-0044
Japan

Email: fibrib@gmail.com

Tianxiang Li
Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: peter416733@gmail.com

