

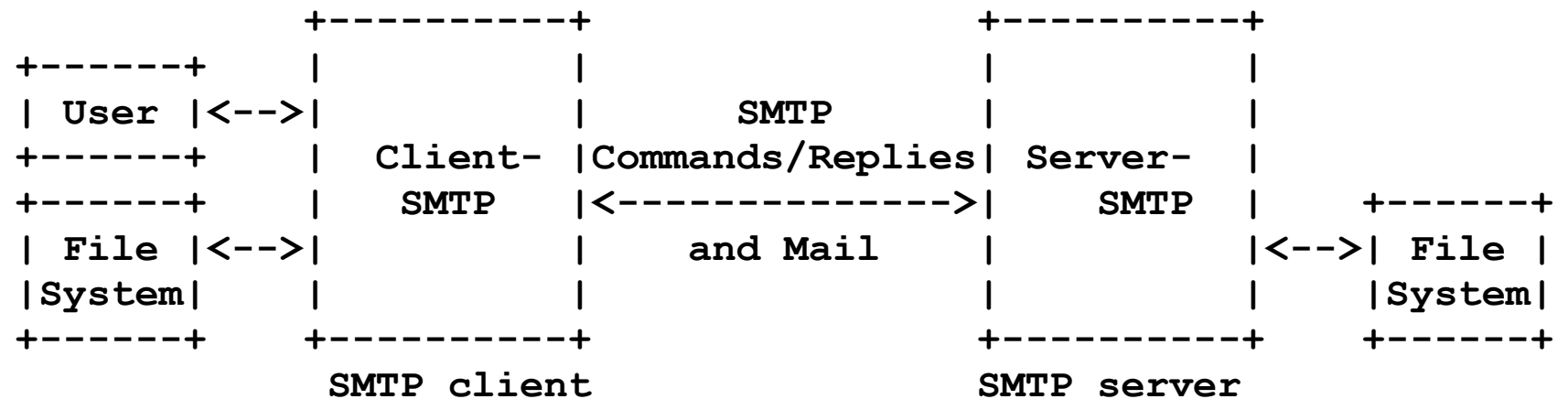
Sieci komputerowe

Wykład 14 Warstwa aplikacji

Przykład protokołu warstwy aplikacji

- Protokół SMTP (ang. Simple Mail Transport Protocol)
 - RFC 821, RFC 5321
- Model klient-serwer
 - MTA (ang. Mail Transfer Agent)
 - klient

Model protokołu



Budowa SMTP

- Komunikacja między dwoma MTA wykorzystuje jedynie tryb 7 bit ASCII
- Polecenia przesyłane są od klienta do serwera, a serwer odpowiada na nie, podając kody odpowiedzi
- SMTP korzysta z transportu TCP

