KUNGL TEKNISKA HÖGSKOLAN
Institutionen för teleinformatik
CCSlab

2G1305 Internetworking/Internetteknik
Winter 2002, Period 3

Module 6: IPSec, Firewalls, and IPv6

Lecture notes of G. Q. Maguire Jr.

Based on lecture notes by Frank Reichert and *IPv6: The New Internet Protocol* by Christian Huitema.

Last modified: 2002.02.09:09:10

Lecture 6: Outline

- IPSec
- Firewalls
- IPv6

IPsec

IPsec in three parts:

- encapsulating security payload (ESP) defines encryption or IP payloads,
- authentication header (AH) defines authentication method, and
- the IP security association key management protocol (ISAKMP) manages the exchange of secret keys between senders and recipients of ESP or AH packets.

IPSec working group
http://www.ietf.org/html.charters/ipsec-charter.html

IPSec resources page
http://web.mit.edu/tytso/www/ipsec

IPSec vendor survey
http://web.mit.edu/tytso/www/ipsec/companies.html

IP Security document roadmap (*RFC 2411*)

OpenBSD's version of IPSec
http://www.openbsd.org/cgi-bin/man.cgi?query=ipsec

ESP packet

Consists of:

- a control header - contains a Security Parameters Index (SPI) and a sequence number field (the SPI + destination IP address unqiuely identifies the Security Association (SA)).
- a data payload - encrypted version of the user's original packet. It may also contain control information needed by the cryptographic algorithms (for example DES needs an initialization vector (IV)).
- an optional authentication trailer - contains an Integrity Check Value (ICV) - which is used to validate the authenticity of the packet.

ESP could use any one of several algorithms: DES, Triple DES, …

For further information see:

RFC2406:IP Encapsulating Security Payload (ESP)

AH header

For authentication purposes only contains:

- an SPI,
- a sequence number, and
- an authentication value.

AH uses either:

- Message Digest 5 (MD5) algorithm,
- Secure Hash Algorithm 1 (SHA-1),
- truncated HMAC (hashed message authentication code), or
- …

For further information see:

- IP Authentication Header (*RFC 2402*)

ISAKMP

ISAKMP is based on the Diffie-Hellman key exchange protocol; it assumes the identities of the two parties are known.

Using ISAKMP you can:

- control the level of trust in the keys,
- force SPIs to be changed at an appropriate frequency,
- identify keyholders via digital certificates
  [requires using a certificate authority (CA)]

For further information see:

- Internet Security Association and Key Management Protocol (ISAKMP) (*RFC 2408*)
- The Internet IP Security Domain of Interpretation for ISAKMP (*RFC 2407*)
- The OAKLEY Key Determination Protocol (*RFC 2412*)
- The Internet Key Exchange (IKE) (*RFC 2409*)

Where can you run ISAKMP?

Mode	Where it runs	Payload
Transport	end-systems	payload data follows the normal IP header
Tunnelling	internetworking device: e.g., router, firewall, or VPN gateway	• end-user's entire packet-IP headers and all-placed within another packet with ESP or AH fields [thus it is encapsulated in another packet] • can hide the original source and destination address information

Security Protocol abbrevs.

- Generic Security Services App. Programming Interface (GSS-API)
- Secured Socket Layer (SSL)
- Internet Protocol Security Protocol (IPSEC)
- Privacy-Enhanced Electronic Mail (PEM)
- Remote Authentication Dial-In User Services (RADIUS)
- Secured Electronic Transaction (SET)
- Secured HyperText Transport Protocol (S-HTTP)
- Pretty Good Privacy (PGP)

GSS-API

Generic Security Services Application Programming Interface (GSS-API)

- provides an abstract interface which provides security services for use in distributed applications
- but isolates callers from specific security mechanisms and implementations.

GSS-API peers establish a common security mechanism for security context establishment either through administrative action, or through negotiation.

GSS-API is specified in:

- J. Linn, "Generic Security Service API", *RFC 1508*, September 1993.
- J. Wray, "Generic Security Service API : C-bindings", *RFC 1509* September 1993.

Security Organizations and Companies

Computer Emergency Response Team (*CERT* ®) Coordination Center

U. S. National Institute of Standards and Technology, Computer Security Division

Forum of Incident Response and Security Teams (*FIRST*)

Swedish Defense Material Administration, Electronics Systems Directorate

http://www.safenet-inc.com/

http://www.netsys.com/ - UNIX and Internet security

…

Firewalls

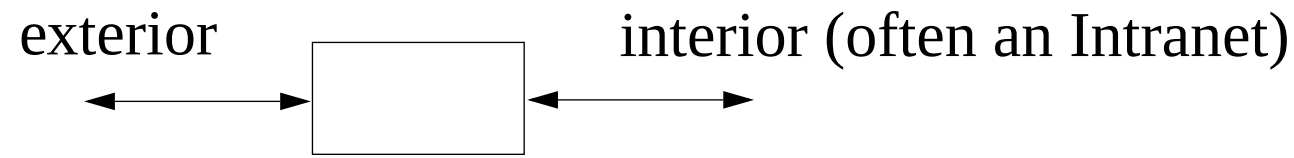


Figure 63: Firewall an internet gateway

The firewall can provide packet by packet filtering of packets coming into the intranet or leaving the intranet. The firewall can decide which packets should be forwarded based on source, destination addresses, and port using an explicitly defined **policy**.

