APAN Network Research Workshop 2010 at the 30th APAN Meeting
August 9, 2010 Hanoi, Vietnam

i-Path Project

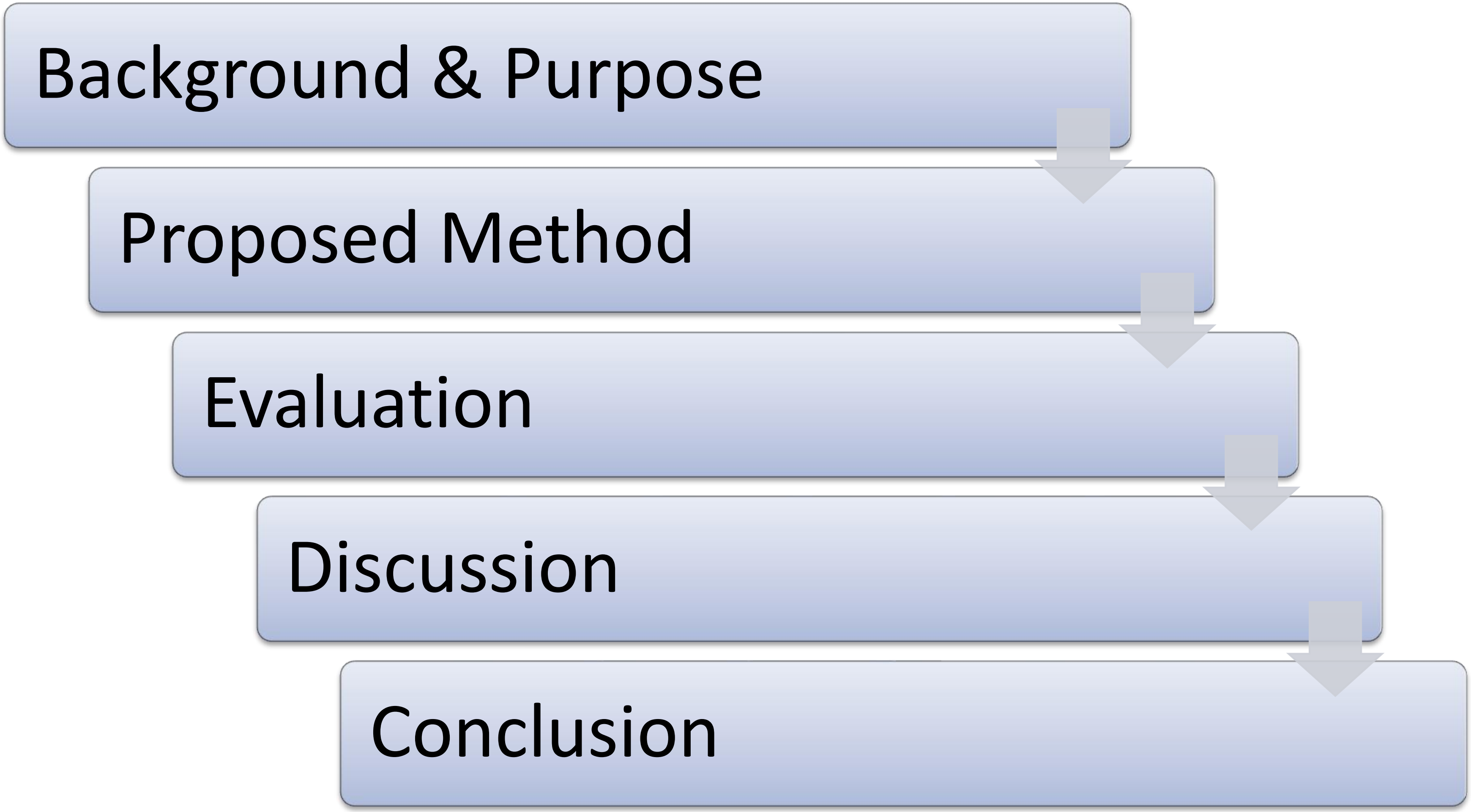
Extended UDP Multiple Hole Punching Method to Traverse Large Scale NATs

Kazuhiro Tobe, Akihiro Shimoda, and Shigeki Goto

Waseda University

{tobe, shimo, goto}@goto.info.waseda.ac.jp

Outline

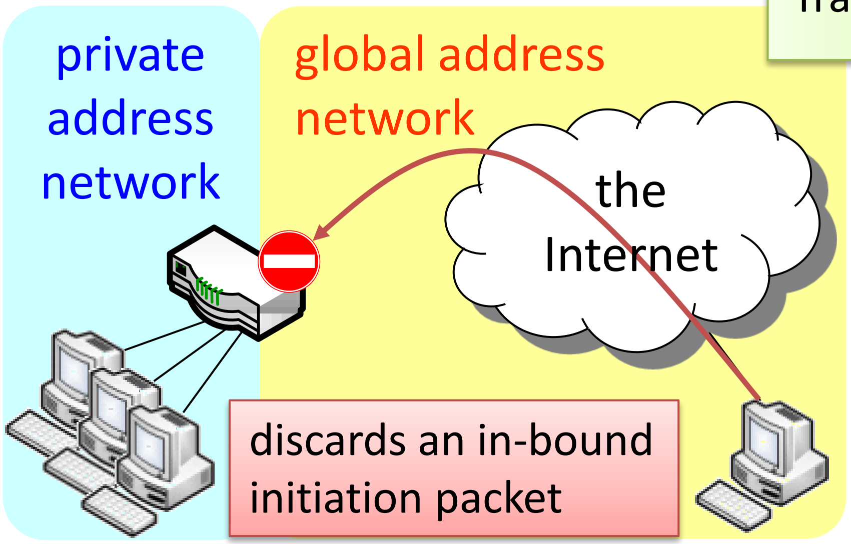


- Background & Purpose
- Proposed Method
- Evaluation
- Discussion
- Conclusion

Background (1) Problem in Network Address Translator (NAT)

