SNS COLLEGE OF TECHNOLOGY
(An Autonomous Institution)
COIMBATORE-35
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

19ITT302 INTERNET OF THINGS

IPsec in IPv6 :

IP security, or IPsec, is a collection of standards for the security of transmitted sensitive information over unprotected networks. At the network level, IPsec protects and authenticates data packets being sent between IPsec devices. IPsec has several optional security features, the use of which can be dictated by local security policies:

- Data confidentiality -sender can encrypt packets before send
- Data integrity - receiver can authenticate packets to ensure data hasn’t been tampered with
- Data origin authentication – receiver can confirm the source of any packets received
- Antireplay -receiver can detect and reject any replayed data packets

IPsec for IPv6 is implemented with Authentication Header and Encapsulating Security Payload. Authentication Header (AH) verifies the source to protect IP header integrity. Encapsulating Security Payload (ESP) “provides confidentiality, authentication of the source, connectionless integrity of the inner packet, antireplay, and limited traffic flow confidentiality.”

IPsec has two different modes of operation: Transport mode and Tunnel mode.

- Transport mode (host to host) uses the IPv6 header of the original packet, then the AH or ESP header, and then the payload
- Tunnel mode (gateway to gateway or gateway to host) uses a new IPv6 header that includes the AH or ESP header, the original IP header, and the payload

IPv6 Encryption

While end-to-end encryption was retroactively added to IPv4, it was built into IPv6. Encryption and integrity-checking, currently used by VPNs, is standard in IPv6 for all devices and systems.

IPv6 is also more secure for name resolution. The Secure Neighbor Discovery (SEND) protocol enables cryptographic confirmation of a host's identity upon connection, making naming-based attacks more difficult. This is not a replacement for verification at the application or service level but offers additional security.

Is IPv6 more secure than IPv4?

The short answer is no. However, this question can mean two different things, and therefore requires a more nuanced answer. This question can mean:

- Whether the specific IPv6 protocols are more secure than their IPv4 equals
- Whether deployments of IPv6 are more secure than their IPv4 equals

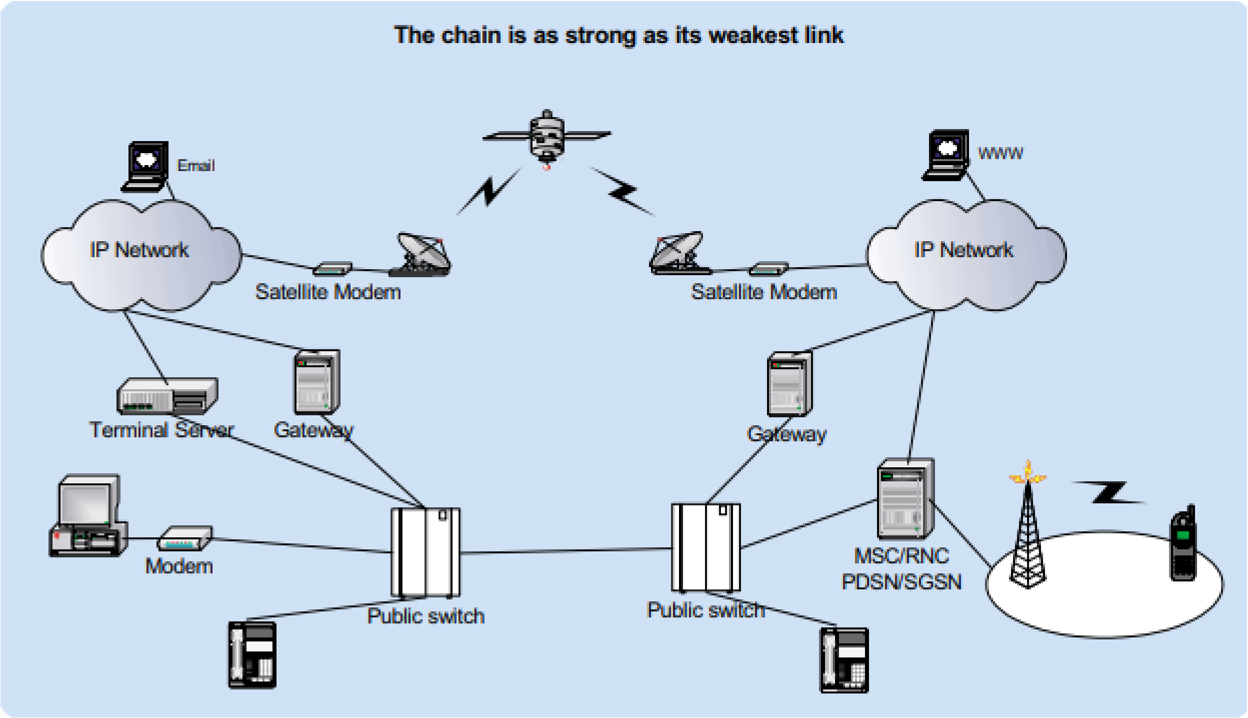
When comparing IPv4 and IPv6 at the protocol level, the complexity of IPv6 could present a higher number of points for attacks. However, it is more practical to compare IPv4 and IPv6 deployments in terms of security. For that, it is important to consider how long protocol specifications and implementations have existed.

