Advanced Computer Networking (ACN)

IN2097 – WiSe 2024–2025

Prof. Dr.-Ing. Georg Carle, Sebastian Gallenmüller

Christian Dietze, Max Helm, Benedikt Jaeger,
Marcel Kempf, Jihye Kim, Patrick Sattler

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

Domain Name System

Domain Name System

Introduction (recap)
Terminology

- Stub resolver:
  - According to RFC 1034 [1]: Provides recursive resolution for a system which lacks resources (e.g. your PC)

1 https://www.iana.org/domains/root/servers

Introduction (recap)
Terminology

- Stub resolver:
  - According to RFC 1034 [1]: Provides recursive resolution for a system which lacks resources (e.g. your PC)
- Forwarder:
  - Forwards DNS queries to another resolver
  - E.g. your routers resolver

1 https://www.iana.org/domains/root/servers

Introduction (recap)
Terminology

- Stub resolver:
  - According to RFC 1034 [1]: Provides recursive resolution for a system which lacks resources (e.g. your PC)
- Forwarder:
  - Forwards DNS queries to another resolver
  - E.g. your routers resolver
- Recursive resolver
  - Handles recursive queries and iteratively resolves them
  - Usually open resolvers are recursive resolver

[1] https://www.iana.org/domains/root/servers

Introduction (recap)
Terminology

- Stub resolver:
  - According to RFC 1034 [1]: Provides recursive resolution for a system which lacks resources (e.g. your PC)
- Forwarder:
  - Forwards DNS queries to another resolver
  - E.g. your routers resolver
- Recursive resolver
  - Handles recursive queries and iteratively resolves them
  - Usually open resolvers are recursive resolver
- Authoritative name server
  - Has authoritative information on a set of zones
  - Gets queried by recursive resolvers

[1] https://www.iana.org/domains/root/servers

Introduction (recap)
Terminology

- Stub resolver:
  - According to RFC 1034 [1]: Provides recursive resolution for a system which lacks resources (e.g. your PC)
- Forwarder:
  - Forwards DNS queries to another resolver
  - E.g. your routers resolver
- Recursive resolver
  - Handles recursive queries and iteratively resolves them
  - Usually open resolvers are recursive resolver
- Authoritative name server
  - Has authoritative information on a set of zones
  - Gets queried by recursive resolvers
- TLD name server
  - Authoritative nameserver for the TLD zones
  - E.g. `a.nic.de` for the `de` zone

1 https://www.iana.org/domains/root/servers

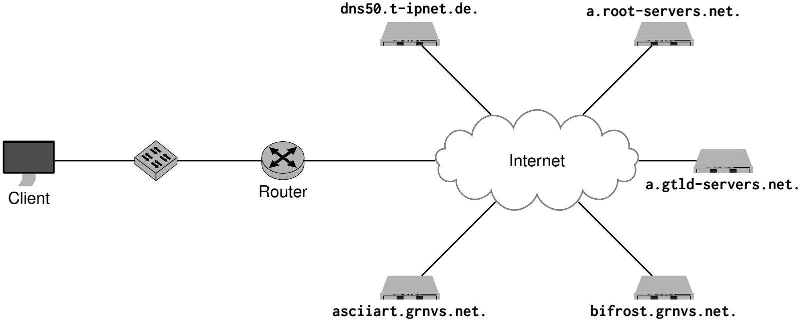
Introduction (recap)
Terminology

- Stub resolver:
  - According to RFC 1034 [1]: Provides recursive resolution for a system which lacks resources (e.g. your PC)
- Forwarder:
  - Forwards DNS queries to another resolver
  - E.g. your routers resolver
- Recursive resolver
  - Handles recursive queries and iteratively resolves them
  - Usually open resolvers are recursive resolver
- Authoritative name server
  - Has authoritative information on a set of zones
  - Gets queried by recursive resolvers
- TLD name server
  - Authoritative nameserver for the TLD zones
  - E.g. `a.nic.de` for the `de` zone
- Root server:
  - Authoritative name servers which serve the DNS root zone[1]
  - 13 authorities manage hundreds of servers: [a-m].root-servers.net
  - E.g. `k.root-servers.net` is managed by RIPE
  - `https://root-servers.org/` tracks the location of many root servers

1 https://www.iana.org/domains/root/servers

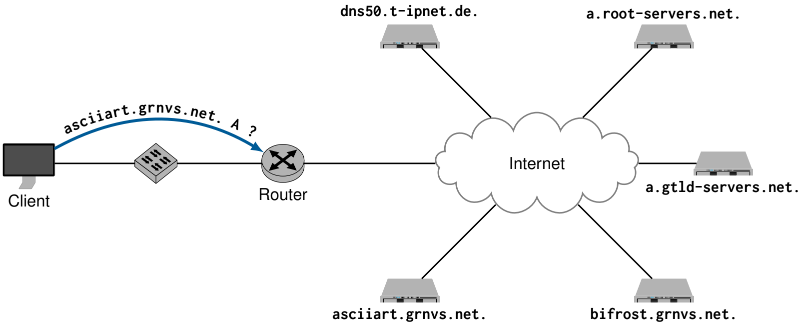
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



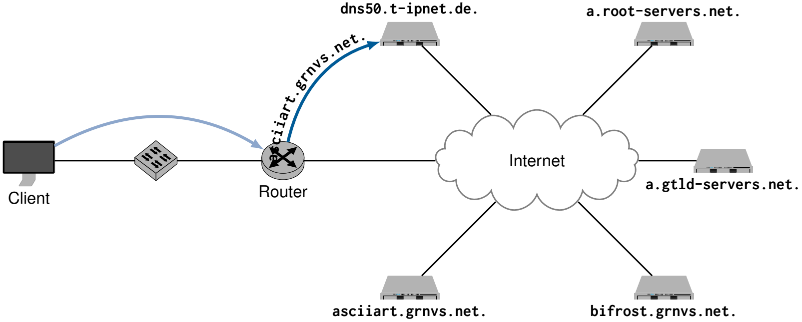
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



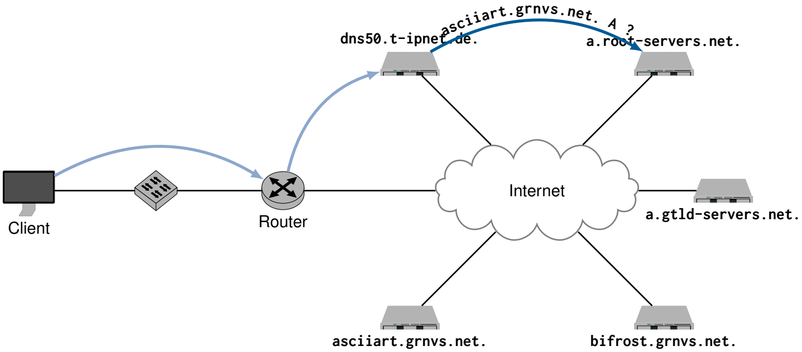
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



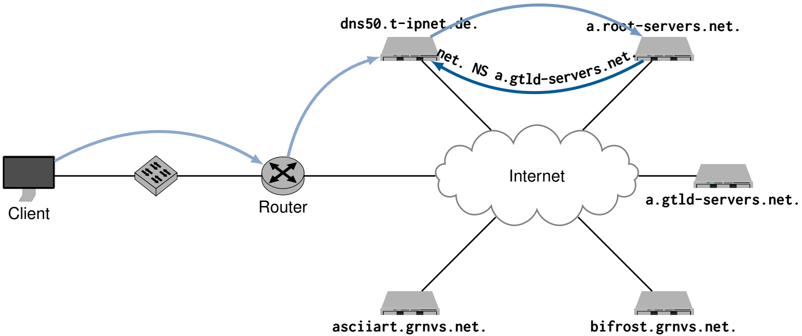
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



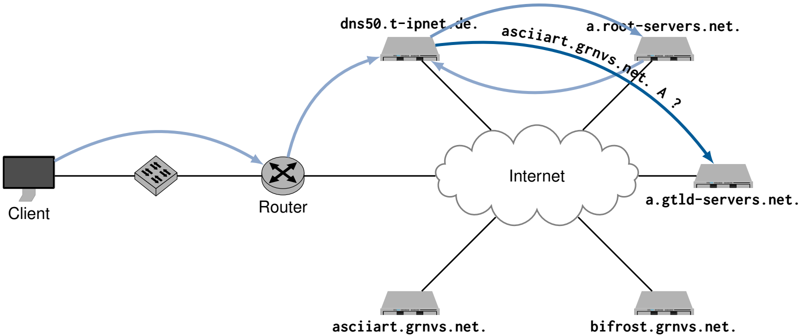
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



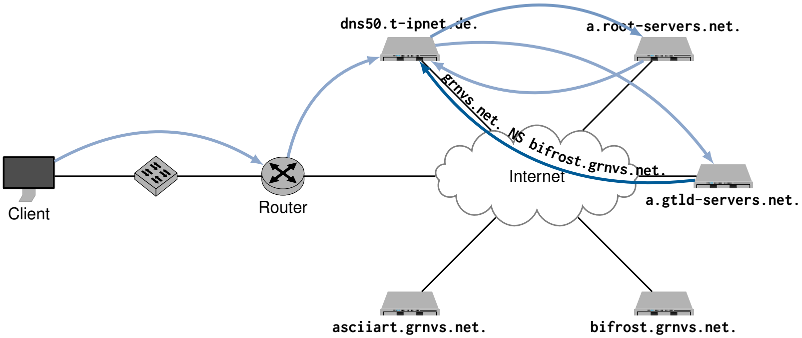
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