The Problem in NATs

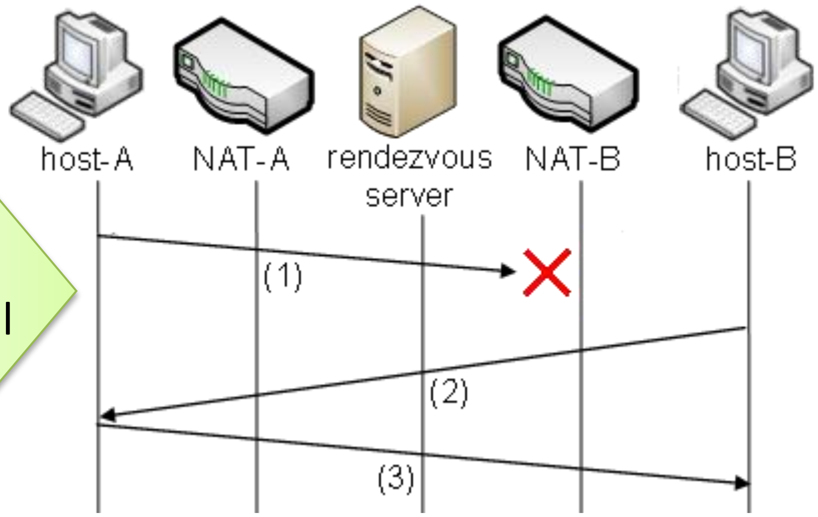
- A network behind a NAT cannot be accessed from external hosts
  - Peer-to-Peer (P2P) apps does not work on a host behind a NAT
  - e.g. VoIP apps, Online games



discards an in-bound initiation packet

NAT Traversal

UDP Hole Punching



End hosts can directly communicate each other beyond NATs by using the UDP Hole Punching. However, they can traverse the only Cone NAT [RFC3489] i.e., cannot traverse a Symmetric NAT.

Ford, B.; Srisuresh, P.; Kegel, D. Peer-to-Peer Communication Across Network Address Translators. USENIX Annual Technical Conference, pp.179-192, 2005.

Our Previous Method

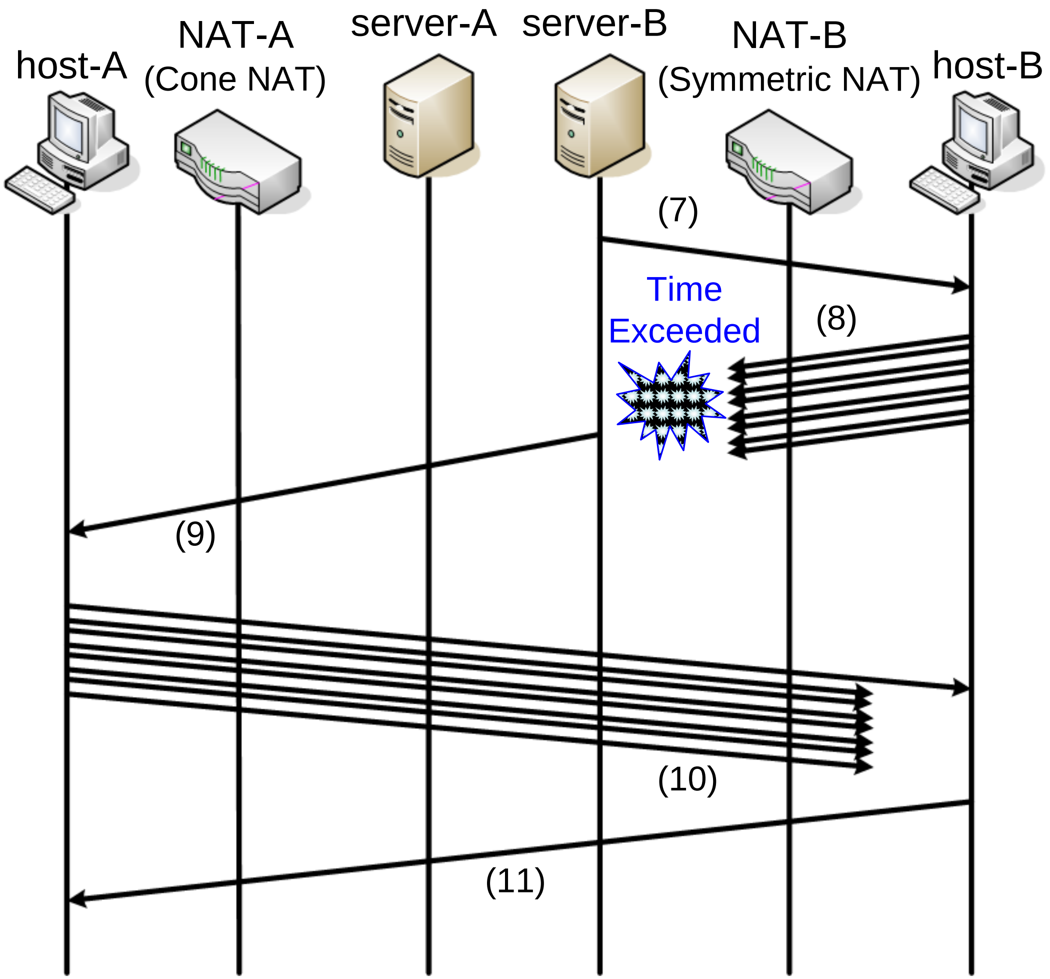
UDP Multiple Hole Punching

Extends the concept of the UDP Hole Punching

- Conducts *Port Prediction* helped by two servers
- Sends numerous UDP packets with low TTL values

Can traverse a Symmetric NAT without relay servers

- Low loads and low-delay
- cf. TURN, ICE, Teredo



Wei, Y.; Yamada, D.; Yoshida, S.; Goto, S. A New Method for Symmetric NAT Traversal in UDP and TCP. APAN Network Research Workshop 2008, pp.11-18, August 2008.

