XFR-over-TLS (XoT)

Making Zone Transfers Private

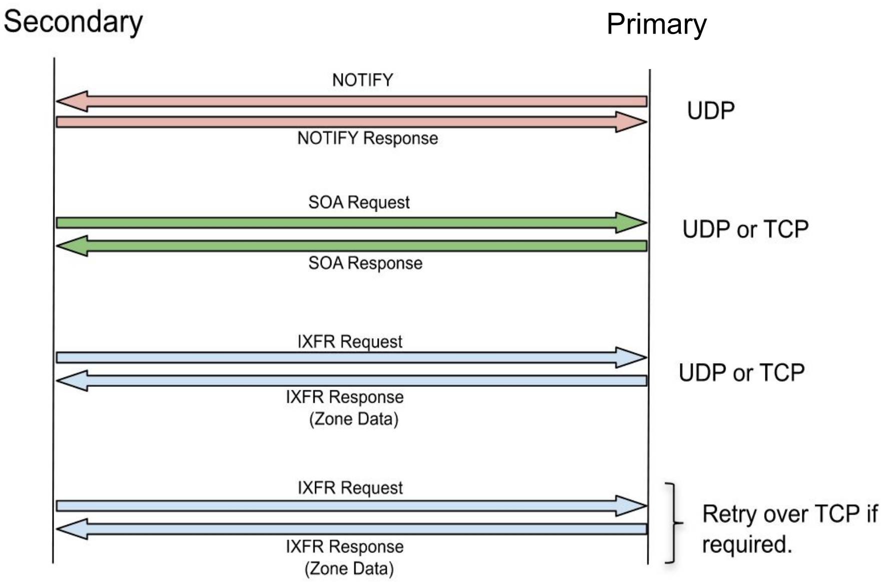
Allison Mankin amankin@gmail.com
Willem Toorop willem@nlnetlabs.nl
Sara Dickinson sara@sinodun.com
Pallavi Aras paras@salesforce.com
Han Zhang hzhang@salesforce.com

Use cases for XoT

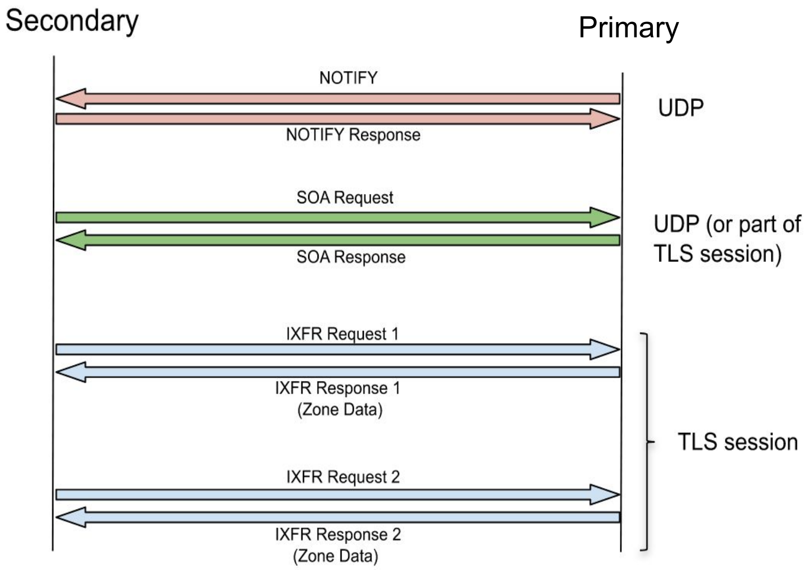
- **Confidentiality**: Encrypting zone transfers will **defeat zone content leakage** that can occur via passive surveillance
- **Authentication**: Use of single or mutual TLS authentication (in combination with ACLs) can complement and potentially be an alternative to TSIG
- **Performance**: Current usage of TCP for IXFR is sub-optimal in many cases e.g. TCP connections are frequently closed after a single IXFR for a single zone

- **SOLUTION**: Encryption of IXFR & AXFR using DNS-over-TLS [RFC7858]
  - Internet-Draft: **draft-hzpa-dprive-xfr-over-tls**