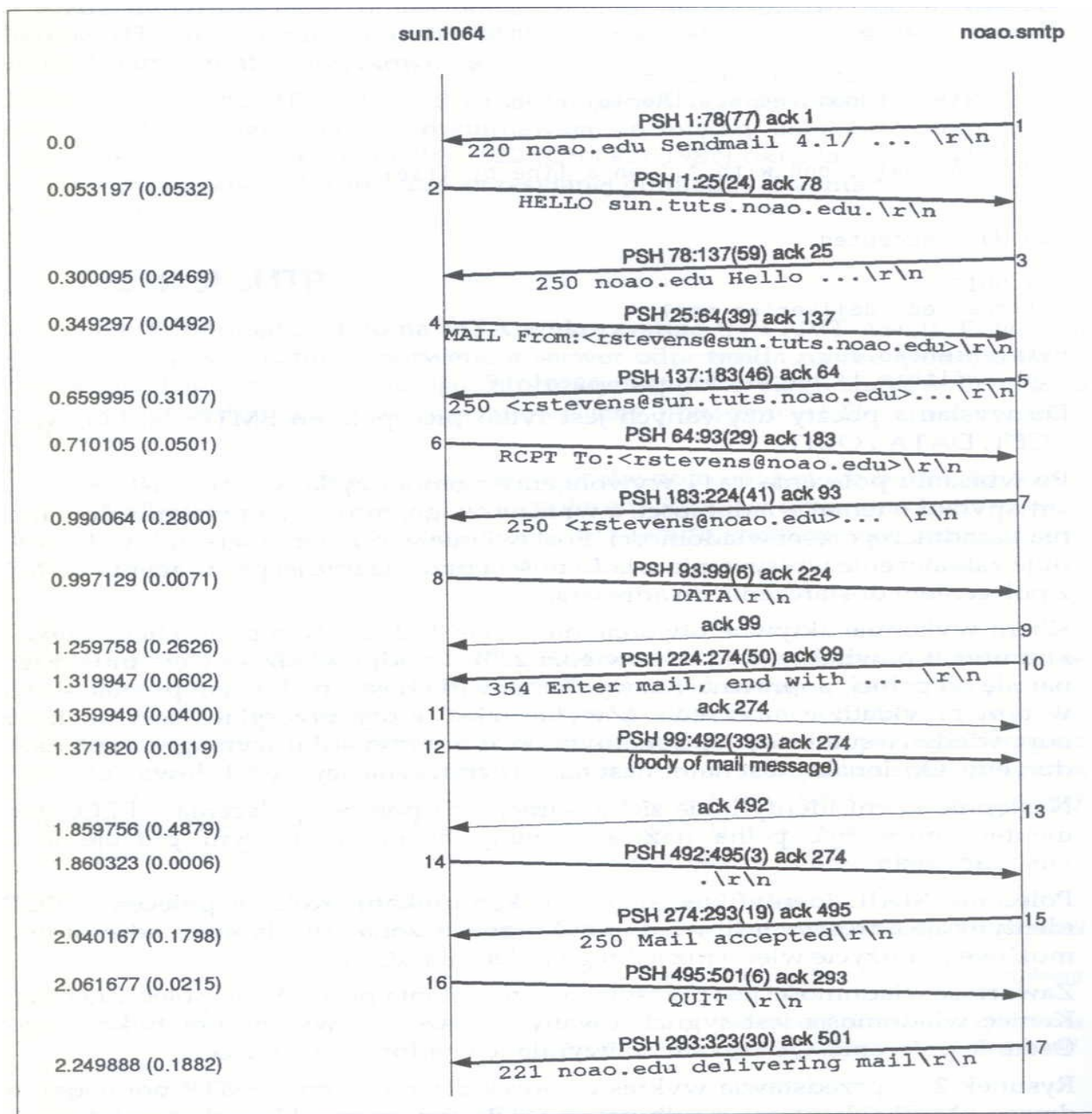
Rozmowa z serwerem SMTP

```
basa@cia:~$ telnet duch 25
Trying 10.1.3.2...
Connected to duch.mimuw.edu.pl.
Escape character is '^]'.
220 duch.mimuw.edu.pl ESMTP Postfix
HELO cia.mimuw.edu.pl.
250 duch.mimuw.edu.pl
MAIL FROM: <m.basa@mimuw.edu.pl.>
250 Ok
RCPT TO: <basa@students.mimuw.edu.pl.>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
To jest tresc wiadomosci.
250 2.6.0 Ok, id=03974-31, from MTA([127.0.0.1]:10025): 250
Ok: queued as 438993380D9
```

Wysyłanie wiadomości

```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.org
S: 250 Hello relay.example.org, I am glad to meet you
C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: Alice Example <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 Jan 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

Transport komunikatów SMTP



Rysunek 28.2 Podstawy działania SMTP przy dostarczaniu poczty

SMTP- prosty protokół warstwy aplikacji

- Jak widać z powyższego przykładu, do wysłania poczty potrzeba jedynie kilku poleceń

Kody odpowiedzi

- Cyfry kodu odpowiedzi mają ściśle określone znaczenie. Pierwsza cyfra określa generalnie, czy odpowiedź jest dobra, zła, czy niekompletna:
 - 2xx - odpowiedz pozytywna,
 - 3xx - odpowiedz częściowo pozytywna, odbiorca oczekuje na dodatkowe dane,
 - 4xx - odpowiedz częściowo negatywna, odbiorca w tej chwili nie może przeprowadzić żadnej akcji,
 - 5xx - odpowiedz jednoznacznie negatywna.

Druga cyfra kodu odpowiedzi

- Druga cyfra przyporządkowuje odpowiedź do określonej kategorii odpowiedzi lub błędu:
- x0x - błąd składniowy polecenia,
- x1x - odpowiedź informacyjna,
- x2x - dotyczy połączenia lub sesji,
- x3x - nie określone,
- x4x - nie określone,
- x5x - dotyczy statusu odbiorcy.

Trzecia cyfra kodu odpowiedzi

- Znaczenie trzeciej cyfry nie jest wyspecyfikowane. Pełni ona rolę uściślającą dla dwóch poprzednich cyfr.

Kody odpowiedzi

- 214 Help message (Information on how to use the receiver or the meaning of a particular non-standard command)
- 220 Service ready
- 221 <domain> Service closing transmission channel
- 250 Requested mail action okay, completed
- 251 User not local, will forward to <forward-path>

Kody odpowiedzi c.d.

- 354 Start mail input
- 450 Requested mail action not taken: mailbox unavailable
- 451 Requested action aborted: local error in processing
- 452 Requested action not taken: insufficient system storage
- 501 Syntax error in parameters or arguments
- 502 Command not implemented
- 503 Bad sequence of command
- 504 Command parameter not implemented
- 550 akcja wysyłania wiadomości nie może być wykonana

Rozszerzenia SMTP (ESMTP)

