Advanced Topics
on Network Socket Programming
Computer Science Department, University of Crete

Manolis Surligas surligas@csd.uoc.gr
October 22, 2015
Custom Packet headers

• Each layer has its own headers

• Application layer may have its own

• Indeed most applications introduce custom packet headers

• Possible application header fields:
  - Application version
  - Sequence number
  - CRC
Custom Packet headers

Example of HTTP header structure

```
Hypertext Transfer Protocol
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Date: Wed, 21 Oct 2015 13:58:33 GMT
Pragma: no-cache
Expires: Fri, 01 Jan 1986 00:00:00 GMT
Cache-Control: no-cache, no-store, must-revalidate
Last-Modified: Sun, 17 May 1998 03:00:00 GMT
X-Content-Type-Options: nosniff
Content-Type: image/gif
Server: Golfe2
Content-Length: 35

[HTTP response 1/1]
[Time since request: 0.055004000 seconds]
[Request in frame: 8509]
CompuServe GIF, Version: GIF89a
Version: GIF89a
Screen width: 1
Screen height: 1
Global settings: (Global color table present) (1 bit per color) (1 bit per pixel)
Background color index: 255
Global color map: fffffff000000
Image
Trailer (End of the GIF stream)
```
Create custom Packet headers

- The problem is that the `send()`, `sendto()`, `recv()`, `recvfrom()` take as argument a single buffer.

- A packet header may consist from integers, shorts, e.t.c.

- How we can efficiently create the header?
Create custom Packet headers

- Assume that the header of each of our packets is:

  Seq Number: 32 bits  Type: 16 bits  Version: 8 bits  Data: X bits

- Create a struct that describes properly the header attributes

```c
typedef struct custom_header{
    uint32_t seq_number;
    uint16_t type;
    uint8_t version;
};
```
Create custom Packet headers

- Fill in the appropriate values

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28

struct custom_header header;
header.seq_number = 12849;
header.type = TYPE_DATA_PACKET;
header.version = 1;
```
Create custom Packet headers

- Fill in the appropriate values

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28

struct custom_header header;
header.seq_number = 12849;
header.type = TYPE_DATA_PACKET;
header.version = 1;
```

- Is the assignment right?
Create custom Packet headers

- Fill in the appropriate values

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28

struct custom_header header;
header.seq_number = 12849;
header.type = TYPE_DATA_PACKET;
header.version = 1;
```

- Is the assignment right? **NO!!**
Create custom Packet headers

- Fill in the appropriate values *always taking care the endianess*

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28
.
.
.
struct custom_header header;
header.seq_number = htonl(12849);
header.type = htons(TYPE_DATA_PACKET);
header.version = 1;
```
Create custom Packet headers

- Fill in the appropriate values always taking care the endianess

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28
.
.
.
struct custom_header header;
header.seq_number = htonl(12849);
header.type = htons(TYPE_DATA_PACKET);
header.version = 1;
```

- Why the `version` field is not needed to be converted in Network Byte Order?
Create custom Packet headers

- Copy the header in the buffer

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28

uint8_t buffer[1024];
struct custom_header header;
header.seq_number = htonl(12849);
header.type = htons(TYPE_DATA_PACKET);
header.version = 1;
memcpy(buffer, &header, sizeof(header));
```
Create custom Packet headers

- After the header fill also the data

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28
.
.
uint8_t buffer[1024];
struct custom_header header;
header.seq_number = htonl(12849);
header.type = htons(TYPE_DATA_PACKET);
header.version = 1;
memcpy(buffer, &header, sizeof(header));
memcpy(buffer + sizeof(header), data, data_len);
```
Create custom Packet headers

- And send them through the network!

```c
/* Packet that carries application data */
#define TYPE_DATA_PACKET 28

uint8_t buffer[1024];
struct custom_header header;
header.seq_number = htonl(12849);
header.type = htons(TYPE_DATA_PACKET);
header.version = 1;
memcpy(buffer, &header, sizeof(header));
memcpy(buffer + sizeof(header), data, data_len);
send(sd, buffer, data_len + sizeof(header), 0);
```
Retrieve custom headers

- At the receiver side, `recv()`, `recvfrom()` place the data in plain buffers

- We want to retrieve easily the packet header fields

- This can be easily accomplished with pointers
Retrieve custom headers

```c
uint8_t buffer[1024];
struct custom_header *header;
uint8_t *data;
recv(sd, buffer, 1024, 0);
header = (struct custom_header *)buffer;
data = buffer + sizeof(struct custom_header);
```
Retrieve custom headers

```c
uint8_t buffer[1024];
struct custom_header *header;
uint8_t *data;
recv(sd, buffer, 1024, 0);
header = (struct custom_header *)buffer;
data = buffer + sizeof(struct custom_header);
```

- Are we done? **NO!!**
Retrieve custom headers

- Header is in network byte order. We must convert it back

```c
uint8_t buffer[1024];
struct custom_header *header;
struct custom_header header_host;
uint8_t *data;
recv(sd, buffer, 1024, 0);
header = (struct custom_header *)buffer;
data = buffer + sizeof(struct custom_header);
header_host.seq_number = ntohl(header->seq_number);
header_host.type = ntohs(header->type);
header_host.version = header->version;
```