IXFR : Existing mechanisms vs IXoT

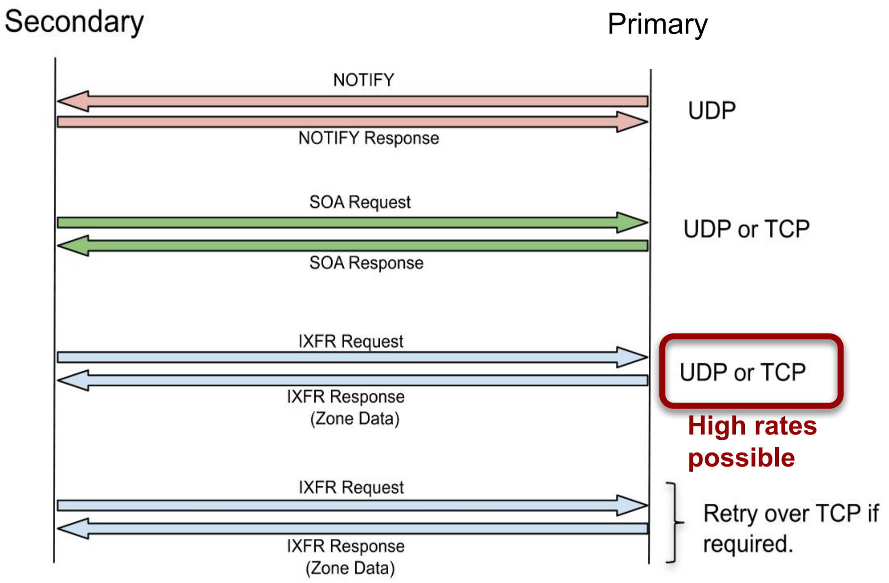


Existing



XOT-Based IXFR

IXFR : Existing mechanisms vs IXoT



Secondary

Primary

NOTIFY

NOTIFY Response

UDP

SOA Request

SOA Response

UDP or TCP

IXFR Request

IXFR Response
(Zone Data)

UDP or TCP

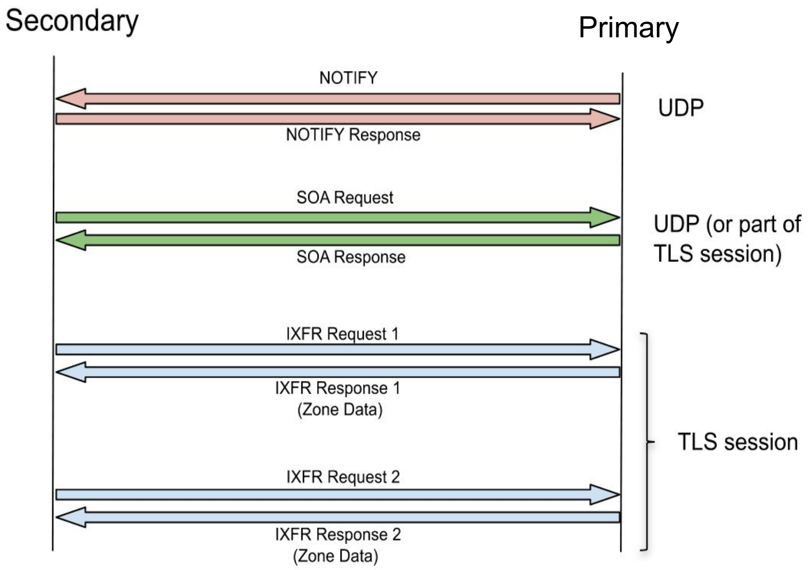
High rates possible

IXFR Request

IXFR Response
(Zone Data)

Retry over TCP if required.

