ZESZYTY ETI ZESPOŁU SZKÓŁ W TARNOBRZEGU
Nr 2 Seria: Teleinformatyka 2013

Mateusz Gaweł

Zespół Szkół im. ks. S. Staszica w Tarnobrzegu

PROTOKÓŁ I SERWER HTTP APACHE JAKO PRZYKŁAD SERWERA HTTP PRZYKŁADY KOMUNIKACJI Z SERWEREM HTTP

Streszczenie

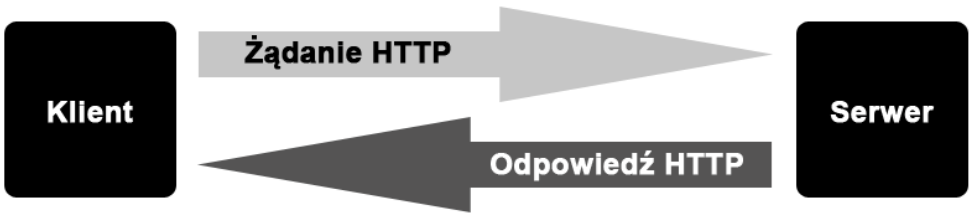
Praca zawiera informacje dotyczące komunikacji z serwerem HTTP (Hypertext Transfer Protocol). Poniżej opisany został protokół HTTP, który używany jest do komunikacji klienta z serwerem HTTP. W pracy przedstawiono sposoby komunikacji z wykorzystaniem protokołu HTTP oraz zaprezentowano jej przykłady. Ponadto w pracy znaleźć można informacje dotyczące bezpieczeństwa podczas korzystania z protokołu HTTP.

1. WSTĘP

W roku 1990 opublikowany został projekt budowy sytemu hipertekstowego WWW (World Wide Web), obsługiwanego przy pomocy przeglądarki internetowej używając architektury klient-serwer. Użycie hipertekstu pozwoliło użytkownikom przeglądającym stronę na podążanie za zamieszczonymi na niej hiperłączami, które przenoszą do innych udostępnionych w sieci dokumentów. System WWW zaprojektowano, aby zbierać zasoby ludzkiej wiedzy i udostępniać je innym ludziom. Przełomem było połączenie WWW z Internetem. Wtedy opracowano język programowania stron WWW (HTML – HyperText Markup Language) oraz protokół do transmisji stron internetowych – HTTP. Dzięki temu krokowi Internet stał się ogólnodostępny i każdy mógł z niego korzystać.

1.1 Protokół HTTP

Protokół HTTP to podstawowy protokół WWW. Schemat działania protokołu wygląda następująco: klient wysyła żądanie do serwera, on je interpretuje i odpowiada na zadane zapytanie. Protokół HTTP jest bezstanowy. Oznacza to, że protokół nie utrzymuje sesji połączeniowej między klientem a serwerem . Po zakończeniu przetwarzania żądania klienta, serwer kończy połączenie, a wszystkie dane przesłane w danej sesji zostają usunięte. Protokół HTTP opiera się na modelu TCP/IP i pracuje na porcie 80.[1]



Rys.1. Schemat funkcjonowania HTTP

1.2 Bezpieczeństwo HTTP

Protokół HTTP jest protokołem niezwykle prostym. Niestety ta prostota niesie problemy z bezpieczeństwem. Dane przesyłane protokołem HTTP są jawne, przez to można „podsłuchać" je specjalnymi programami. Jest to niebezpieczne, gdyż poufne dane (np.: hasła dostępu, numery kont, tajemnice służbowe) mogą zostać przechwycone przez osoby niepowołane. W celu poprawy bezpieczeństwa opracowani szyfrowaną wersją protokołu HTTP o nazwie HTTPS (Secure HTTP). Protokół HTTPS szyfruje przesyłane dane za pomocą systemu SSL (Secure Socket Layer). Mechanizm udoskonalonego protokołu przedstawia się następująco: klient, gdy nawiązuje połączenie, otrzymuje certyfikat, który zawiera dane indentyfikacyjne i klucz szyfrujący. Następnie rozpoczyna się transmisja danych zaszyfrowanych tym kluczem. Odszyfrowywanie danych odbywa się dopiero wtedy, gdy trafią one do odbiorcy (klienta lub serwera). Warto wspomnieć, że certyfikat może mieć ograniczoną datę ważności. Po upłynięciu takiego czasu musi zostać wygenerowany nowy certyfikat z nowymi danymi. Krytycznym momentem komunikacji poprzez HTTPS jest wysłanie przez serwer klucza, który może być przechwycony przez osoby postronne. Protokół HTTPS pracuje na porcie 443.[2]

