

第三讲 HTTP/HTTPS

盛羽

中南大学计算机学院

shengyu@csu.edu.cn

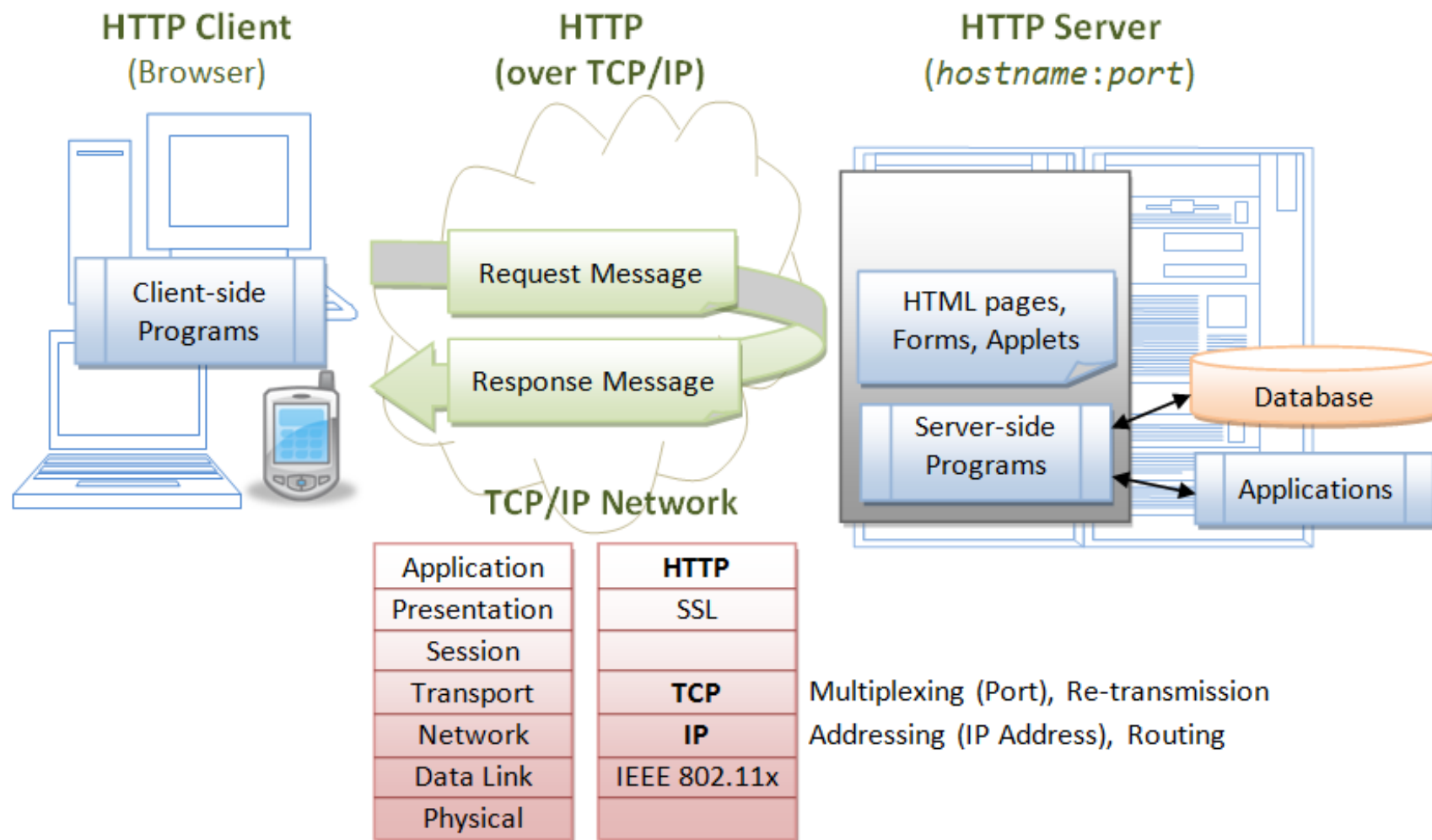
1. HTTP
2. HTTP的发展历程
3. HTTPS

1. HTTP协议

- `www.csu.edu.cn`
- `http://www.csu.edu.cn:80/index.htm`
- `https://www.csu.edu.cn:443/index.html`
- `scheme://host.domain:port/path/filename`