Existing



Secondary

Primary

NOTIFY

NOTIFY Response

UDP

SOA Request

SOA Response

UDP (or part of TLS session)

IXFR Request 1

IXFR Response 1
(Zone Data)

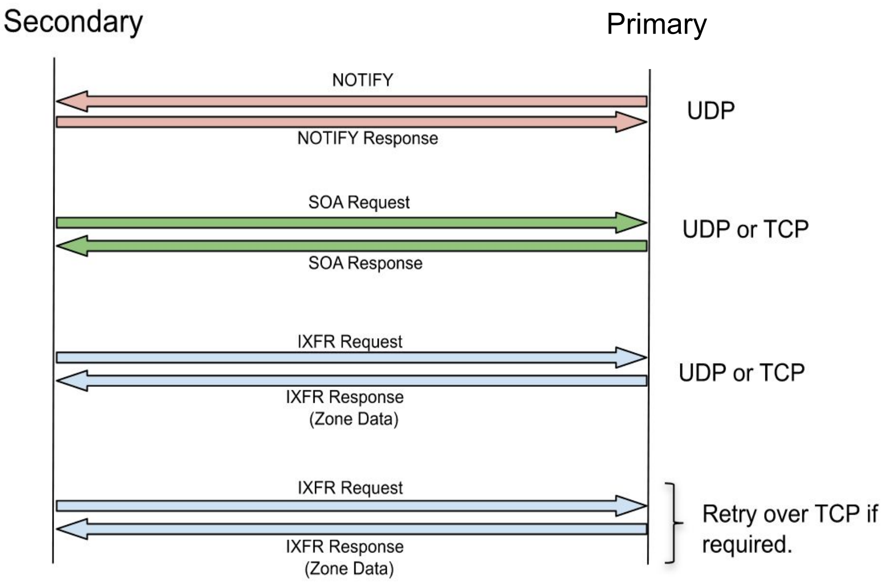
IXFR Request 2

IXFR Response 2
(Zone Data)

TLS session

XOT-Based IXFR

IXFR : Existing mechanisms vs IXoT



Secondary

Primary

NOTIFY

NOTIFY Response

UDP

SOA Request

SOA Response

UDP or TCP

IXFR Request

IXFR Response
(Zone Data)

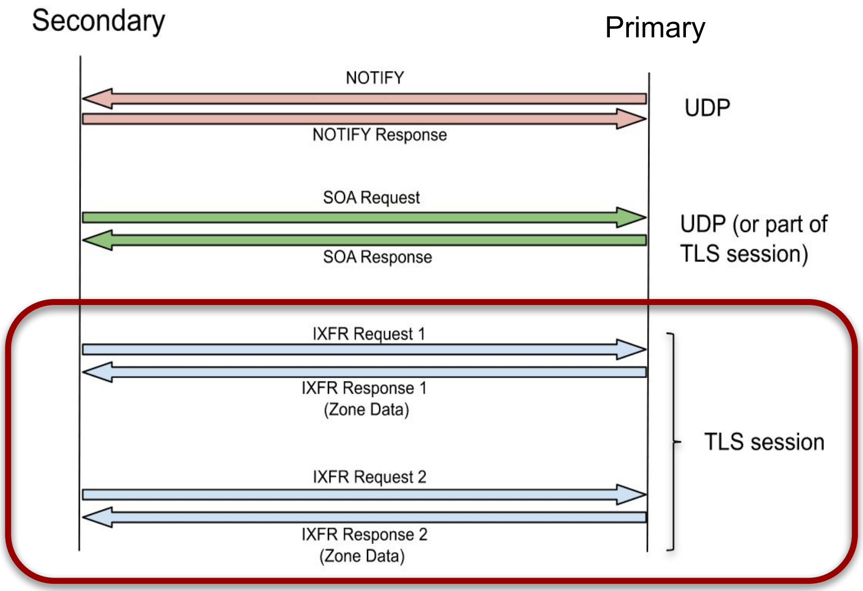
UDP or TCP

IXFR Request

IXFR Response
(Zone Data)

Retry over TCP if required.