Background (2)

Large Scale NAT/Carrier Grade NAT

[Huston] G. Huston, "IPv4 Address Report", http://www.potaroo.net/tools/ipv4/index.html

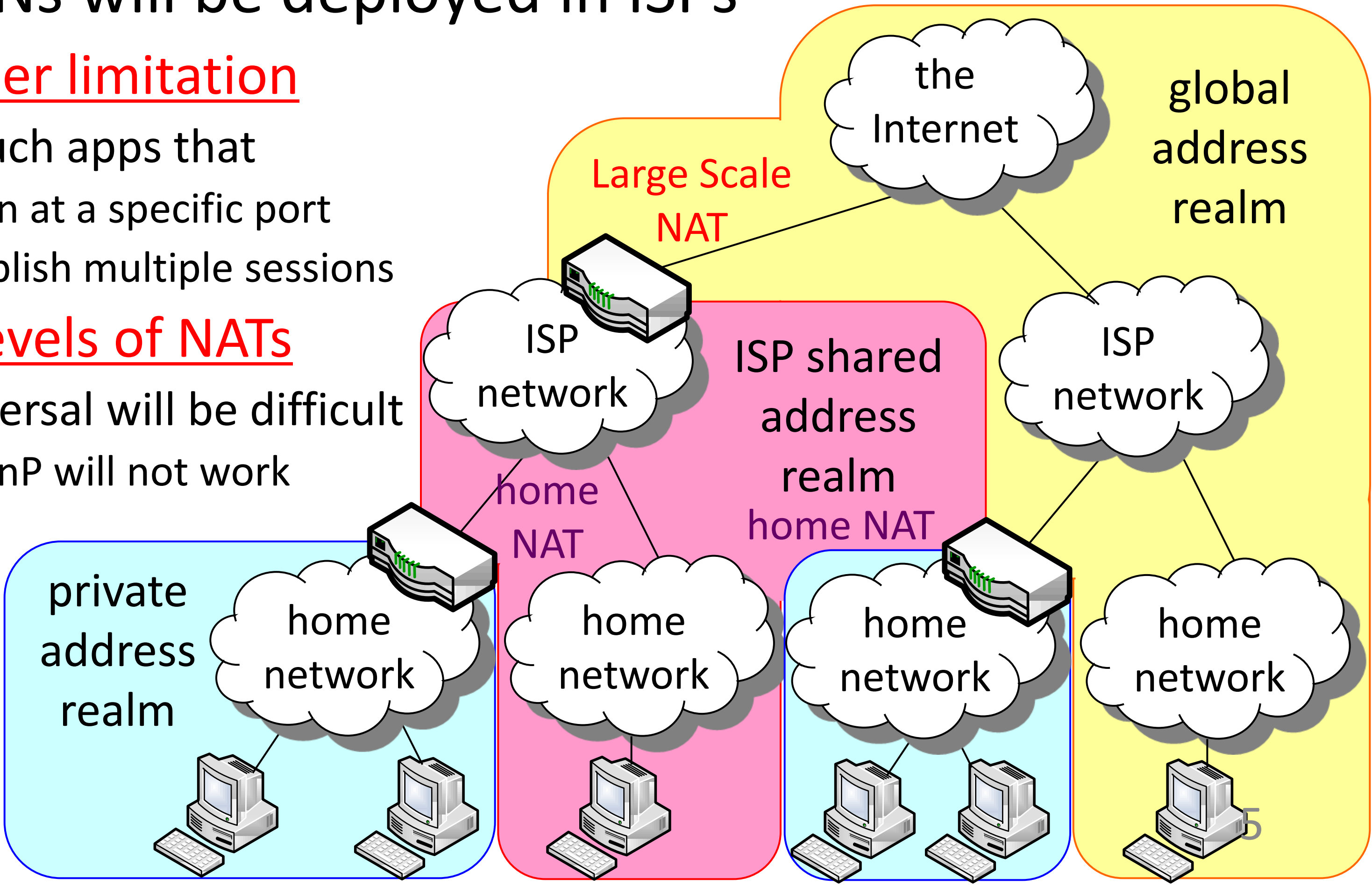
- IPv4 address exhaustion will occur (IANA:2011, RIR:2012) [Huston]

