Fakulta Elektrotechniky a Informatiky Slovenskej Technickej Univerzity

Katedra Telekomunikácií

Komunikačné protokoly

HTTP a Secure HTTP protokol.

Šk. rok 2007/2008

Ivana Ferenczyová, Matej Fister

Krúžok TLK 5

HTTP protokol

Prehľad :

Hypertext Transfer Protocol (HTTP) je protokol aplikačnej vrstvy pre distribuované, spolupracujúce informačné systémy pre hypermédia. Je to všeobecný, jednoduchý protokol, ktorý môže byť použitý na rôzne úlohy, ktoré využívajú hypertext ako sú menné servery a systémy na distribuovanú správu objektov, prostredníctvom rozšírenia jeho požiadavkových metód, chybových kódov a hlavičiek. Hlavnou črtou protokolu HTTP je vytváranie a prevod reprezentácie dát, umožňujúca vytvárať systémy nezávislé od prenášaných dát.

HTTP sa používa World-Wide Web globálnou informačnou iniciatívou od roku 1990. Táto špecifikácia definuje protokol uvedený ako „HTTP/1.1“ , a je aktualizáciou dokumentu RFC 2068.

Účel :

Hypertext Transfer Protocol (HTTP) je protokol aplikačnej vrstvy pre distribuované, spolupracujúce informačné systémy pre hypermédia. HTTP sa používa World-Wide Web globálnou informačnou iniciatívou od roku 1990. Prvá verzia protokolu HTTP uvedená ako HTTP/0.9, bol jednoduchý protokol na prenos neupravených dát cez internet. HTTP/1.0 definovaný v RFC 1945, vylepšil protokol umožňujúcim správam, byť vo formáte ako MIME správy, obsahujúce metainformácie o prenášaných dátach a zmenách v sémantike požiadaviek/odpovedí. Avšak HTTP/1.0 dostatočne neberie do úvahy efekty hierarchických proxy, vyrovnávaciu pamäť, potrebu stáleho spojenia, alebo zdanlivých hostiteľov. Ako doplnok, rozšírenie neúplne implementovaných aplikácií volajúcich samých seba „HTTP/1.0“ bol nútený urobiť zmeny vo verzii protokolu, za účelom každej z dvoch komunikujúcich aplikácií určiť možnosti tej druhej.

Táto špecifikácia definuje protokol uvedený ako „HTTP/1.1“. Tento protokol zahrňuje prísnejšie požiadavky ako HTTP/1.0 za účelom zaistenia spoľahlivej implementácie jeho vlastností.

Praktický informačný systém potrebuje väčšiu funkčnosť ako jednoduchý prijímač, obsahujúcu vyhľadávanie , koncovú aktualizáciu a anotáciu. HTTP umožňuje otvorené konečné skupiny metód a hlavičiek, ktoré indikujú účel požiadavky. Vytvára disciplínu odkazov poskytovanú uniformovaným

identifikátorom zdrojov (Uniform Resources Identifier – URI) buď ako lokácií (Uniform Resources Locator - URL) alebo názvov (Uniform Resources Name - URN), na určenie zdroju, ktorá z metód bude použitá. Správy sú vkladané vo formáte podobnom tomu, ktorý používa internetová pošta definovanom vo viacúčelových rozšíreniach internetovej pošty (Multipurpose Internet Mail Extensions – MIME).

HTTP je taktiež využívaný ako všeobecný protokol na komunikáciu medzi užívateľskými agentmi a zástupcami (proxies)/bránami k ďalším internetovým systémom, ktoré obsahujú podporu pre SMTP, NNTP, FTP, Gopher a WAIS protokoly. V tomto smere, HTTP umožňuje jednoduchý prístup hypermedií k zdrojom prístupných z rozličných aplikácií.

Požiadavky :

Kľúčové slová „MUSÍ“, „NESMIE“, „POTREBNÉ“, „BUDE“, „NEBUDE“, „MAL BY“, „NEMAL BY“, „ODPORÚČANÉ“, „MOCT“, „VOLITELNÉ“ v tomto dokumente sú interpretované ako je popísané v RFC 2119.

