

Networking “Intro”

CPSC 1181 – O.O.

Jeremy Hilliker
Summer 2017

Overview

- Networks
 - LAN
 - Topologies
 - WAN
- Protocols
 - Layering
 - OSI 7 Layer Model
 - TCP
 - IP
 - URIs
 - URLs
 - URNs
 - DNS

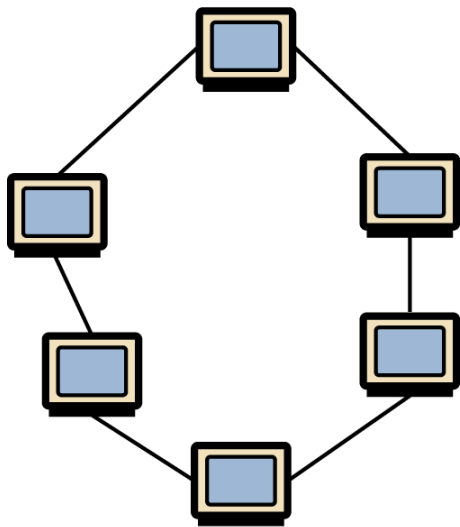
Networking

- Def'n: **Network**
 - A collection of nodes connected together using a common set of protocols so that they can
 - Send messages
 - Offer & Use service
 - Share information & resources
 - Precise meaning often depends on context
- Def'n: **Node** or **Host**
 - A reachable device on the network
- Def'n: **Protocol**
 - Agreement on how data is transferred

LAN (bonus)

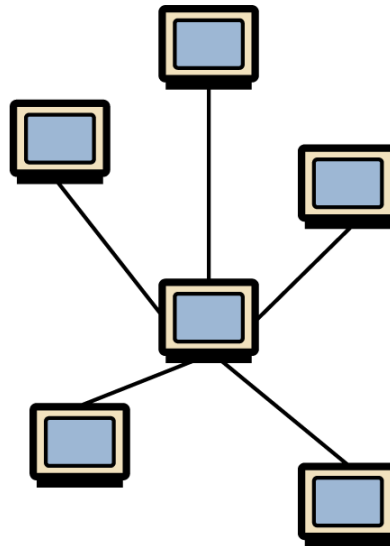
- Def'n: a **LAN** is
 - A “local area network”
 - Informally:
 - a shared network of nodes than communicate with each other over a network in the same building
 - Formally:
 - A network of nodes that can communicate with each other without using a router or gateway

Topologies (bonus)



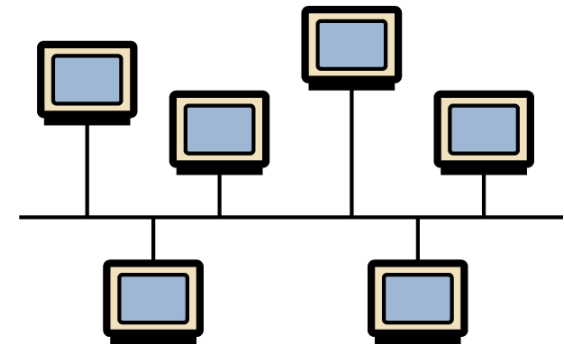
Ring topology

Token Ring (1984)
FDDI (early 80s - 1990)



Star topology

Gigabit Ethernet (1999-2008)

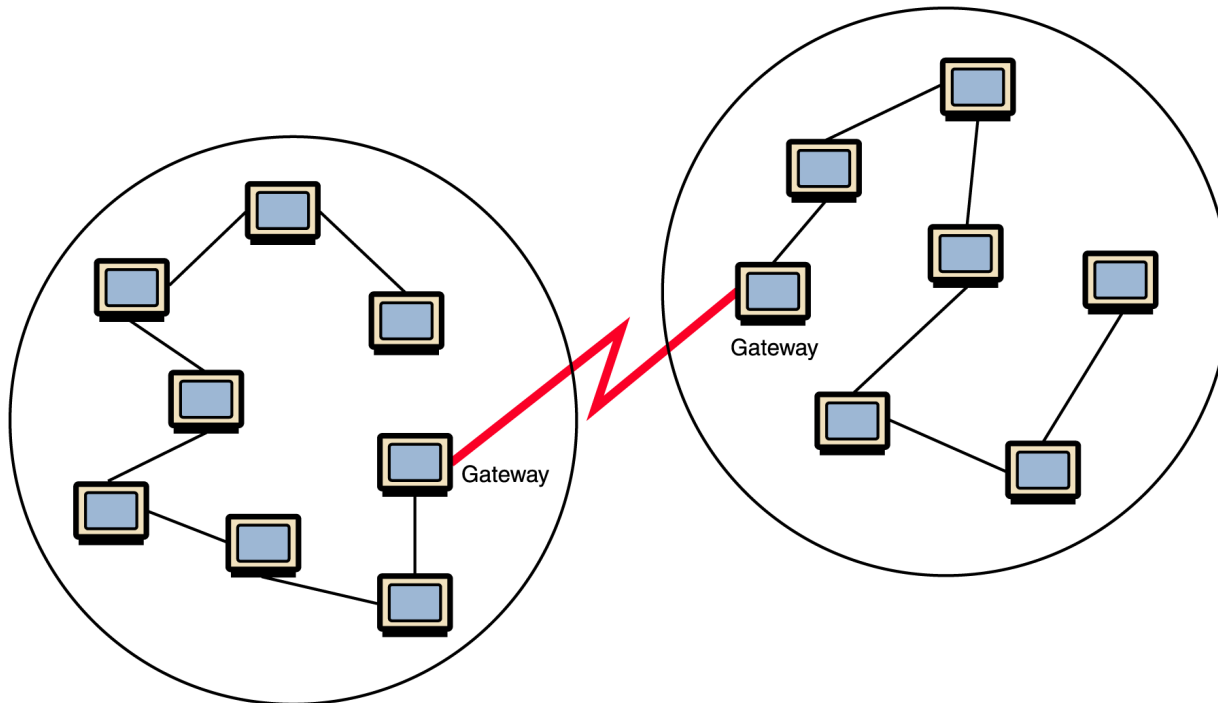


Bus topology

100Base-TX (1995)
10Base-T (1987-1990)
10Base-5 & 2 (1980s)