=> LSNs or CGNs will be deployed in ISPs

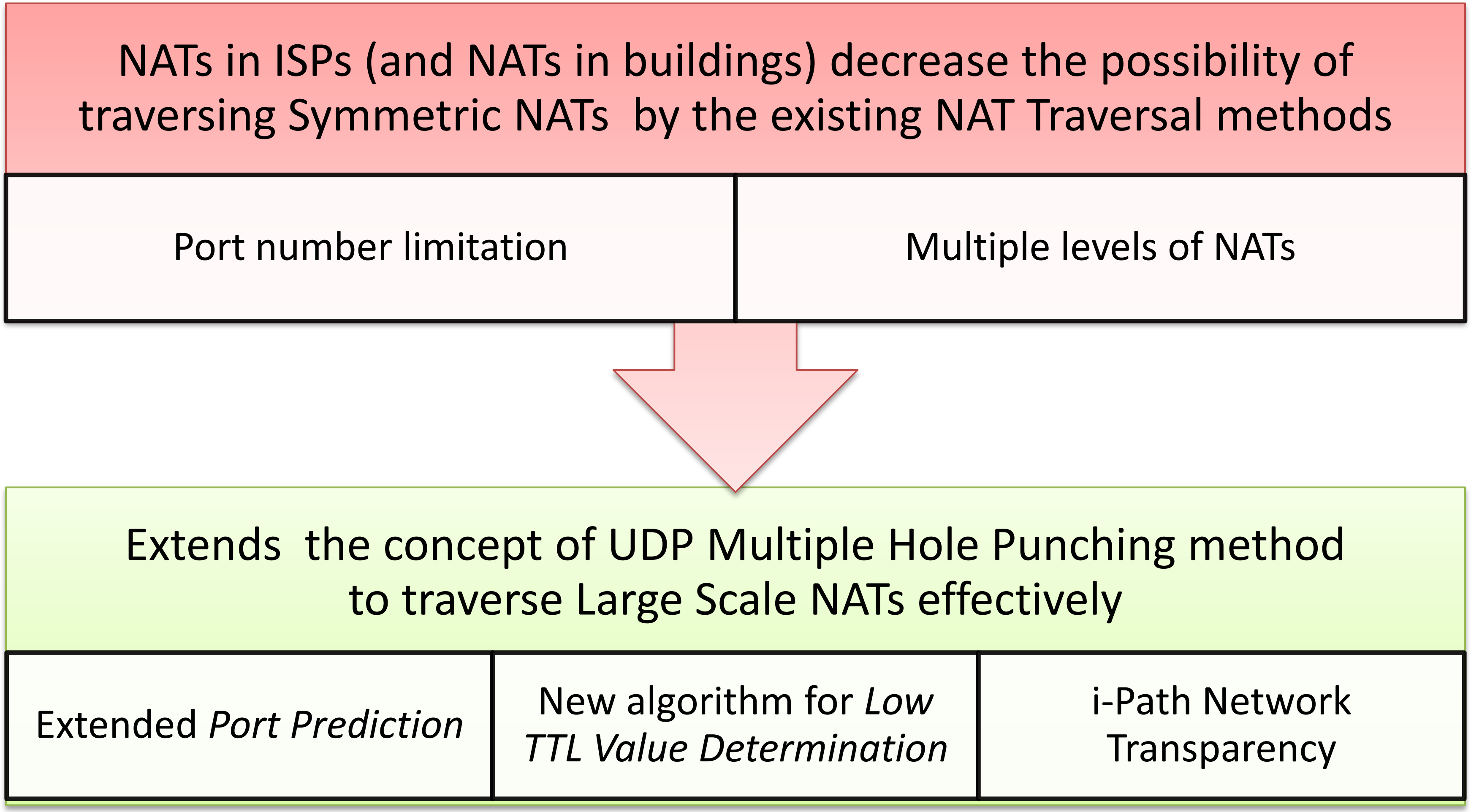
- Port number limitation
  - Blocks such apps that
    - Listen at a specific port
    - Establish multiple sessions
- Multiple levels of NATs
  - NAT Traversal will be difficult

    e.g. UPnP will not work

UDP Multiple Hole Punching needs improving



Summary of Background & Purpose



NATs in ISPs (and NATs in buildings) decrease the possibility of traversing Symmetric NATs by the existing NAT Traversal methods

Port number limitation

Multiple levels of NATs

Extends the concept of UDP Multiple Hole Punching method to traverse Large Scale NATs effectively

Extended *Port Prediction*

New algorithm for *Low TTL Value Determination*

i-Path Network Transparency

Proposed Method

Extended Port Prediction

- Capturing Method
- Scanning Method

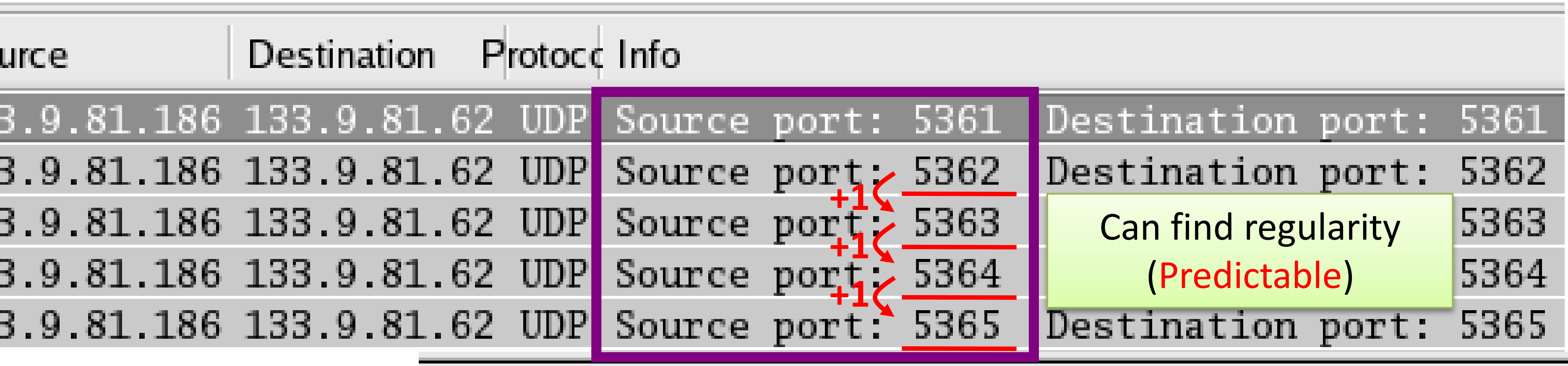
New Algorithm
for Low TTL Value Determination

i-Path Network Transparency

A Part of Our Previous Method (1)

Port Prediction

- A technique to examine ports assigned by a NAT and to predict the next assigned port
  - Success => A host can traverse a Symmetric NAT by using Hole Punching