1.3 Zapytania HTTP

Protokół HTTP do komunikacji z serwerem używa zapytań i nagłówków. Polecenia od nagłówków różnią się tym, że te pierwsze wysyłają główne żądania do serwera (np.: pobranie dokumentu), a te drugie do polecenia dołączają dodatkowe informacje (np.: dane o przeglądarce klienta.) W protokole HTTP do komunikacji z serwerem używane są poniższe polecenia:[3]

- GET – metoda, która pozwala na pobranie zasobu wskazanego przez adres,
- HEAD – pobiera informacje o zasobie, stosowane do sprawdzenia dostępności zasobu,
- PUT – przyjęcie danych w postaci pliku przesyłanego od klienta do serwera,
- POST – przyjęcie danych od klienta do serwera,
- DELETE – usunięcie żądanego zasobu,
- OPTIONS – informacja o opcjach i wymaganiach istniejących w kanale komunikacyjnym,
- TRACE – diagnostyka kanału komunikacyjnego,
- CONNECT – żądanie przeznaczone dla serwerów pośredniczących, pełniących funkcję tunelowania.

1.4 Nagłówki HTTP

Aby klient mógł „porozmawiać" z serwerem używane są nagłówki HTTP:

- Accept – służy do określania listy akceptowalnych przez przeglądarkę typów dokumentów,
- Accept-Charset – określa preferowane przez przeglądarkę formaty kodowania,
- Accept-Encoding – ten nagłówek określa kodowanie, za pomocą którego zostanie przesłana zawartość,
- Accept-Language – określa w jakim języku użytkownik przeglądarki chce czytać strony,
- Accept-Ranges – ten nagłówek jest ustawiany przez przeglądarki i programy ułatwiające pobieranie plików. Określa czy klient potrafi odczytywać pliki przesyłane w częściach,
- Allow – określa metody HTTP obsługiwane przez serwer,
- Authorization – potwierdzenie uwierzytelnienia dla autoryzacji HTTP,
- Cache-Control – określa, czy przeglądarka może przechowywać dane podręczne,
- Connection – ten nagłówek mówi, czy połączenia ma być zamknięte po obsłużeniu zapytania, czy ma czekać na kolejne polecenia,
- Content-Encoding – typ kompresji stosowany prze przesyle przez HTTP,
- Content-Language – język dokumentu przesłanego przez serwer,
- Content-Length – określa długość przesyłanej zawartości w bajtach. Nagłówek przesyłany przez serwer,
- Content-Location – alternatywna lokacja pliku ze zwróconą treścią,
- Content-Range – informacja, jaki zakres pliku został przesłany, stosowany z kodem odpowiedzi 206,
- Content-Type – ten nagłówek informuje przeglądarkę w jakim formacie i stronie kodowej wysyłany jest dokument,
- Cookie – w tym nagłówku przesyłane są ciasteczka przesyłane w przeglądarce,
- Date – ten nagłówek określa datę na serwerze,
- Expires – data powyżej której dokument będzie już nieaktualny,
- Host – nagłówek określający dla jakiej domeny określony został adres,
- If-Modified-Since – nagłówek nakazujący serwerowi przesłanie dokumentu, tylko wtedy, gdy został zmodyfikowane od podanej daty,