Linux firewall

For example, for the software firewall used in Linux systems called “ipfwadm”:

- all ports are typically closed for inbound traffic,
- all outbound traffic is “IP masqueraded”, i.e., appears to come from the gateway machine; and
- For bi-directional services required by the users, “holes” may be punched through the firewall - these holes can reroute traffic to/from particular ports:
  - to specific users or
  - the most recent workstation to request a service.

Firewall Design

apply basics of security:

- least privilege:
  - don’t make hosts do more than they have to (implies: specialize servers)
  - use minimum privileges for the task in hand
- fail safe
  - even if things break it should not leave anything open
- defence in depth
  - use several discrete barriers - don’t depend on a single firewall for all security
- weakest links
  - know the limitations of your defences - understand your weakest link

Firewalls should have sufficient performance to keep the pipes full - i.e., a firewall should not limit the amount of traffic flowing across the connection to the external network, only **what** flows across it!

Proxy Access Through A Firewall

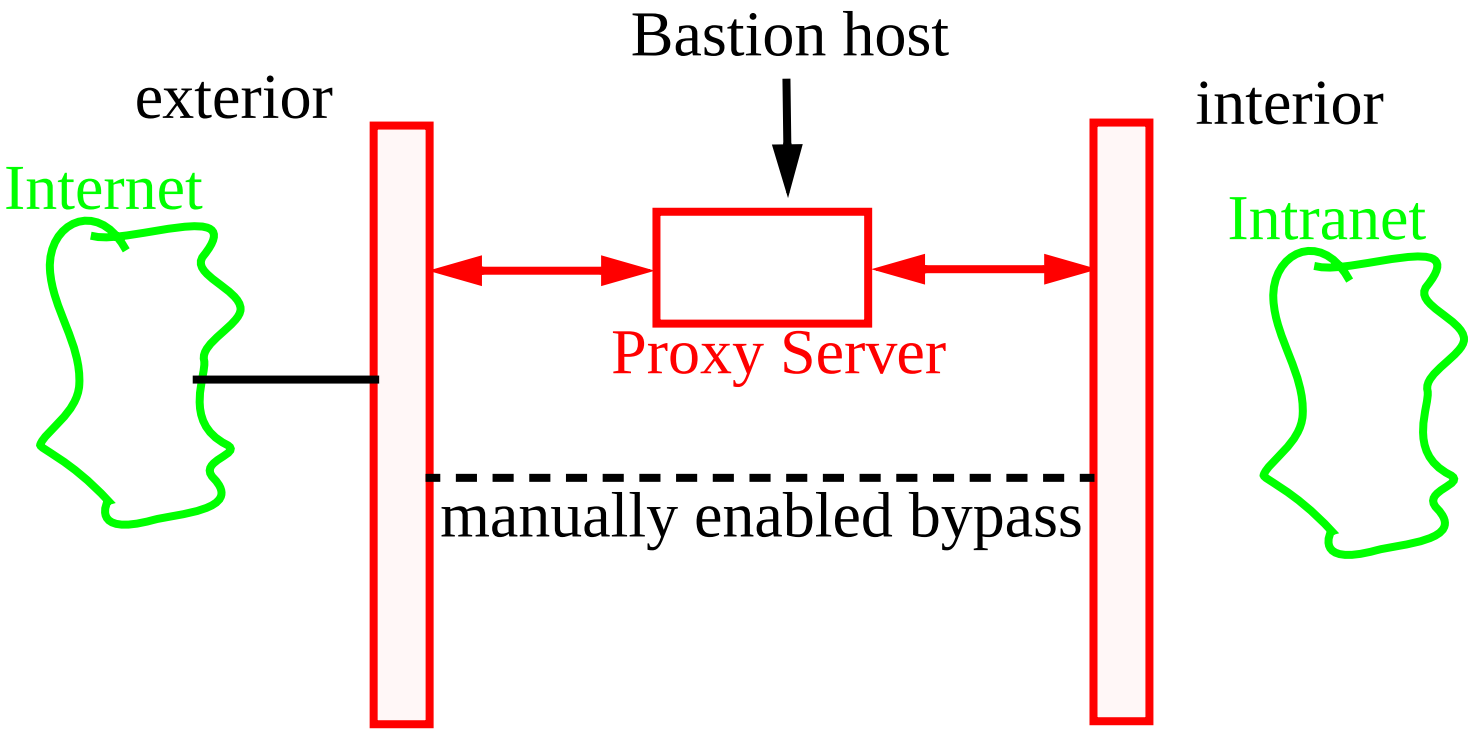


Figure 64: Firewall an internet gateway

Often you need application level proxies (i.e., they undertand details of the application protocol) -- an example is to proxy RealAudio's streaming audio.

Further information about Firewalls

Books

Firewalls and Internet Security: Repelling the Wily Hacker
Bill Cheswick and Steve Bellovin, Addison Wesley, 1994,ISBN: 0-201-63357-4

Building Internet Firewalls
D. Brent Chapman and Elizabeth Zwicky, O'Reilly, 1995,ISBN: 1-56592-124-0

Linux Routers: A Primer for Network Administrators
Tony Mancill, Prentice-Hall, 2001, ISBN 0-13-086113-8.