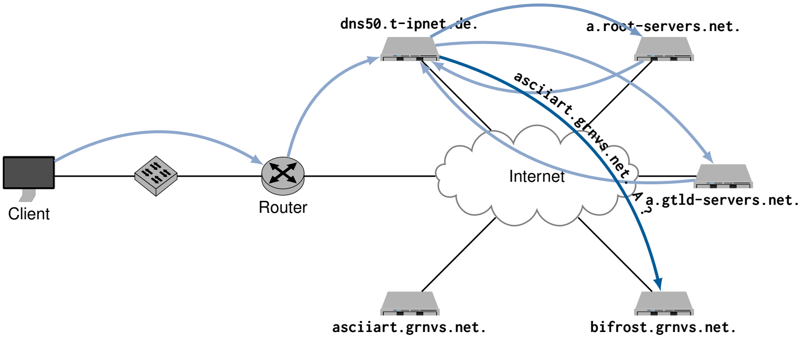
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



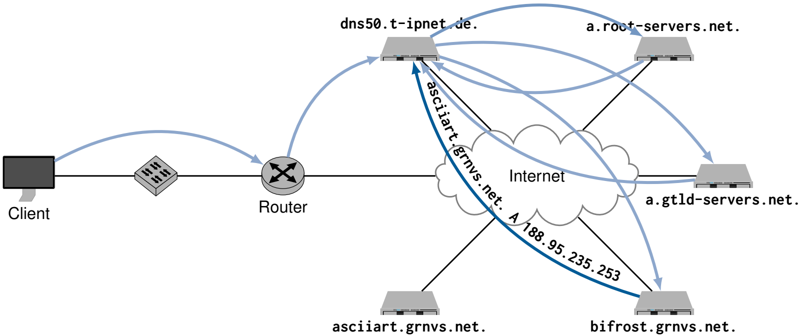
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



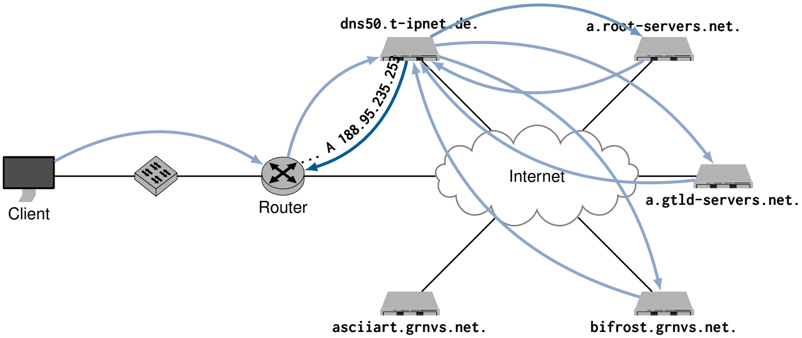
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



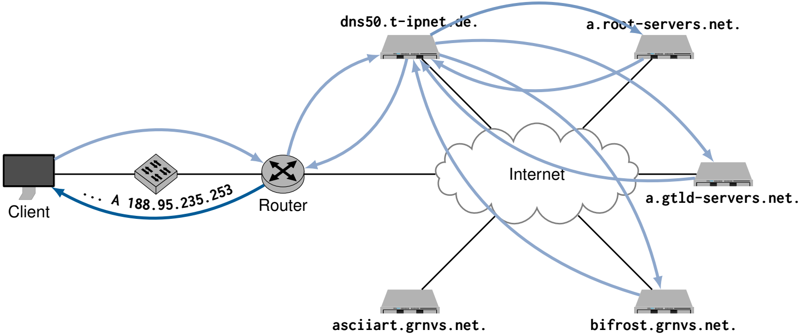
Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



Introduction (recap)
Function

- First standardized in RFC 1034 [1] and 1035 [2]
- System to resolve Fully Qualified Domain Name (FQDN) to IP addresses
- Original concept focused on high scalability → distributed database



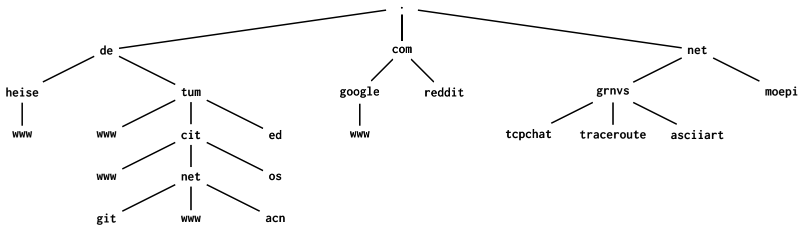
Introduction (recap)
Hierarchical Structure

- The distributed concept of DNS is based on delegations
- Starting at the root zone a tree of delegations is build

[2] List of eTLDs by Mozilla `publicsuffix.org`

Introduction (recap)
Hierarchical Structure

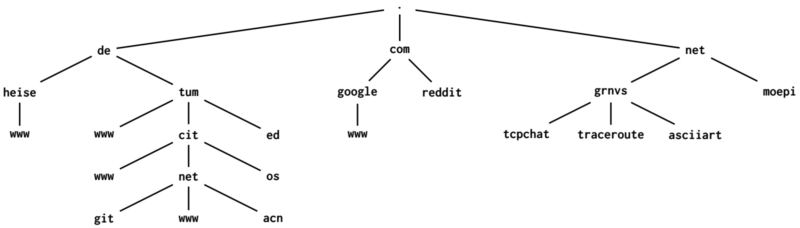
- The distributed concept of DNS is based on delegations
- Starting at the root zone a tree of delegations is build



2 List of eTLDs by Mozilla `publicsuffix.org`

Introduction (recap)
Hierarchical Structure

- The distributed concept of DNS is based on delegations
- Starting at the root zone a tree of delegations is build



eTLD

- Effective top-level domain[2]
- co.uk, com.br, gov.br, ...

---

2 List of eTLDs by Mozilla publicsuffix.org

Domain Name System

DNS Basics
Messages

Header	
Question	The question of the name server
Answer	RRs answering the question
Authority	RRs pointing toward an authority
Additional	RRs holding additional information

- Query and response use same message format
- Header indicates type of message
- The answer, authority, and additional section are arrays of resource records (RR)

DNS Basics
Message Header

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">ID</td></tr>
<tr><td>QR</td><td colspan="4">Opcode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td colspan="16">QDCOUNT</td></tr>
<tr><td colspan="16">ANCOUNT</td></tr>
<tr><td colspan="16">NSCOUNT</td></tr>
<tr><td colspan="16">ARCOUNT</td></tr>
</table>

[3] https://www.iana.org/assignments/dns-parameters contains all defined opcode an rcode values

DNS Basics
Message Header

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">ID</td></tr>
<tr><td>QR</td><td colspan="4">Opcode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td colspan="16">QDCOUNT</td></tr>
<tr><td colspan="16">ANCOUNT</td></tr>
<tr><td colspan="16">NSCOUNT</td></tr>
<tr><td colspan="16">ARCOUNT</td></tr>
</table>

ID Unique query ID to identify the corresponding response

[3] `https://www.iana.org/assignments/dns-parameters` contains all defined opcode an rcode values

DNS Basics
Message Header

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ID															
QR	Opcode				AA	TC	RD	RA	Z			RCODE			
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

ID Unique query ID to identify the corresponding response

QR Set if the message is a response

[3] https://www.iana.org/assignments/dns-parameters contains all defined opcode an rcode values

DNS Basics
Message Header

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">ID</td></tr>
<tr><td>QR</td><td colspan="4">Opcode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td colspan="16">QDCOUNT</td></tr>
<tr><td colspan="16">ANCOUNT</td></tr>
<tr><td colspan="16">NSCOUNT</td></tr>
<tr><td colspan="16">ARCOUNT</td></tr>
</table>

- **ID** Unique query ID to identify the corresponding response
- **QR** Set if the message is a response
- **Opcode** Specifies kind of query (e.g. query, status, notify, update)[3]

[3] https://www.iana.org/assignments/dns-parameters contains all defined opcode an rcode values

Message Header

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">ID</td></tr>
<tr><td>QR</td><td colspan="4">Opcode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td colspan="16">QDCOUNT</td></tr>
<tr><td colspan="16">ANCOUNT</td></tr>
<tr><td colspan="16">NSCOUNT</td></tr>
<tr><td colspan="16">ARCOUNT</td></tr>
</table>

ID Unique query ID to identify the corresponding response

QR Set if the message is a response

Opcode Specifies kind of query (e.g. query, status, notify, update)[3]

AA: Authoritative Answer Set if the responding name server is an authority for the requested domain

[3] `https://www.iana.org/assignments/dns-parameters` contains all defined opcode an rcode values

DNS Basics
Message Header

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">ID</td></tr>
<tr><td>QR</td><td colspan="4">Opcode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td colspan="16">QDCOUNT</td></tr>
<tr><td colspan="16">ANCOUNT</td></tr>
<tr><td colspan="16">NSCOUNT</td></tr>
<tr><td colspan="16">ARCOUNT</td></tr>
</table>