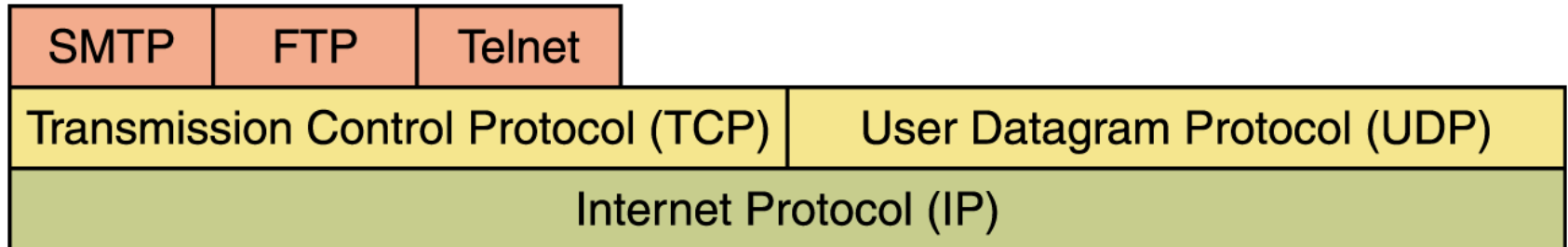
WAN (bonus)

- Def'n: a *router* / *gateway* is:
 - A device that connects different networks together.
- Def'n: a **WAN** is
 - A “wide area network”
 - A collection of networks belonging to one organization linked together (by routers)
 - Typically spans multiple buildings, or multiple units



Protocol Layering

- Sometimes referred to as the:
 - Network / Protocol stack



OSI 7 Layer Model

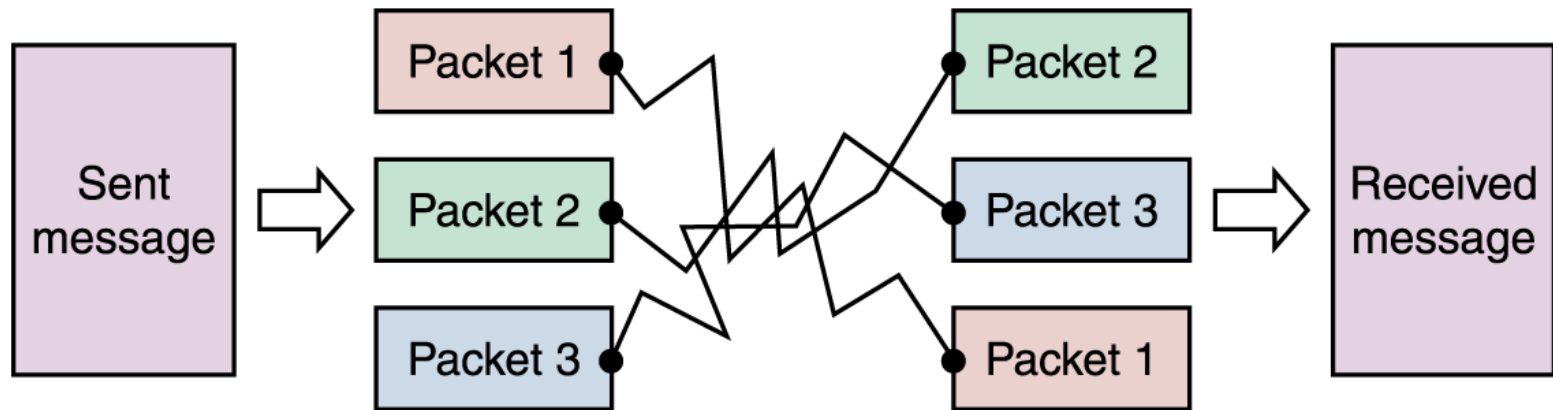
	Name	Responsibility	Eg.
All [Away]	Application	<ul style="list-style-type: none"> Resource sharing and redirection IPC, remote resource access 	<ul style="list-style-type: none"> HTTP, FTP, NNTP SIP, NTP, DNS, NFS
People [Pizza]	Presentation	<ul style="list-style-type: none"> Formats data <ul style="list-style-type: none"> Marshalling, Serialization Encryption 	<ul style="list-style-type: none"> MIME, XDR ASN.1 TLS, SSL
Seem [Sausage]	Session	<ul style="list-style-type: none"> Session establishment, maint, termination Security (login / authentication) etc. 	<ul style="list-style-type: none"> [Sockets] TCP, RTP, PPTP
To [Throw]	Transport	<ul style="list-style-type: none"> Data Transfer <ul style="list-style-type: none"> Guaranteed delivery? Exactly once? In order? Error detection? Message segmentation Message acknowledgement Traffic/flow control Multiplexing (ports) 	<ul style="list-style-type: none"> TCP UDP SCTP DCCP
Need [Not]	Network	<ul style="list-style-type: none"> Routing Datagram fragmenting 	<ul style="list-style-type: none"> IP, IPsec, ICMP IGMP, MPLS, ARP
Data [Do]	Data-link	<ul style="list-style-type: none"> Control access to transmission medium Node-to-node transfer Low level error correction 	<ul style="list-style-type: none"> MPLS, ARP SLIP, PPP Ethernet, 802.11
Promptly / Processing [Please]	Physical	<ul style="list-style-type: none"> Encoding bits Sends frames and control signals 	<ul style="list-style-type: none"> Ethernet, 802.11 FDDI Bluetooth

How?

