

A New QoS Controllable Security Protocol

Mahmoud MOSTAFA, Anas ABOU EL KALAM, Christian Fraboul

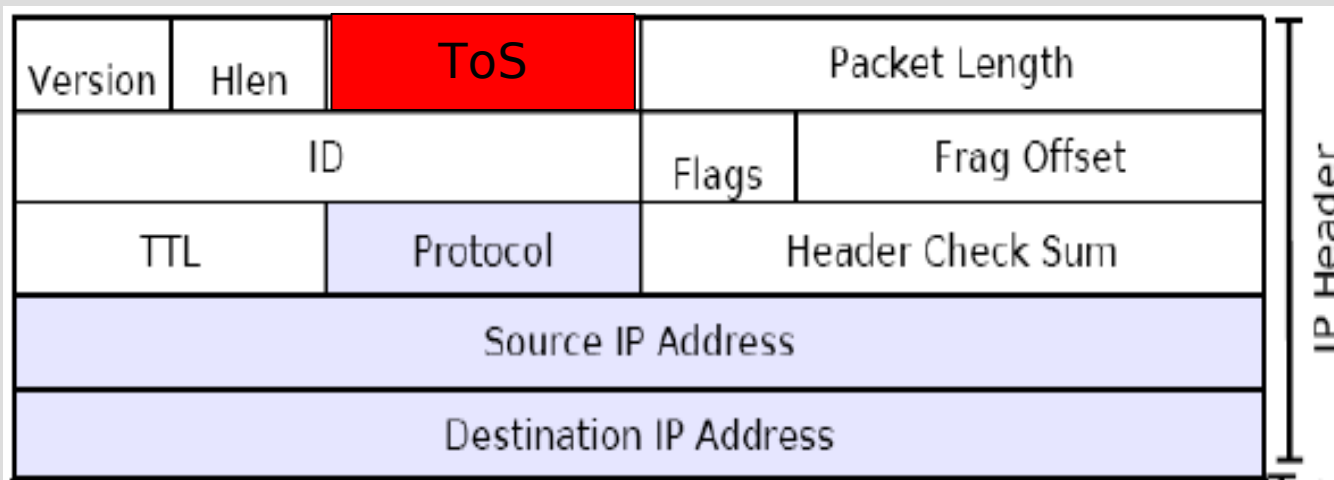
Université de Toulouse, INP, IRIT

Toulouse – France



QoS: Class of Service concept

- ♦ Divide the network traffic into different classes
- Each packet is assigned a priority value stored in the “Type of Service” field



- **QoS: Class of Service concept**

Multi-field packet classifiers inspect headers' fields & set priorities

- **TCP/UDP header**

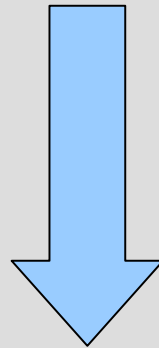
- Source Port number
- Destination Port number

- **IP header**

- Protocol identifier
- Source IP address
- Destination IP address

Motivation

Existing **security protocols** such as **IPSec ESP**
(Encapsulating Security Payload) encrypt / hide these fields