Implementácia nie je vyhovujúca pokiaľ nesplňuje jednu alebo viac z „MUSÍ“ alebo „POTREBNÉ“ úrovní požiadaviek implementácie protokolu. Implementácia, ktorá spĺňa všetky „MUSÍ“ alebo „POTREBNÉ“ a všetky „MAL BY“ úrovne požiadaviek pre protokol sa nazýva „nepodmienene vyhovujúca“; tá, ktorá vyhovuje všetkým „MUSÍ“ úrovniam požiadaviek, ale nie všetkým „MAL BY“ úrovniam požiadaviek sa nazýva „podmienene vyhovujúca“.

Terminológia :

Táto špecifikácia používa viacero termínov vzťahujúcich sa na úlohy hrané účastníkmi a objektmi HTTP komunikácie.

Spojenie (connection)

Virtuálny okruh transportnej vrstvy ustanovený medzi dvomi programami za účelom komunikácie.

Správa (message)

Základná jednotka HTTP komunikácie, pozostávajúca zo štruktúrovanej postupnosti osmíc odpovedajúcej syntaxi definovanej v sekcii 4 a prenášanej cez spojenie.

Požiadavka (request)

HTTP správa požiadavky definovaná v sekcii 5.

Odozva (response)

HTTP správa odpovede definovaná v sekcii 6.

Zdroj (resource)

Objekt alebo služba sieťových dát, ktoré môžu byť identifikované pomocou URI, ako je definované v sekcii 3.2. Zdroje môžu byť prístupné v mnohonásobných reprezentáciách (napr. viaceré jazyky, formáty dát, veľkosti a rozlíšenia) alebo rôzne v iných smeroch.

Entita (entity)

Informácia prenášaná ako užitočný náklad požiadavky alebo odozvy. Entita pozostáva z metainformácie vo forme polí hlavičky entity, a obsahu vo forme tela entity ako je opísané v sekcii 7.

Reprezentácia (representation)

Entita zahrňujúca odozvu, ktorá je objektom na prevod obsahu, ako je opísané v sekcii 12. Môžu existovať viacnásobné reprezentácie priradené k jednotlivým stavom odoziev.

Prevod obsahu (content negotiation)

Mechanizmus na výber vhodnej reprezentácie pri obsluhe požiadavky ako je popísané v sekcii 12. Reprezentácia entít môže byť prevedená v akejkoľvek odozve (zahrňujúc chybové odozvy).

Variant (variant)

Zdroj môže mať v každom momente jednu, alebo viac ako jednu, jemu priradenú reprezentáciu. Každá z týchto reprezentácií je označovaná ako variant. Použitie označenia variant nutne nevyjadruje, že zdroj je objektom na prevod obsahu.

Klient (client)

Program, ktorý ustanovuje spojenie za účelom posielania požiadaviek.

Užívateľský agent (user agent)

Klient, ktorý zaháji požiadavku. Takýmito sú často prehliadače, editory alebo iné nástroje koncového užívateľa.

Server (server)

Aplikačný program, ktorý prijíma spojenia za účelom obsluhy požiadaviek posielaním odoziev. Akýkoľvek program môže mať schopnosť byť aj klientom aj serverom. Naše použitie týchto výrazov poukazuje iba na úlohu

vykonávanú programom pre individuálne spojenie, nie na schopnosti programu všeobecne. Podobne, ktorýkoľvek server môže vystupovať ako pôvodný server, proxy, brána alebo tunel, prepínaním režimu založenom na základe typu každej požiadavky.

Pôvodný server (origin server)

Server, na ktorom pobýva daný zdroj alebo bude vytvorený.