Existing



Secondary

Primary

NOTIFY

NOTIFY Response

UDP

SOA Request

SOA Response

UDP (or part of TLS session)

IXFR Request 1

IXFR Response 1
(Zone Data)

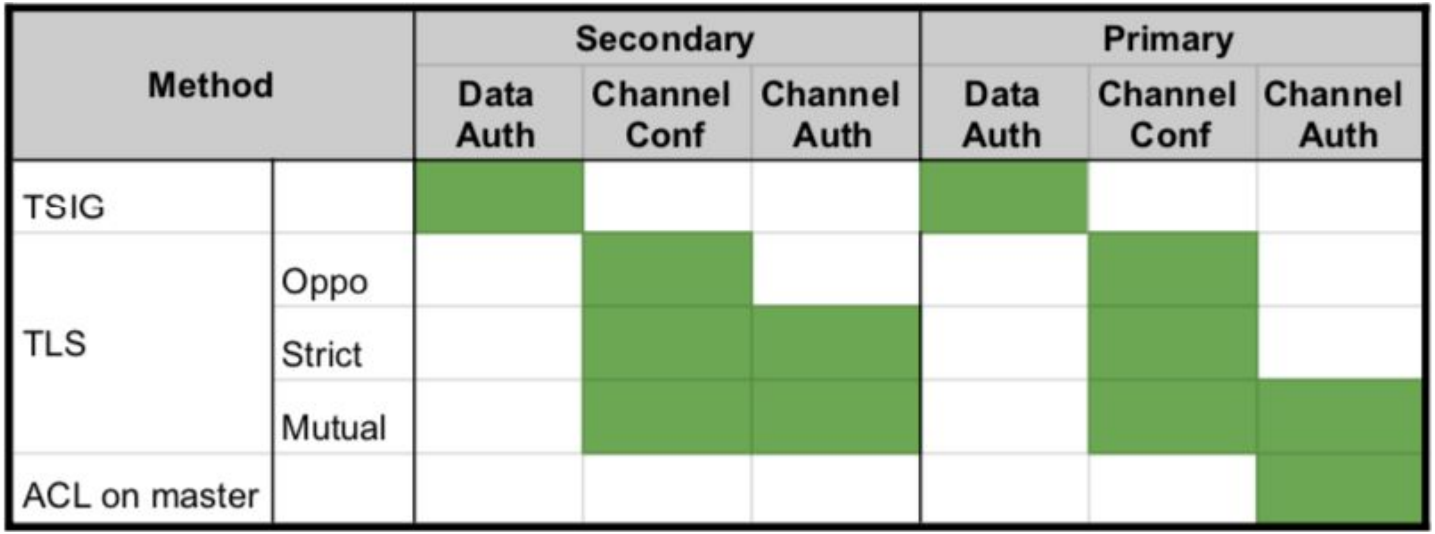
IXFR Request 2

IXFR Response 2
(Zone Data)