- Last-Modified – nagłówek serwera, informujący o ostatniej aktualizacji dokumentu,
- Location – wymusza przekierowanie na podany adres,
- Proxy-Authenticate – żądanie autoryzacji dostępu do serwera pośredniczącego,
- Proxy-Authorization – potwierdzenie uwierzytelnienia dla serwera pośredniczącego,
- Range – określa na jaką cześć pliku oczekuje przeglądarka,
- Referer – nagłówek, który informuje o stronie, z której nastąpiło przekierowanie,
- Refresh – ustawia automatyczne przekierowanie w przeglądarce na podany adres po określonym czasie,
- Retry-After – używany razem z kodem odpowiedzi 503, określa czas po którym serwer będzie w stanie odpowiedzieć,
- Server – nagłówek identyfikujący serwer i użyte w nim oprogramowanie,
- Set-Cookie – nagłówek wysłany od serwera, nakazuje przeglądarce ustawienie określonych ciasteczek,
- Transfer-Encoding – określa w jaki sposób serwer przesłał zawartość dokumentu,
- User-Agent – identyfikuje przeglądarkę,
- WWW-Authenticate – przesyłany razem z kodem 401, określa w jaki sposób ma być przeprowadzona identyfikacja użytkownika.

Istnieją jeszcze inne nagłówki HTTP, które nie zostały umieszczone w pracy. Można je znaleźć w sieci.

1.5 Odpowiedzi HTTP

Oprócz treści żądanego dokumentu serwer HTTP odsyła komunikaty składające się z trzech cyfr. Pierwsza z nich oznacza rodzaj kodu odpowiedzi. Wzorce znajdują się poniżej:[4]

- 1XX – kody informacyjne,
- 2XX – kody powodzenia,
- 3XX – kody przekierowań,
- 4XX – kody błędu aplikacji klienta,
- 5XX – kody błędu serwera.

Teraz czas na dokładne wyjaśnienie poszczególnych kodów:

- 100 – kontynuuj, prośba o dalsze wysyłanie zapytania,
- 101 – zmiana protokołu,
- 110 – przekroczono czas połączenia, serwer zbyt długo nie odpowiada,
- 111 – serwer odrzucił połączenie,
- 200 – zawartość żądanego dokumentu,
- 201 – wysłany dokument został zapisane na serwerze,
- 202 – zapytanie zostało przyjęte do obsłużenia, lecz jego zrealizowanie jeszcze się nie odbyło,
- 203 – zwrócona informacja nie odpowiada dokładnie odpowiedzi pierwotnego serwera, lecz została utworzona z lokalnych bądź zewnętrznych kopii,
- 204 – serwer zrealizował zapytanie i nie potrzebuje zwracać żadnej treści,
- 205 – serwer zrealizował zapytanie i klient powinien przywrócić pierwotny wygląd dokumentu,
- 206 – serwer zrealizował tylko część zapytania typu GET,
- 300 – istnieje więcej niż jeden sposób obsługi danego zapytania,
- 301 – żądany zasób zmienił swój adres i w przyszłości powinien być on szukany pod nowym wskazanym adresem,
- 302 – żądany zasób jest chwilowo dostępny pod innym adresem, a przyszłe odwołania do zasobu powinny być kierowane pod adres pierwotny,
- 303 – odpowiedź na żądanie znajduje się pod innym adresem i tam klient powinien się skierować.
- 304 – zawartość zasobu nie podległa zmianie według warunku przekazanego przez klienta,
- 305 – do żądanego zasobu trzeba odwołać się przez serwer proxy,
- 306 – kod aktualnie nieużywany, zarezerwowany dla starszych wersji,
- 307 – żądany zasób znajduje się chwilowo pod innym adresem,
- 308 – zbyt wiele przekserowań,
- 400 – żądanie nie może być obsłużone przez serwer z powodu błędnej składni zapytania,