Proxy (proxy)

Sprostredkovateľský program, ktorý vystupuje ako server aj klient zároveň za účelom vytvárania požiadaviek v záujme ďalších klientov. Požiadavky sú obsluhované vnútorne alebo s možným prekladom prepúšťané ďalším serverom. Proxy MUSÍ mať implementované požiadavky špecifikácie pre klienta aj server. „Transparentný proxy“ je proxy, ktorý nemení požiadavky alebo odozvy okrem tých, ktoré sú potrebné pre overenie proxy a identifikáciu. „Netransparentný proxy“ je proxy, ktorý mení požiadavky a odozvy za účelom pridania niektorých služieb užívateľským agentom, ako sú služby popisujúce skupiny, zmena typu média, redukcia protokolu alebo anonymná filtrácia. Okrem prípadov, kde je explicitne určené transparentné alebo netransparentné chovanie, HTTP proxy vyžaduje oba typy proxy.

Brána (gateway)

Server, ktorý pôsobí ako sprostredkovateľ pre ďalšie iné servery. Na rozdiel od proxy, brána prijíma požiadavky ako keby bola pôvodným serverom pre požadované zdroje. Klient čo dal požiadavku si nemusí byť vedomý, že komunikuje s bránou.

Tunel (tunnel)

Sprostredkovateľský program, ktorý vystupuje ako transparentný prenos medzi dvomi spojeniami. Odkedy je aktívny, nepovažuje sa za spoločníka HTTP komunikácie, napriek tomu že tunel môže byť založený pomocou HTTP požiadavky.

Rýchla pamäť (cache)

Lokálne úložisko programu pre správy odoziev; podsystém, ktorý riadi ukladanie, vyhľadávanie a mazanie správ. Rýchla pamäť ukladá odozvy, ktoré sa do nej môžu uložiť, za účelom skrátenia času odozvy a zníženia obsadenia šírky pásma pre budúce ekvivalentné požiadavky. Hociktorý

klient alebo server môže obsahovať rýchlu pamäť, aj keď rýchlu pamäť nemôže používať server pôsobiaci ako tunel.

Možnosť ukladania do rýchlej pamäte (cacheable)

Odozva sa môže uložiť do vyrovnávacej pamäte, pokiaľ je povolené uloženie kópie správy odozvy, na použitie v odpovedi pre nasledujúce požiadavky. Pravidlá pre určovanie možnosti ukladania do rýchlej pamäte HTTP odoziev sú definované v sekcii 13. Pokiaľ má zdroj možnosť uloženia do vyrovnávacej pamäti, môžu nastať ďalšie obmedzenia či je možné použiť uloženú kópiu v rýchlej pamäti pre konkrétnu požiadavku.

Z prvej ruky (first-hand)

Odozva je z prvej ruky, pokiaľ prichádza priamo a bez zbytočného zdržania z pôvodného servera, prípadne cez jeden alebo viac proxy. Odozva je taktiež z prvej ruky pokiaľ bola overená jej platnosť priamo s pôvodným serverom.

Jednoznačný čas vypršania (explicit expiration time)

Čas, za ktorý pôvodný server posúdi či by už entita nemala byť ďalej vracaná rýchlou pamäťou bez ďalšieho overenia.

Heuristický čas vypršania (heuristic expiration time)

Čas vypršania určený rýchlou pamäťou , keď nie je dostupný žiadny Jednoznačný čas vypršania.

Vek (age)

Vek odozvy je čas, odkedy bola odozva odoslaná alebo úspešne overená pôvodným serverom.

Doba platnosti (freshness lifetime)

Dĺžka času medzi odoslaním odozvy a času jej vypršania.

Platnosť (fresh)

Odozva je platná ak jej vek neprekročil dobu platnosti.

Staroba (stale)

Odozva je stará, keď jej vek prekročí jej dobu platnosti.

Sémantická transparentnosť (semantically transparent)

Rýchla pamäť pôsobí v „Sémantická transparentnosť“ režime, s prihliadnutím na konkrétnu odozvu, keď neovplyvňuje žiadneho žiadajúceho klienta ani pôvodný server, s výnimkou zvýšenia výkonnosti.

