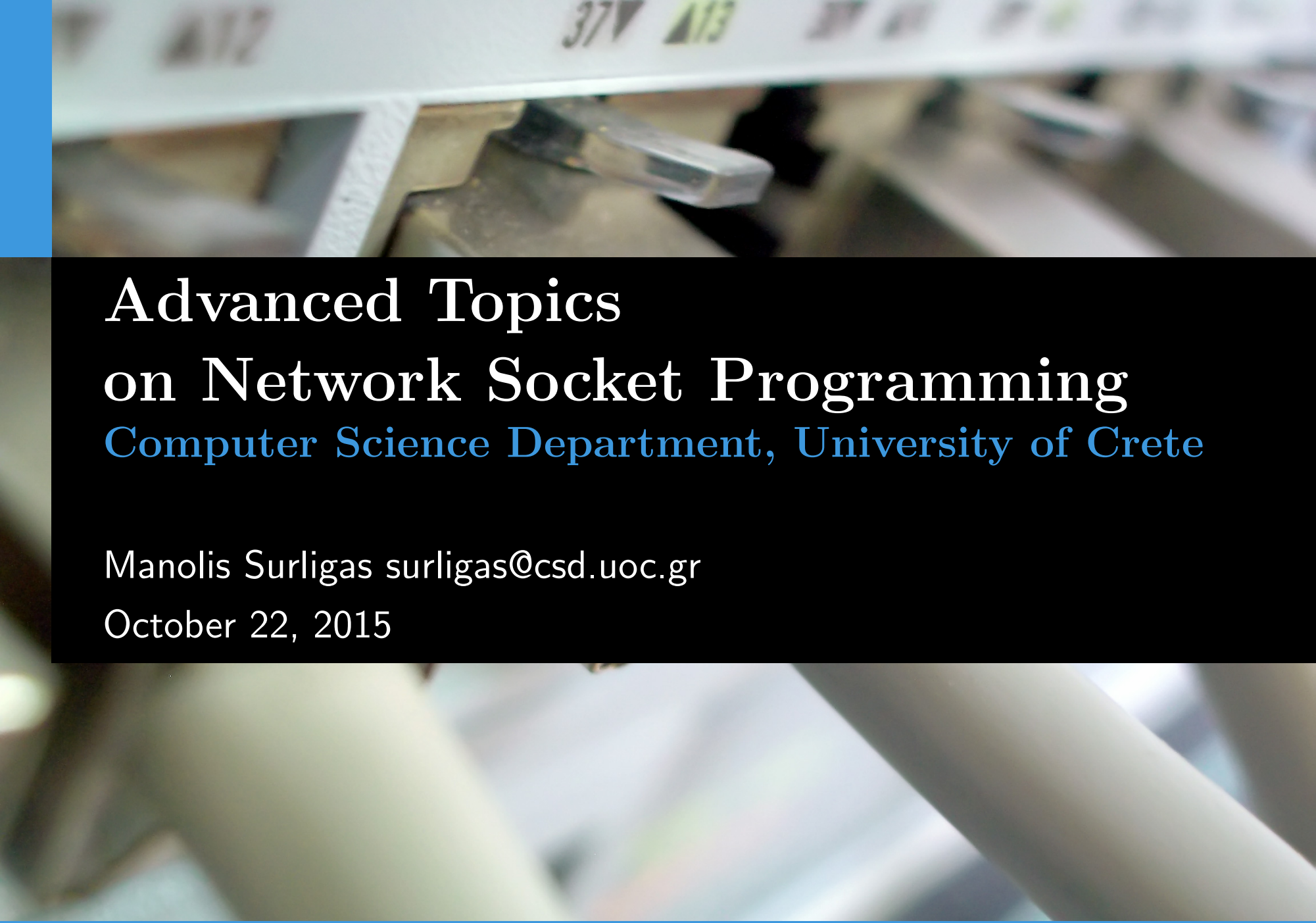


Advanced Topics on Network Socket Programming

Computer Science Department, University of Crete

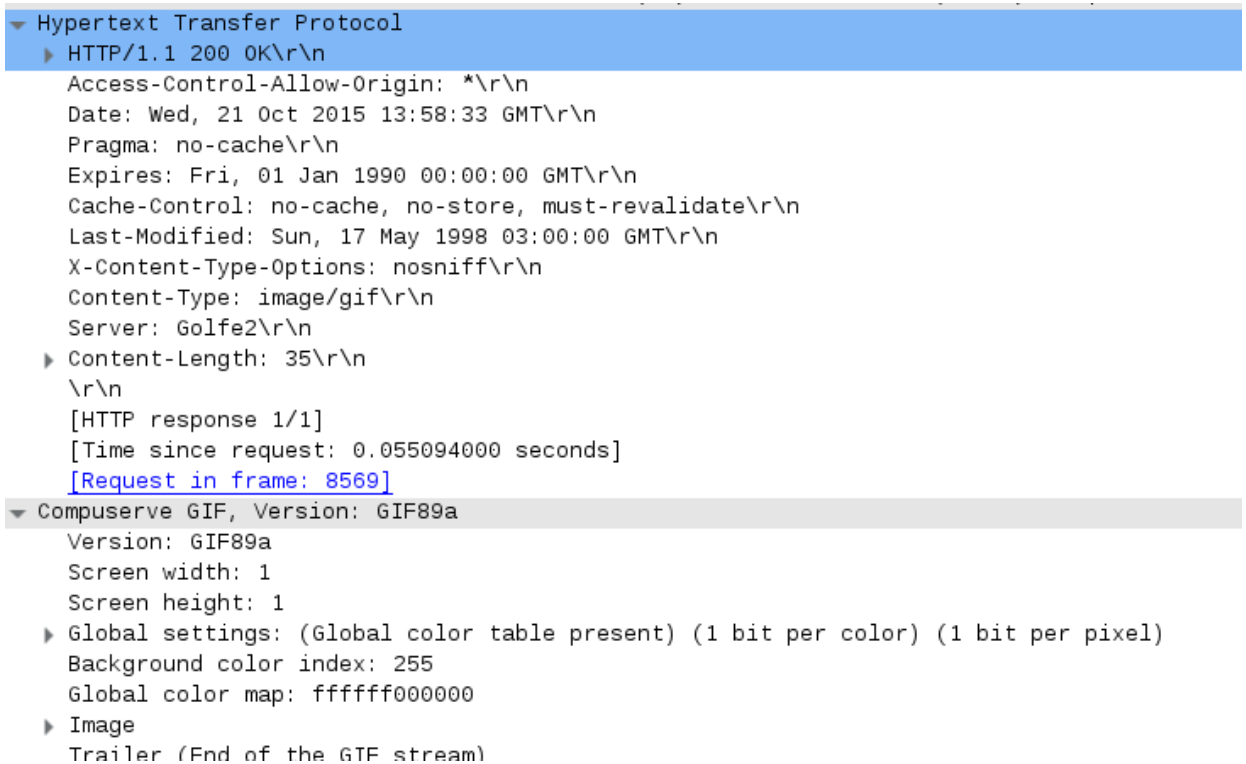
Manolis Surligas surligas@csd.uoc.gr

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

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

```
struct custom_header{
    uint32_t seq_number;
    uint16_t type;
    uint8_t  version;
};
```

Create custom Packet headers

- Fill in the appropriate values

```
/* 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

```
/* 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

```
/* 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**

```
/* 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**

```
/* 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

```
/* 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

```
/* 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!

```
/* 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

```
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

```
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

```
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;
```