Firewalls mailing list *http://lists.gnac.net/firewalls/*

Those interested in firewalls, such as Computer Security Institute (CSI) at *http://www.gocsi.com/*

SOCKs

Permeo Technologies, Inc.'s SOCKS *http://www.socks.nec.com/*

In order to bridge a firewall we can use a proxy:

- the proxy will appear to be all external hosts to those within the firewall.
  - for example, If a user attached to the intranet requests a webpage, the request is sent to the proxy host where the same request is duplicated and sent to the real destination. When data is returned the proxy readdresses (with the user's intranet address) the returned data and sends it to the user.
- widely used to provide proxies for commonly used external services (such as Telnet, FTP, and HTTP).

M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones, "SOCKS Protocol V5", *RFC 1928*, April 1996.

P. McMahon, "GSS-API Authentication Method for SOCKS Version 5", Network Working Group, *RFC 1961* , June 1996

Newping

http://www.dbnet.ece.ntua.gr/~george/newping.html

based on ftp://ftp.nec.com/pub/security/socks.cstc/util/newping.c which is not available any longer.

- a “ping” for SOCKS
- depends on the target host not blocking the service on the appropriate port (in this case “time”). And this version is primarily for checking “Is it alive?” rather than gathering statistics on the average response time of several echo requests.
- It uses the “time” TCP port to verify that a host is up, rather than using ICMP. It is thus usable through a firewall that blocks ICMP.

MBONE through firewalls

http://www.cs.virginia.edu/mngroup/projects/firewalls/

Their firewall features:

- Source host checking (allowing only certain hosts to transmit through the firewall, or denying specific hosts)
- Destination port checking
- Packet contents (unwrapping encapsulated IP)
- Regulating bandwidth allocated to a specific multicast group's traffic

Their Mbone gateway is based on a modified multicast routing daemon.

IP address management

Lucent Technologies NavisRadius™ commercial AAA servers

http://www.lucent.com/

http://www.lucent.com/security/

Secure Mailer (aka Postfix)

Wietse Venema's attempt to provide an alternative to the widely-used Sendmail program

70% of all mail sent via the Internet is sent via Sendmail

"Security. *Postfix* uses multiple layers of defense to protect the local system against intruders. Almost every Postfix daemon can run in a chroot jail with fixed low privileges. There is no direct path from the network to the security-sensitive local delivery programs - an intruder has to break through several other programs first. Postfix does not even trust the contents of its own queue files, or the contents of its own IPC messages. Postfix avoids placing sender-provided information into shell environment variables. Last but not least, no Postfix program is set-uid."

Network Security Tools

From U.S. DOE's Computer Incident Advisory Capability
http://ciac.llnl.gov/ciac/ToolsUnixNetSec.html

- System Administrator Tool for Analyzing Networks (SATAN), network security analyzer designed by Dan Farmer and Wietse Venema; scans systems connected to the network noting the existence of well known, often exploited vulnerabilities.
- ipacl - forces all TCP and UDP packets to pass through an access control list facility
- logdaemon - modified versions of rshd, rlogind, ftpd, rexecd,login, and telnetd that log significantly more information -- enabling better auditing of problems via the logfiles
- improved versions of: portmap , rpcbind,
- screend - a daemon and kernel modifications to allow all packets to be filtered based on source address, destination address, or any other byte or set of bytes in the packet

- securelib - new versions of the accept, recvfrom, and recvmsg networking system calls
- TCP Wrappers - allows monitoring and control over who connects to a host's TFTP, EXEC, FTP, RSH, TELNET, RLOGIN, FINGER, and SYSTAT ports + a library so that other programs can be controlled and monitored in the same fashion
- xinetd - a replacement for inetd which supports access control based on the address of the remote host and the time of access + provides extensive logging capabilities

NMAP

NMAP -- The Network Mapper from *http://www.insecure.org/nmap/*

This page also has a link to "*Remote OS detection via TCP/IP Stack FingerPrinting*" by Fyodor <fyodor@dhp.com> (www.insecure.org), October 18, 1998 - a means of identifying which OS the host is running by noting its TCP/IP behavior.

Network Security Exercises

You will find a nice set of exercises by Ramesh Govindan at USC's ISI for Kerberos, S/Key, and firewalls at:

http://www.isi.edu/~govindan/cs558/netsec/index.html

Note that you should **not** use their machines for these exercises, but I think you will find this useful reading.

Internet Protocol Version 6 (IPv6)

- Successor of current IPv4
- Internet needs to change IP in order to continue growth
- Defines a transition from IPv4 to IPv6

Specified by *RFC2460: Internet Protocol, Version 6 (IPv6) Specification*, December 1998.

Growth

- Currently IPv4 serves a market doubling every ~12 months
- In addition, new and very large markets are developing rapidly:
  - Nomadic Computing
  - Networked Entertainment
  - Device Control

Nomadic Computing

- Wireless computers
  - supporting multimedia
  - replacing pagers, cellular telephones, …
- IPv6 includes support for mobility
  - low overhead (?)
  - auto configuration
  - mobility

Networked Entertainment

Your TV will be an Internet Host!
[Olivetti and Thompson (RCA) are both currently selling such systems]

- 500 channels of television
- large scale routing and addressing
- auto-configuration
- requires support for real-time data

SonicBlues's ReplayTV 4000 a networked Digital Video Recorder (DVR) {i.e., coder/decoder + very big disk) that takes advantage of your broadband Internet connection - enables you to capture and transfer videos.

