DNS LLQ

IETF 91 Honolulu
Tom Pusateri

Updated: 11/12/14

Introduction

- LLQ is a form of PubSub following the Observer design pattern
- Interested unicast clients subscribe to updates for a DNS query
- DNS servers publish the updates
- Unnecessary for multicast clients
- LLQ Client implemented in OSX, iOS, Bonjour for Windows
- LLQ Server implemented in `mdnsextd` to extend DNS server

LLQ Goals

- Detect when new instances appear
- Detect when existing instances disappear
- Monitor changes to a service
- More efficient and more timely than polling
- Minimal changes to existing DNS servers
- Handle moderate data set sizes and change rates

Discovery

```
C:SOA Query _ipp._tcp.floor1.foo.com.
S*:SOA Resp _ipp._tcp.floor1.foo.com.
   Answer: 0, Authority: 1
   SOA floor1.foo.com.
       proxy.floor1.foo.com.
C:SOA Query _tcp.floor1.foo.com.
S*:SOA Resp _tcp.floor1.foo.com.
   Answer: 0, Authority: 1
   SOA floor1.foo.com.
       proxy.floor1.foo.com.
```

```
C:SOA Query floor1.foo.com.
S*:SOA Resp floor1.foo.com.
   Answer: 1, Authority: 0
   SOA floor1.foo.com.
       proxy.floor1.foo.com.
C:SRV Query _dns-llq._udp.floor1.foo.com.
S*:SRV Resp _dns-llq._udp.floor1.foo.com.
   Answer: 1, Authority: 0
   SRV 0 0 53 proxy.floor1.foo.com.
```

LLQ Setup Goals

- Resilience to packet loss
- Ensure client reachability
- reduce DoS attacks
- reduce client packet storms

LLQ Setup (4-way)

Initial Request

```
C:Query:IN PTR _ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-SETUP ID:0
```

Challenge

```
S1:Query:IN PTR _ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-SETUP ID:RND64 ERR:0
```

Challenge Response

```
C:Query:IN PTR _ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-SETUP ID:RND64
```

ACK + Answers

```
S1:Query:IN PTR _ipp._tcp.floor1.foo.com
Answer:RDATA hp._ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-SETUP ID:RND64 ERR:0
```

LLQ Add Events

Add Event

```
S1:DNS Header ID:RND32
Query:IN PTR _ipp._tcp.floor1.foo.com Resp:IN
PTR _ipp._tcp.floor1.foo.com TTL 600
    RDATA xerox._ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-EVENT ID:RND64 ERR:0
```

Gratuitous Response Ack

```
C:DNS Header ID:RND32
Additional:OPT-LLQ LLQ-EVENT ID:RND64 ERR:0
```

LLQ Delete Event

Remove Event

```
S1:DNS Header ID:RND32
Query:IN PTR _ipp._tcp.floor1.foo.com Resp:IN
PTR _ipp._tcp.floor1.foo.com TTL -1
    RDATA xerox._ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-EVENT ID:RND64 ERR:0
```

Gratuitous Response Ack

```
C:DNS Header ID:RND32
Additional:OPT-LLQ LLQ-EVENT ID:RND64 ERR:0
```

LLQ Refresh

Refresh Request

```
C:Query:IN PTR _ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-REFRESH ID:RND64
Lease Life: 600
```

Refresh Ack

```
S1:Query:IN PTR _ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-REFRESH ID:RND64 ERR:0
Lease Life: 600
```

Terminate Request

```
C:Query:IN PTR _ipp._tcp.floor1.foo.com
Additional:OPT-LLQ LLQ-REFRESH ID:RND64
Lease Life: 0
```

LLQ Analysis

- Replicating the features of a connection oriented protocol complicates LLQ over UDP considerably
- protocol could be greatly simplified by moving to TCP
- Previous concerns over servers handling large numbers of TCP connections no longer valid
  - 10’s of thousands of connections seem possible
- maintaining DNS packet format adds complication
- there is not widespread adoption of the current LLQ draft
- encryption is easily achieved with TLS over TCP, but DTLS possible

TCP Existence Proof

- dnsextd can do LLQ over TCP
- TCP 3-way handshake replaces 4-way UDP LLQ setup procedure
- Supports Private LLQ - TLS using TSIG
- Supports Semi-Private LLQ - server can drop TLS connection after initial setup sending subsequent events over un-encrypted UDP

WG Options

- Pick up draft-sekar-dns-llq and standardize UDP
- Deprecate UDP and only standardize TCP
- Use existing DNS packet format with LLQ option but drop setup and possibly refresh
- Create new protocol for connection oriented LLQ only, not bound by existing DNS packet format
- Should TLS be required for the connection?
- What about signing the contents with TSIG or SIG0?

Related Work

- DNS PRIVate Exchange (dprive) WG
    - Client / Server encryption for UDP/TCP
        - draft-dempsky-dnscurve-01

        “TCP is considerably more expensive for clients and servers than UDP is, and TCP has no protection against denial of service, so server administrators are advised to stay below 512 bytes if possible. DNSCurve adds some denial-of-service protection for UDP but cannot do anything to help TCP.”

        - qualified: TCP SYN floods and un-encrypted TCP reset attacks
        - TCP 3-way handshake connection time results in 2x round trip time for queries/responses (according to RIPE Atlas Probes in 2012)
- DNS over DTLS
    - draft-wing-dnsop-dnsodtls-01
- TLS for DNS: Initiation and Performance Considerations
    - draft-hzhwm-dprive-start-tls-for-dns-00

Further Reading

- https://tools.ietf.org/html/draft-sekar-dns-llq-01
- LLQ in Zeroconf Book - http://zeroconf.org
- LLQ over TCP in mDNSResponder source - PrivateDNS.txt