- Application defines some application layer protocol
- Application establishes a connection* to remote host by specifying:
 - Mode of transport
 - Presentation, session, and transport choices
 - Destination (host, port)
- Application then sends Data
 - The lower layers take this, do their jobs, and send the data
 - Receiving end does the same in opposite order

TCP (Session, Transport)

- Transmission Control Protocol
- Provides:
 - Message segmentation
 - Turns streams into packets (and vice versa)
 - Guaranteed, in-order, exactly once delivery
 - Acknowledgement, error detection, retransmission, sequencing
 - Multiplexing
 - Ports (allows multiple services and connections on one host)
 - Flow control
 - Transmission windows (how much data can be “in flight”)
 - Congestion control



Message is divided
into packets

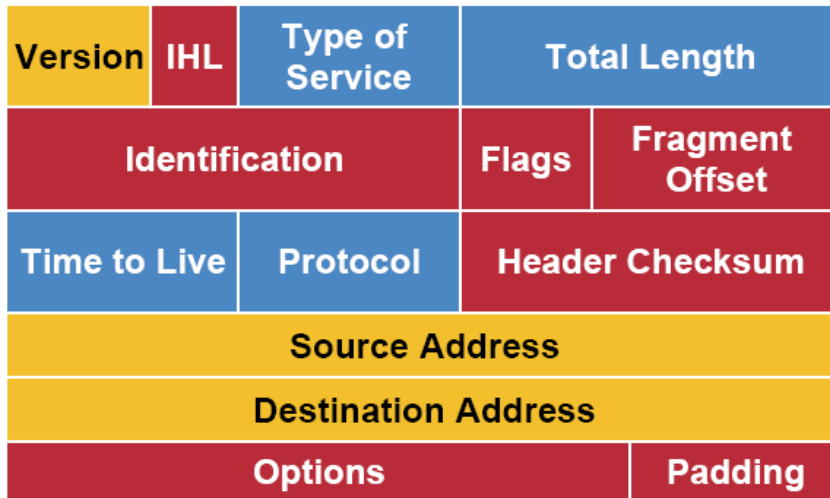
Packets are sent over the Internet
by the most expedient route

Packets are reordered
and then reassembled

IP (Network Layer)

- Internet Protocol
- Provides:
 - Datagram fragmenting
 - If input is too big, break it into pieces
 - Avoid this because IP is not reliable
 - Routing
 - Getting the datagram across the network(s) to the right destination node
 - An address for every node
 - IP4:
 - “Dotted Quad”: 127.0.0.1
 - 4 parts, each between 0 and 255 (4 x 8 bits = 32 bits)
 - IP6:
 - 8 groups of 4 hex digits, delimited by colons
 - **2001:0db8:0000:0042:0000:8a2e:0370:7334** (64 + 64 bits)