Device Control

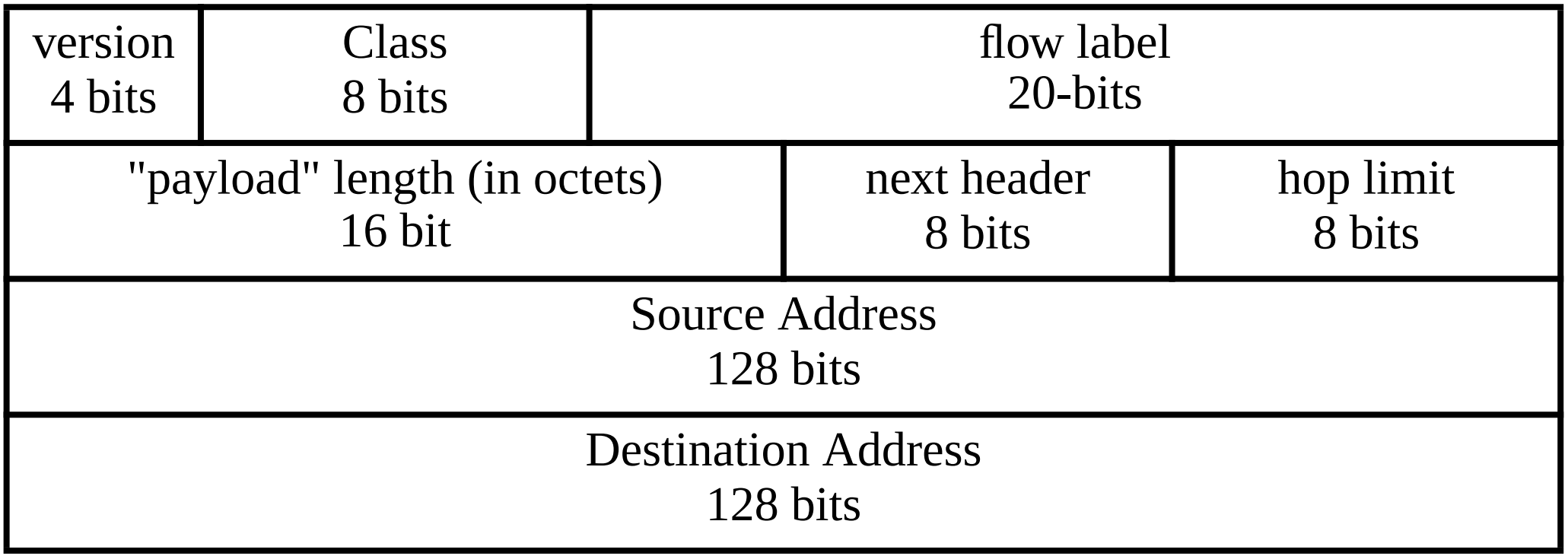
- Control everyday devices for
  - lightning, heating and cooling, motors, ...
  - new street light controllers already have IP addresses!
  - electrical outlets with addresses
- Market size is enormous
- Solution must be
  - simple, robust, easy to use
  - very low cost
  - potential power savings by (remote) network management based control may be quite large

There is already a networked: Toaster, a Coke machine, … .

IPv6 features

- Expanded Addressing Capabilities
  - 128 bit address length
  - supports more levels of hierarchy
  - improved multicast routing by using a scope field
  - new cluster addresses to identify topological regions
- Header Format Simplification
  - some IPv4 fields have been dropped, some made optional
  - header is easier to compute
- Improved Support for Extensions and Options
  - more efficient for forwarding of packets
  - less stringent limits to length of options
  - greater flexibility for introduction of future options
- Flow Labeling Capability
  - labeling of packets belonging to a particular “flow”
  - allows special handling of, e.g., real-time, packets
- Authentication and Privacy Capabilities
  - Extensions to support authentication, data integrity, and (optional) data confidentiality

IPv6 header format



IPv6 header (total length = 40 bytes)

IPv6: 6 fields + 2 addresses

versus

IPv4: 10 fixed fields + 2 addresses + options

Demultiplexing

Initially, it was assumed that by keeping the version field the same that IPv4 and IPv6 could be mixed over the same links with the same link drivers.

However, now IPv6 will be demultiplexed at the link layer:
hence, IPv6 been assigned the Ethernet type 0x86DD (instead of IPv4's 0x8000)

Simplifications

IPv6 builds on 20 years of internetworking experience - which lead to the following simplifications and benefits:

Simplification	Benefits
Use fixed format headers	Use extension headers instead, thus no need for a header length field, simpler to process
Eliminate header checksum	Eliminate need for recomputation of checksum at each hop (relies on link layer or higher layers to check the integrity of what is delivered)
Avoid hop-by-hop segmentation	No segmentation, thus you must do **Path MTU discovery** or only send small packets (1996: 536 octets, 1997: proposed 1500 octets) • This is because we should have units of **control** based on the units of **transmitted** data.
Eliminate Type of Service (ToS) field	Instead use (labeled) flows

Quality-of-Service Capabilities

- for packet streams
- Flow characterized by flow id + source address
- unique random flow id for each source

- Class field

D (1 bit)	**Network-wide priority (3 bits)**	**Reserved (4 bits)**
Delay sensitive	Encodes the priority of traffic, can be used to provide “Differentiated services”	Researchers would like to use two of these bits for congestion avoidance control: • one bit which could be set by routers to indicate that congestion was experienced; • the other bit could be used by the source to mark that it is “ready to adapt”.

- Flow ID - indicates packets which should all be handled the same way.