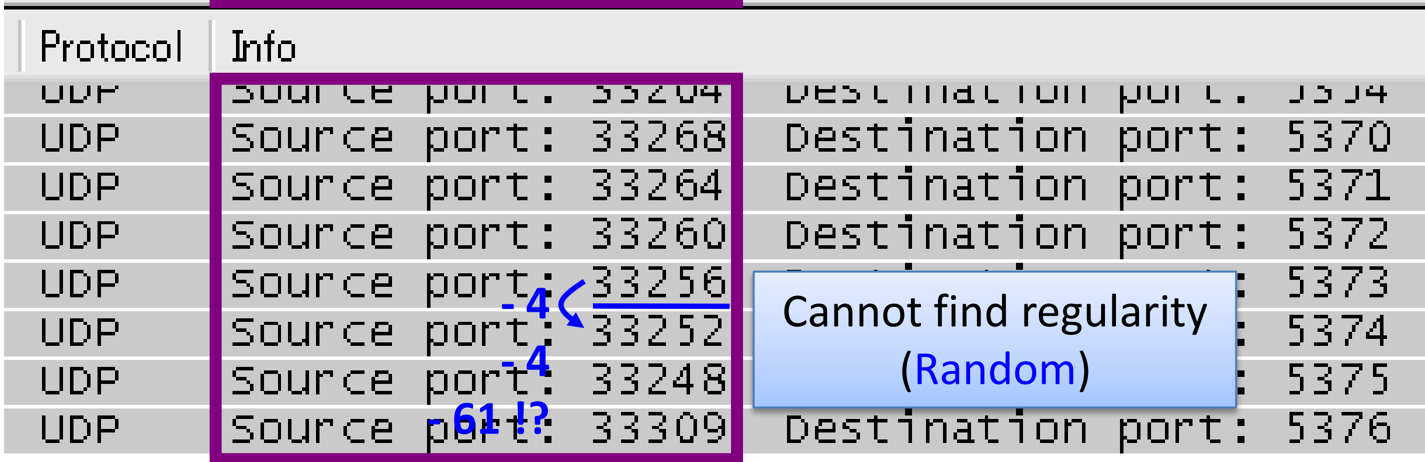
urce	Destination	Protocc	Info	
3.9.81.186	133.9.81.62	UDP	Source port: 5361	Destination port: 5361
3.9.81.186	133.9.81.62	UDP	Source port: 5362	Destination port: 5362
3.9.81.186	133.9.81.62	UDP	Source port: 5363	5363
3.9.81.186	133.9.81.62	UDP	Source port: 5364	5364
3.9.81.186	133.9.81.62	UDP	Source port: 5365	Destination port: 5365

+1

+1

+1

Can find regularity
(Predictable)



Protocol	Info	
UDP	Source port: [illegible]	Destination port: [illegible]
UDP	Source port: 33268	Destination port: 5370
UDP	Source port: 33264	Destination port: 5371
UDP	Source port: 33260	Destination port: 5372
UDP	Source port: 33256	: 5373
UDP	Source port: 33252	: 5374
UDP	Source port: 33248	: 5375
UDP	Source port: 33309	Destination port: 5376

-4

-4

-61 !?

Cannot find regularity
(Random)

Case: Random

A UDP Multiple Punching host send numerous packets (the last resort)

Problem (1) in Port Prediction

Problem (1) A possibility that a UDP Multiple Hole Punching host fails in *Port Prediction*

– Symmetric NATs may assign new port numbers for other hosts during Port Prediction

=> Estimated to be random,
while it is really a predictable algorithm

A host may open more ports than necessary
⇒ wastes port numbers of Large Scale NATs

Proposed Method (1)

Extended Port Prediction

Capturing Method

- captures packets in the network behind NATs
- counts the number of initiation packets of UDP sessions during *Port Prediction*

Scanning Method

- counts the number of running hosts (*N*) in the network before *Port Prediction*
- estimates the potential error ([0, *E*])
  - $E = w * N$, where *w* refers to a weight

Proposed Method

Extended Port Prediction

- Capturing Method
- Scanning Method

New Algorithm
for Low TTL Value Determination

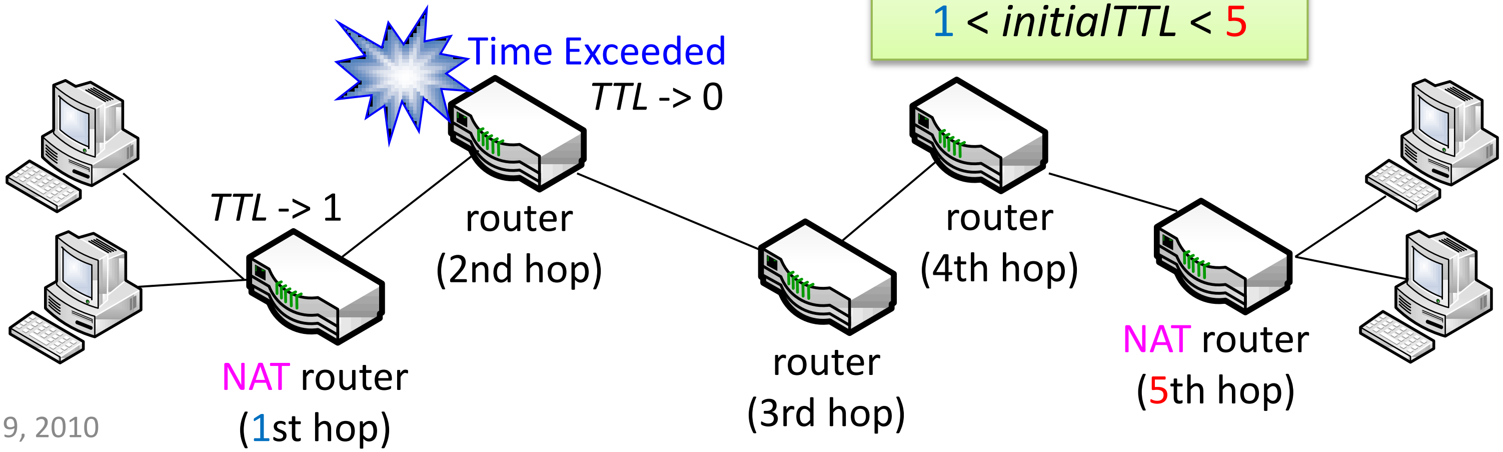
i-Path Network Transparency

A Part of Our Previous Method (2)