Keď je rýchla pamäť sémanticky transparentná , klient jednoznačne prijme takú istú odozvu (okrem hop-by-hop hlavičiek), ako by prijal odozvu zabezpečenú priamo pôvodným serverom.

Overovač (validator)

Časť protokolu (napr. značka entity alebo čas poslednej zmeny), ktorý sa používa na určenie toho, kedy je záznam v rýchlej pamäti ekvivalentná kópia entity.

Vysielaný prúd/prijímaný prúd (upstream/downstream)

Vysielaný a prijímaný prúd opisujú tok správ. Všetky správy tečú z vysielaného do prijímaného prúdu.

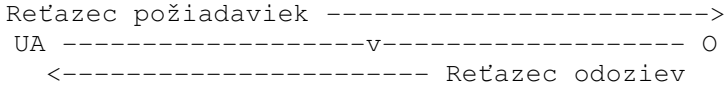
Prichádzajúci / odchádzajúci (inbound/outbound)

Prichádzajúci a odchádzajúci odkazujú na cestu požiadavky a odozvy. „Prichádzajúci“ znamená „putujúci k pôvodnému serveru“ a „odchádzajúci“ znamená „putujúci k užívateľskému agentovi“.

Spôsob komunikácie :

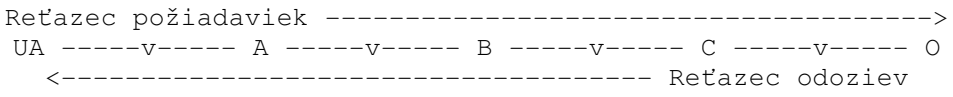
HTTP protokol je protokol požiadaviek a odpovedí. Klient pošle požiadavku serveru pozostávajúcu z metódy požiadavky, URI a verzie protokolu, nasledovanú MIME podobnou správou obsahujúcu činitele požiadaviek, informácie o klientovi a možné telo správy prostredníctvom spojenia so serverom. Server odpovie so stavovým riadkom, obsahujúcim verziu protokolu správy a potvrdzujúci alebo chybový kód, následne MIME podobnú správu obsahujúcu informáciu o serveri, metainformáciu entity, a prípadný obsah tela entity. Vzťah medzi HTTP a MIME je popísaný v kapitole 19.4.

Väčšina HTTP komunikácie začína užívateľským agentom a pozostáva z požiadavky na zdroj na nejakom pôvodnom serveri. V najjednoduchšom prípade to môže byť úspešne vykonané cez jedno spojenie medzi agentom a pôvodným serverom.

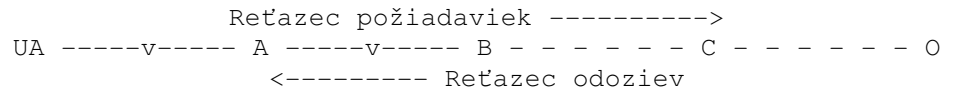


Zložitejšia situácia nastane, keď sa v reťazci požiadaviek / odoziev nachádza jeden alebo viac sprostredkovateľov. Máme tri hlavné formy sprostredkovateľov : proxy, brána a tunel. Proxy je zasielajúci agent, prijímajúci

požiadavky pre URI vo všeobecnom tvare, prepisujúci celú alebo časť správy, a posielajúci naformátovanú správu ďalej k serveru identifikovaného z URI. Brána je prijímajúci agent, pôsobiaci ako vrstva nad nejakým iným serverom a pokiaľ je to nutné prekladá požiadavky do základného protokolu servera. Tunel pôsobí ako relačný bod medzi dvomi spojeniami bez menenia správy; tunely sa používajú keď je potrebné komunikovať cez sprostredkovateľov (ako sú napríklad firewally) vtedy, keď sprostredkovatelia nerozumejú obsahu správ.