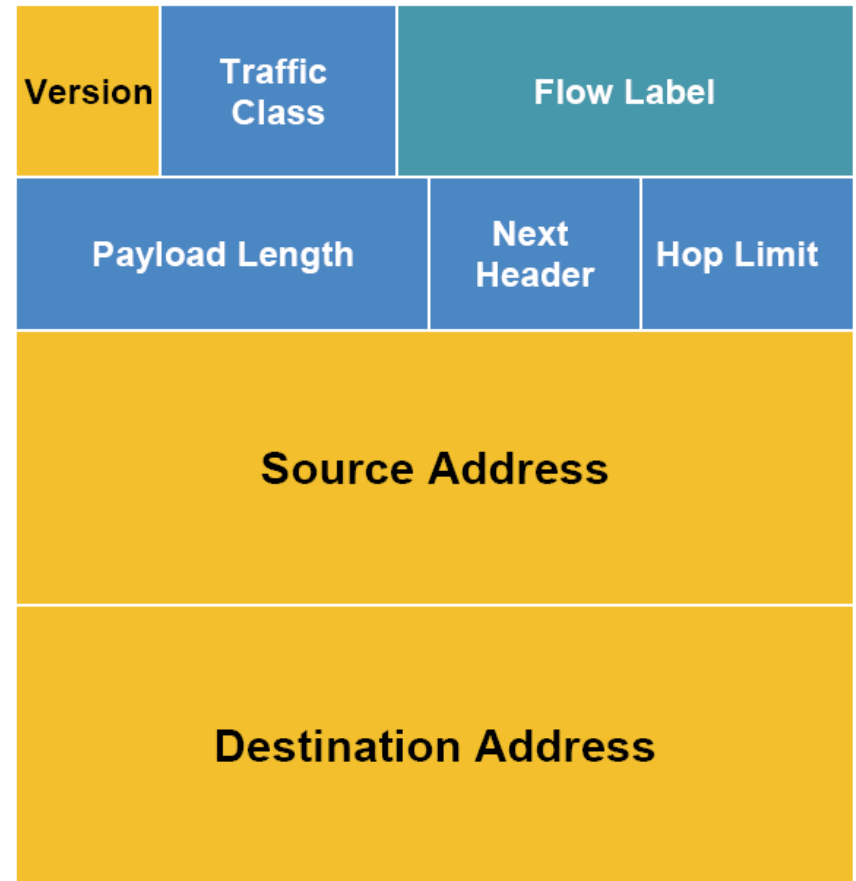
IPv4 Header



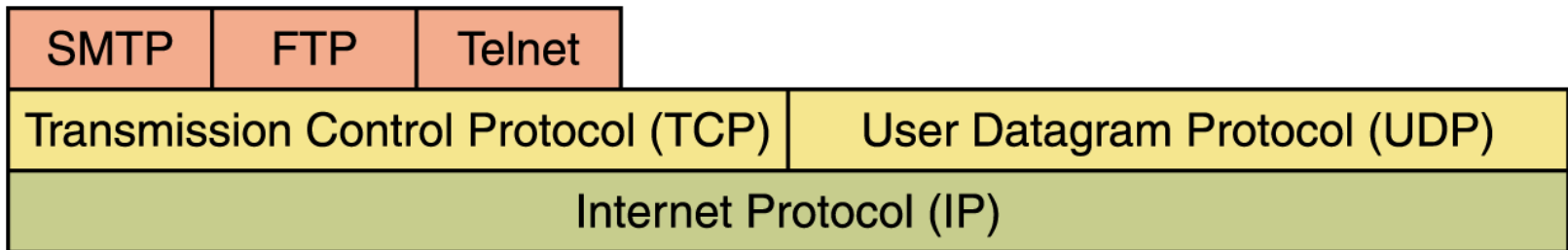
Legend

- Field's Name Kept from IPv4 to IPv6
- Fields Not Kept in IPv6
- Name and Position Changed in IPv6
- New Field in IPv6

IPv6 Header



Recall: Layering



URI

- Def'n: a **URI** is
 - A Uniform Resource Identifier
 - Identifies a resource (high-level abstraction)

`mailto:someone@example.com`

- Def'n: a **URN** is
 - A Uniform Resource Name
 - The “name” of a resource

urn:example:mammal:monotreme:echidna

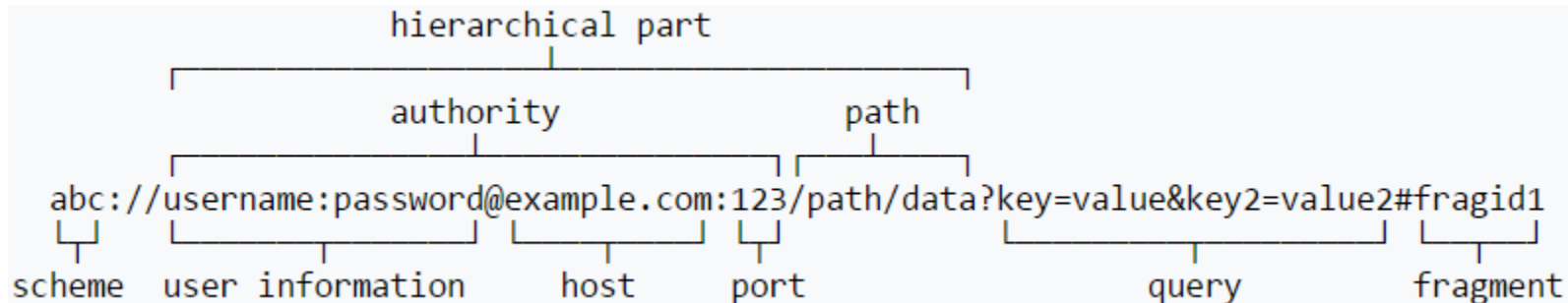
scheme path

urn:isbn:0451450523
doi:10.1000/1

- Def'n: a **URL** is
 - A Uniform Resource Locator
 - The “location” of a resource

`scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]`

URL

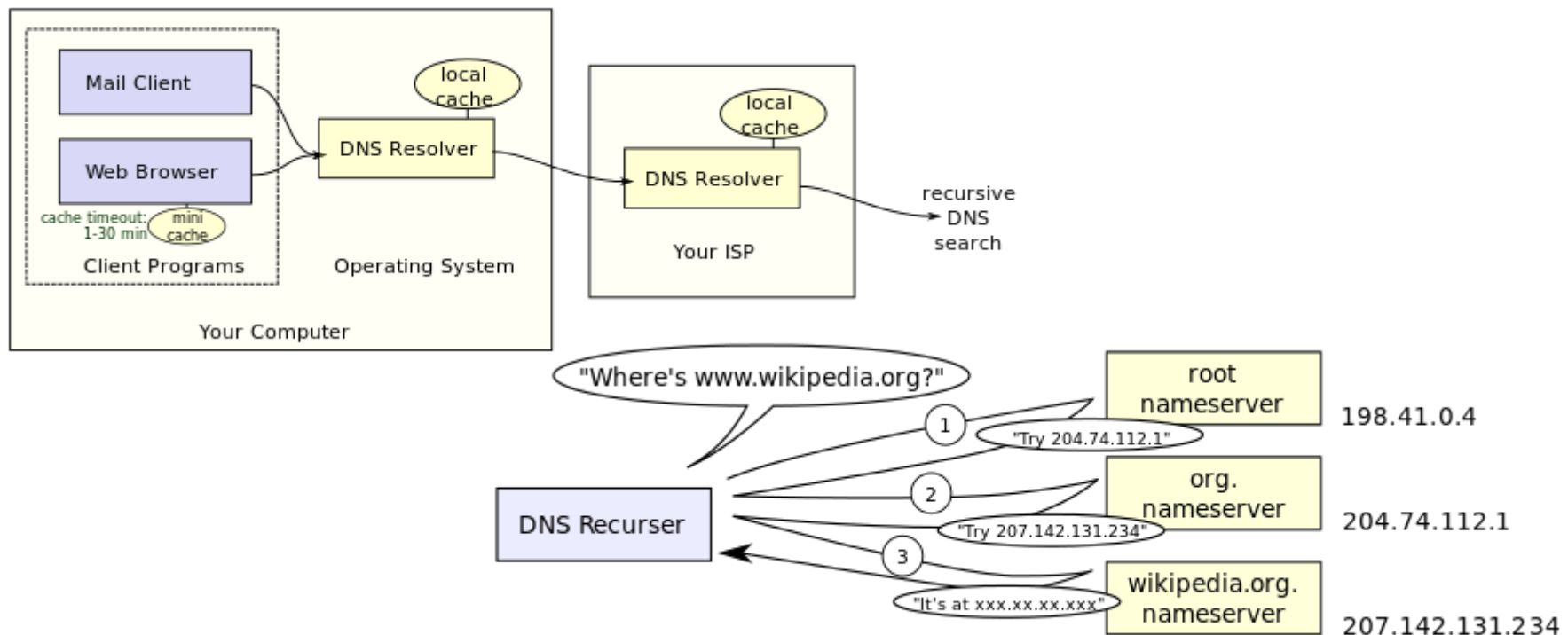


- Read:
 - Scheme (indicates application protocol)
 - User info (usually omitted – not relevant)
 - Host (right to left)
 - Port (usually omitted – uses default)
 - Path (left to right)
 - Query
 - Fragment

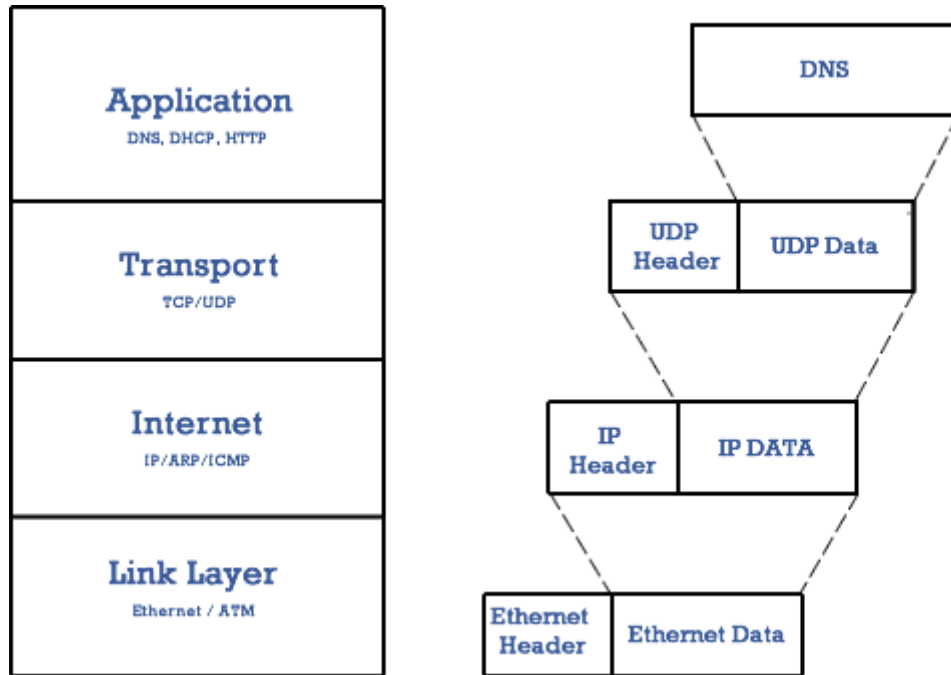
- http://
- swing.langara.bc.ca
- /pls/prod/hzgkfcls.P_Sel_Crse_Search
- ?term=201710