The original specified in *RFC 1809: Using the Flow Label Field in IPv6*

Subsequently updated - see Chapter 6 of Huitema, **2nd edition**; this change occurred because of Steve McCanne's SigComm'96 paper:
McCanne, S., Jacobson, V., and Vetterli, M., *Receiver-driven Layered Multicast*. ACM SIGCOMM, August 1996, Stanford, CA, pp. 117-130.

Payload length

Payload length is the length of the data carried after the header.

As the length field is 16 bits ⇒ maximum packet size of 64 kilobytes; but there is a provision for "jumbograms" [via the Hop-by-Hop option header with option type 194].

IPv4 Protocol type ==> IPv6 Next Header type

Tells how to interpret the next header which follows, it is either the payload type or the type of the next header. [Payload types use the IPv4 protocol type values]

Decimal	Keyword	Header type
0	HBH	Hop-by-hop options
2	ICMP	IPv6 ICMP
3	GGP	Gateway-to-Gateway Protocol
5	ST	Stream
6	TCP	Transmission Control Protocol
17	UDP	User Datagram Protocol
43	RH	IPv6 Routing Header
44	FH	IPv6 Fragmentation Header
45	IDRP	Inter-domain Routing Protocol
51	AH	Authentication Header
52	ESP	Encrypted Security Payload
59	Null	No next Header (IPv6)
60		IPv6 Destination Options Header
88	IGRP	IGRP
89	OSPF	Open Shortest Path First
255		Reserved

Extension headers

- Each header is a multiple of 8 octets long
- order (after IPv6 header):
  - Hop-by-hop option,
  - Destination options header (1)
  - Routing header,
  - Fragment header,
  - Authentication header,
  - Encapsulating security payload header,
  - Destination options header (2)
  - Followed by the upper layer header (e.g., TCP, UDP, …)

So a TCP packet looks like:

IPv6 header Next Header = **TCP**	TCP header + data

If we wanted to explicitly route the above packet, we simply add a routing header:

IPv6 header Next Header = **Routing**	Routing header Next Header = **TCP**	TCP header + data

Addressing

- 128 bits long
- three types: unicast, multicast, anycast

Unicast	identifies exactly one interface
Multicast	identifies a group of interfaces; a packet sent to a multicast address will be delivered to all members of the group
Anycast	delivered to the nearest member of the group

- 2^{96} times more addresses than IPv4 are available !!!

IPv6 addresses per m^2

Earth: 511,263,971,197,990 m^2
$\Rightarrow$ 665,570,793,348,866,943,898,599 / m^2

- pessimistic estimate with hierarchies: ~1,564 addresses / m^2
- optimistic: 3,911,873,538,269,506,102 /m^2

Writing an IPv6 address

The 128 bit IPv6 address is written as eight 16 bit integers using hexadecimal digits.

The integers are separated by colons.

for example: FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

A number of abbreviations are allowed:

- leading zeros in integers can be suppressed
- a single set of consecutive 16 bit integers with the value null, can be replaced by double colon, i.e.,
  1080:0:0:0:8:800:200C:417A becomes 1080::8:800:200C:417A
- When an IPv4 address is turned into an IPv6 address we prepend 96 bits of zero; but we can write it as:
  ::10.0.0.1 - hence combining dotted-decimal and IPv6 forms
- Prefixes can be denoted in the same manner as for IPv4, i.e.,
  FEDC:BA98:7600::/40 - for a 40 bit long prefix

Address Allocation

0000 0000	1/256	Reserved
0000 0001	1/256	Unassigned
0000 001	1/128	NSAP Allocation
0000 010	1/128	IPX
0000 011	1/128	Unassigned
0000 1	1/32	Unassigned
0001	1/16	Unassigned
001	1/8	Unassigned
010	1/8	Aggregated Global Unicast Addresses
011	1/8	Unassigned
100	1/8	Geographic-based Unicast Addresses
101	1/8	Unassigned
110	1/8	Unassigned
1110	1/16	Unassigned
1111 0	1/32	Unassigned
1111 10	1/64	Unassigned
1111 110	1/128	Unassigned
1111 1110 0	1/512	Unassigned
1111 1110 10	1/1024	Link Local Use Addresses
1111 1110 11	1/1024	Site Local Use Addresses
1111 1111	1/256	Multicast Addresses

Aggregate Global Unicast Addresses

010 (3 bits)	TLA (13 bits)	NLA (32 bits)	SLA (16 bits)	Interface ID (64 bits)

Formerly called “Provider based Unicast Addresses”, which has variable length bits fields, but this was recognized as too difficult to deal with (especially if you had to renumber).

TLA	Top Level Aggregator In the backbone, there will be one routing entry per TLA
NLA	Next Level Aggregator Can be divided into top tier provider and second tier provider, etc. Replaces: subscriber id
SLA	Site Local Aggregator Generally allocated to a link within a site. Replaces: subnet
Interface ID	Identifies an interface Replaces: Node id

Interface ID

Must be unique to the link, but there are some advantages of making it more globally unique.

Hence, most will be based on the IEEE EUI-64 format, but with the “u” (unique) bit inverted.

- The “u” bit is the 7th most significant bit of a 64 bit EUI.
- The inversion was necessary because 0:0:0:0 is a valid EUI, but this would collide with one of the IPv6 special addresses.
- u=1, when the address comes from a valid EUI, and is 0 otherwise.

To go from a 48 bit IEEE 802, you insert 0xFFFE in between the 3rd and 4th octets of an IEEE 802 address, i.e., 123456789abc becomes 123456FFFE789abc.