ID	Unique query ID to identify the corresponding response
QR	Set if the message is a response
Opcode	Specifies kind of query (e.g. query, status, notify, update)[3]
AA: Authoritative Answer	Set if the responding name server is an authority for the requested domain
TC: Truncated	Indicates that the DNS message is truncated due to the permitted length

[3] `https://www.iana.org/assignments/dns-parameters` contains all defined opcode an rcode values

Message Header (2)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ID															
QR	Opcode				AA	TC	RD	RA	Z			RCODE			
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

RD: Recursion desired If set the nameserver resolves the query recursively

DNS Basics
Message Header (2)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ID															
QR	Opcode				AA	TC	RD	RA	Z			RCODE			
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

RD: Recursion desired — If set the nameserver resolves the query recursively

RA: Recursion available — Set by the nameserver if it supports recursive queries

DNS Basics
Message Header (2)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ID															
QR	Opcode				AA	TC	RD	RA	Z			RCODE			
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

RD: Recursion desired If set the nameserver resolves the query recursively

RA: Recursion available Set by the nameserver if it supports recursive queries

Z 1 bit future use; 2 bits for DNSSEC (authentic data (AD) and checking disabled (CD))

DNS Basics
Message Header (2)

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td colspan="16">ID</td></tr>
<tr><td>QR</td><td colspan="4">Opcode</td><td>AA</td><td>TC</td><td>RD</td><td>RA</td><td colspan="3">Z</td><td colspan="4">RCODE</td></tr>
<tr><td colspan="16">QDCOUNT</td></tr>
<tr><td colspan="16">ANCOUNT</td></tr>
<tr><td colspan="16">NSCOUNT</td></tr>
<tr><td colspan="16">ARCOUNT</td></tr>
</table>

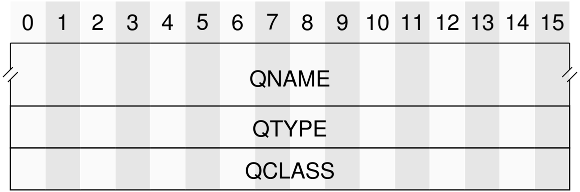
RD: Recursion desired	If set the nameserver resolves the query recursively
RA: Recursion available	Set by the nameserver if it supports recursive queries
Z	1 bit future use; 2 bits for DNSSEC (authentic data (AD) and checking disabled (CD))
RCODE: Response code	Code indicating query status (e.g. NOERROR, NXDOMAIN, SERVFAIL)

DNS Basics
Message Header (2)

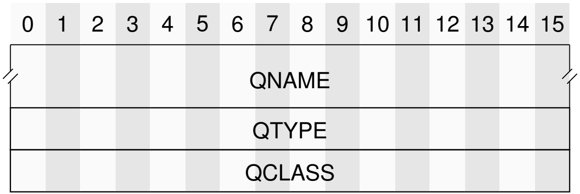
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ID															
QR	Opcode				AA	TC	RD	RA	Z			RCODE			
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

RD: Recursion desired	If set the nameserver resolves the query recursively
RA: Recursion available	Set by the nameserver if it supports recursive queries
Z	1 bit future use; 2 bits for DNSSEC (authentic data (AD) and checking disabled (CD))
RCODE: Response code	Code indicating query status (e.g. NOERROR, NXDOMAIN, SERVFAIL)
*COUNT	Number of RR in the corresponding message section

DNS Basics
Question Section

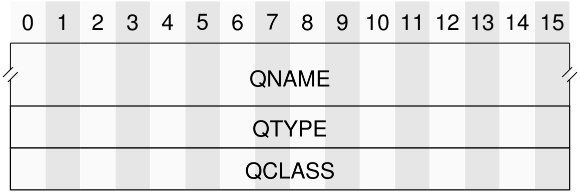


DNS Basics
Question Section



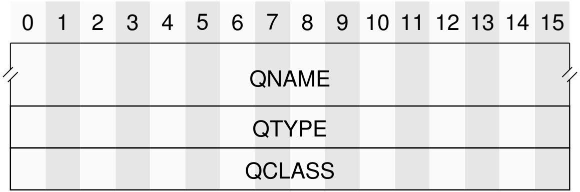
QNAME Requested Name, variable length

DNS Basics
Question Section



QNAME	Requested Name, variable length
QTYPE	Requested RR Type (e.g., A, NS)

DNS Basics
Question Section

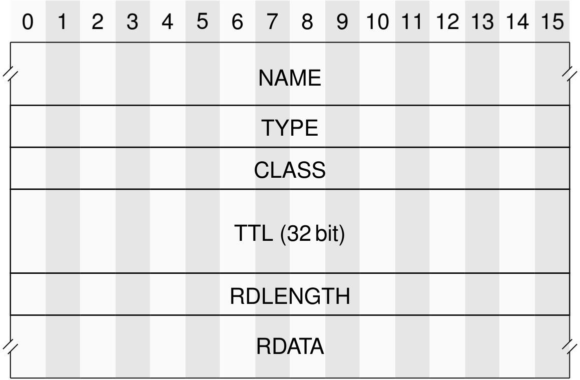


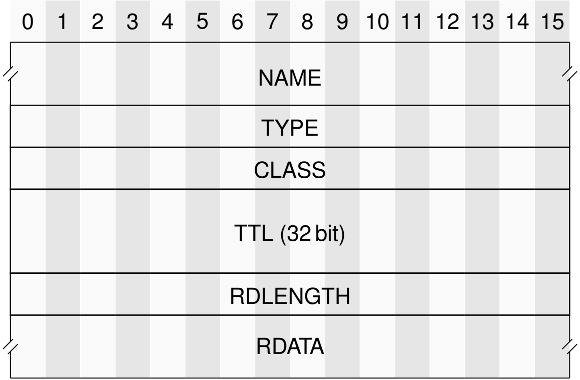
QNAME Requested Name, variable length

QTYPE Requested RR Type (e.g., A, NS)

QCLASS Normally Internet (IN)

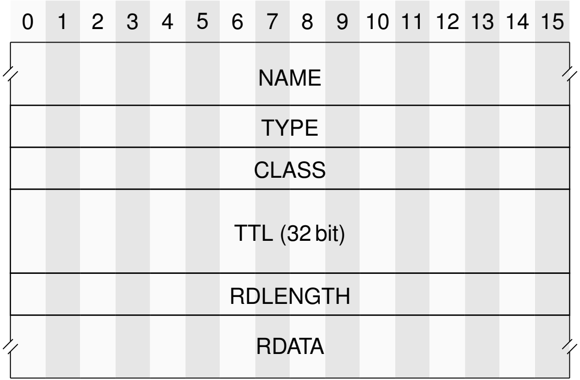
DNS Basics
RR Sections





TTL Valid lifetime of the RR in seconds

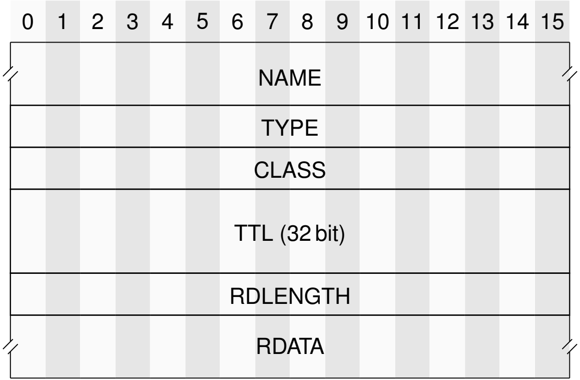
DNS Basics
RR Sections



TTL Valid lifetime of the RR in seconds

RDLENGTH Length of the following data

DNS Basics
RR Sections



TTL	Valid lifetime of the RR in seconds
RDLENGTH	Length of the following data
RDATA	Data of the RR mapped to the name

DNS Basics
Resource Record Types

Type	Meaning	Representation
A	an IPv4 host address	32 bit address
AAAA	an IPv6 host address	128 bit address

[4] https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-https/

DNS Basics
Resource Record Types

Type	Meaning	Representation
A	an IPv4 host address	32 bit address
AAAA	an IPv6 host address	128 bit address
CNAME	canonical name for an alias	a domain name
NS	autorithative name server	domain name
SOA	start of zone authority	Various fields

[4] https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-https/

DNS Basics
Resource Record Types

Type	Meaning	Representation
A	an IPv4 host address	32 bit address
AAAA	an IPv6 host address	128 bit address
CNAME	canonical name for an alias	a domain name
NS	autorithative name server	domain name
SOA	start of zone authority	Various fields
MX	Mail exchange address	Preference and mail server domain name
TXT	TXT record	Arbitrary text

4 https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-https/

DNS Basics
Resource Record Types

Type	Meaning	Representation
A	an IPv4 host address	32 bit address
AAAA	an IPv6 host address	128 bit address
CNAME	canonical name for an alias	a domain name
NS	autorithative name server	domain name
SOA	start of zone authority	Various fields
MX	Mail exchange address	Preference and mail server domain name
TXT	TXT record	Arbitrary text
SVCB	service binding record	Information on services[4]
HTTPS	HTTPS service binding record	Information on the HTTPS service

4 https://datatracker.ietf.org/doc/draft-ietf-dnsop-svcb-https/

DNS Basics
Zone

- RFC 8499 defines a zone:

  Authoritative information is organized into units called
  ZONEs, and these zones can be automatically distributed to the
  name servers which provide redundant service for the data in a zone.

