Generic UDP Encapsulation

draft-herbert-gue-03

Tom Herbert <therbert@google.com>
Lucy Yong <lucy.yong@huawei.com>
Osama Zia <osamaz@microsoft.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.

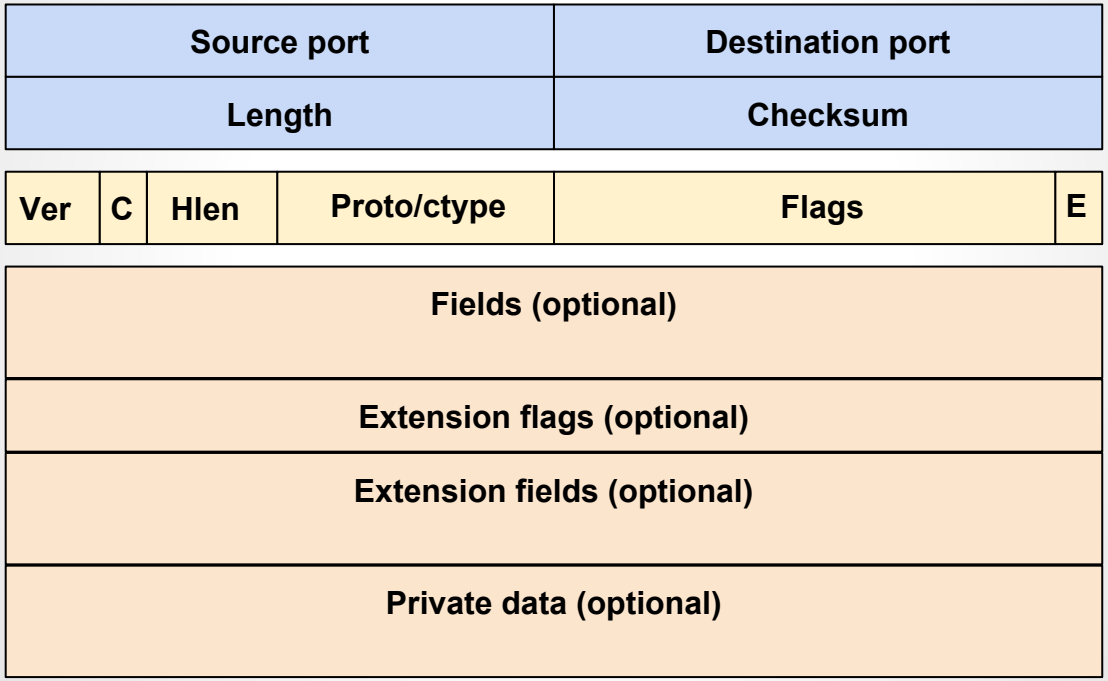
GUE's roots are in GRE

- GRE is established, well deployed, & **simple**
- Unfortunately, we've hit the wall in trying to extend GRE
- GUE as a "successor" to GRE
  - Retain same model of simplicity and extensibility
  - Allow more opportunity to extend the protocol
  - A few other "improvements"

Features

- Flag-fields like GRE for extensibility
- Header length allows middle box deep parsing
- IP protocol number indicates next header
- UDP encapsulation to facilitate ECMP
- Data messages as well as control messages (e.g. OAM)
- Security to provide integrity or authentication of header
- Checksum like UDP-lite, tunnel fragmentation
- Hardware friendliness considerations
- Support for network virtualization

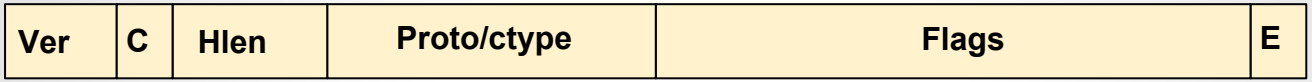
The GUE headers



UDP fields

- Ports
  - Destination GUE port (6080)
  - Source port used to provide entropy for ECMP
- UDP checksum
  - Requirement is one of
    - Enable it
    - Use GUE header checksum
    - Apply exceptions based on RFC6936
  - Sometimes performance advantage to enable