Special Address Formats

Unspecified address

“::” == “0:0:0:0:0:0:0:0 - can only be used as a source address by a station which does not yet have an address

Loop-back address

0:0:0:0:0:0:0:1 - used to send an IPv6 datagram to yourself

IPv4-based address

prefix the 32 bit IPv4 address with 96 zero bits

Site local addresses

Site local address can not be routed on the global internet, but they can be used by sites that are not connected to the internet or for communication within the site.

1111111011 (10 bits)	0 (38 bits)	SLA (16 bits)	Interface ID (64 bits)

Link local addresses

Link local address are simply unique to a given link - they can be used by stations that have not yet been assigned a provider-based address.

1111111010 (10 bits)	0 (54 bits)	Interface ID (64 bits)

Multicast Addresses

	4 bit	4 bit	
1111 1111	Flags xxxT	Scope	112 bit - group id

T == Transient

T = 0	well-known permanent - assigned by the IANA
T = 1	non-permanent

Scope	
0	reserved
1	node local scope
2	link local scope
3, 4	unassigned
5	site local scope
6, 7	unassigned
8	organization local scope
9, A, B, C, D	unassigned
E	global scope
F	reserved

Permanently assigned groups

For example, group 0x43 has been assigned to the Network Time Protocol (NTP), hence:

FF01::43	represents all NTP servers on the same node as the sender
FF02::43	represents all NTP servers on the same link as the sender
FF05::43	represents all NTP servers on the same site as the sender
FF08::43	represents all NTP servers within the same organization as the sender
FF0E::43	represents all NTP servers in the Internet

IANA has assigned a whole series of group identifiers, including:

FF0X:0:0:0:0:0:0	Reserved multicast address - this can not be used within any scope
FF01:0:0:0:0:0:1	All Nodes on this node address
FF02:0:0:0:0:0:1	All Nodes on this link address
FF01:0:0:0:0:0:2	All Routers on this node address
FF02:0:0:0:0:0:2	All Router address on this link

FF02:0:0:0:0:0:3	unassigned
FF02:0:0:0:0:1:1	Link Name
FF02:0:0:0:0:1:2	All DHCP agents on this link
FF02:0:0:0:0:1:3	All DHCP servers on this link
FF02:0:0:0:0:1:4	All DHCP relays on this link
FF05:0:0:0:0:1:2	All DHCP agents at this site
FF05:0:0:0:0:1:3	All DHCP servers at this site
FF05:0:0:0:0:1:4	All DHCP relays at this site
FF0X:0:0:0:0:2:7FFE	Session Announcement Protocol (SAP) v1 Announcements

Multimedia conferences:

FF0X:0:0:0:0:2:8000 .. FF0X:0:0:0:0:2:FFFF	multimedia conferences

X=2 -- this link; X=5 -- this site

Use SAP to announce the conference - repeatedly until the end of the conference.

Anycast

Sending a packet to a generic address to get a specific service from the “nearest” instance. This puts the burden of determining which instance to deliver it to on the routing system.

Requires defining a router entry for each anycast address.

Subnet Anycast Address:.

Thus the host ID of zero is treated as the subnet.

IPv6 Routing

- all standard routing protocols
- routing extensions
  - Provider Selection
  - Host Mobility (route to current location)
  - Auto-Readdressing (route to new address)

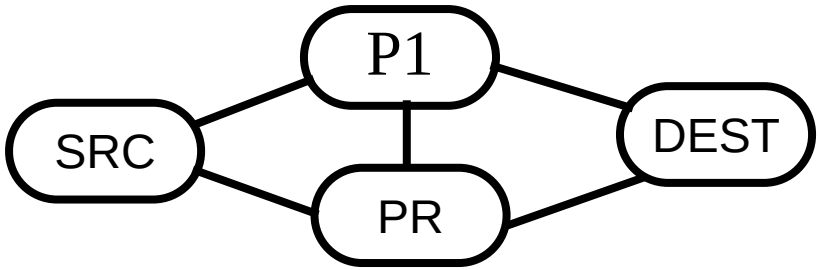


Figure 65: IPv6 Routing Option: provider specifies: SRC, PR, P1, Dest
reply: Dest, PR, P1, SRC

Routing header

Next Header (8 bits)	Header Ext Length (8 bits)	Routing Type=0 (8 bits)	Segments Left (8 bits)
reserved (32 bits)			
address[1] (128 bits)			
address[2]			
...			
address[n]			

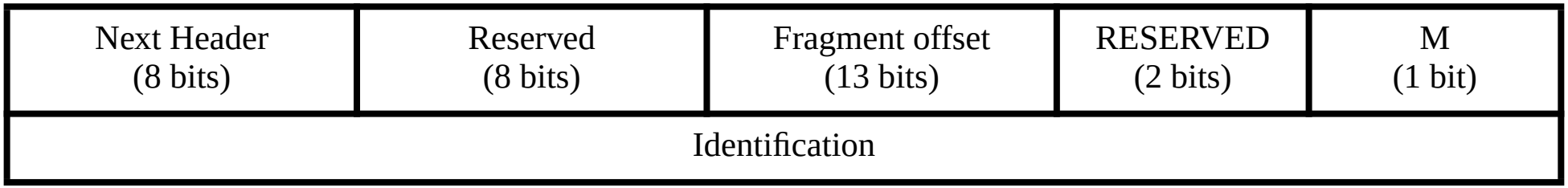
Next Header identifies the next header in the chain of headers.