TLS session

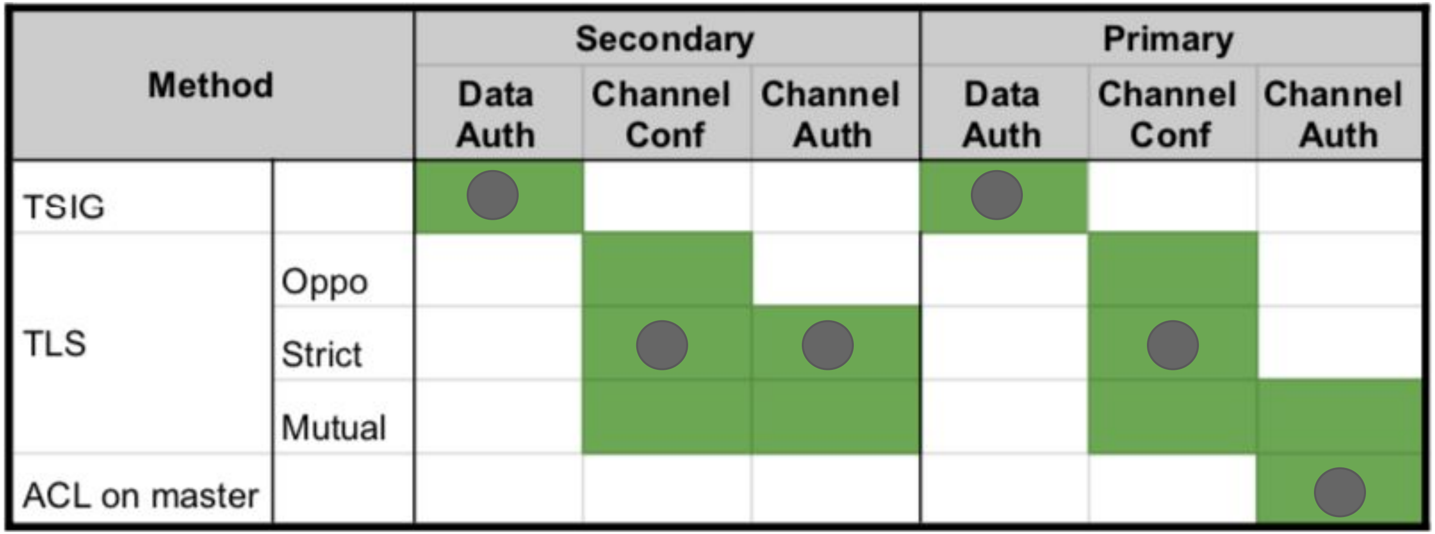
XOT-Based IXFR

XoT - Authentication mechanisms



Method		Secondary			Primary		
		Data Auth	Channel Conf	Channel Auth	Data Auth	Channel Conf	Channel Auth
TSIG		🟩			🟩		
TLS	Oppo		🟩			🟩	
	Strict		🟩	🟩		🟩	
	Mutual		🟩	🟩		🟩	🟩
ACL on master							🟩

XoT - Authentication mechanisms



Method		Secondary			Primary		
		Data Auth	Channel Conf	Channel Auth	Data Auth	Channel Conf	Channel Auth
TSIG		🟩 ●			🟩 ●		
TLS	Oppo		🟩			🟩	
	Strict		🟩 ●	🟩 ●		🟩 ●	
	Mutual		🟩	🟩		🟩	🟩
ACL on master							🟩 ●

Analysis: Using **TSIG, Strict TLS and an ACL** on the primary provides all 3 properties for both parties with reasonable overhead

Policy Management for XoT

- 'Transfer Group' - entire group of servers involved in transfers of a given zone (all primaries, all secondaries)

- The entire transfer group SHOULD have the same policy wrt (no weak point):
  - TSIG, TLS (O, S or m), IP ACL

- CHALLENGE: How to configure, enforce and test policy implementation?
  - Often involves different operators, different software, hidden servers
  - Feedback please ☺

Ongoing work

- **Latest implementation**
  - Unbound release 1.9.2 includes secondary-side AXFR XoT
  - NOTE: Server side XoT can be deployed using a TLS proxy

- **Open questions on the draft**
  - SHOULD/MUST
    - SOA query be on a TLS connection?
    - ‘Condensation’ of changes be required (optional in IXFR)?
    - Use only TLS 1.3 or later?
  - Padding - what policy?

Padding Policy

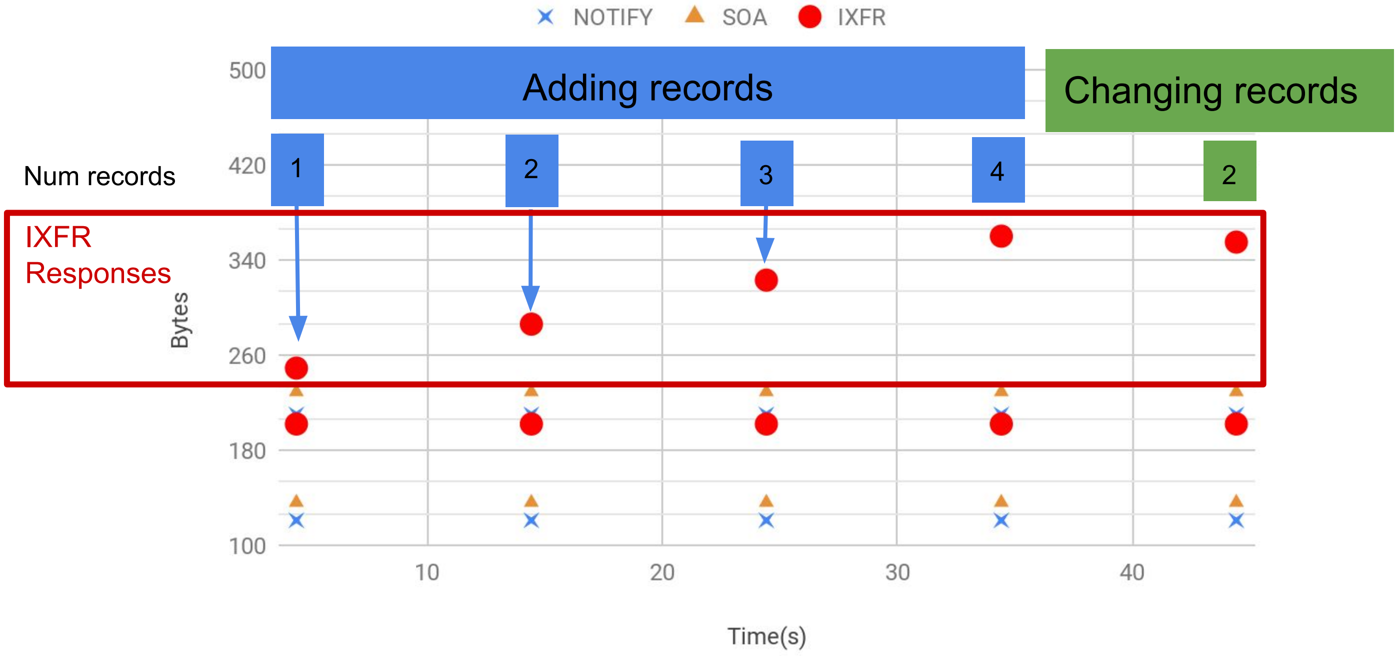
- **Requirements could be context specific**