Low TTL Value Determination

- End hosts send UDP packets whose TTL is set so low that the packets are dropped between the NAT on the sender side and the NAT on the destination side
  - Existing algorithms for Low TTL Value Determination
    - UDP Multiple Hole Punching: manual (by the experimenter)
    - NATBLASTER: Traceroute (only proposed, not implemented)
  - case *initialTTL* == 2

InitialTTL must be …
1 < *initialTTL* < 5

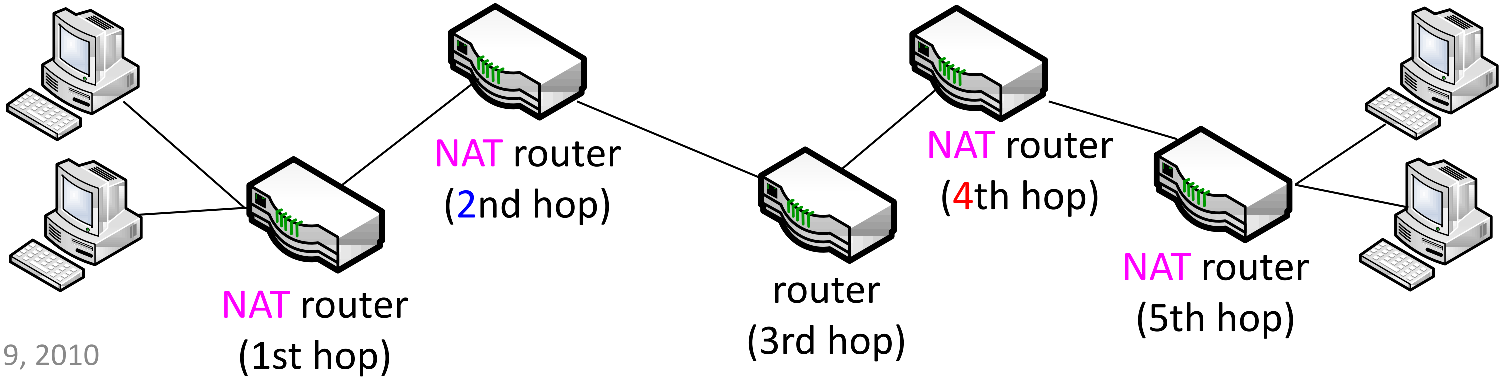


Problem (2) in Port Prediction

- NATs are cascaded
  => The packets must be discarded between the NAT on the sender side and the NAT on the destination side
  ∴ $2 < initialTTL < 4$

Problem (2) impossible to know how many NATs are cascaded

  - Traceroute/Tracert provides end hosts with routers' IP addresses
  - It can be a hint but some routers do not return ICMP messages



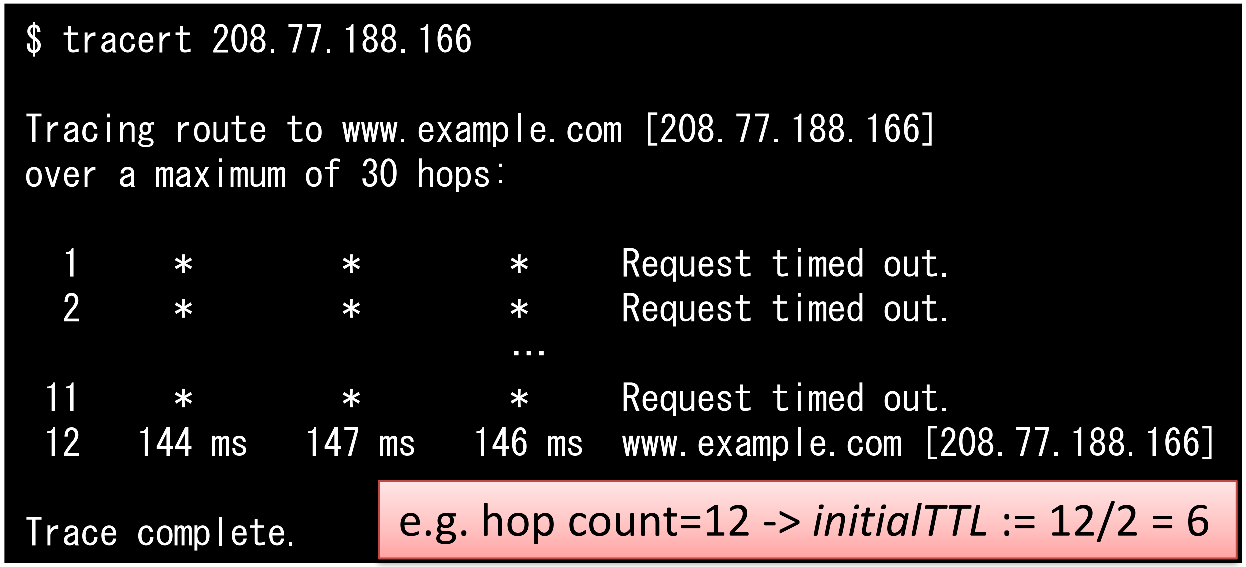
Proposed Method (2)

Low TTL Value Determination

Solution (2) sets the initial TTL value to half of the end-to-end hop count

[assumption] NATs are concentrated close to end hosts and do not exist in the center part of a network