Tento obrázok zobrazuje troch sprostredkovateľov (A, B a C) medzi užívateľským agentom a pôvodným serverom. Správa požiadavky alebo odozvy, ktorá putuje celým reťazcom je prepúšťaná cez štyri samostatné spojenia. Toto rozlíšenie je dôležité, pretože niektoré nastavenia HTTP komunikácie môžu byť aplikované iba na spojenie s najbližším, nie tunelovým susedom, iba na koncové body reťazca, alebo na všetky spojenia pozdĺž reťazca. Keďže diagram je lineárny, každý účastník môže byť zaangažovaný vo viacerých komunikáciách prebiehajúcich súčasne. Napríklad B môže prijímať požiadavky od viacerých klientov iných ako A a/alebo posúvať požiadavky serverom iným ako C, v tom istom čase ako vybavuje požiadavku od A. Každý člen v komunikácii, ktorý nevystupuje ako tunel, môže použiť vnútornú rýchlu pamäť. Efekt v použití rýchlej pamäti je ten, že skráti reťazec požiadaviek alebo odoziev, ak jeden z účastníkov pozdĺž reťazca má uloženú odozvu v rýchlej pamäti použiteľnú na danú požiadavku. Nasledujúci obrázok zobrazuje výsledný reťazec ak B má uloženú kópiu zo skoršej odozvy od O (cez C) pre požiadavku, ktorá nebola uložená v pamäti A alebo UA.



Nie všetky odozvy sa dajú využiteľne uložiť do rýchlej pamäti, a niektoré požiadavky môžu obsahovať činitele, ktoré vyžadujú špeciálne podmienky na správanie rýchlej pamäti. HTTP podmienky na správanie rýchlej pamäti a odozvy, ktoré sa dajú uložiť sú definované v sekcii 13.

V skutočnosti je tu veľa architektúr a konfigurácii rýchlych pamätí a proxy zapojených naprieč World Wide Web. Tieto systémy zahrňujú národné

hierarchie rýchlych pamätí proxy na ušetrenie cezoceánskej šírky pásma, systémy, ktoré rozposielajú zoznamy rýchlych pamätí a iné. Výhoda HTTP/1.1 je, že podporuje veľa odlišných konfigurácií.

HTTP komunikácia je väčšinou prevádzaná cez TCP/IP spojenie. Východzí port je TCP 80, ale môžu byť použité aj iné porty. Toto nezabraňuje HTTP aby bol implementovaný nad hociktorým iným protokolom na internete alebo iných sieťach. HTTP uvažuje iba so spoľahlivým prenosom; každý protokol, ktorý toto garantuje, môže byť použitý.

V HTTP/1.0 väčšina implementácií používala nové spojenie pre každú výmenu požiadavky alebo odozvy. V HTTP/1.1 môže byť spojenie použité na jednu alebo viac výmen požiadaviek/odoziev, taktiež môže byť spojenie uzavreté z rozličných dôvodov.

Uniformované identifikátory zdrojov (URI)

URI sú známe pod mnohými názvami : WWW adresy, univerzálne identifikátory dokumentov, univerzálne identifikátory zdrojov a nakoniec kombináciou URL a URN. Od začiatku HTTP, URI sú jednoducho formátované reťazce, ktoré sú určené menom, miestom a ďalšou charakteristikou zdroja.

URL

Skratka „HTTP“ sa používa na určenie zdroja na sieti pomocou HTTP protokolu. Všeobecný formát je daný takto :

```
http_URL = "http:" "//" host [ ":" port ] [ abs_path [ "?" query ]]
```

Ak nebol zadaný žiadny port, tak sa použije port číslo 80. Zápis určuje zdroj, ktorý sa nachádza na serveri naslúchajúcom na TCP spojení na danom porte a danom hostiteľovi, a odozva je absolútna cesta k zdroju.