Header Ext. Length. - number of 64 bit words (not including the first 64 bits).

Routing type=0, is the generic routing header which all IPv6 implementations must support.

Number of Segments is the number of segments left in the list (between 0 and 23).

Fragment header

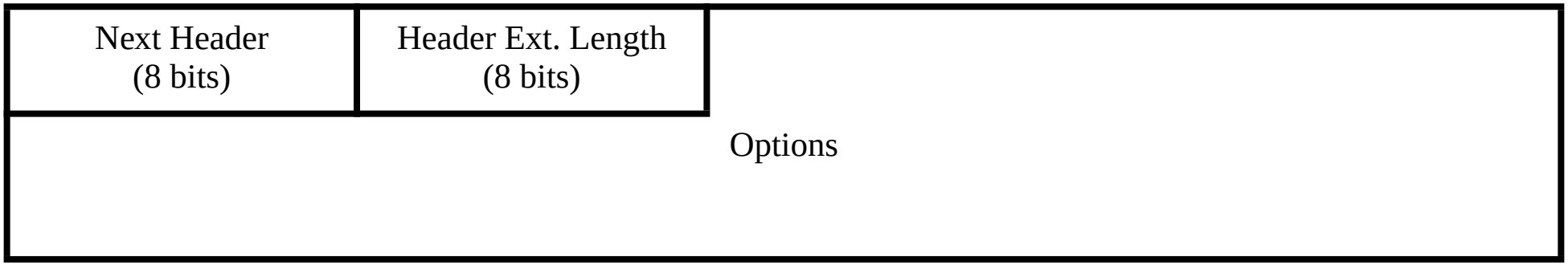


Fragment offset - in units of 64 bit words, the field is the most significant 13 bits of a 16 bit words.

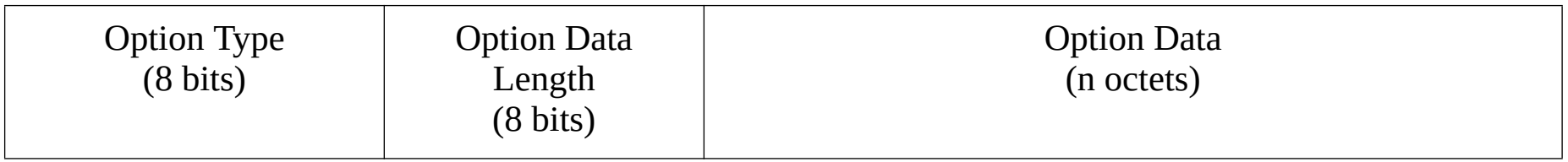
M == More fragment bit, set in all but the last fragment

Identification - a 32 bit number

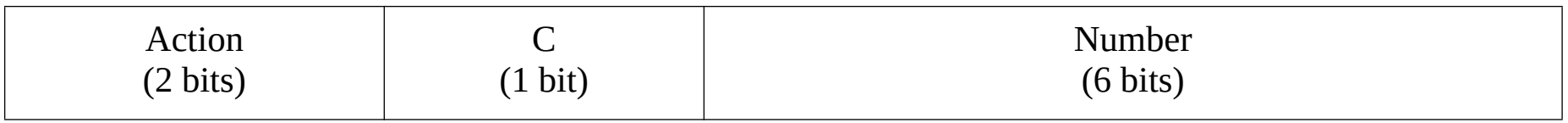
Destination Options header



Each options field is encoded as:



The option type:



Action tells what action must be taken if the processing nodes does not recognize

the option.

Bits	Action
00	Skip over this option
01	Discard packet silently (i.e., without sending an ICMP report)
10	Discard packet and send an ICMP report - even if destination is multicast
11	Discard packet and send an ICMP report - only if destination is **not** multicast

C == change en route bit -- indicates that this option may be changed by intermediate relays on the way to the destination

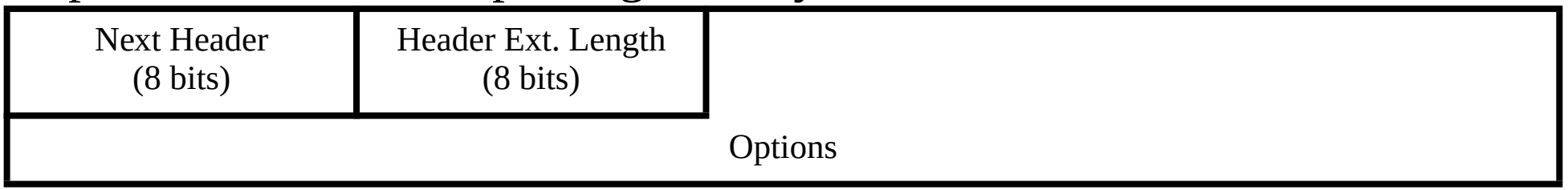
Currently only two options are defined:

Pad1 == a null byte - for use in padding to a 64-bit boundary; note it does not have a null option length field after it - as it is the whole field

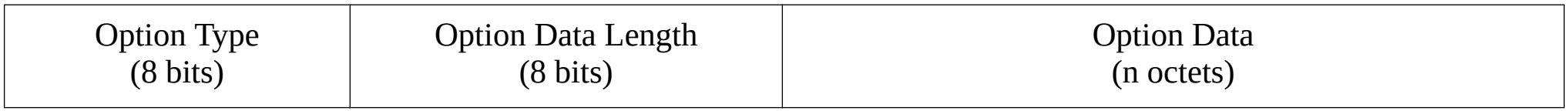
PadN - the length field says how many null bytes are needed to file to a 64-bit boundary.

