

Generic UDP Encapsulation

draft-herbert-gue-02

Tom Herbert <therbert@google.com>

Lucy Yong <lucy.yong@huawei.com>

Goal

An efficient, extensible, and generic encapsulation mechanism to facilitate packet transport in data center networks for non-virtualization as well as virtualization use cases.

Requirements

- Control, security, and performance
- Extensible, including private extensions
- Multi-protocol encapsulation
- Amenable to SW and HW implementation
- Compatibility with switches, middle boxes
- Integrates well with security protocols
- Control (OAM) messages

Generic UDP encapsulation

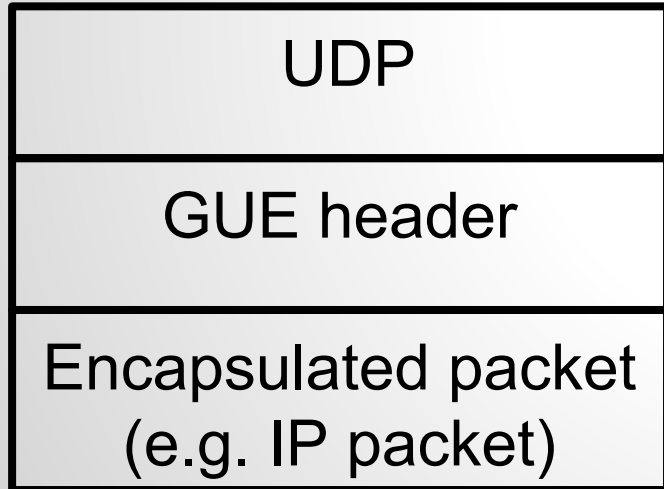
- Encapsulation of network protocols with an encapsulation header in UDP
- Encapsulate by IP protocol number
 - IP, IPv6, EtherIP, ESP, AH, GRE, MPLS, etc
- Includes data/control messages, version, header length, optional flags and fields

Extensibility

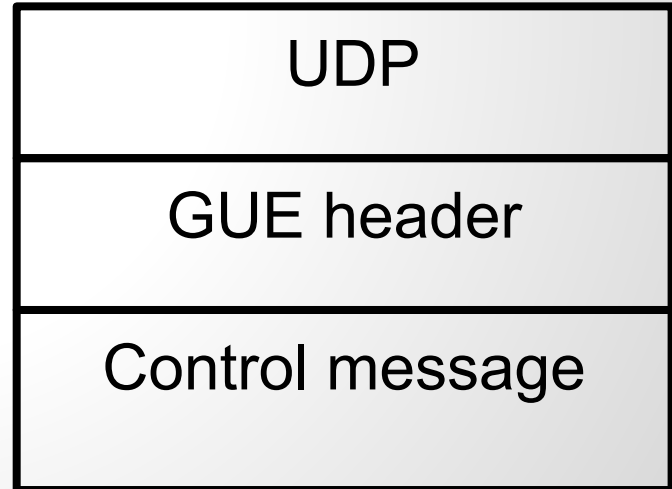
- Extensibility based on GRE model
- Flags and optional fields
- Possible uses
 - Virtualization (nvo3)
 - Security
 - Congestion control
 - Offload emulation