Protocol Example: DNS

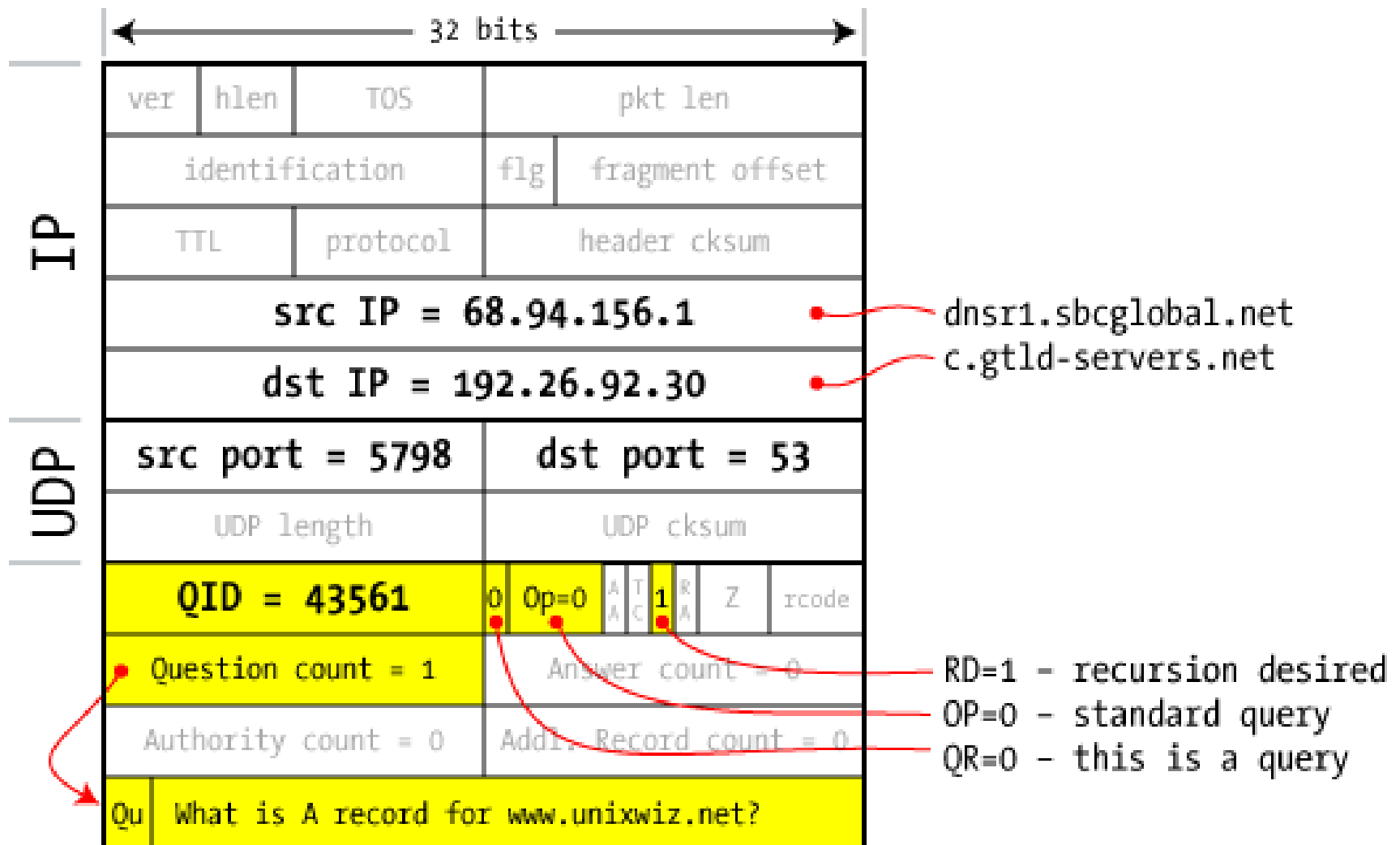
- Def'n: DNS is
 - The Domain Name System
 - A way to convert host **names** into IP addresses
- DNS is a hierarchical distributed system



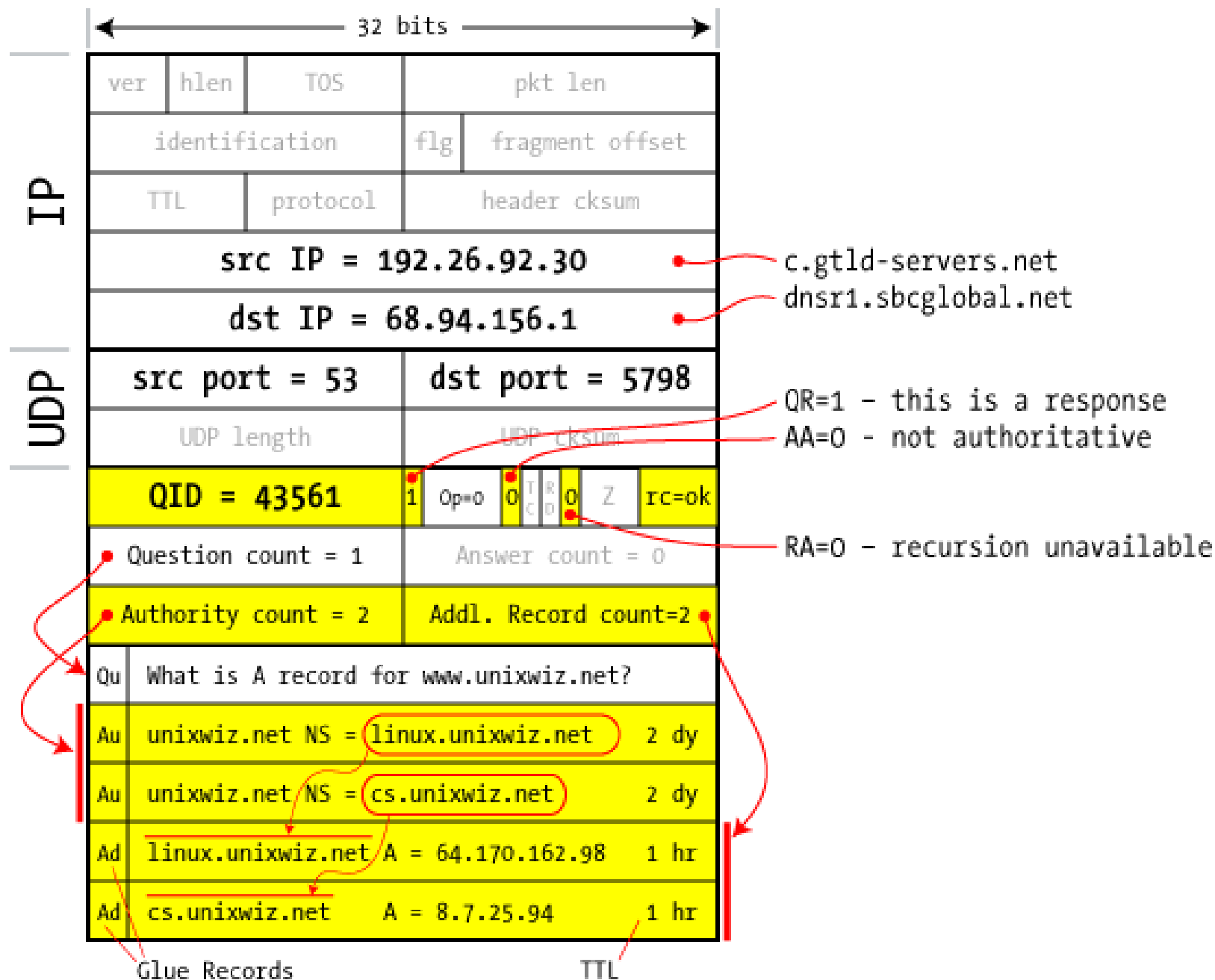
SMTP	FTP	Telnet	
Transmission Control Protocol (TCP)			User Datagram Protocol (UDP)
Internet Protocol (IP)			