Porovnávanie URI

Keď porovnávame dve URI či sa zhodujú alebo nie, klient BY MAL použiť rozpoznávanie malých a veľkých písmen s týmito výnimkami :

- Port, ktorý nebol zadaný je rovnaký ako východzí port
- Porovnanie mien hostiteľov NESMIE posudzovať malé a veľké písmená
- Porovnanie typov NESMIE posudzovať malé a veľké písmená
- Prázdna absolútna cesta je ekvivalentná s „ / “

Znaky iné ako tie v sadách „reserve“ a „unsafe“ (uvedené v RFC 2396) sú ekvivalentné ich ““%” HEX HEX” kódovaniu. Napríklad tieto URL su ekvivaletné :

http://abc.com:80/~smith/home.html
http://ABC.com/%7Esmith/home.html
http://ABC.com:/%7esmith/home.html

stavové kódy

stavový kód je trojciferné číslo, ktoré určuje ako sa vyhovelo požiadavke. Prvé číslo určuje triedu odozvy. Existuje 5 hodnôt pre prvé číslo :

1xx : informačné – požiadavka prijatá, proces pokračuje.

2xx : vyhovujúce – akcia bola úspešne prijatá, pochopená a akceptovaná.

3xx : presmerovanie – nasledujúce akcie musia byť obdržané za účelom splnenia požiadavky.

4xx : chyba klienta – požiadavka má zlú syntax alebo nemôže byť splnená.

5xx : chyba servera – server zlyhal pri plnení zrejme platnej požiadavky.

Súhrnný výpis jednotlivých kódov :

```
"100"  : Continue
"101"  : Switching Protocols
"200"  : OK
"201"  : Created
"202"  : Accepted
"203"  : Non-Authoritative Information
"204"  : No Content
"205"  : Reset Content
"206"  : Partial Content
"300"  : Multiple Choices
"301"  : Moved Permanently
"302"  : Found
"303"  : See Other
"304"  : Not Modified
"305"  : Use Proxy
"307"  : Temporary Redirect
"400"  : Bad Request
"401"  : Unauthorized
"402"  : Payment Required
"403"  : Forbidden
"404"  : Not Found
"405"  : Method Not Allowed
"406"  : Not Acceptable
"407"  : Proxy Authentication Required
"408"  : Request Time-out
```

```
"409"  : Conflict
"410"  : Gone
"411"  : Length Required
"412"  : Precondition Failed
"413"  : Request Entity Too Large
"414"  : Request-URI Too Large
"415"  : Unsupported Media Type
"416"  : Requested range not satisfiable
"417"  : Expectation Failed
"500"  : Internal Server Error
"501"  : Not Implemented
"502"  : Bad Gateway
"503"  : Service Unavailable
"504"  : Gateway Time-out
"505"  : HTTP Version not supported extension-code
```

SHTTP protokol

Prehľad :

Zabezpečený HTTP (secure HTTP) poskytuje bezpečný komunikačný mechanizmus medzi párom HTTP klient - server za účelom umožniť samočinné komerčné transakcie pre široké spektrum aplikácii. Zámerom je poskytnúť pružný protokol, ktorý podporuje viaceré operačné módy, správu kľúčov, overovacie modely, kryptografické algoritmy a enkapsulačné formáty, cez voľby prevodov medzi účastníkmy počas celej transakcie.

Súhrn vlastností :

SHTTP je bezpečný správovo orientovaný komunikačný protokol navrhnutý na používanie s HTTP. Je navrhnutý na koexistenciu s HTTP modelom správ a nato aby bol ľahko integrovateľný s HTTP aplikáciami.

SHTTP poskytuje viacero bezpečnostných mechanizmov pre HTTP klientov a serverom, poskytovaním bezpečnostných služieb určených na široké spektrum koncových riešení na WWW. Protokol poskytuje súmerné možnosti pre oboch klienta aj server, zatiaľ čo udržuje transakčný model a implementačné vlastnosti HTTP.

Niekoľko kryptografických formátov správ môže byť zabudovaných do SHTTP klientov a serverov ako sú CMS (cryptographics message syntax) a MOSS (MIME object security services)

Použitá literatúra

[1] http://tools.ietf.org/html/rfc2616
[2] http://tools.ietf.org/html/rfc2660