- Has a set of name server records (authoritative nameserver)
- Starts with a SOA record, ends at the next SOA record
- Child zone:

  The entity on record that has the delegation of the domain from the Parent.

- Parent zone:

  The domain in which the Child is registered.

- Delegation:

  The process by which a separate zone is created in the
  name space beneath the apex of a given domain.

DNS Basics
Delegations

Parent zone:

- The zone of the domain name excluding the last label (except ENTs)
- E.g. `de` for `tum.de` and `in.tum.de` for `net.in.tum.de`
- `acn.net.in.tum.de` has no SOA record. I.e. it is not in the zone apex

DNS Basics
Delegations

Parent zone:

- The zone of the domain name excluding the last label (except ENTs)
- E.g. `de` for `tum.de` and `in.tum.de` for `net.in.tum.de`
- `acn.net.in.tum.de` has no SOA record. I.e. it is not in the zone apex
  - → it is part of `net.in.tum.de`

DNS Basics
Delegations

Parent zone:

- The zone of the domain name excluding the last label (except ENTs)
- E.g. `de` for `tum.de` and `in.tum.de` for `net.in.tum.de`
- `acn.net.in.tum.de` has no SOA record. I.e. it is not in the zone apex

  → it is part of `net.in.tum.de`

Delegations:

- The parent zone has the NS records which delegate the query to the authoritative name server of a zone
- Recap:
  - NS record points to a domain name
  - Either a domain name in the same zone – called in-bailiwick
    - E.g. `ns1.google.com` for `google.com`
  - Or any other domain name (e.g. `dns1.lrz.de` for `net.in.tum.de`)

Delegations

Parent zone:

- The zone of the domain name excluding the last label (except ENTs)
- E.g. `de` for `tum.de` and `in.tum.de` for `net.in.tum.de`
- `acn.net.in.tum.de` has no SOA record. I.e. it is not in the zone apex
  → it is part of `net.in.tum.de`

Delegations:

- The parent zone has the NS records which delegate the query to the authoritative name server of a zone
- Recap:
  - NS record points to a domain name
  - Either a domain name in the same zone – called in-bailiwick
    - E.g. `ns1.google.com` for `google.com`
  - Or any other domain name (e.g. `dns1.lrz.de` for `net.in.tum.de`)
    → Resolver needs to query A/AAAA record of name server name

Parent zone:

- The zone of the domain name excluding the last label (except ENTs)
- E.g. `de` for `tum.de` and `in.tum.de` for `net.in.tum.de`
- `acn.net.in.tum.de` has no SOA record. I.e. it is not in the zone apex
  → it is part of `net.in.tum.de`

Delegations:

- The parent zone has the NS records which delegate the query to the authoritative name server of a zone
- Recap:
  - NS record points to a domain name
  - Either a domain name in the same zone – called in-bailiwick
    - E.g. `ns1.google.com` for `google.com`
  - Or any other domain name (e.g. `dns1.lrz.de` for `net.in.tum.de`)
    → Resolver needs to query A/AAAA record of name server name
  - Problem: How can in-bailiwick records (e.g. of `ns1.google.com`) be retrieved?

Glue Records

- An A/AAAA record in the parent zone for the name server name of a child zone
- Glue records are non authoritative records in the parent zone

Empty Non Terminals (ENT)

- Nodes with children but no RRs of their own (RFC2136 Section 7.16)
- Queries for ENTs return NOERROR but RR in the answer section
- This behavior is important for QNAME minimization and rDNS walking
- E.g.
  - `www.ent.example.com` contains a SOA and an A record
  - `ent.example.com` contains no record
  - `example.com` contains at least a SOA record
  - `ent.example.com` is an ENT

DNS Basics
Delegations Issues

- NS record points to an IP address

5 The .io Error – Taking Control of All .io Domains With a Targeted Registration https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/

DNS Basics
Delegations Issues

- NS record points to an IP address
  - Not reachable and not valid
  - Reliabilty issue (e.g. other name server are not reachable/overloaded)

[5] The .io Error – Taking Control of All .io Domains With a Targeted Registration https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/

Delegations Issues

- NS record points to an IP address
  - Not reachable and not valid
  - Reliabilty issue (e.g. other name server are not reachable/overloaded)

- NS record contains a typo in the domain name
  1. Domain name is not registrable

[5] The .io Error – Taking Control of All .io Domains With a Targeted Registration https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/

DNS Basics
Delegations Issues

- NS record points to an IP address
  - Not reachable and not valid
  - Reliabilty issue (e.g. other name server are not reachable/overloaded)

- NS record contains a typo in the domain name
  1. Domain name is not registrable
     - Reliabilty issue

[5] The .io Error – Taking Control of All .io Domains With a Targeted Registration https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/

Delegations Issues

- NS record points to an IP address
  - Not reachable and not valid
  - Reliabilty issue (e.g. other name server are not reachable/overloaded)

- NS record contains a typo in the domain name
  1. Domain name is not registrable
     - Reliabilty issue
  2. Domain name is open registrable

     $\rightarrow$ Hijacking possibility[5]

[5] The .io Error – Taking Control of All .io Domains With a Targeted Registration https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/

DNS Basics
Delegations Issues

- NS record points to an IP address
    - Not reachable and not valid
    - Reliabilty issue (e.g. other name server are not reachable/overloaded)

- NS record contains a typo in the domain name
    1. Domain name is not registrable
        - Reliabilty issue
    2. Domain name is open registrable

        $\rightarrow$ Hijacking possibility[5]
- NS record points to a host without a DNS service or without authoritative information on the zone (lame delegation)

[5] The .io Error – Taking Control of All .io Domains With a Targeted Registration https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/

DNS Basics
Delegations Issues

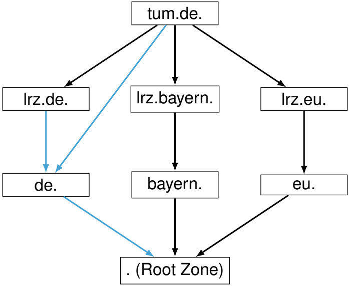
- NS record points to an IP address
  - Not reachable and not valid
  - Reliabilty issue (e.g. other name server are not reachable/overloaded)

- NS record contains a typo in the domain name
  1. Domain name is not registrable
     - Reliabilty issue
  2. Domain name is open registrable
     → Hijacking possibility[5]
- NS record points to a host without a DNS service or without authoritative information on the zone (lame delegation)
  - E.g. `net.in.tum.de 3600 IN NS ns1.google.com`
  - `ns1.google.com` has no authoritative information on `net.in.tum.de`
  - Try `dig soa net.in.tum.de @ns1.google.com`

[5] The .io Error – Taking Control of All .io Domains With a Targeted Registration https://thehackerblog.com/the-io-error-taking-control-of-all-io-domains-with-a-targeted-registration/

Trusted Computing Base (TCB)

- A set of all components critical to a systems security
- First defined in the context of the kernel and trusted processes by John Rushby
- Ramasubramanian et al. defines[6]:
  The nameservers in the delegation graph of a domain name form the trusted computing base(TCB) of that name.
- More general: A zones TCB consists of all zones in the delegation graph

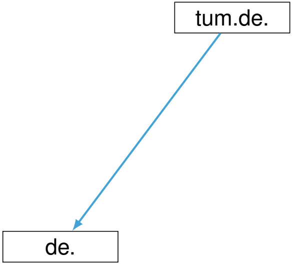


6 Ramasubramanian et al., Perils of Transitive Trust in the Domain Name System ACM IMC 2005

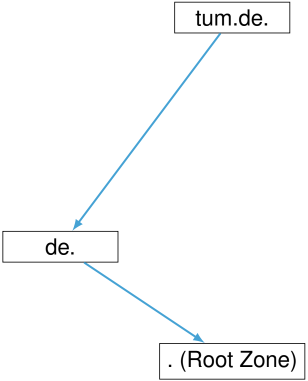
DNS Basics
Example tum.de.

tum.de.

DNS Basics
Example tum.de.

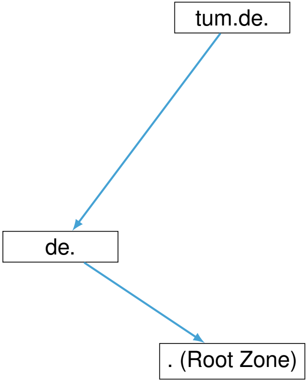


DNS Basics
Example tum.de.



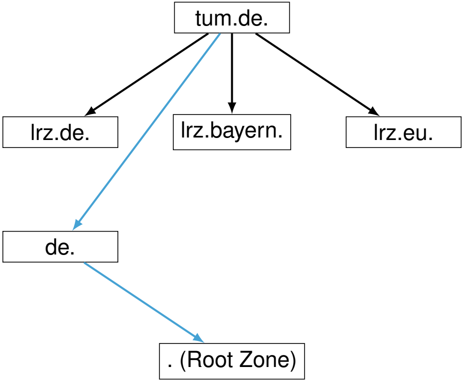
DNS Basics
Example tum.de.

tum.de.	86400	IN	NS	dns1.lrz.de.
tum.de.	86400	IN	NS	dns2.lrz.bayern.
tum.de.	86400	IN	NS	dns3.lrz.eu.