Hop-by-Hop Options header

Same basic format as Destination option header, but the hop-by-hop header will be processed at each hop along the way.



Each options field is encoded as:



Currently three options are defined: Pad1, PadN, and

- Jumbo payload option (option type =194) - the option Data Length is 4 and is followed by a 32 bit Jumbo Payload Length value.

RFC 2113 "IPv6 Router Alert Option", by Dave Katz, February 1997

Security

- Header Authentication with signatures
  - Must have support for Message Digest 5 (MD5) algorithm (*RFC1321*)
- *RFC1810*: Report on MD5 Performance by J. Touch. June 1995 - is a report on MD5 performance.
- Packet Encapsulation with e.g., DES

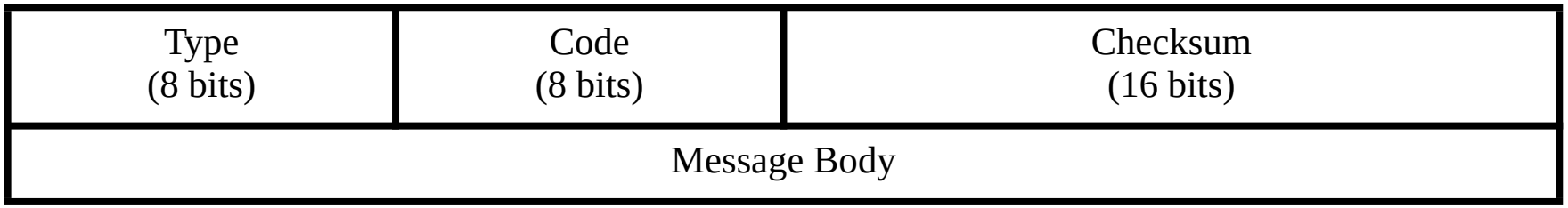
For more information see Chapter 5 of *IPv6*, 2nd edition, by Christian Huitema.

IPSEC IPv6 implementation

The NRL IPv6/IPsec Software Distribution

- a reference implementation of IPv6 and IP Security for the 4.4BSD-Lite networking software.
- Freely distributable (subject to U.S. export controls) and usable for commercial and non-commercial purposes (you must adhere to the NRL and UC Berkeley license terms) see also: *http://web.mit.edu/network/isakmp*
  - DOD ISAKMP Distribution
  - Cisco's ISAKMP Distribution
  - NRL's IPv6 + IPSEC Alpha 7.1 Distribution (Dec '98)
  - Portland State University's Mobile IP with IPSEC for FreeBSD 2.2.1.

IPv6 ICMP



Currently defined ICMP Types

Type	Purpose
1	Destination Unreachable
2	Packet too big
3	Time exceeded
4	Parameter problem
128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect

IPv6 ICMP Error Messages

Type: 1, 2, 3, or 4:

Type	Code	Checksum
Parameter		
As much of invoking packet as will fit - without the overall ICMP packet exceeding 576 octets		

IPv6 ICMP Echo Request/Reply (PING)

Type: Echo Request = 128, Echo Reply = 129

Type	Code	Checksum
Identifier		Sequence number
Data		

IPv6 ICMP and groups

Three group membership messages (type 130, 131, and 132):

Type	Code	Checksum
Maximum Response Delay		Unused
Multicast Address		

The Group Membership Reduction is used when a node leaves group.

Reports are always sent to the same group address that is reported.

Maximum response delay is the time in milliseconds that the responding report messages can be delayed. Responding stations are supposed to spread their responses uniformly over this range of delays (to prevent everyone from responding at once).

For more information about ICMP see: *RFC1885*: “Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)”, by A. Conta, S. Deering. December 1995.

DNS and IPv6

A new record type “AAAA” which contains a 128 bit address.

Just as for the “in-addr.arpa” domain used for converting Ipv4 addresses into names, IPv6 defines an “ipv6.int” domain:
thus the address 4321:0:1:2:3:4:567:89ab is represented as:
b.a.9.8.7.6.5.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.INT

For further information see *RFC1886*: “DNS Extensions to support IP version 6”. by S. Thomson & C. Huitema. December 1995.

IPv6 Transition Mechanisms

- Incremental update and deployment
  - first step: dual mode hosts and routers
  - Encapsulation of IPv6 in IPv4 packets
- Minimal upgrade dependencies (must first upgrade DNS)
- Easy addressing (upgraded routers can use IPv4 address)

Why IPv6?

- solves Internet scaling problem
  - eliminates the problem of running out of addresses
  - allows route aggregation - which allows the size of the routing tables in the backbone routers to decrease
- flexible transition (interworks with IPv4)
- meets the needs of new markets
- new functionality
- real-time flows
- provider selection
- host mobility
- end-to-end security
- auto-configuration - chapter 4, “Plug and Play” in *IPv6*, 2nd edition, by Christian Huitema - this a **very major** advantage of IPv6.

IPv6 networks

6Bone - *http://www.6bone.net/* a testbed for deployment of IPv6

vBNS - *http://www.vbns.net*

Further information

See:

http://www.ipv6.org/

http://www.ipv6forum.com/

Summary

This lecture we have discussed:

- IPSec
- Firewalls
- IPv6