1. HTTP协议

■ HTTP: Hypertext Transfer Protocol (超文本传输协议)



■ 协议格式

METHOD /path - to - resource HTTP/Version-number
Header-Name-1: value
Header-Name-2: value
Optional request body

请求报文

Http/version-number	status code	message
Header-Name-1: value		
Header-Name-2: value		
Optional	Response body	

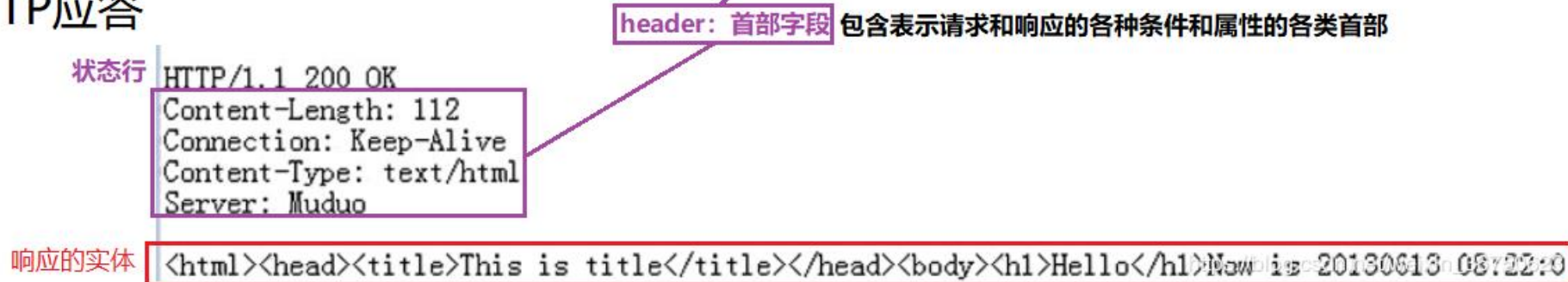
响应报文

■ 协议格式

HTTP请求

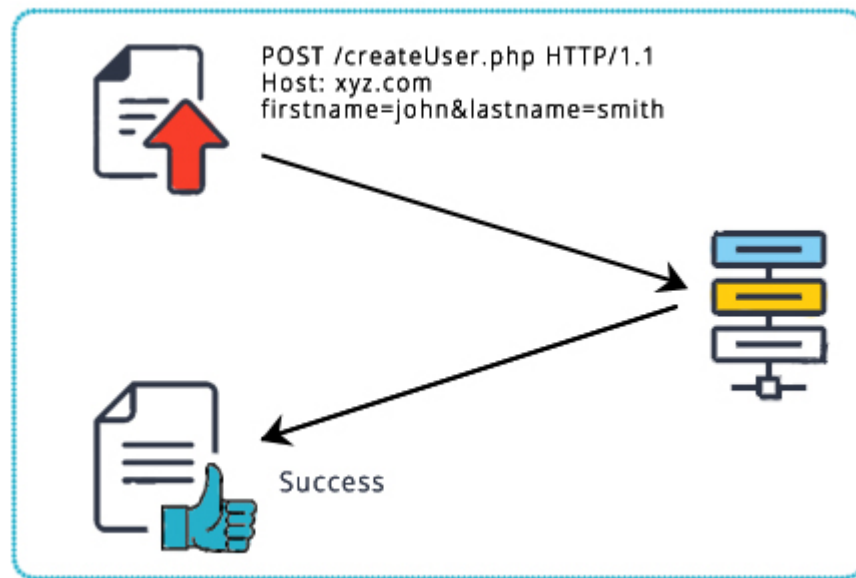
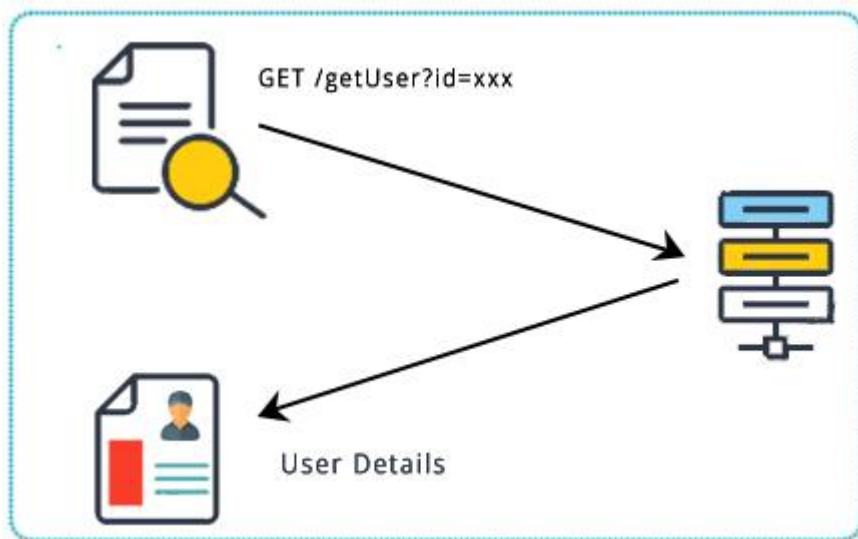


HTTP应答



■ 请求方法

■ GET vs POST



■ 响应状态码

分类	分类描述
1**	信息，服务器收到请求，需要请求者继续执行操作
2**	成功，操作被成功接收并处理
3**	重定向，需要进一步的操作以完成请求
4**	客户端错误，请求包含语法错误或无法完成请求
5**	服务器错误，服务器在处理请求的过程中发生了错误

1. HTTP协议

■ 响应状态码

- 200 OK //客户端请求成功
 - 400 Bad Request //客户端请求有语法错误, 不能被服务器所理解
 - 304 Not Modified //所请求的资源未修改, 不会返回任何资源