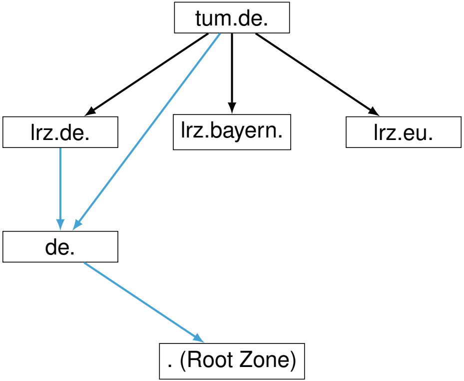
DNS Basics
Example tum.de.

tum.de.	86400	IN	NS	dns1.lrz.de.
tum.de.	86400	IN	NS	dns2.lrz.bayern.
tum.de.	86400	IN	NS	dns3.lrz.eu.



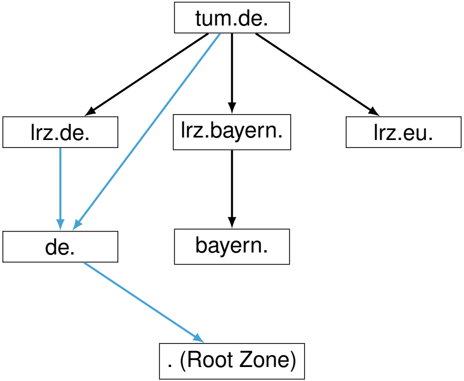
DNS Basics
Example tum.de.

tum.de.	86400	IN	NS	dns1.lrz.de.
tum.de.	86400	IN	NS	dns2.lrz.bayern.
tum.de.	86400	IN	NS	dns3.lrz.eu.



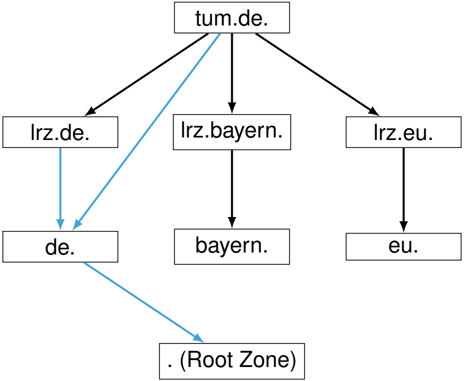
DNS Basics
Example tum.de.

tum.de.	86400	IN	NS	dns1.lrz.de.
tum.de.	86400	IN	NS	dns2.lrz.bayern.
tum.de.	86400	IN	NS	dns3.lrz.eu.



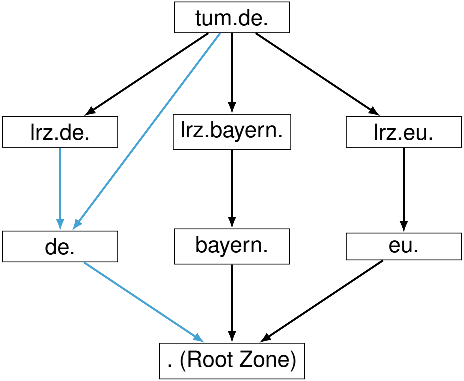
DNS Basics
Example tum.de.

tum.de.	86400	IN	NS	dns1.lrz.de.
tum.de.	86400	IN	NS	dns2.lrz.bayern.
tum.de.	86400	IN	NS	dns3.lrz.eu.



DNS Basics
Example tum.de.

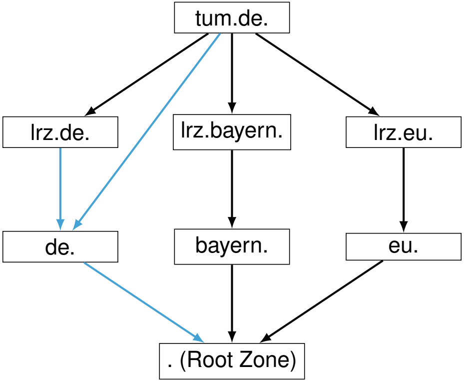
tum.de.	86400	IN	NS	dns1.lrz.de.
tum.de.	86400	IN	NS	dns2.lrz.bayern.
tum.de.	86400	IN	NS	dns3.lrz.eu.



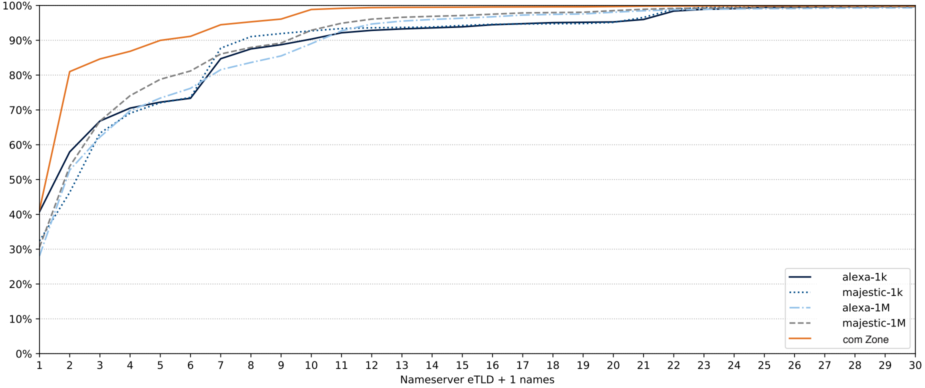
DNS Basics
Example tum.de.

tum.de.	86400	IN	NS	dns1.lrz.de.
tum.de.	86400	IN	NS	dns2.lrz.bayern.
tum.de.	86400	IN	NS	dns3.lrz.eu.

.bayern setup is more complex in reality

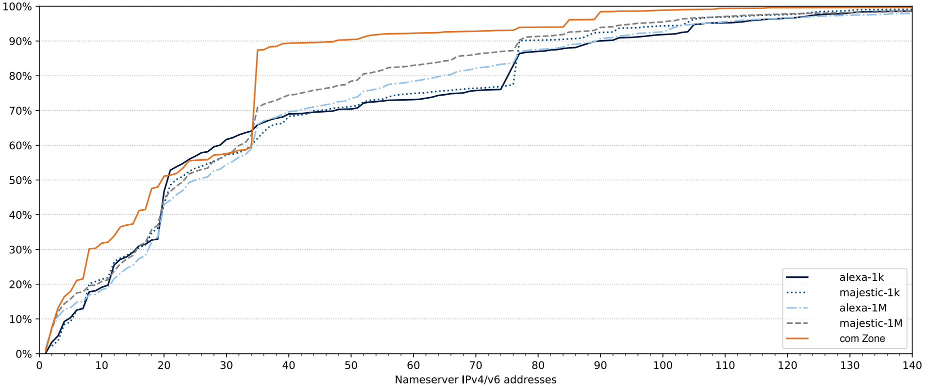


DNS Basics
TCB Statistics



- Idea: Number of eTLD + 1 label gives us an idea on the number of parties involved
- The more parties involved the higher is the attack surface
- Caveat: Some DNS provider use name server names in different eTLDs (e.g. AWS) → more eTLD + 1 names per provider
- Therefore: a lower number of eTLD + 1 names is better

DNS Basics
TCB Statistics



- Number of IP addresses per TCB is a more accurate representation for the number of hosts in the TCB (not considering anycast)
- Significant increases in the graph stem from DNS providers

DNS Basics
Name Server Resilience Requirements

RFC 8499 on zones:

```
 Authoritative information is organized into units called
ZONEs, and these zones can be automatically distributed to the
name servers which provide redundant service for the data in a zone.
```

DNS Basics
Name Server Resilience Requirements

RFC 8499 on zones:

```
Authoritative information is organized into units called
ZONEs, and these zones can be automatically distributed to the
name servers which provide redundant service for the data in a zone.
```

Name servers which provide redundant service for the data in a zone.

DNS Basics
Name Server Resilience Requirements

RFC 8499 on zones:

```
Authoritative information is organized into units called
ZONEs, and these zones can be automatically distributed to the
name servers which provide redundant service for the data in a zone.
```

Name servers which provide redundant service for the data in a zone.

RFC 2182 in 3.1:

```
Secondary servers must be placed at both topologically and
geographically dispersed locations on the Internet, to minimise the
likelihood of a single failure disabling all of them.
```

DNS Basics
Name Server Resilience Requirements

RFC 8499 on zones:

```
Authoritative information is organized into units called
ZONEs, and these zones can be automatically distributed to the
name servers which provide redundant service for the data in a zone.
```

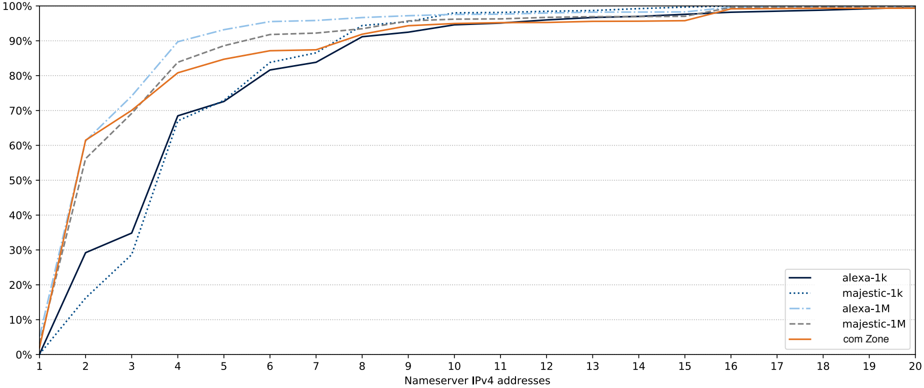
Name servers which provide redundant service for the data in a zone.