- Packet sizes and timings vary depending on several factors:
    - Frequency of updates (manual reload vs steady dynamic updates vs batch dynamic)
    - ‘Condensation’ of changes
    - DNSSEC signed (NSEC/NSEC3)
        - Ongoing resigning of records as signatures expire (spikes or jittered)
        - Updates trigger resigning -> new RRSIGs

- Next slides show two extremes of patterns/packet sizes

BIND 9.12

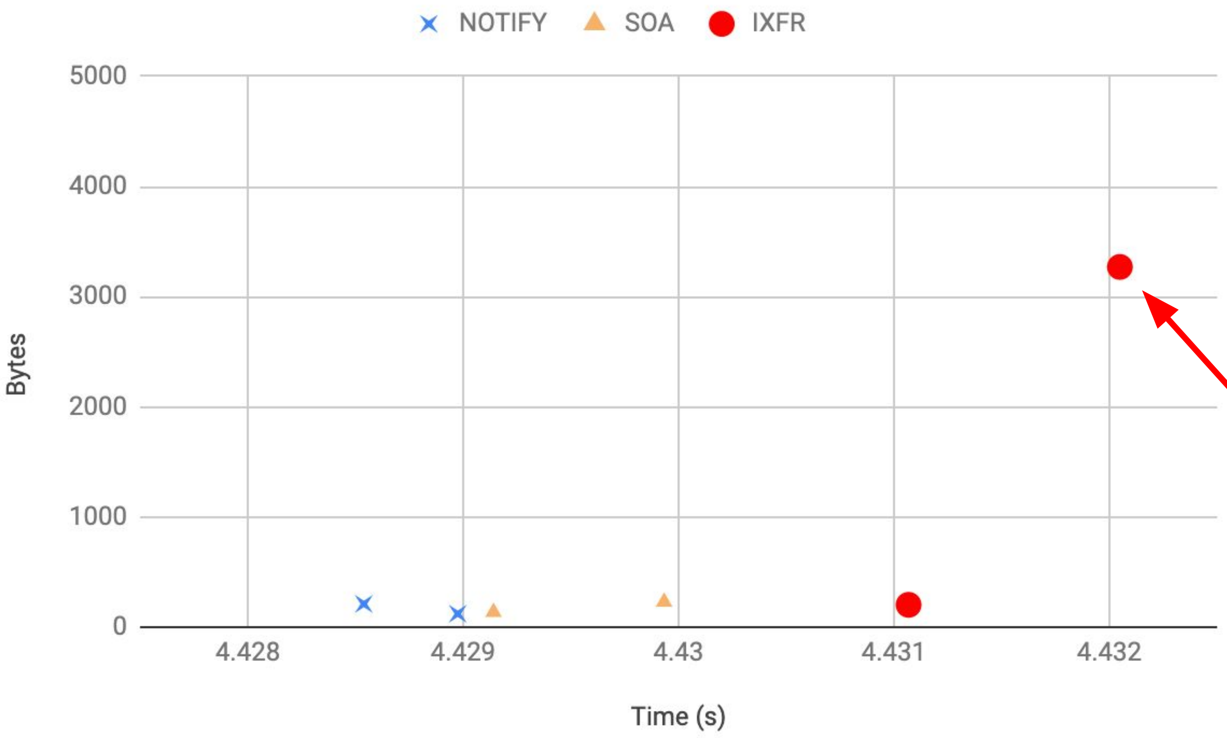
Simplest IXFR pattern (unsigned zone with regular updates)