 - 404 Not Found //请求资源不存在, eg: 输入了错误的URL
 - 500 Internal Server Error //服务器发生不可预期的错误
-
- eg: HTTP/1.1 200 OK (CRLF)

1. HTTP
2. HTTP的发展历程
3. HTTPS

2. HTTP的发展历程

■ 发展历程

HTTP/0.9

HTTP/1.0

HTTP/1.1

HTTP/2.0

HTTP/3.0

2. HTTP的发展历程

■ HTTP/0.9 — 单行协议

■ HTTP 的首个版本

— 一个简单的 客户端/服务器、请求/响应、对 telnet 友好的协议

■ 请求: 单行 (method + 所请求文档的 path)

■ 支持的方法: 仅 GET

■ 响应类型: 仅超文本

■ 响应后马上结束的连接

■ 没有 HTTP headers (无法传输其他内容类型的文件), 没有 status/error 代码, 没有 URLs, 没有版本控制

```
GET /index.html
```

```
<html>  
  <body>Hello World</body>  
</html>
```

■ HTTP/1.0 — 构建可扩展性

- 对浏览器友好的协议
- 提供了对请求和响应都包含丰富元数据的 header 域 (HTTP 版本号、status code 和 content type)
- 响应: 不再只限于超文本 (Content-Type 头部提供了传输 HTML 之外文件的能力 — 如脚本、样式或媒体文件)
- 支持的方法: GET, HEAD, POST
- 响应后马上结束的连接

