Enabling Ethernet devices without Global MAC Addresses

Pat Thaler, Senior Technical Director, Broadcom
20 June 2014

Problem statement

- **Network ports moving into smaller and smaller things**
  - Sensors and actuators – e.g. light switches and thermostats
  - Potentially dozens of ports per home, car or machine
  - Some things may be disposable or short lived
- **Should all these things consume global MAC address space?**
  - With cell phones and tablets, the consumption rate of MAC addresses has increased dramatically
  - The 48-bit MAC address space is supposed to last for at least 100 years
- **What about using Local Addresses?**
  - User configuration isn't reasonable – often no local interface and too large a potential for error
  - Existing automatic protocols configure addresses for virtual ports and rely existence of a physical port MAC address
- **Consider a protocol to get an address from the network with no initial address**

Alternatives for a protocol without a MAC address

- **Assign an address before the port comes up**
  - E.g. using auto-negotiation for Ethernet, from the Access Point on WiFi
  - But this would be MAC and, in some cases, PHY type dependent.
  - Only some Ethernet PHYs have auto-negotiation
  - Assumes the directly attached device (bridge or access point) will be the address server
- **Define a Null address value to use as a source address for the address acquisition protocol**
  - This address is never allowed as a destination address
  - New bridges can ignore it for learning when seen as a source address. For existing bridges, it will move around in learning, but since it never is a source address, it won't matter where they think it is.
  - Could use well-known group addresses for the destination address
    - Possibly one for address servers and one for client nodes
    - Possibly existing LAN scoped addresses – e.g. nearest non-TPMR

Identifying the right response

- **With a multicast destination address, how does a client know which reply PDUs are for it?**
- **Client PDUs include a Client ID with identifier type and value; examples of identifier types:**
  - EUI-64
  - ICC ID (from SIM card)
  - A random number for those devices that have no configured unique ID
- **Response PDU includes the Client ID from the client's PDU**
  - Client processes PDUs received with its Client ID and discards ones with other Client IDs

Who’s the address server?

- **Claiming protocol without a server**
  - Client generates a proposed address and initiates a claim, waits for response and uses address if no conflict detected
  - Proposed address might have a set value for the first 24 bits and a randomly generated value for the other 24.
  - Most suited to small* networks which can operate without a server
  - Requires that all nodes receive each other’s traffic (or something in the network can proxy for nodes that don’t receive the claim).
  - Similar protocols exist for IPv6 (RFC 4862) and FCoE (FC-BB-6 VN2VN)
- **Address Server**
  - Address requests go to a server which responds with an address
  - Default address range can be defined for operation without configuration
  - Multiple servers can operate by each having an address range.
- **Bridges as servers**
  - Address range could be divided between bridges with a distribution protocol – possibly starting from the spanning tree root
  - Reduces multicast traffic but all bridges might need to participate

* Small could be ~ 1000 ports

Claiming, servers, bridges

- **Claiming and server protocols could coexist**
  - Claiming protocol and server protocol can operate on different address ranges
  - Server could listen for Claims and reply with an address assignment
  - Allows the network to have a server or not as dictated by its size and nature and clients to adapt to either without configuration.
- **Bridge Relay**
  - Node transmits with Null Source Address
  - Bridge encapsulates in a relay PDU with the bridge's address for source address
  - Encapsulation may include a port identifier.
  - Responses go to bridge which relays to send to the well-know client multicast address
  - Bridge can use the port identifier to choose the output port for the relayed message.
  - Reduces multicast traffic for responses but requires changes to bridges

Address stability

- **Client may store the last used address**
- **On re-initializing, client may request the same address**
- **For server-less, it sends that address in the first claim**
  - If the claim fails, the client picks random address component as usual
- **For server, the address request can have a field to carry a proposed address**
  - The server assigns the proposed address if it is available and assigns another address if it isn't.

Quicker start up for specialized stable networks

- **Some applications such as automotive networks have strict requirements on latency to start the network.**
  - E.g. automotive network should work within on the order of 100 ms after power is applied
  - Changes to these networks would be rare
  - Potentially the learned address could be stored in non-volatile memory
  - If necessary, a message could be broadcast indicating that the existing addresses can still be used or a message can be sent to invalidate the existing address and restart address acquisition

Conclusion

- **Obtaining a MAC address over the network is possible**
- **This allows nodes to operate without a global MAC address and without configuration**
- **Such a protocol could protect the 48-bit MAC address space from exhaustion**
- **May also simplify the production of small inexpensive devices**
  - Removes need to configure with a global address at production time.
- **It may be desirable to standardize two mechanisms –**
  - Address server-based
  - Server-less claiming, and
  - Provide for coexistence of the two.