- Unsigned zone with records added every 10 seconds
- **Smallest XFR response packet possible** would be 5 records:
  - 1 new record
  - 4 SOAs
- Order of few hundred bytes (**~250 in this case**)
- Packet size can indicate record changes but adding and changing are hard to distinguish (and name compression happens)

BIND 9.12

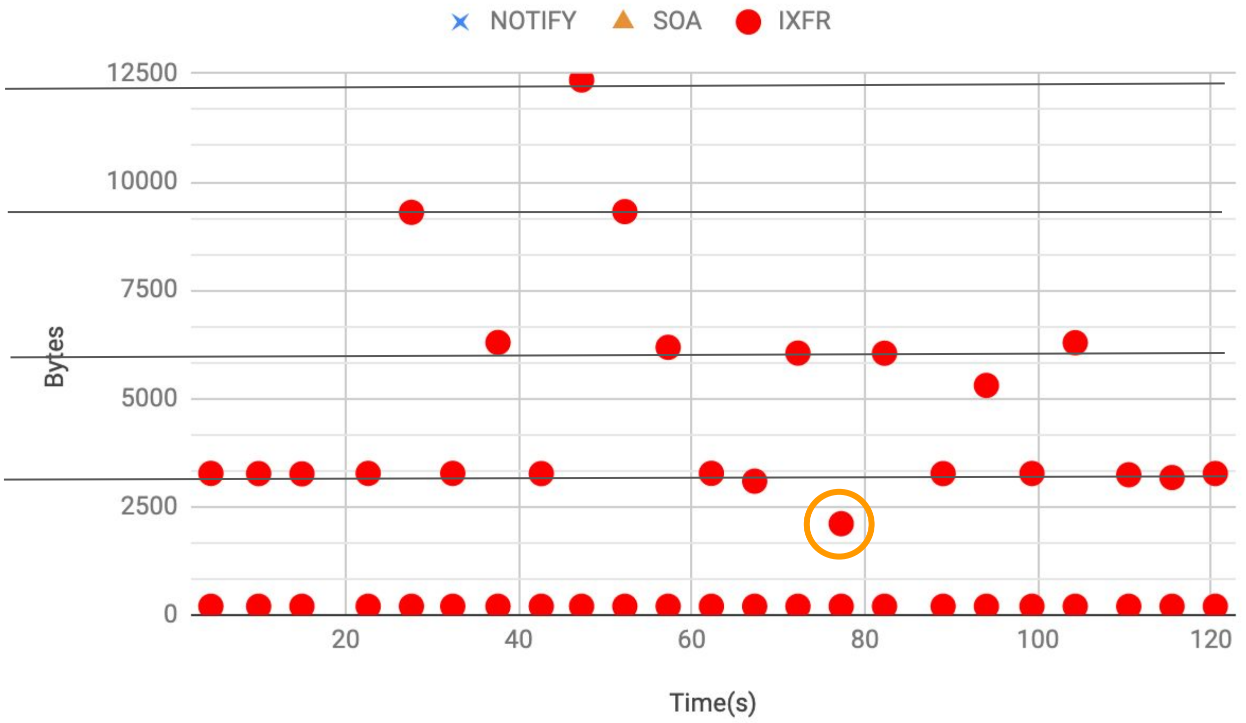
Single IXFR exchange for large DNSSEC NSEC3 signed zone (no updates)