2. HTTP的发展历程

```
GET / HTTP/1.0
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5)
Accept: */*
```

```
HTTP/1.0 200 OK
Content-Type: text/plain
Content-Length: 137582
Expires: Thu, 05 Dec 1997 16:00:00 GMT
Last-Modified: Wed, 5 August 1996 15:55:28 GMT
Server: Apache 0.84

<html>
  <body>Hello World</body>
</html>
```

2. HTTP的发展历程

■ HTTP/0.9 和 HTTP/1.0 的主要问题

— 为每个请求建立一个新连接

- HTTP/0.9 和 HTTP/1.0 都需要为每次请求建立一个新的连接（并在收到对应的响应后立即关闭该连接）。
- 每次新连接建立时，都要经历一遍 TCP 三次握手。

2. HTTP的发展历程

■ HTTP/1.1 — 标准化的协议

■ 这是当前仍广泛使用的 HTTP 版本

■ 特点

- 持久连接
- 管道机制：FIFO
- Content-Length 字段
- 分块传输编码

■ 问题

- "队头堵塞" (Head-of-line blocking)
 - 管道FIFO, 可以发多个请求, 但是响应是根据请求顺序返回
- 必须是幂等请求
 - 意外中断时候, 客户端需要把未收到响应的请求重发, 非幂等请求, 会造成资源破坏

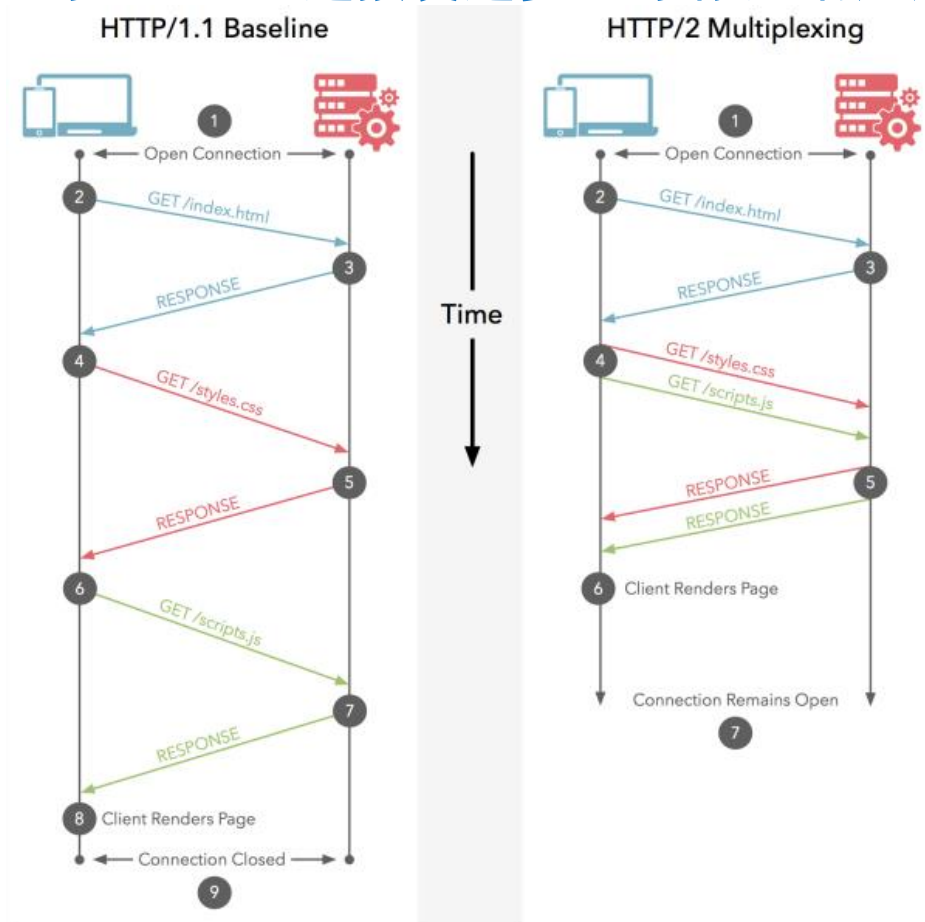
■ 解决方案

- 一个域名6个链接
- 域名分片

■ HTTP2.0

■ 多路复用 (Multiplexing)