Two basic message types

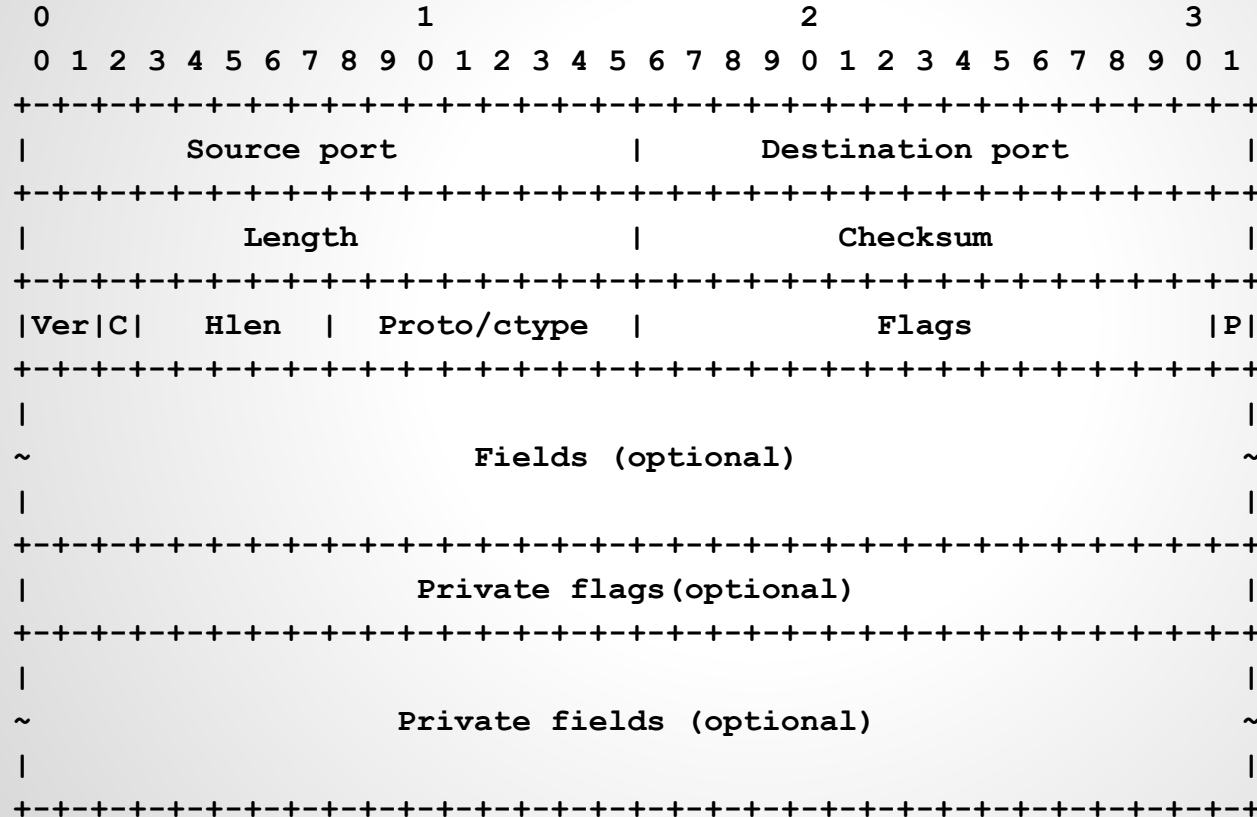
Data Message
(encapsulation)



Control Message (e.
g. OAM)



Header format (with UDP)



UDP

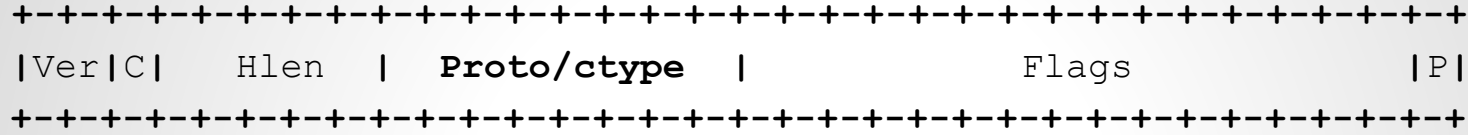
- Ports

- Destination GUE port (request IANA assignment)
- Source port used to provide entropy for RSS and ECMP (hash of inner flow tuple)

- Checksum

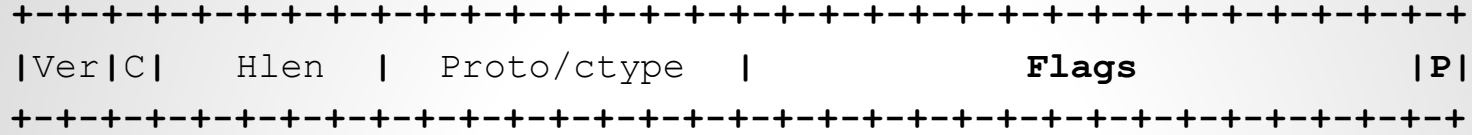
- IPv6: Must use (RFC6936 exception)
- IPv4: Must use if corruption can cause misdelivery (e.g. virtual network ID is corrupted)
- UDP checksum offload widely supported (in hosts)

Proto/ctype



- Proto/ctype
 - C bit not set: contains protocol number of encapsulated packet (standard assigned IP protocol numbers)
 - C bit set: contains type of control message

Flags



- **Flags:** Flag bits that can be defined. Flags may indicate presence of optional fields
- **P bit:** Private flags present

Flag properties

- Flags may be paired together to create a field with optional sizes
- New flags are defined contiguously from high to low order bits
- Flags are idempotent to maintain backwards compatibility
 - New flags cannot redefine meaning of older ones
 - Flags cannot redefine meaning of other header elements

Fields

- Mechanism of extensibility in GUE
- Immediately follow basic four byte header
- Presence of a field is indicated by flags
- Fields have size of multiple of four bytes
- Fields are fixed size
- Fields can be repurposed by negotiation or configuration (e.g. security)

Flags and fields reception

- Decapsulator **must** drop packet with unknown flag
- Decapsulator **may** drop packet that does not have required flags set
- Middle box **may** inspect flags and fields, but **must not** drop packets due to unknown flags

Private flags and fields

- Private flags: An optional field that is present when P bit flag is set
- Contains a set of 32 bit flags that can be defined by a site or implementation
- Private flags may indicate presence of private optional fields
- Same requirements in size and receive processing as GUE flags

Implementation status

- GUE is in Linux 3.17
 - IPIP, GRE, and SIT tunnels supported