Most frequently, the security vulnerabilities in a network protocol stem from flaws in implementation. These flaws are later patched, and over time the discovery and patching of vulnerabilities strengthens the security of the network protocol. Because IPv4 protocols have benefitted from this process much longer than IPv6 protocols, there are more robust in their security.

Sometimes, these vulnerabilities stem from flaws in the protocol specifications. In this case, IPv4 protocol specifications once again benefit from having been around longer, as the IPv6 protocol specifications are newer and have not yet received the same level of scrutiny.

IPv6 Header Compression Scheme :

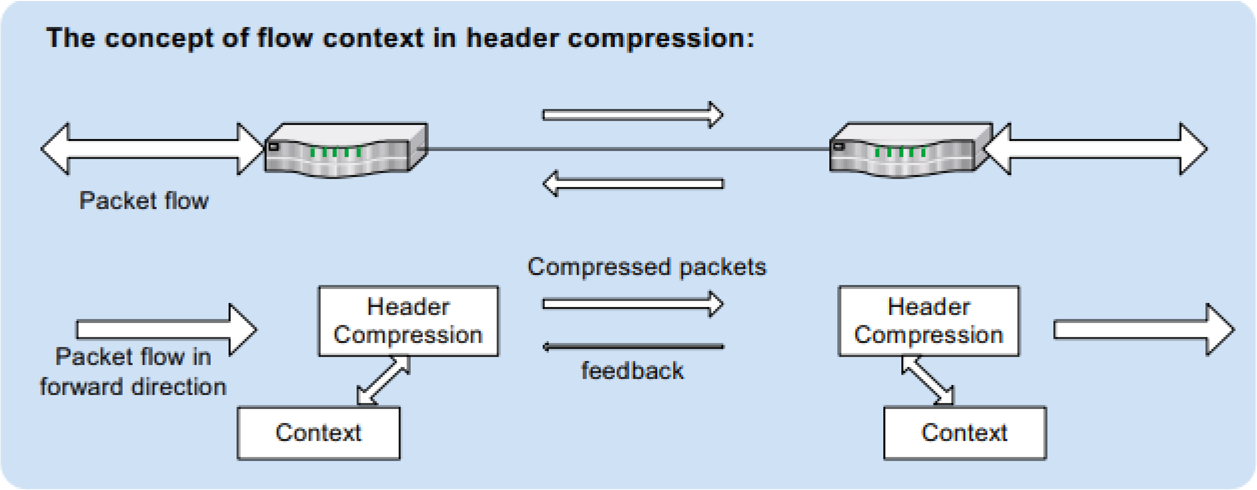
The need for IP header compression The Internet Protocol (IP) is the choice of transport protocol both on wired and wireless networks and this choice is leading to the convergence of telecommunication and data networks. These converged networks will be the building blocks of the All-IP vision. As the networks evolve to provide more bandwidth, the applications, services and the consumers of those applications all compete for that bandwidth. For the network operators it is important to offer a high quality of service (QoS) in order to attract more customers and encourage them to use their network as much as possible, thus providing higher average revenue per user (ARPU). As for wireless networks with their high bit error rates (highly prone to interference) and high latency (long round trip times), it is difficult to attain those high bandwidths required. When all these factors are taken into account it means that the available resources must be used as efficiently as possible.



In many services and applications e.g., Voice over IP, interactive games, messaging etc, the payload of the IP packet is almost of the same size or even smaller than the header. Over the end-to-end connection, comprised of multiple hops, these protocol headers are extremely important but over just one link (hop-to-hop) these headers serve no useful purpose. It is possible to compress those headers, providing in many cases more than 90% savings, and thus save the bandwidth and use the expensive resources efficiently. IP header compression also provides other important benefits, such as reduction in packet loss and improved interactive response time. In short, IP header compression is the process of compressing excess protocol headers before transmitting them on a link and uncompressing them to their original state on reception at the other end of the link. It is possible to compress the protocol headers due to the redundancy in header fields of the same packet as well as consecutive packets of the same packet stream.

Header compression:

The IP protocol together with transport protocols like TCP or UDP and optional application protocols like RTP are described as a packet header. The information carried in the header helps the applications to communicate over large distances connected by multiple links or hops in the network. The information comprises of source and destination addresses, ports, protocol identifiers, sequence numbers, error checks etc. As long as the applications are communicating most of this information carried in packet headers remains the same or changes in specific patterns. By observing the fields that remain constant or change in specific patterns it is possible either not to send them in each packet or to represent them in a smaller number of bits than would have been required originally. This process is described as compression.



The process of header compression uses the concept of flow context, which is a collection of information about field values and change patterns of field values in the packet header. This context is formed on the compressor and the decompressor side for each packet flow. The first few packets of a newly identified flow are used to build the context on both sides. These packets are sent without compression. The number of these first few packets, which are initially sent uncompressed, is closely related to link characteristics like bit error rate (BER) and round trip time (RTT). Once the context is established on both sides, the compressor compresses the packets as much as possible. By taking into account the link conditions and feedback from the decompressor, the compressed packet sizes vary. At certain intervals and in the case of error recovery, uncompressed packets are sent to reconstruct the context and revert back to normal operational mode, which is sending compressed packets. The header compression module is a part of the protocol stack on the devices. It is a feature, which must be negotiated before it can be used on a link. Both end points must agree if they support header compression and on the related parameters to be negotiated.