- 允许同时通过单一的 HTTP/2 连接发起多重的请求-响应消息



■ HTTP 3.0

■ QUIC (Quick UDP Internet Connections)

- 基于UDP的传输层协议

■ 传输与建立连接上的优势

- 避免前序包阻塞 (HOL阻塞)

- 零RTT建立连接

■ 优雅的丢包处理

- FEC前向纠错

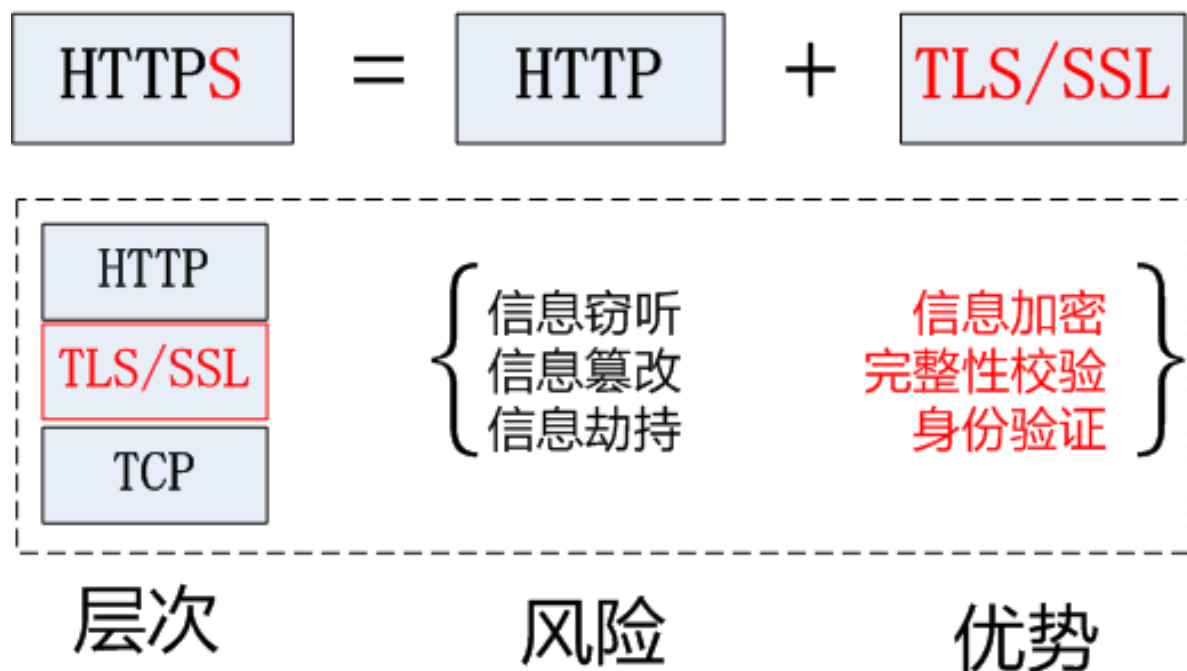
- 关键包发送多次

- 快速重启会话