- **Update triggered purely by resigning of signatures** (zone signed with jitter)
- 1 SOA change -> 12 RRSIGs regenerated
- 28 records in response
  - 12 removes
  - 12 adds
  - 4 SOA records
- Each record averages just over 100 bytes, **response is ~3000 bytes**

BIND 9.12

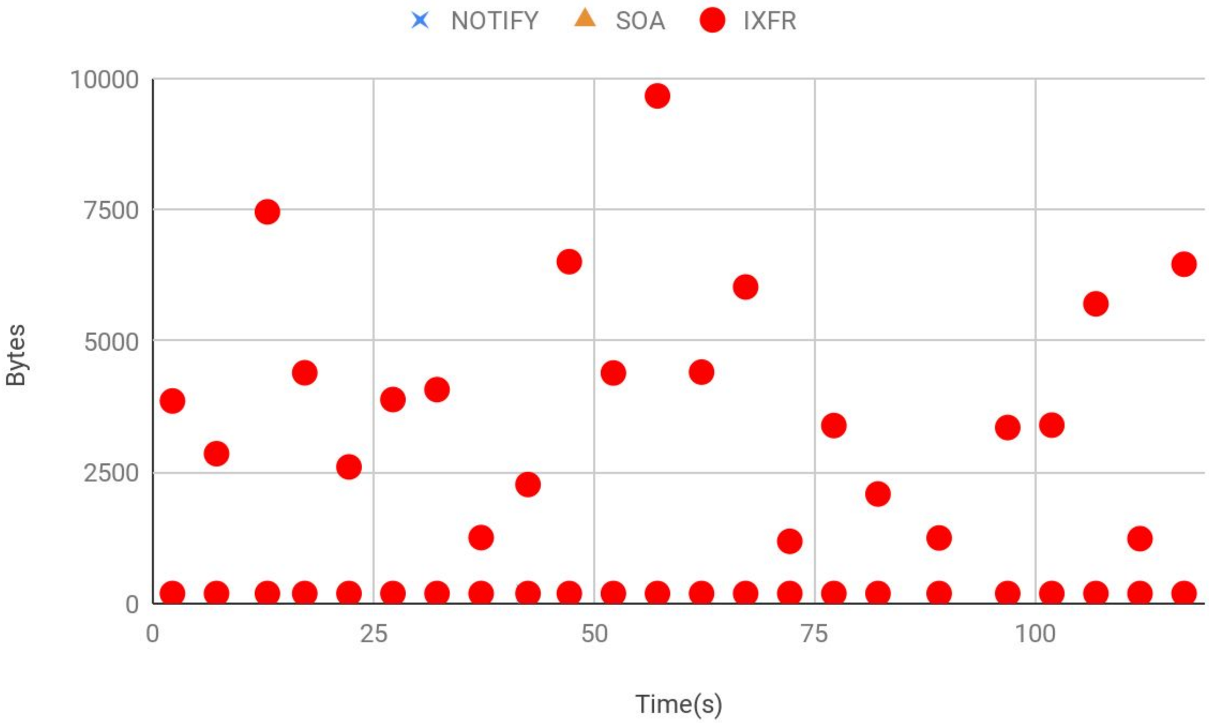
Multiple IXFRs for large DNSSEC NSEC3 signed zone (one update shown)



- **Periodic resigning dominates**
- Transfers every 5s, on a **separate TCP connection**
- Responses clustered around **multiples of 3k** bytes (1 SOA change) - note no condensation of changes
- Anomaly at 77s is caused by a **single record update to the zone**

BIND 9.12

Multiple IXFRs - large dynamic DNSSEC NSEC3 signed zone (many updates)



- **Updates to zone every few seconds**
- If updates are frequent, size pattern is more complex
- **But answers still dominated by RRSIG records**
- Still see 5s intervals

Take aways

- **Padding specifics**
  - Unsigned zones can directly leak number of record updates even when encrypted
  - Re-using a single connection for multiple zones would disguise the update pattern (as well as being a performance gain)
  - DNSSEC signing with jitter disguises the actual updates, but pattern varies with zone size and signing details
- **Future work for XoT in general**
  - Should some signalling be added (using EDNS0)? Useful for multiple aspects...