- 401 – żądanie zasobu, który wymaga uwierzytelnienia,
- 402 – wymagana opłata, odpowiedź zarezerwowana na przyszłość,
- 403 – serwer zrozumiał zapytanie lecz konfiguracja bezpieczeństwa zabrania mu zwrócić żądany zasób,
- 404 – serwer nie odnalazł zasobu według podanego adresu, ani niczego co by wskazywało na istnienie takiego zasobu w przeszłości,
- 405 – metoda zawarta w żądaniu nie jest dozwolona dla wskazanego zasobu,
- 406 – zażądany zasób nie jest w stanie zwrócić odpowiedzi mogącej być obsłużonej przez klienta według informacji podanych w zapytaniu,
- 407 – wymagane uwierzytelnienie do serwera proxy,
- 408 – koniec czasu oczekiwania na żądanie, klient nie przesłał zapytania do serwera w określonym czasie,
- 409 – żądanie nie może być zrealizowane, ponieważ występuje konflikt z obecnym statusem zasobu,
- 410 – zażądany zasób nie jest dłużej dostępny i nie znany jest jego ewentualny nowy adres,
- 411 – serwer odmawia zrealizowania zapytania ze względu na brak informacji o długości treści,
- 412 – serwer nie może spełnić przynajmniej jednego z warunków zawartych w zapytaniu,
- 413 – całkowita długość zapytania jest zbyt długa dla serwera,
- 414 – długość zażądanego URI jest większa niż maksymalna oczekiwana przez serwer,
- 415 – serwer odmawia przyjęcia zapytania, ponieważ jego składnia jest niezrozumiała dla serwera,
- 416 – klient podał w zapytaniu zakres bajtowy, który nie może być zastosowany do wskazanego zasobu,
- 417 – oczekiwana wartość nie może być zwrócona,
- 500 – serwer napotkał niespodziewane trudności, które uniemożliwiły zrealizowanie żądania,
- 501 – serwer nie dysponuje funkcjonalnością wymaganą w zapytaniu,
- 502 – serwer – spełniający rolę bramy lub pośrednika – otrzymał niepoprawną odpowiedź od serwera nadrzędnego i nie jest w stanie zrealizować żądania klienta,
- 503 – serwer nie jest w stanie w danej chwili zrealizować zapytania klienta ze względu na przeciążenie,
- 504 – serwer – spełniający rolę bramy lub pośrednika – nie otrzymał w ustalonym czasie odpowiedzi od wskazanego serwera HTTP,
- 505 – serwer nie obsługuje bądź odmawia obsługi wskazanej przez klienta wersji HTTP.

Podsumowując HTTP jest protokołem prostym, który umożliwia komunikację typu klient-serwer. Komunikacja między nimi odbywa się za pomocą poleceń, nagłówków i zwracanych kodów odpowiedzi. Jest to praktyczne, ponieważ polecenia są zrozumiałe dla człowieka, dzięki temu można poznać działanie protokołu HTTP.

2. SERWERY HTTP

2.1 Apache jako serwer HTTP

Wcześniej opisany został serwer HTTP. Przykładem takiego serwera jest serwer Apache. Apache jest najpopularniejszym serwerem HTTP na świecie, pracuje na około 65% wszystkich serwerów stron internetowych. Główne cechy Apache'a to wielowątkowość, skalowalność i bezpieczeństwo. Wielowątkowość serwera polega na tym, że może on obsłużyć wielu użytkowników naraz. Serwer ma dużo możliwości konfiguracji, wszystkie dokonuje się w pliku „httpd.conf". W tym pliku można uruchomić też inne moduły, np. mod_rewrite, który służy do przepisywania adresów. W środowisku Apache można uruchomić parser PHP oraz serwer MySQL. Możliwości konfiguracji serwera Apache przedstawiają się następująco:

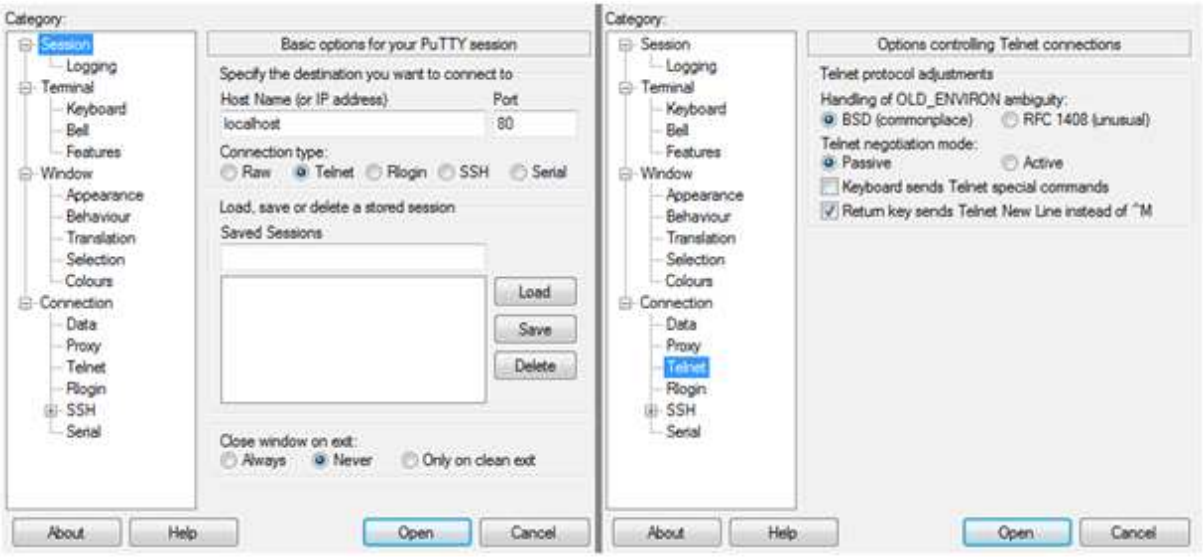
- środowisko serwera,
- podstawowe parametry sieciowe,
- lista dołączonych modułów,
- aliasy,
- określenie dostępu do zasobów,
- formaty i położenie dzienników serwera,
- moduły CGI (Common Gateway Interface),
- opcje automatycznego indeksowania katalogów,

- negocjacja treści,
- własne dokumenty błędów serwera,
- odwrotny DNS,
- konfiguracja wirtualnych hostów.

3. PRZYKŁADY

3.1 Komunikacja z serwerem HTTP za pomocą terminalu

Połączenie z serwerem HTTP zostanie nawiązane za pomocą programu Putty. Przed połączeniem jednak trzeba wprowadzić odpowiednie ustawienia:



Rys.2. Ustawienia Putty

Teraz czas na rozpoczęcie „rozmowy" z serwerem HTTP. W tym referacie rozpoczęta zostanie poleceniem GET, w celu pobrania zawartości strony. Warto zwrócić uwagę na to, że polecenie i wszystkie wpisane nagłówki zostaną wysłane dopiero po dwukrotnym wciśnięciu klawisza enter:

```
GET /http/http.html HTTP/1.1
Host: localhost

HTTP/1.1 200 OK
Date: Sat, 05 Jan 2013 18:43:08 GMT
Server: Apache/2.2.22 (Win32) PHP/5.3.13
Last-Modified: Sat, 05 Jan 2013 18:21:50 GMT
ETag: "8000000000032-61-4d28eab728c69"
Accept-Ranges: bytes
Content-Length: 97
Content-Type: text/html

<html>
        <head>
                <title>Test</title>
        </head>
        <body>
                <p>To dziala!</p>
        </body>
</html>
```

Rys.3. Pobranie zawartości strony

Jak widać powyżej po wydaniu polecenia został zwrócony kod odpowiedzi, nagłówki i treść dokumentu. Teraz czas na próbę wywołania dokumentu, który nie istnieje na serwerze:

```
GET /http/404.html HTTP/1.1
Host: localhost

HTTP/1.1 404 Not Found
Date: Thu, 03 Jan 2013 20:10:52 GMT
Server: Apache/2.2.22 (Win64) PHP/5.3.13
Content-Length: 211
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
                                                <html><head>
                                                            <title>404 Not Fou
nd</title>
          </head><body>
                      <h1>Not Found</h1>
                                            <p>The requested URL /http/404.html was
 not found on this server.</p>
                               </body></html>
```

Rys.4. Komunikat informujący o braku strony na serwerze

Teraz można spróbować czegoś innego. Przykładowo można wysłać nagłówek, którego serwer nie rozpozna:

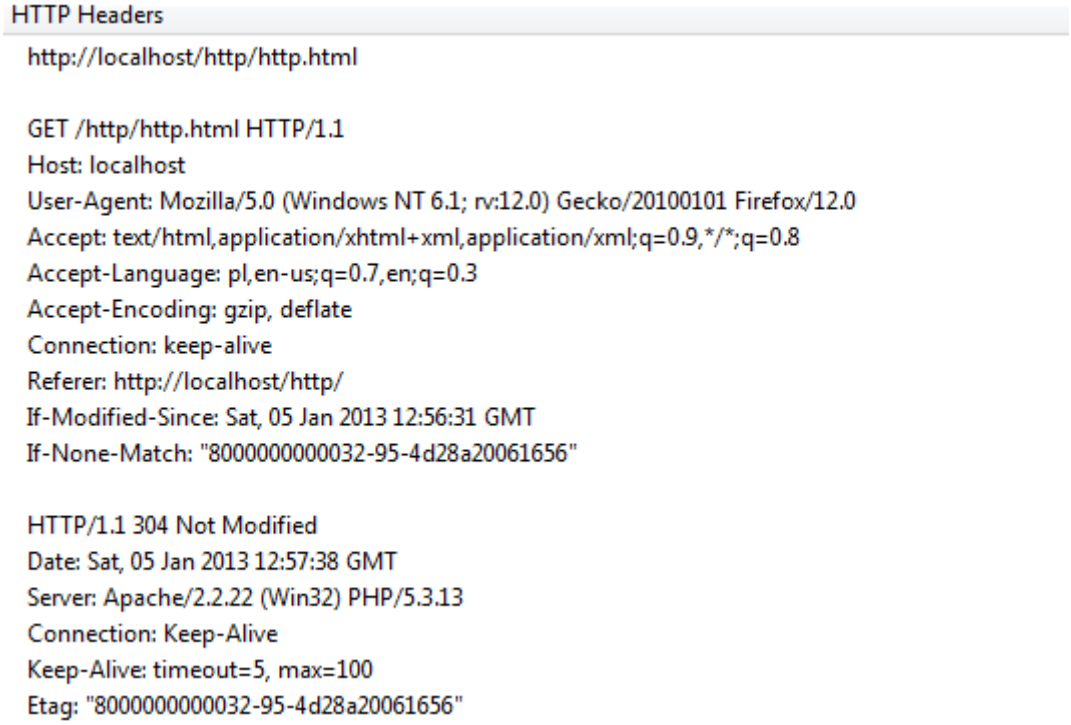
```
GET http/http.html HTTP/1.1
Bzdury: Bzdurki

HTTP/1.1 400 Bad Request
Date: Thu, 03 Jan 2013 20:13:37 GMT
Server: Apache/2.2.22 (Win64) PHP/5.3.13
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
                                                <html><head>
                                                            <title>400 Bad Req
uest</title>
             </head><body>
                         <h1>Bad Request</h1>
                                               <p>Your browser sent a request that
 this server could not understand.<br />
                                           </p>
                                              </body></html>
```

Rys.5. Komunikat informujący o błędzie składniowym

Zwrócony został stosowny błąd. Warto także zwrócić uwagę na część „HTTP/1.1”. Klient i serwer wysyłają ją, aby sprawdzić czy korzystają z tej samej wersji HTTP i czy mogą się porozumieć.

3.2 Komunikacja przeglądarki z serwerem HTTP

We wcześniejszym podrozdziale pokazana została komunikacja z serwerem HTTP poprzez terminal. Teraz ukazana zostanie komunikacja przeglądarki z serwerem. Przebiega ona podobnie do tej z użyciem terminalu, jednak wysyłane są dodatkowe nagłówki (np.: identyfikujące przeglądarkę).



Rys.6. Komunikaty wysyłane przy komunikacji przeglądarki z serwerem

Po przesłaniu nagłówków zostaje wyświetlona zawartość żądanego dokumentu.