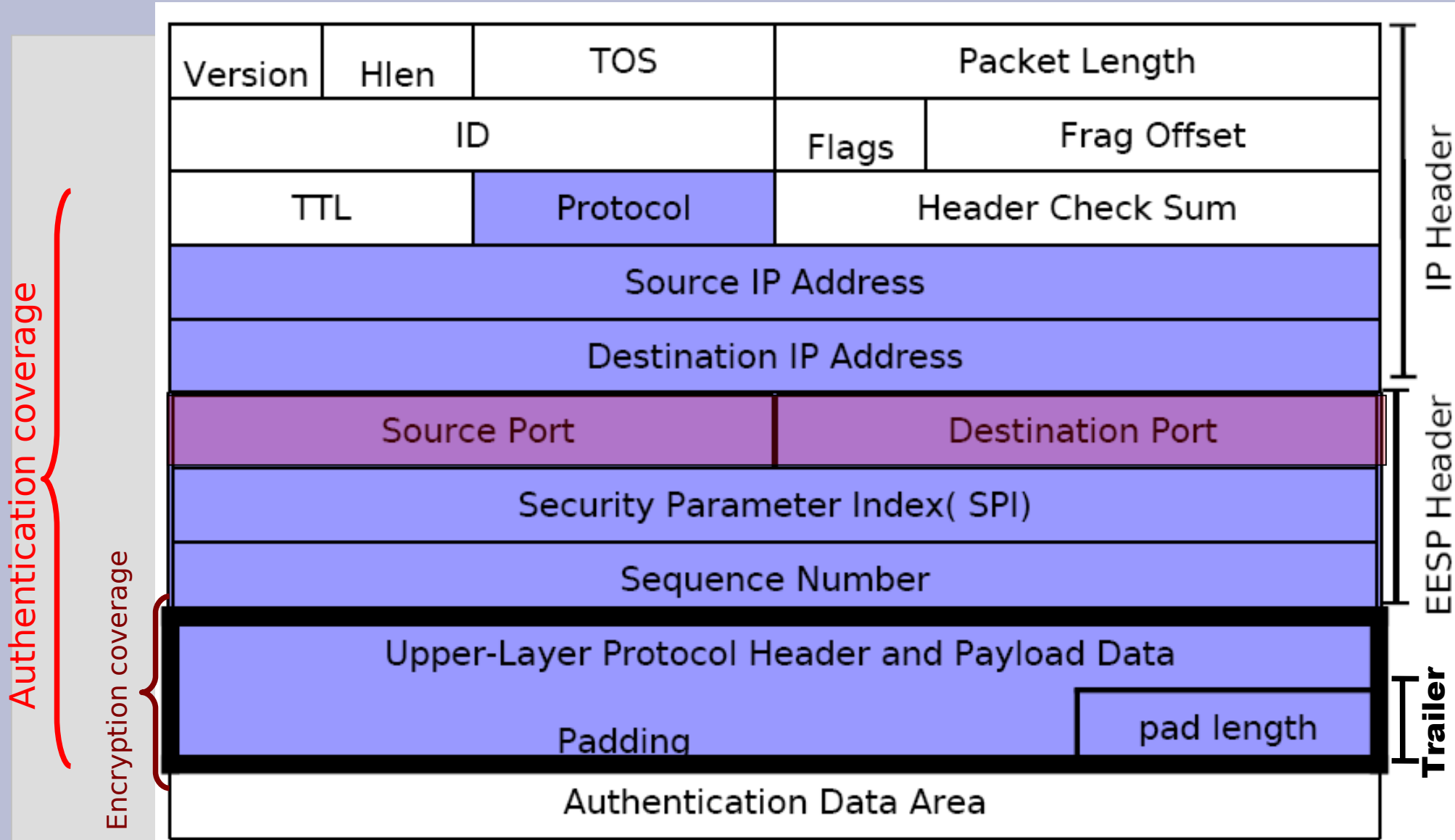


Active admission control / Priorities / QoS
can not be provided!

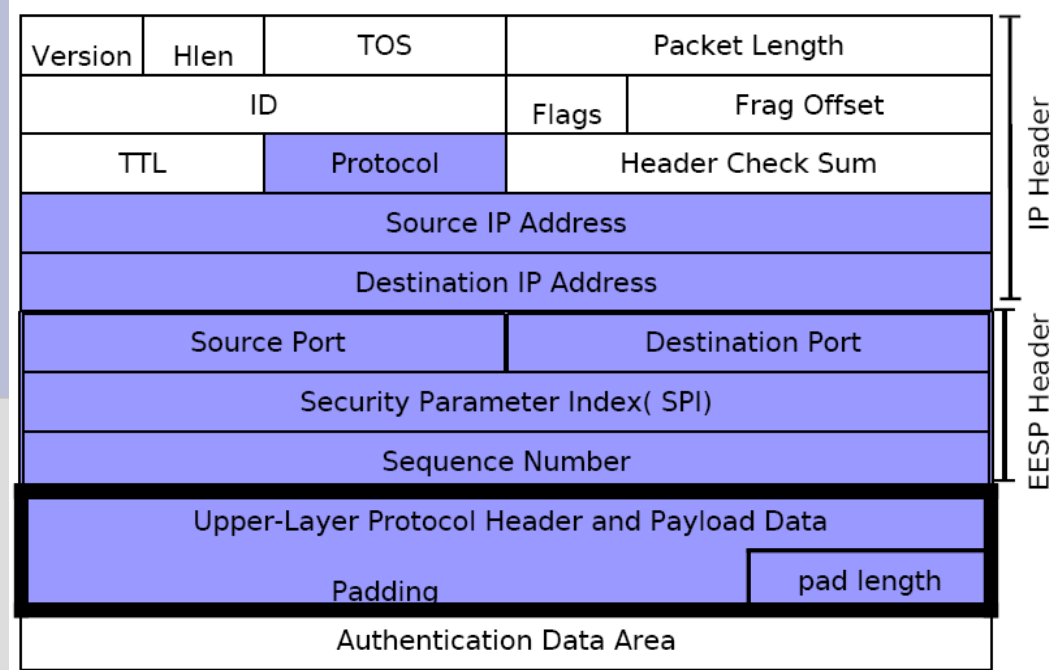
Our Solution

The **EESP** security protocol
(Enhanced Encapsulated Security Payload)

EESP (Enhanced Encapsulated Security Payload)