RFC 2182 in 3.1:

```
Secondary servers must be placed at both topologically and
geographically dispersed locations on the Internet, to minimise the
likelihood of a single failure disabling all of them.
```

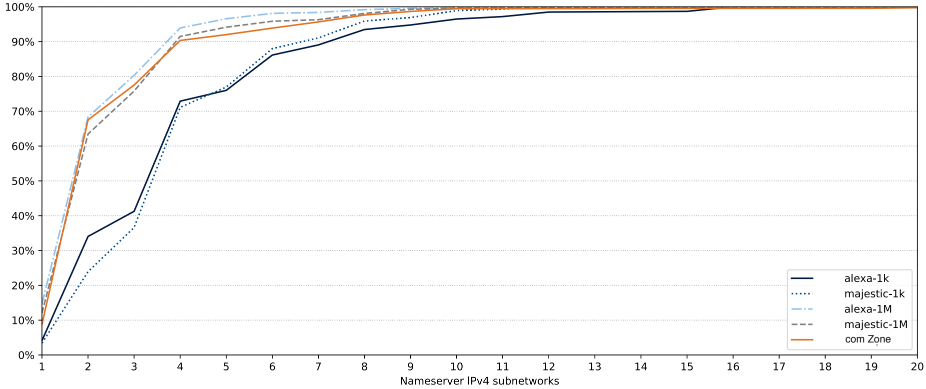
Servers must be placed at both topologically and geographically dispersed

Nameserver IPv4 addresses per zone



DNS Basics
RFC Compliance

Nameserver IPv4 /24 subnets per zone



- Most zones have two nameserver IP addresses
- Aggregated on /24 subnets for topological diversity we can find up to 10 % of non compliant zones

Domain Name System

EDNS
Transport Protocol

- Default is UDP
- DNS UDP supports messages up to 512 byte payload
- With additions such as DNSSEC and EDNS0 the boundary of 512 bytes is easily broken
- UDP-Fragmentation does not work reliable
- When it works it can be abused [1]

Fallback to TCP

- DNS standard included TCP from the beginning (optional)
- DNS Flag Day 2020 tries to force all DNS infrastructure provider to support TCP [2]
- TCP needs an extra RTT to setup connection

[1] A. Herzberg and H. Shulman, Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org, 2013 IEEE CNS
[2] DNS Flag Day 2020, `https://dnsflagday.net/2020/`

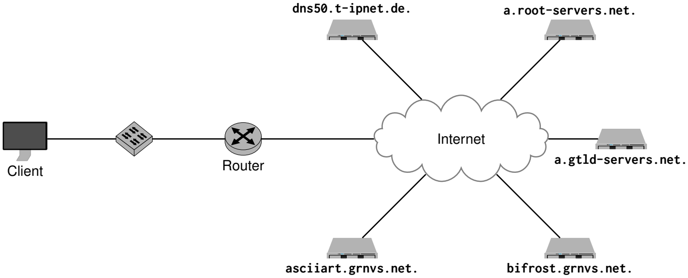
Extension mechanisms for DNS (EDNS(0))

- Defined in RFC6891
- Backwards compatible (Fallback mechanism required)
- Advertises size of maximum UDP payload size
- Extend 4 bit RCODE
- Adds new label types
- Adds the OPT pseudo-RR

Extension mechanisms for DNS (EDNS(0))

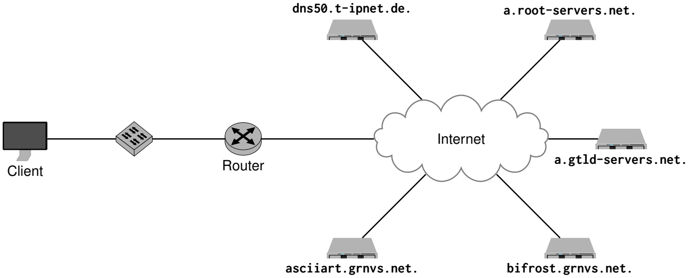
- Defined in RFC6891
- Backwards compatible (Fallback mechanism required)
- Advertises size of maximum UDP payload size
- Extend 4 bit RCODE
- Adds new label types
- Adds the OPT pseudo-RR
  - RR in the *Additional* section (maximum one is allowed)
  - Always related to the message it is in
  - Shall never be cached
  - TTL is partly used for extenden RCODE
  - RDATA contains key-value pairs

EDNS
EDNS Client Subnet (ECS)



- Defined in RFC7871 with EDNS OPTION-CODE 8
- Resolver forwards the client IP address to the authoritative name server
- Sends:

EDNS
EDNS Client Subnet (ECS)



- Defined in RFC7871 with EDNS OPTION-CODE 8
- Resolver forwards the client IP address to the authoritative name server
- Sends:
    - IP address family
    - Source prefix length (number of relevant bits in the IP address)
    - Scope prefix length (number of bits the response covers)
    - IP address

EDNS
ECS Advanced

- Recursive resolvers can forward ECS requests

    Usefull for architectures including forwarder

EDNS
ECS Advanced

- Recursive resolvers can forward ECS requests

  Usefull for architectures including forwarder

- Caching policy:
  - Source prefix length denotes the maximum cachable

EDNS
ECS Advanced

- Recursive resolvers can forward ECS requests

    Usefull for architectures including forwarder

- Caching policy:
    - Source prefix length denotes the maximum cachable
    - Source prefix length > Scope prefix length
        - Less bits are needed for the best respone
        - Cache answer for address with scope prefix length

EDNS
ECS Advanced

- Recursive resolvers can forward ECS requests

    Usefull for architectures including forwarder

- Caching policy:
    - Source prefix length denotes the maximum cachable
    - Source prefix length > Scope prefix length
        - Less bits are needed for the best respone
        - Cache answer for address with scope prefix length
    - Source prefix length < Scope prefix length
        - Source prefix length was not specific enough to select the most appropriate response
        - Resolver can retry query with longer prefix → better user experience
        - Or cache the answers for request matching the exact prefix and source prefix length

Problem

Original design of DNS does not include any security features

- Focus on scalability and distribution
- DNS does not provide a mechanism to authenticate replies
- The integrity of replies is not protected
- Client privacy is not given
  - Queries are sent in plain text
  - Queries reveal information about client behavior/traffic

DNS Security
Solutions

Protocols have been developed to solve different security issues:

- DNSSEC
  - Provides authenticity and integrity of DNS responses
- DNS Encryption
  - Protects the privacy of a client
  - Encrypts the traffic between client and resolver
  - E.g., DNS over TLS (DoT), DNS over HTTPS (DoH)
- QNAME Minimization
  - Protects the privacy of a client
  - Reduces the information sent to name servers

Domain Name System Security Extensions (DNSSEC)

- Sign DNS records
  - Public-key cryptography
  - Verified public keys of the DNS root zone (Trusted Third Party)
  - Authentication chain of trust from root zone to child zone
- Additional DNS RRs to integrate DNSSEC, e.g.,
  - RRSIG (Resource Record Signature)
  - DNSKEY (Public Key)
  - NSEC/NSEC3 (Next secure record (v3))

DNS Encryption resulted in a heated discussion in the media:

- What are possible solutions?
- Which properties do they promise?
- What are the advantages and disadvantages?
- What is **not** solved by these solutions?

Problem Statement:

- Queries in plain text reveal user behavior and accessed services
- Nearly everything in the Internet relies on DNS
- Intercepting client traffic enables detailed fingerprinting

DNS Encryption

Goals:

- DNS encryption only targets the communication between client and resolver
- Recursive queries from resolver to name servers are still plain text
- These queries should not contain client information
- DNS resolution itself is not altered

Assumptions:

- Resolvers can be trusted
- Resolvers are used by a large number of clients

DNS Encryption - Protocols

DNSCrypt[7]

- Development started in 2008
- Own protocol for encryption and authentication
- Supports UDP and TCP with port 443

DNS-over-TLS [3]

- Uses existing protocol TLS for encryption
- Based on TCP instead of UDP
- Uses port 853 (Critics: can be blocked)

DNS-over-HTTPS [4]

- Uses HTTPS for communication and encryption
- Based on TCP instead of UDP
- Uses port 443 (hard to block)
- Can be configured individually by applications in user space

7 https://dnscrypt.info/

DNS Security
DNS Encryption

Pros:

- Client traffic is encrypted

Cons:

- Internal DNS configurations might be overwritten

Debatable:

- DoH/DoT is faster?
  - TLS/HTTPS is fast and well studied but DNS (UDP/53) as well
- DoH/DoT prevents censorship?
  - The behavior of a resolver is unchanged
  - Probably more clients use large, international resolvers in the future
  - **But** they can censor as well or might be forced to by governments
- DoH/DoT prevents collection of your data?
  - Data can still be collected by the resolver

DNS Encryption

Encrypting DNS traffic between a client and resolver improves the privacy of clients by preventing the effectiveness of eavesdropping traffic, but:

- You still have to trust the resolver
- Data can still be collected
- Censorship is still possible
- Only eavesdropping traffic is limited

QNAME Minimization

Problem:

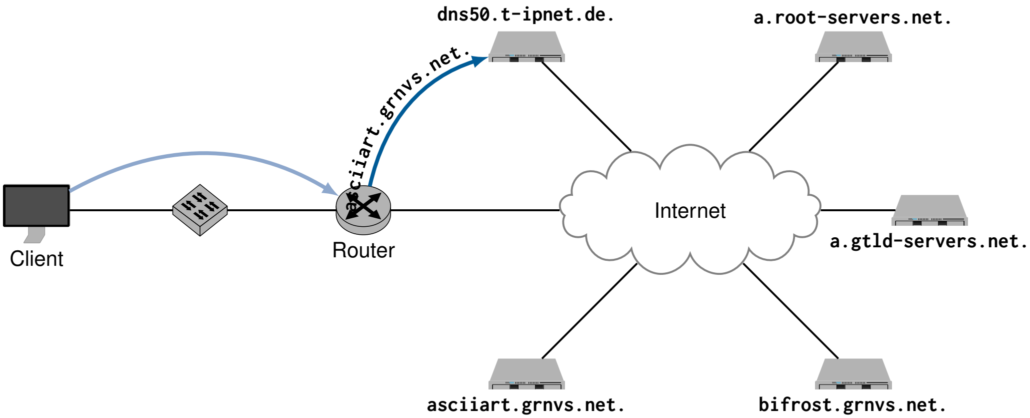
- Resolvers initially sent the complete QNAME and requested QTYPE to all name servers
- Each name server during the recursive resolution learns about the QNAME and QTYPE

Solution:

- DNS Query Name Minimisation RFC7816 [5]
- Send the exact QNAME and QTYPE only to the authoritative NS
- Only resolve the authoritative NS for each label during the recursive resolution

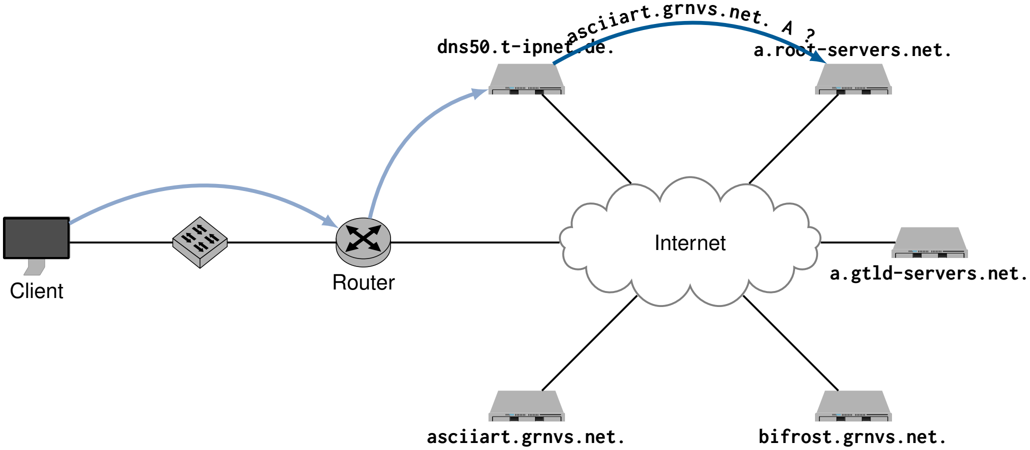
DNS Security
QNAME Minimization

Example:



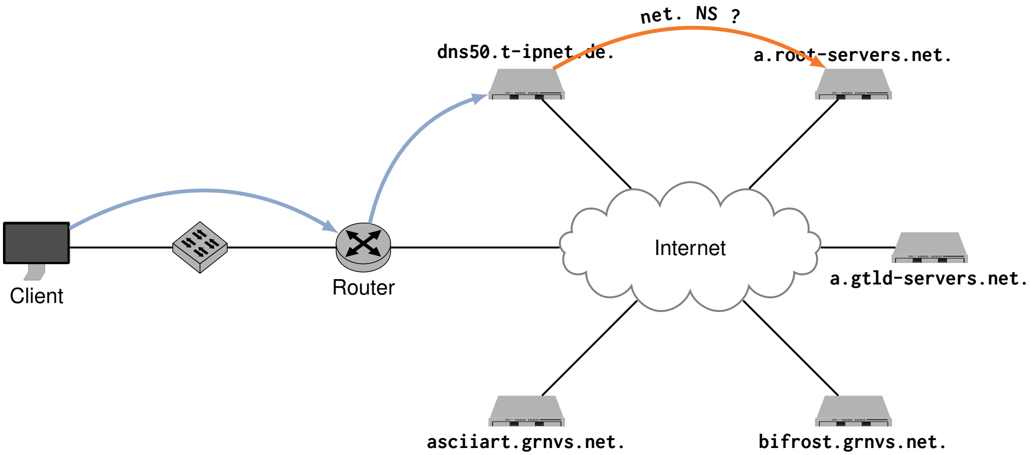
DNS Security
QNAME Minimization

Example:

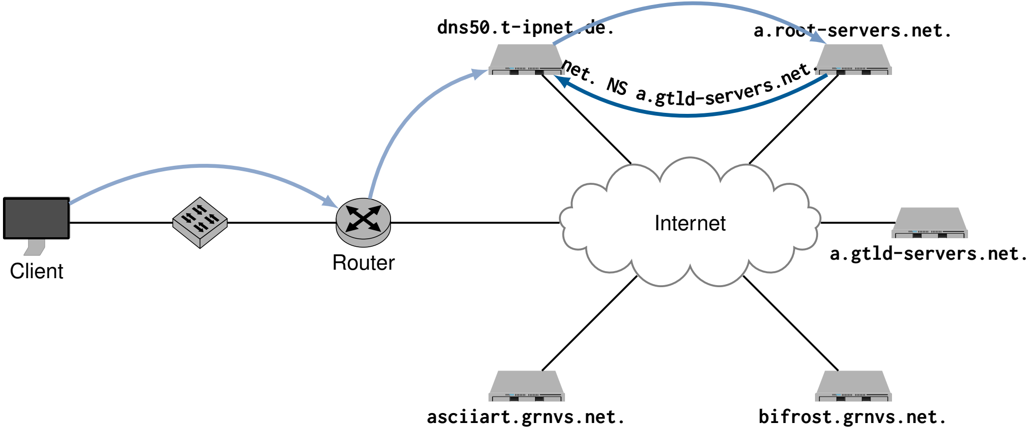


DNS Security
QNAME Minimization

Example:

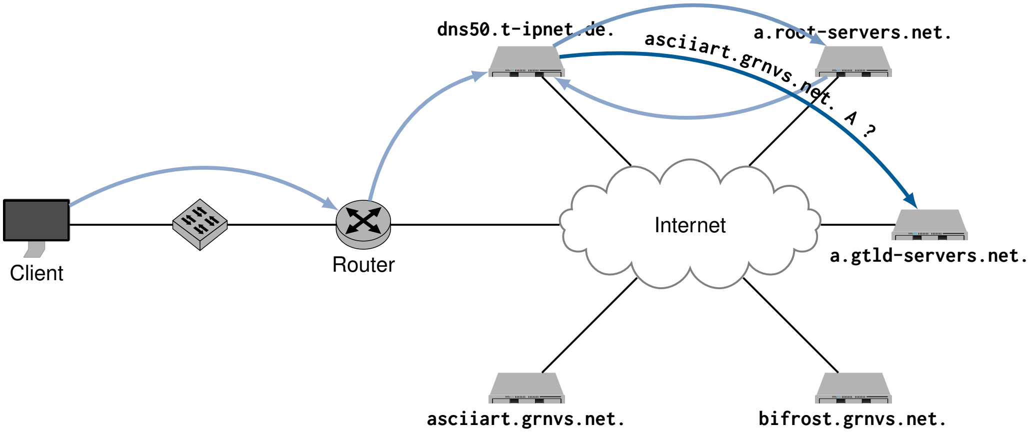


Example:


dns50.t-ipnet.de.
a.root-servers.net.
net. NS a.gtld-servers.net.
Internet
a.gtld-servers.net.
Client
Router
asciiart.grnvs.net.
bifrost.grnvs.net.

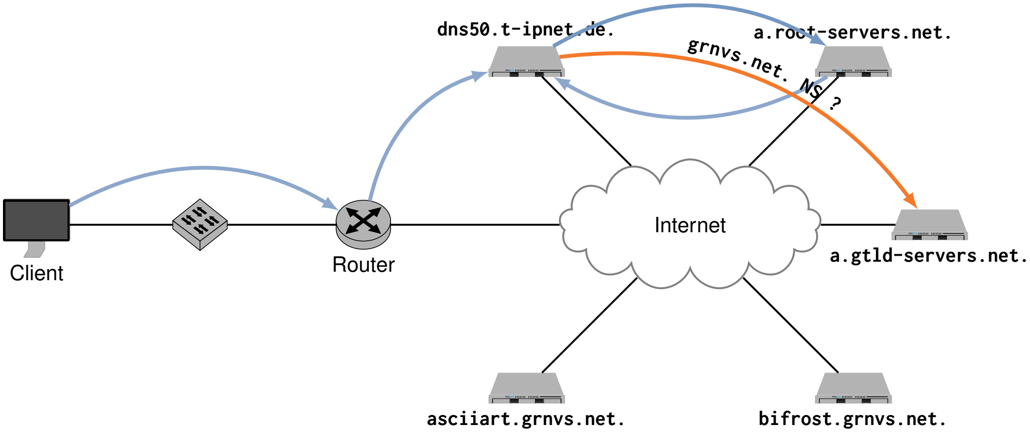
DNS Security
QNAME Minimization

Example:



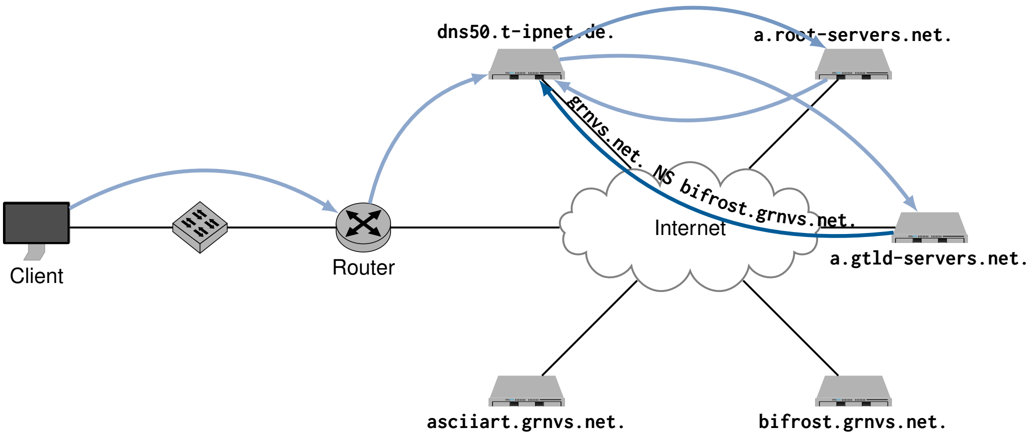
DNS Security
QNAME Minimization

Example:



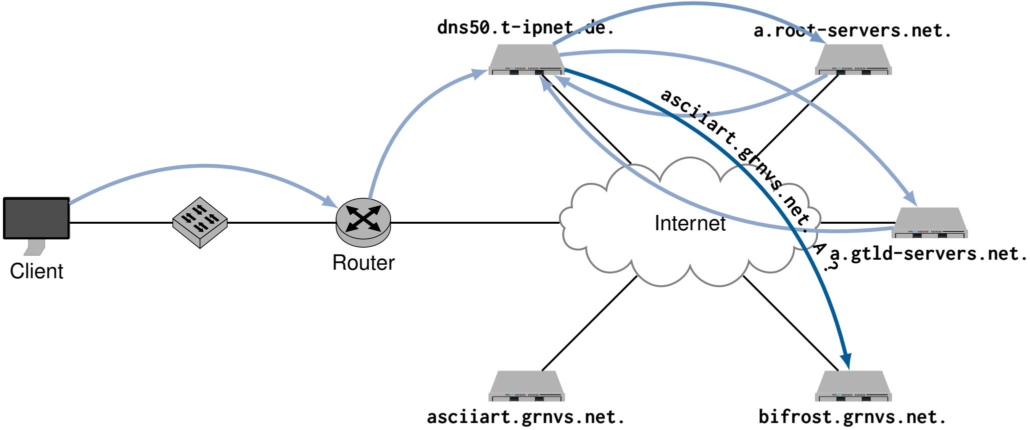
DNS Security
QNAME Minimization

Example:



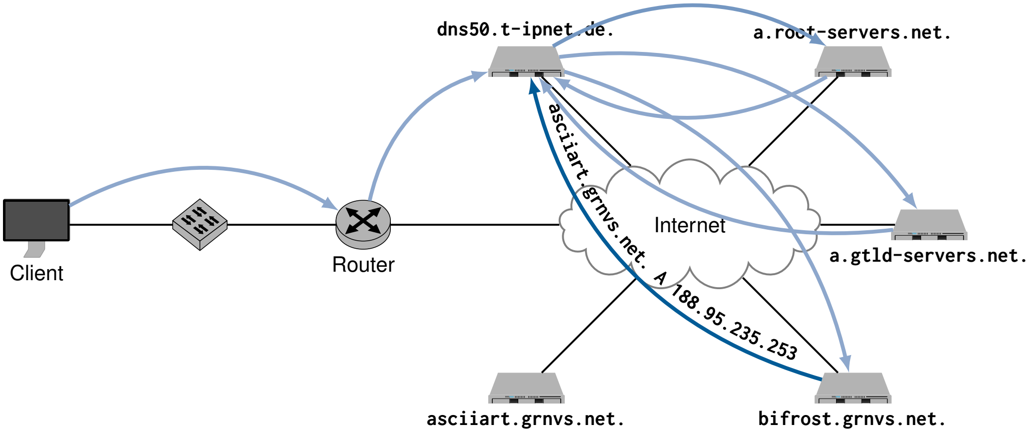
DNS Security
QNAME Minimization

Example:



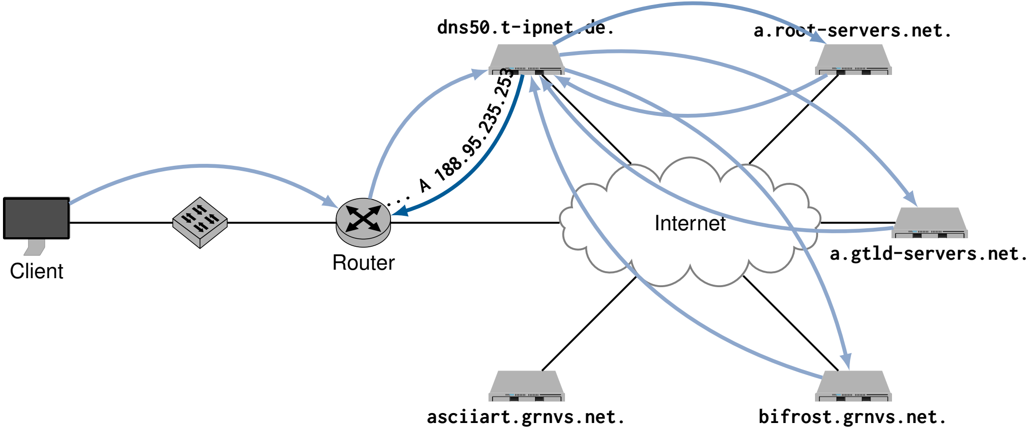
DNS Security
QNAME Minimization

Example:



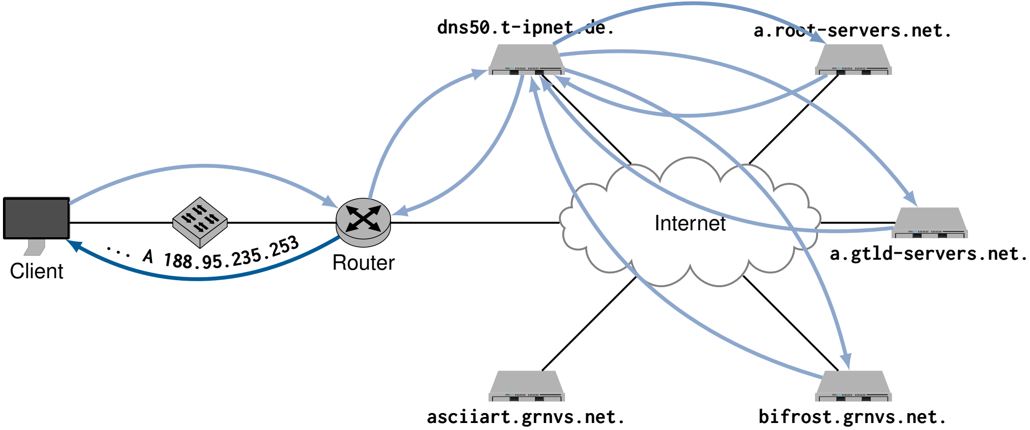
DNS Security
QNAME Minimization

Example:



DNS Security
QNAME Minimization

Example:



DNS Security
QNAME Minimization

QNAME Minimization only changes the resolver behavior and basically follows the DNS specification, but:

- Increased rate of unsuccessful queries (up to 5%[6])
  - Some NS incorrectly reply to NS queries (REFUSED)
  - → Use different QTYPE (A, AAAA)
  - Some NS incorrectly reply to emtpy labels (no data for name)
  - → Fallback to query with all labels
- Increased query load (up to 26% [6])
  - All labels have to be queried one by one
  - A NS authoritative for multiple labels could reply with most significant reply if full name is known
  - → Fallback to query with all labels when same NS is queried

→ Deployment of QNAME minimization is hindered by NS miss-configurations

→ Resolver implement algorithms with different fallback behavior

[1] P. Moackapetris, Domain Names – Concepts and Facilities, https://tools.ietf.org/html/rfc1034, 1987.

[2] P. Moackapetris, Domain Names – Implementation and Specification, https://tools.ietf.org/html/rfc1035, 1987.

[3] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for dns over transport layer security (tls)," RFC 7858, 2016.

[4] P. Hoffman and P. McManus, "Dns queries over https (doh)," RFC 8484, 2018.

[5] S. Bortzmeyer, "DNS Query Name Minimisation to Improve Privacy," RFC 7816, 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7816.txt.

[6] W. B. de Vries, Q. Scheitle, M. Müller, W. Toorop, R. Dolmans, and R. van Rijswijk-Deij, "A first look at qname minimization in the domain name system," in Passive and Active Measurement, D. Choffnes and M. Barcellos, Eds., Cham: Springer International Publishing, 2019, pp. 147–160, ISBN: 978-3-030-15986-3.