Stateless Reconfiguration in DHCPv6

draft-jiang-dhc-stateless-reconfiguration-01

IETF 89 DHC WG

Mar 3rd, 2014

Sheng Jliang (Speaker)
Bing Liu

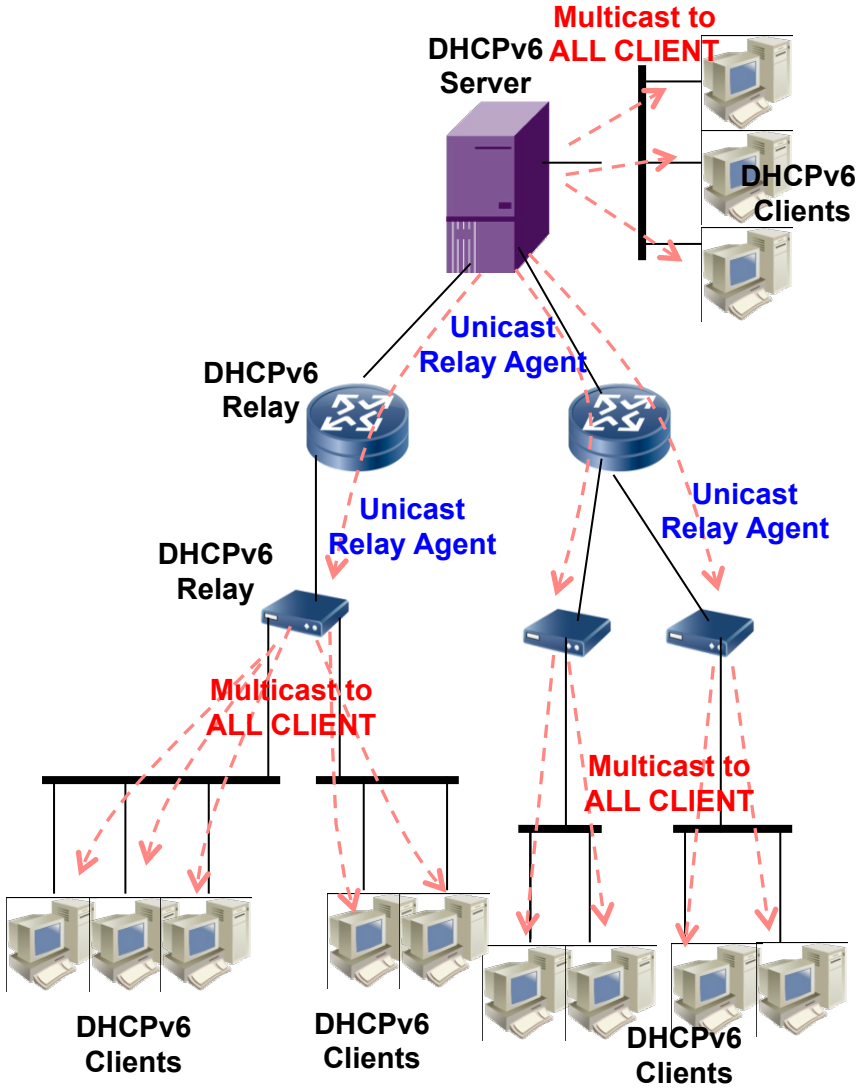
Discussion in last meeting

- **In ietf88, we introduced the basic principles of the mechanism and discussed the requirement of stateless reconfiguration**
- **The WG reached consensus on the requirement of stateless reconfiguration**
  - E.g. mistaken configuration (either caused by human or program) needs to be corrected
  - The network needs to initially update the already configured parameters within a short period due to some emergent events
- **This 01 version added above use cases**
- **Today we're going to discuss the mechanism designs (Marked as "Question to WG" in the draft)**

A brief review of the proposed mechanism

New specification

- A new “Stateless Reconfiguration” message
- A new Link-local scope well-known all-client multicast address
- Other potential new specification according to the following design alternatives



Solution Design -1

- **There are three possible mechanisms to create relay agent destinations on the DHCPv6 server {Question to WG No.1}**
  - **a) Static configuration**: network administrators manually configure static unicast addresses of all relay agents on the DHCPv6 server. (current draft is based this mechanism)
  - **b) Define an ALL_RELAY_AGENT multicast address**: The DHCPv6 server could send the stateless reconfiguration messages directly to the new multicast address.
  - **c) DHCPv6 server dynamically learning**: the DHCPv6 server dynamically records unicast addresses of all relay agents from client Information-request messages. The dynamic records need a keepalive mechanism between relay agents and servers.

	Pros	Cons
Static Configuration	- No need to update any protocol/ function implementation in relays	- Cost significant human management burden - Error-prone, mistakenly configuring the relay addresses or leaving out some relays are expected.
All_Relay _Agent Multicast Address	- A solid coverage of all relays	- Network administrators need to maintain an all-relay-agent multicast group - All relays and DHCPv6 servers need to be updated to know the new multicast address.
DHCPv6 Server Dynamically Learning	- Automatic processing without human intervene.	- Requires more function update to the DHCPv6 server - The keep alive mechanism requires more function/protocol

Solution Design -2

There are two modes to propagate the reconfiguration content. {Question to WG No.2}

- **a) Trigger mode**: The server sends out a multicast Stateless-Reconfiguration message to trigger clients initiating an Information-Request message back to the server. The server can then inform the changed configuration information to clients. (current draft is based this mode)
- **b) Push mode**: The server directly advertise new configuration to the clients. The clients then update the parameters accordingly.
- **c) Or we support both?**

  Then the “Trigger mode” and the “Push mode” messages need to be distinguished. Then there comes another design choice {Question to WG No.4} :

  - Use two type of stateless configuration messages
  - Use a flag in the one stateless configuration message

	Pros	Cons
Trigger Mode	- Obeys the traditional DHC behavior that client is the initiator - Allows the server to response to information-request differently according to various user policies.	- Might not be suitable for some efficiency sensitive scenarios (e.g. 802.15.4 mesh networks)
Push Mode	- Better efficiency, especially for the resource constrained networks. - Simpler protocol state machine	- Violates the traditional DHC behavior - More security issues regarding malicious server
Support Both	- Flexible and more functions	- Additional protocol complexity

{Question to WG No.3}

- Upon receipt of a valid Stateless-Reconfigure message, after a **random delay time**, the client responds with an Information-request message. **Should we define a maximum time of random delay time? If yes, should it come from server by a new option?**

Comments are welcomed!

Adopted as a WG item?

Thank You!