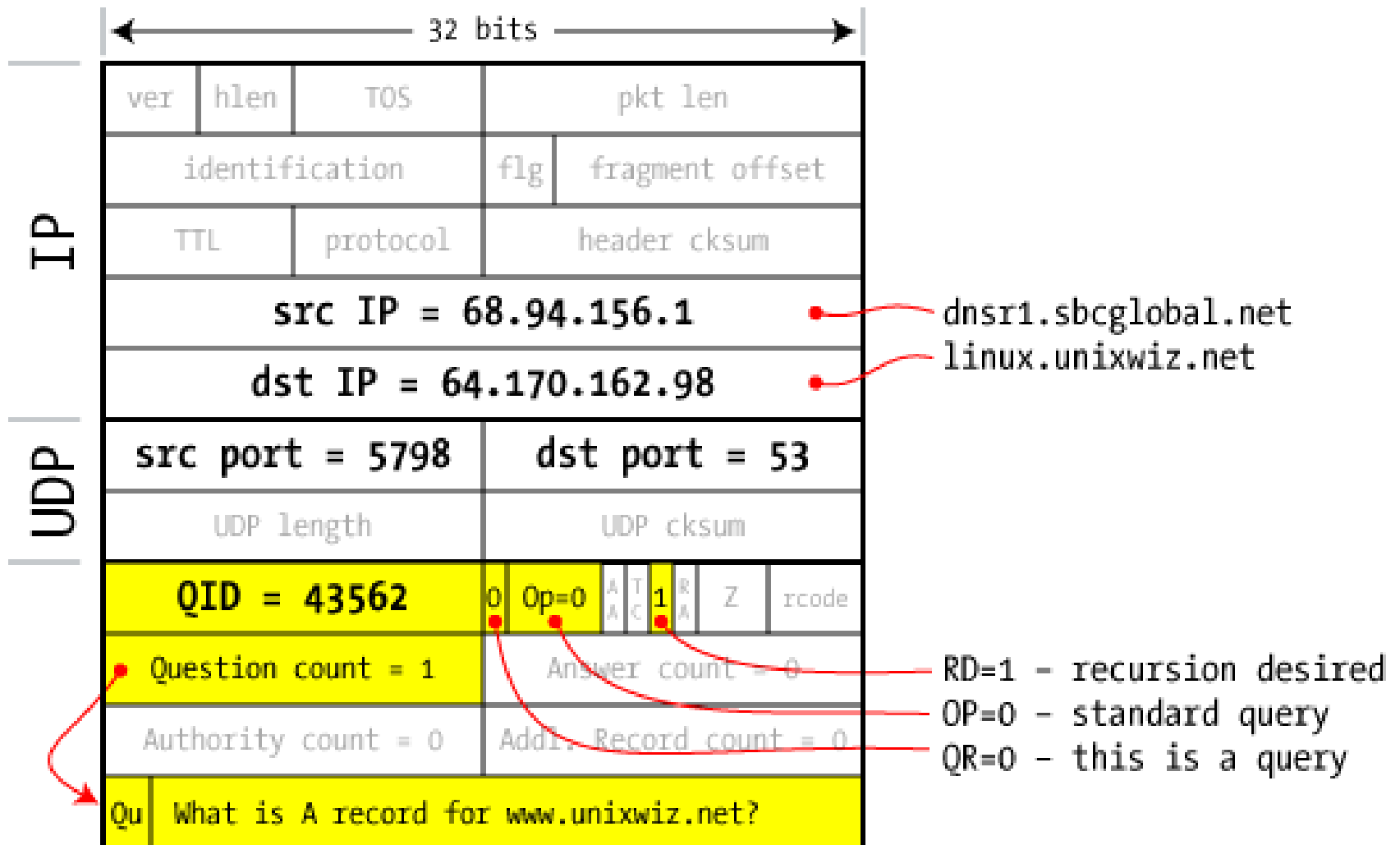


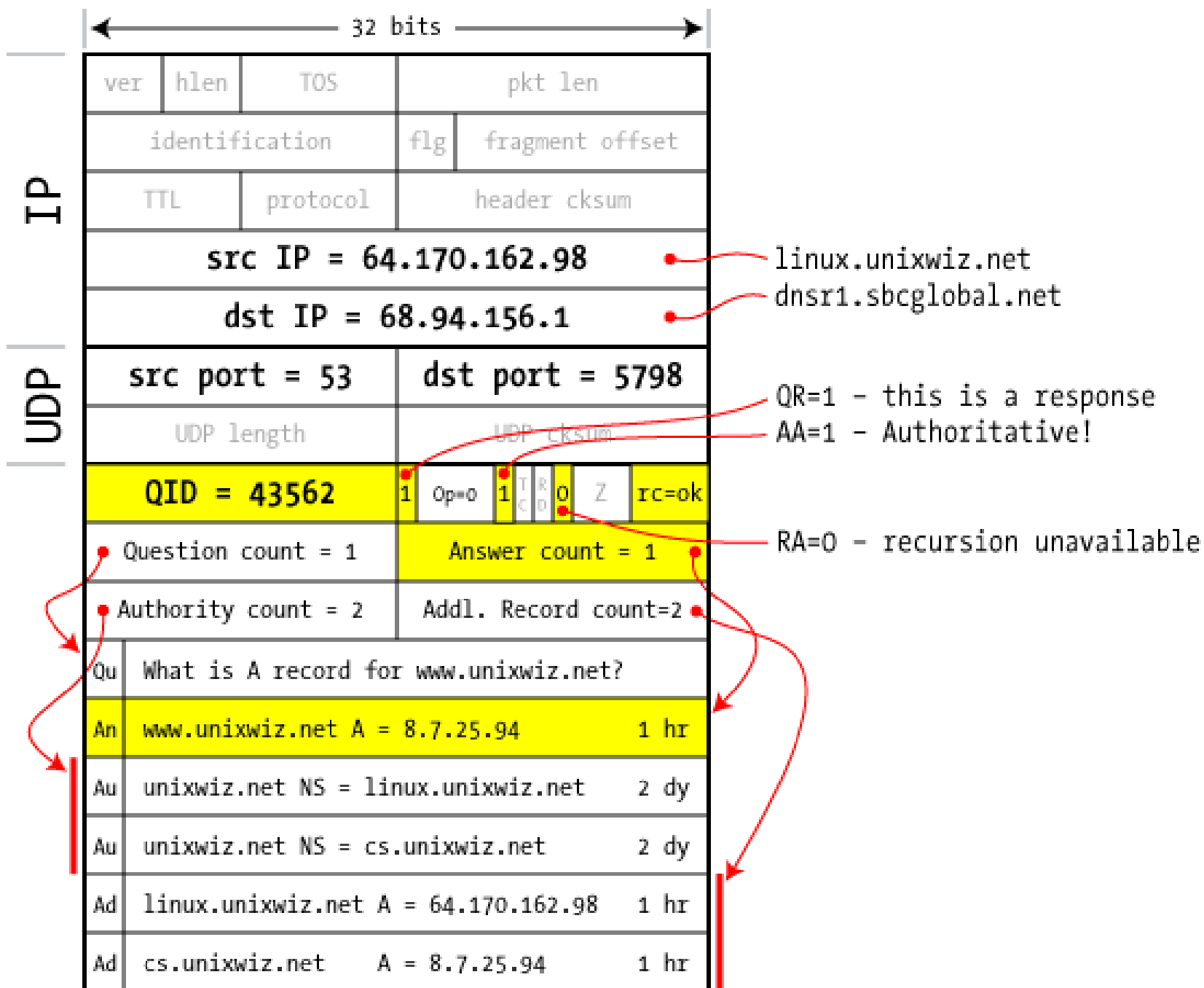
Edgis Sec



<http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html>







name	class	type	data		time to live	
gmail.com	IN	A	172.217.17.101		299s	(00:04:59)
gmail.com	IN	AAAA	2a00:1450:400e:806::2005		299s	(00:04:59)
gmail.com	IN	NS	ns4.google.com		86399s	(23:59:59)
gmail.com	IN	NS	ns1.google.com		86399s	(23:59:59)
gmail.com	IN	MX	preference:	5	3599s	(00:59:59)
			exchange:	gmail-smtp-in.l.google.com		
gmail.com	IN	MX	preference:	10	3599s	(00:59:59)
			exchange:	alt1.gmail-smtp-in.l.google.com		
gmail.com	IN	NS	ns3.google.com		86399s	(23:59:59)
gmail.com	IN	MX	preference:	30	3599s	(00:59:59)
			exchange:	alt3.gmail-smtp-in.l.google.com		
gmail.com	IN	TXT	v=spf1 redirect=_spf.google.com		299s	(00:04:59)
gmail.com	IN	SOA	server:	ns4.google.com	59s	(00:00:59)
			email:	dns-admin@google.com		
			serial:	150581370		
			refresh:	900		
			retry:	900		
			expire:	1800		
			minimum ttl:	60		
gmail.com	IN	MX	preference:	40	3599s	(00:59:59)
			exchange:	alt4.gmail-smtp-in.l.google.com		
gmail.com	IN	NS	ns2.google.com		86399s	(23:59:59)
gmail.com	IN	MX	preference:	20	3599s	(00:59:59)
			exchange:	alt2.gmail-smtp-in.l.google.com		

Recap

- Networks
 - LAN
 - Topologies
 - WAN
- Protocols
 - Layering
 - OSI 7 Layer Model
 - TCP
 - IP
 - URIs
 - URLs
 - URNs
 - DNS