– Requires only the hop count to the destination



```
$ tracert 208.77.188.166

Tracing route to www.example.com [208.77.188.166]
over a maximum of 30 hops:

  1     *        *        *     Request timed out.
  2     *        *        *     Request timed out.
                          …
 11     *        *        *     Request timed out.
 12   144 ms   147 ms   146 ms  www.example.com [208.77.188.166]

Trace complete.
```

e.g. hop count=12 -> *initialTTL* := 12/2 = 6

Proposed Method

Extended Port Prediction

- Capturing Method
- Scanning Method

New Algorithm
for Low TTL Value Determination

i-Path Network Transparency

Our Previous Method

i-Path Routers

Provide end hosts
with the network status info along a path

- In addition to info traceroute/tracert provides, …
- e.g., geographical location, traffic volume, etc

Observe the info disclosure policy of routers

- Disclose only the info all of the stakeholders* allow to disclose

*stakeholders = e.g. the sender/receiver and ISPs

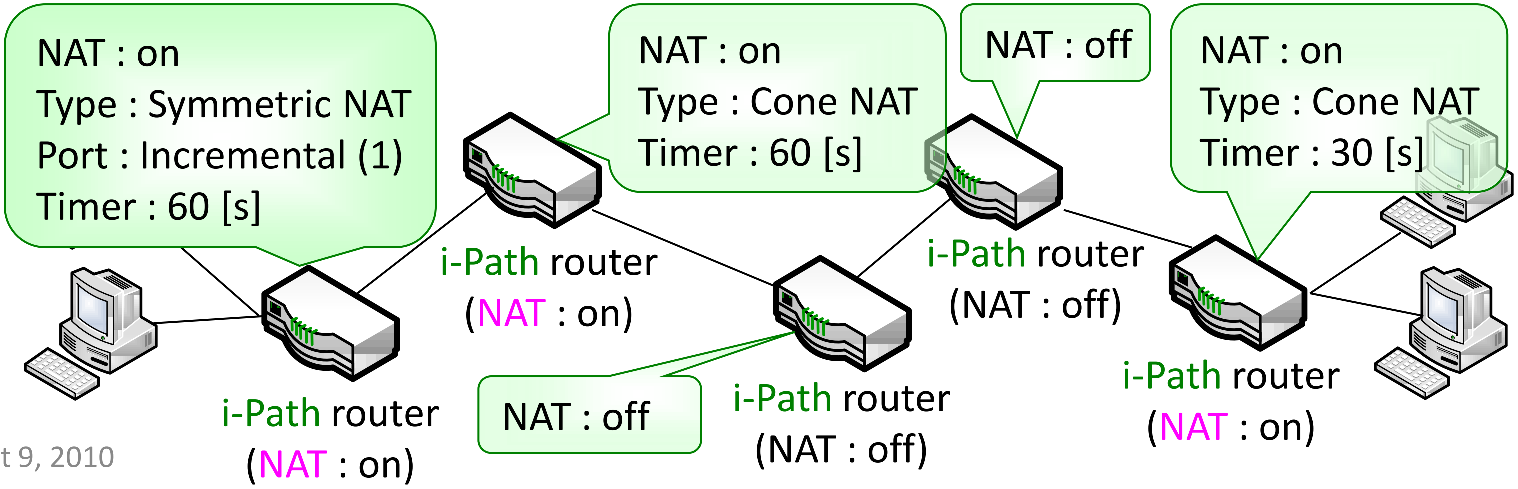
Kobayashi, K.; Goto, S.; Murase, I.; Mochinaga, D. Design for an End-to-end Cross-layer Measurement Protocol and its API toward Network Visibility Respecting Disclosure Policies. IEICE Technical Report Internet Architecture 108(409), pp.11-16, January 2009.

Proposed Method (3)

NAT Traversal by i-Path Network Transparency

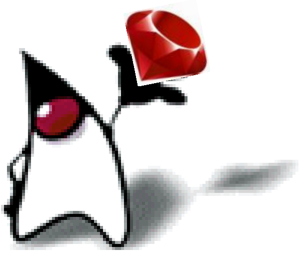
- i-Path routers disclose NAT information
  - NAT on/off: helps *Low TTL Value Determination*
  - NAT property: improves the *Port Prediction* accuracy
- End Hosts can obtain all the routers along a path
  - Can Work in networks behind Multiple levels of NATs