1. HTTP
2. HTTP的发展历程
3. HTTPS

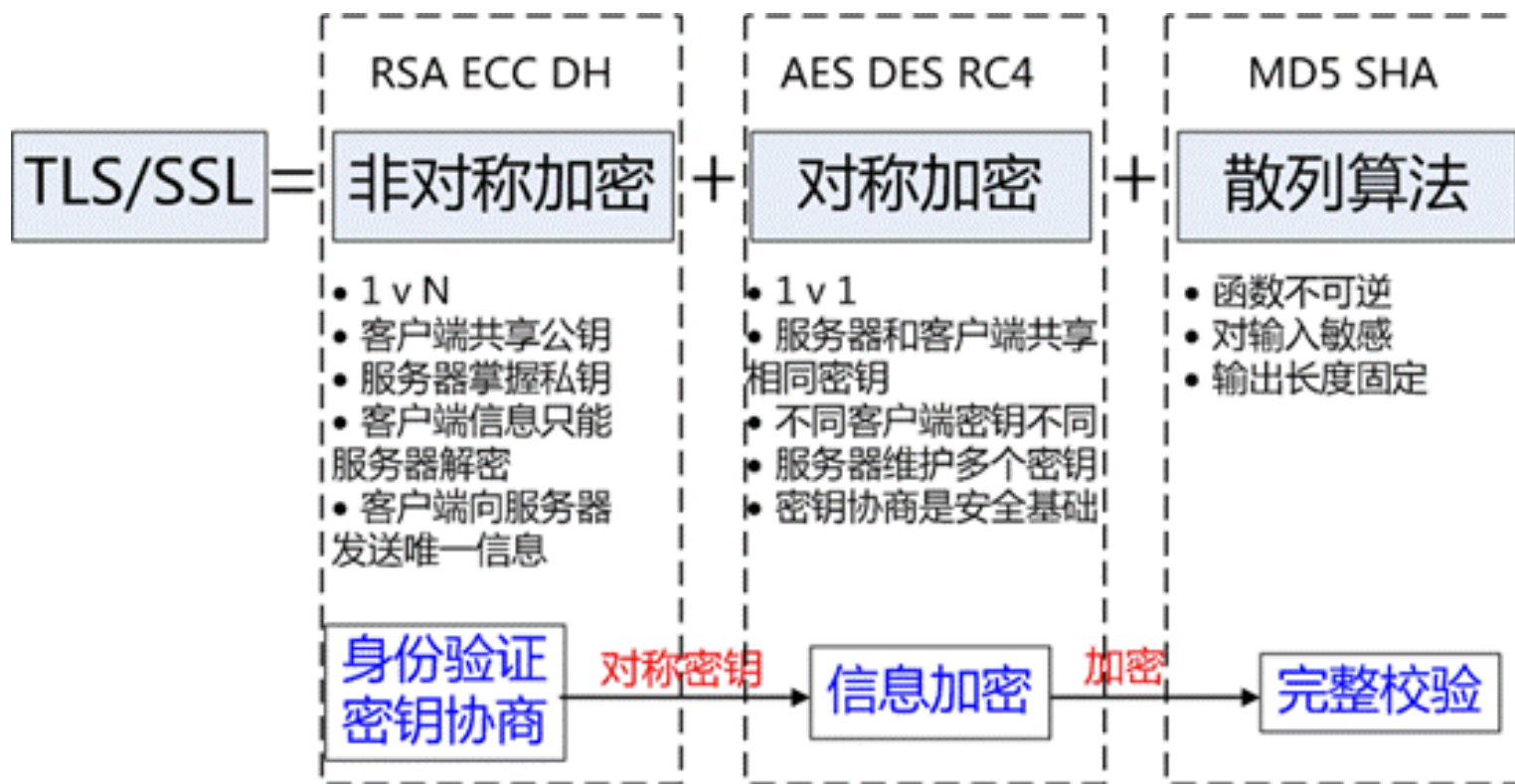
■ HTTPS

- Hypertext Transfer Protocol over Secure Socket Layer
- HTTP的安全版，其安全基础是SSL协议
- 默认使用TCP端口443



■ TLS/SSL的功能实现

- 主要依赖于三类基本算法：散列函数 Hash、对称加密和非对称加密



身份验证CA和证书