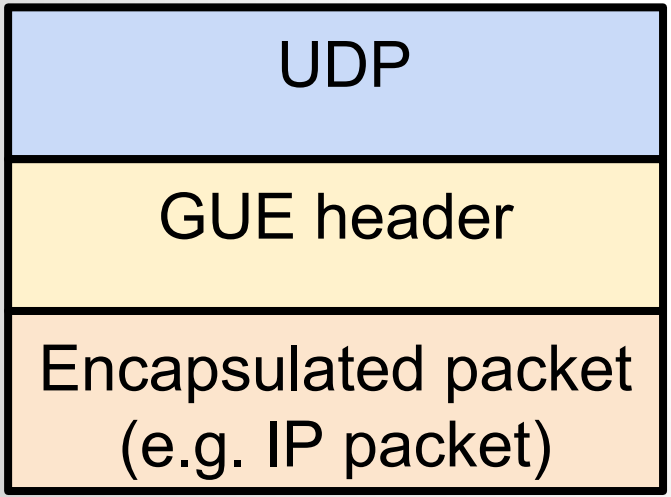
Primary GUE header



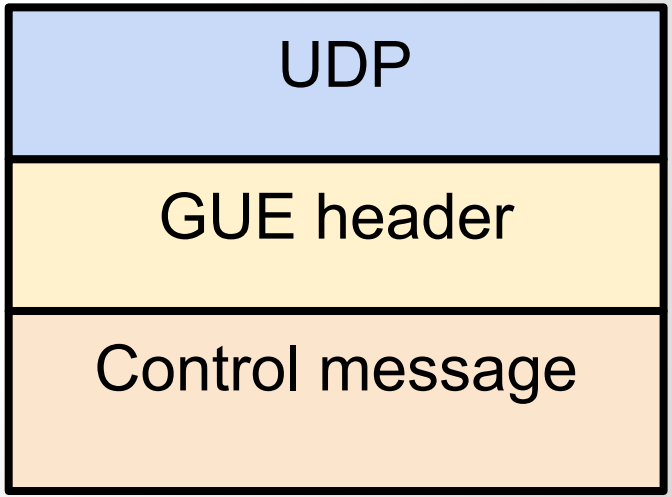
- Ver (version): Defines format of rest of the message
- C bit
  - Set: indicates control message (e.g. OAM message)
  - Not set: indicates data message (protocol encapsulation)
- Hlen: Length in 32-bit words of the GUE header not including first four bytes the header

Two message types

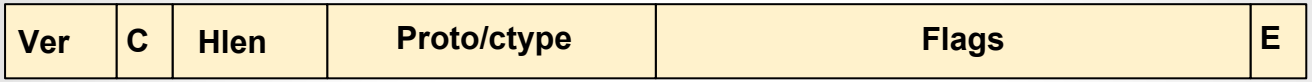
Data Message (encapsulation)



Control Message (e. g. OAM)

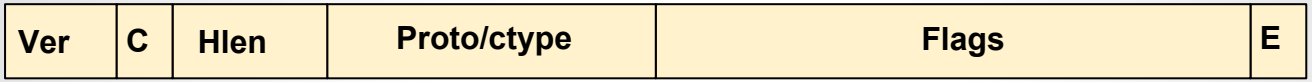


Primary GUE header



- 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

Primary GUE header



- Flags: Flag bits that can be defined. Flags may indicate presence of optional fields
- E bit: Extension flags present (32 more header flags)

Flag properties

- Flags may paired together to create a field with optional sizes
- New flags are defined contiguously from low to high order bits
- Flags are idempotent for forward compatibility
  - New flags cannot redefine meaning of older ones
  - Flags cannot redefine meaning of other pre-existing 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 receive processing

- Decapsulator (terminates UDP)
  - **Must** drop packet with unknown flags
  - **May** drop packet that does not have required flags
- Middle box
  - **May** inspect flags and fields
  - **Must not** drop packets due to unknown flags
  - **Should not** change flags or fields of packets

Protocol extensions

Defined

- Virtual network identifier
- Security field
- Header checksum
- Remote checksum offload
- Tunnel fragmentation

Possibly

- Passive OAM
- Outer/inner TTL mapping
- Congestion control
- Group based policy
- Remote segmentation offload

Probably not

- CRC
- Reliability layer
- QoS
- QCN
- Pseudo wire related
- Routing related
- Inband negotiation

Private data region

- Area data between last field and end of header (indicated by Hlen)
- Unstructured, private use
- Could contain flag-fields, TLVs, etc.

Status

- Three primary I-Ds
  - draft-herbert-gue-03
  - draft-hy-nvo3-gue-4-nvo-01
  - draft-hy-nvo3-gue-4-secure-transport-01
- IANA assigned port number 6080
- GUE is in Linux 3.17
  - IPIP, GRE, and SIT tunnels supported
- What WG should GUE be in?

Thankyou!