```
basa@cia:~$ telnet duch 25
Trying 10.1.3.2...
Connected to duch.mimuw.edu.pl.
Escape character is '^]'.
220 duch.mimuw.edu.pl ESMTP Postfix
ehlo cia.mimuw.edu.pl.
250-duck.mimuw.edu.pl
250-PIPELINING
250-SIZE 10485760
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250 8BITMIME
```

Serwer warstwy aplikacji, konfiguracja i funkcje na przykładzie Apache

- Obsługuje protokołów http, https,
- wirtualne hosty,
- kontrolę dostępu,
- ...

Źródła

- Można pobrać z *httpd.apache.org*

Kompilacja ze źródeł

- `configure --prefix=/home/staff/lab/basa/apache --enable-ssl`
- `make`
- `make install`
- główny plik konfiguracyjny:
 - `apache/conf/httpd.conf`

Uruchomienie serwera

- `./apache/bin/apachectl -f
./apache/conf/httpd.conf -k start`
- `-k restart` restartuje uruchomiony serwer

Konfiguracja

- conf/httpd.conf
- Jedna z najważniejszych dyrektyw:
<Directory ...>
 - pozwala określać reguły dostępu do katalogu
 - można stosować listy dostępu bazujące na adresach IP lub nazwach
 - wewnątrz można używać dyrektyw Allow, Deny

Kontrola dostępu

- Order Allow,Deny
 - porządek sprawdzania reguł – najpierw są sprawdzanie reguły *Allow*
- Order Deny,Allow
 - najpierw są sprawdzanie reguły *Deny*

Order

Match	Allow,Deny result	Deny,Allow result
Match Allow only	Request allowed	Request allowed
Match Deny only	Request denied	Request denied
No match	Default to second directive: Denied	Default to second directive: Allowed
Match both Allow & Deny	Final match controls: Denied	Final match controls: Allowed

Przykłady

Wszystkie hosty w domenie apache.org mają dostęp; wszystkie inne nie.

```
Order Deny,Allow  
Deny from all  
Allow from apache.org
```

Kolejny przykład

Wszystkie hosty w apache.org mają dostęp, z wyjątkiem tych w poddomenie foo.apache.org. Wszystkie hosty poza apache.org nie mają dostępu.

```
Order Allow,Deny  
Allow from apache.org  
Deny from foo.apache.org
```

Kolejny przykład

Jaki będzie efekt poniższego zapisu?

```
<Directory /www>  
Order Allow,Deny  
</Directory>
```


Autoryzacja

```
AuthType basic  
AuthName "Chronione zasoby"  
AuthUserFile /home.../.htpasswd  
require valid-user
```

Virtual hosts

- Umożliwia obsługę wielu domen przez jeden serwer
- Dyrektywa virtualhost
- Jeśli chcemy korzystać z https, wymagany jest osobny adres IP na każdą domenę

Virtualhost - przykład

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.example.com
    DocumentRoot
"/home/staff/lab/basa/apache2/docs/dummy-
host.example.com"
    ServerName dummy-host.example.com
    ServerAlias www.dummy-host.example.com
    ErrorLog "logs/dummy-host.example.com-error_log"
    CustomLog "logs/dummy-host.example.com-
access_log" common
</VirtualHost>
```