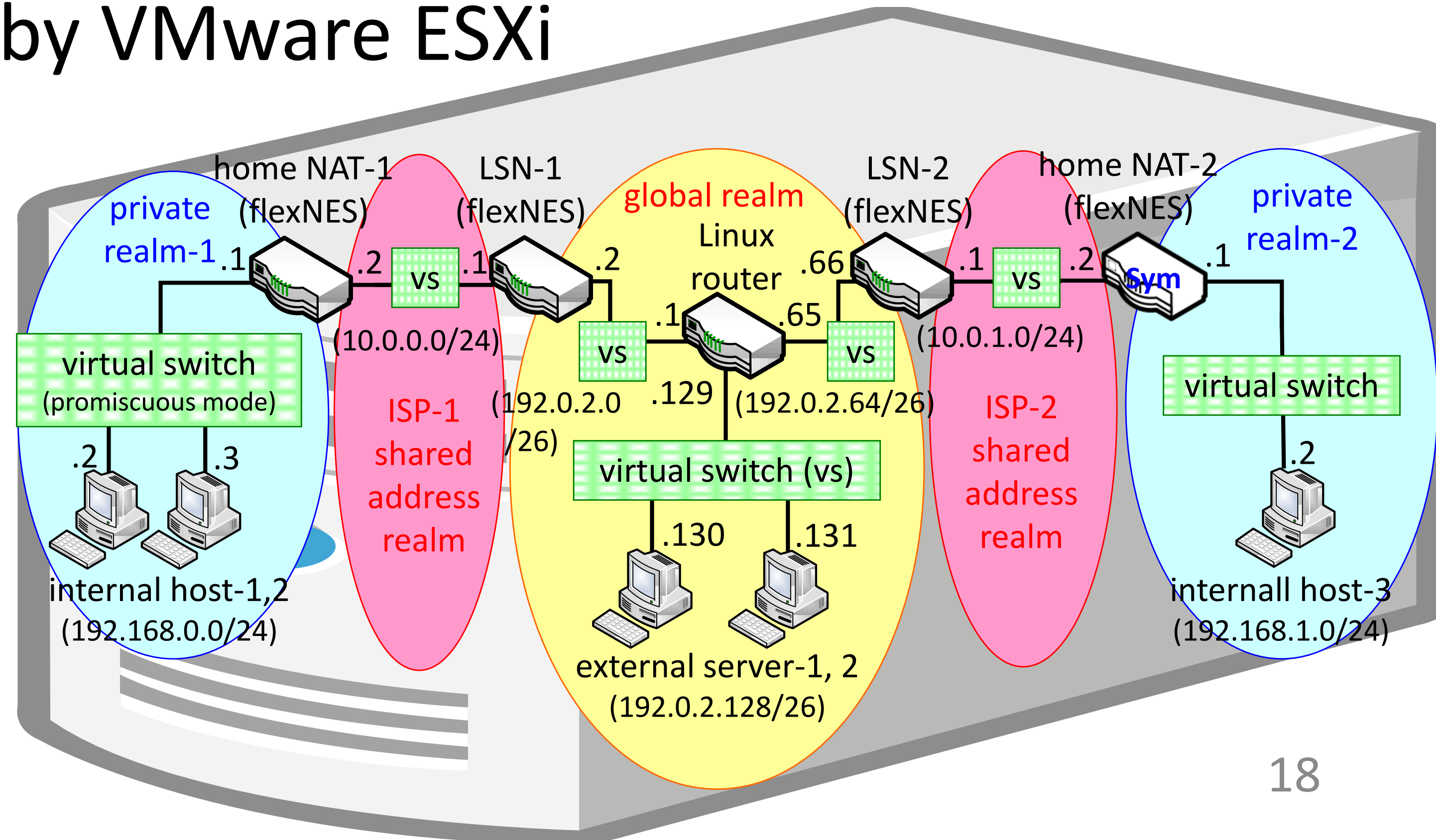
cf. UPnP (Universal Plug and Play)



Evaluation

- Several Java programs
  - – Invoke a Ruby program
- Testbed by VMware ESXi





Discussion

- Capturing Method vs. Scanning Method

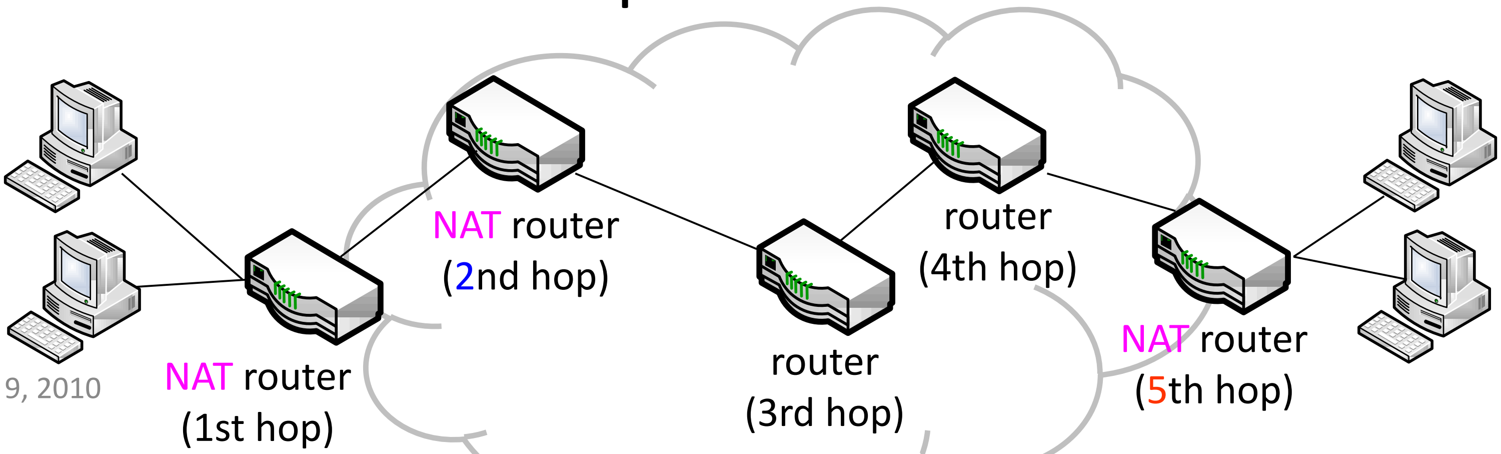
	Capturing Method	Scanning Method
Accuracy	O+	O
Required privilege	Root/Administrator	User mode

- Universal Plug and Play (UPnP) vs. i-Path

	UPnP	i-Path
Multiple levels of NAT	X	O
Authentication mechanism	X	O

Conclusion

- Extends UDP Multiple Hole Punching method
  - Improves the accuracy of Port Prediction
  - Proposes a practical Low TTL Value Determination
  - Discloses the info of NATs by the i-Path framework
- Future Work
  - Verifies the assumption that our LTVD is based on



Acknowledgement

- This research is supported by the National Institute of Information and Communications Technology, Japan.

- i-Path Project
  - http://i-path.goto.info.waseda.ac.jp/trac/i-Path/

Thank You

Q & A