3.3 Najważniejsze ustawienia konfiguracyjne serwera Apache (httpd.conf)

Wspomniano już, że serwer Apache posiada duże możliwości konfiguracji. Wszystkie znajdują się w pliku „httpd.conf". Najważniejsze dyrektywy konfiguracyjne oraz przykłady ich użycia:[6]

- ServerRoot „C:/serwer/WWW" – określa położenie plików serwera w drzewie katalogów,
- TimeOut 300 – określa w sekundach czas oczekiwania na żądanie klienta,
- KeepAlive On – określa, czy serwer ma podtrzymywać połączenie pomiędzy poleceniami klienta,
- MaxAliveRequest 100 – określa maksymalną ilość żądań, które mogą być obsłużone podczas jednego połączenia z klientem,
- KeepAliveTimeout 15 – zmienna, która określa ile sekund serwer będzie czekał na kolejne polecenia od klienta. Jeśli nie otrzyma żadnego polecenia po tym czasie połączenie zostanie zamknięte,
- Listen 80 Port nasłuchiwania serwera, domyślny port HTTP to 80,
- DirectoryIndex index.html index.php – ustawienie, określające jaki dokument ma zostać przesłany do klienta w przypadku, gdy żądanie dotyczy katalogu,
- AccesFileName .htaccess – dyrektywa określająca nazwę pliku konfiguracyjnego odczytywanego, przy każdym żądaniu. To ustawienie pozwala na dynamiczną zmianę konfiguracji serwera przez każdego użytkownika,
- <Files ~ „^\.ht">
Order allow, deny
Deny from all
</Files> – ten blok, nakazuje serwerowi blokowanie dostępu klientom, dla plików rozpoczynających się od „.ht".
- Alias /error/ „C:/serwer/WWW/inny_katalog/bledy – dyrektywa pozwala na przechowywanie dokumentów w innym miejscu, niż określone w DocumentRoot,
- ScriptAlias /php/ „C:/serwer/php/ – dyrektywa działa podobnie do Alias, przy czym określa, że w docelowym katalogu znajdują się skrypty CGI,
- ErrorDocument 404 /error/404.html – zmienna, która zmienia domyślne strony błędów,
- NameVirtualHost 127.0.0.1 – zmienna, która określa, że po podanym adresem IP znajdują się skonfigurowane serwery wirtualne,

- Include conf/test.conf – dyrektywa, która do głównego pliku dołącza określony plik, np.: konfigurację serwera wirtualnego.

4. ZAKOŃCZENIE

Protokół HTTP jest prostym protokołem, który służy do komunikacji w sieci WWW. Do jego obsługi używa się przeglądarek internetowych, jednak nic nie stoi na przeszkodzie, aby łączyć się z serwerami HTTP za pomocą terminalu. Serwer HTTP „rozmawia z klientem" za pomocą komend, nagłówków i kodów odpowiedzi. Istnieje wiele serwerów HTTP. Jedne są szybsze, ale mniej funkcjonalne, a inne zupełnie odwrotnie. Ich dobór zależy od potrzeb administratora serwisu.

BIBLIOGRAFIA

[1] Itpedia.pl (*Protokół HTTP*). *Ostatnia modyfikacja sie 2008.*
[2] karnet.up.wroc.pl/~jasj (*Jan Jełowicki*).
[3] Wikipedia.org (*Hypertext Transfer Protocol*) *Ostatnia modyfikacja lis2012.*
[4] Wikipedia.org (*Kod odpowiedzi HTTP*) *Ostatnia modyfikacja paź 2012.*
[5] Wikipedia.org (*Apache HTTP Server*) *Ostatnia modyfikacja gru 2012.*
[6] Jarosław Mężyk (*Konfiguracja serwera Apache*).

HTTP SERVER AND PROTOCOL
APACHE AS EXAMPLE OF HTTP SERVER
EXAMPLES OF COMMUNICATION WITH HTTP SERVER

Summary

The paper contains information about communication with HTTP (Hypertext Transfer Protocol) server. It's describes HTTP protocol, which is used to communication: client with server HTTP. It's presented ways how to communicate using the HTTP protocol, and examples. In addition, the work provides information about safety when using the HTTP protocol.