EESP Outbound processing



Encryption

$$M_E = E (M_P || Tr)_{KE}$$

- M_P : EESP payload

- *transport mode*: the upper layer *transport* protocol and its *payload* data
- *tunnel mode*: ...+ the original IP header

- Tr : EESP trailer

- *Padding* as well as the *pad length* field

Authentication

$$ICV = H-MAC (M_H || M_E || protocol || Src IP @ || Dst IP @)_{KA}$$

- M_H EESP header

- Source port, Dest port, SPI, sequence number

EESP vs IPsec ESP

		IPSEC ESP	EESP
QoS	Admission control	No	Yes
Security	Data confidentiality	yes	yes
	Authentication	Optional	yes
	Anti-Replay	Optional	yes
	Connectionless integrity	yes	yes
	Head Overhead	8 + 12 bytes	12 bytes

Performance analysis

- **Throughput**

- ◆ rate at which IP packets can be sent to the network

- **Delay**

- ◆ period of time it takes an IP packet to traverse from one point in the network to another

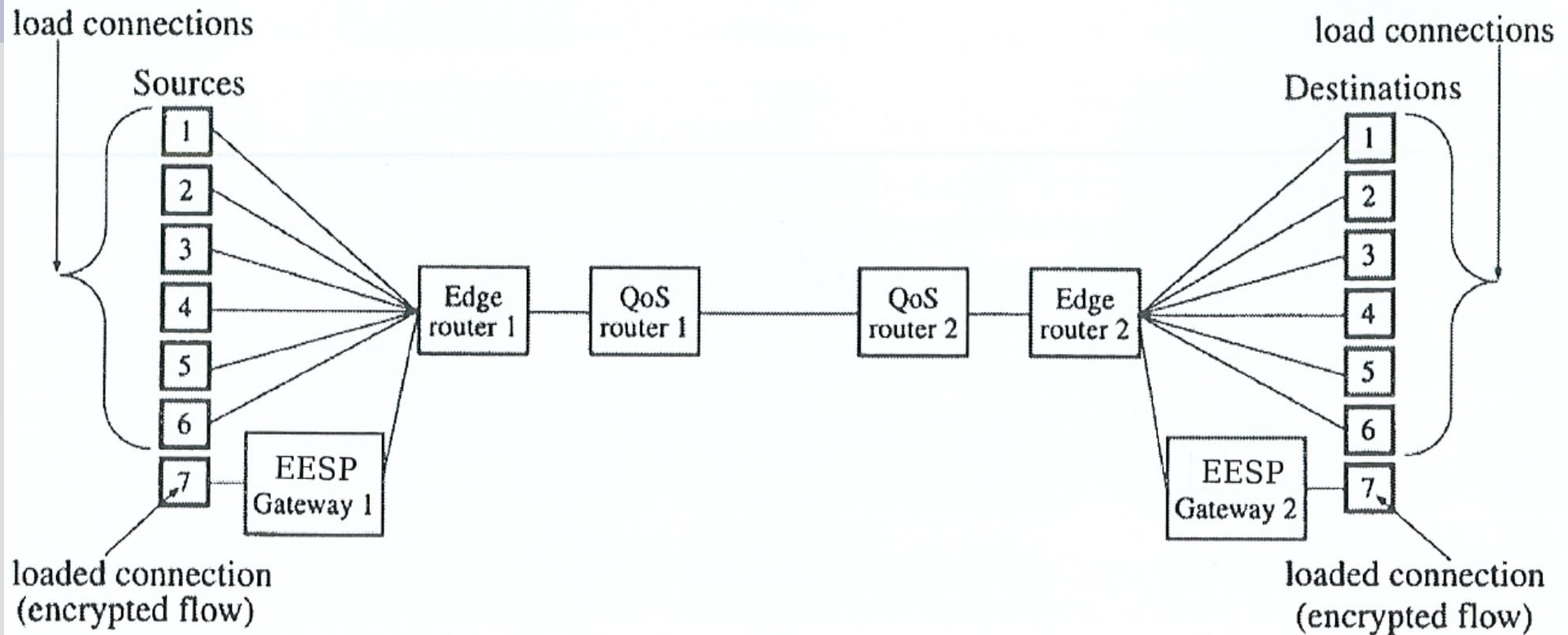
- **Jitter**

- ◆ variation in end-to-end transit delay

- **Packet loss rate**

- ◆ ratio of the number of packets lost over the total number of packets sent to the network

EESP Test bed



- The experiments will be performed For **EESP** and **ESP**
- with and without QoS **priorities**
- in situations of network **congestion**

Performance analysis

- Modify the Linux kernel 2.6 IPsec implementation to build EESP
- Use **DBS** (*Distributed Benchmark System*)
 - Transfer data **simultaneously** among multiple hosts
 - Generate the load traffic
 - create situations of network **congestions**
 - Record **transmission/reception times** of individual packets
 - **Measure** the time variation

Questions

