

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentace k projektu do předmětu ISA
KLIENT SIP

21.listopadu 2015

Autor: Lukáš Pelánek
xpelan03@stud.fit.vutbr.cz

Obsah

Úvod.....	3
Uvedení do problematiky SIP	3
Požadavky	3
Odpovědi	4
Registrace	4
Zásílání zpráv	5
Návrh a implementace	6
Návod na použití.....	6
Závěr	7

Úvod

Tento dokument obsahuje uvedení do problematiky komunikace pomocí protokolu SIP, návrhu aplikace, popis implementace, základní informace o programu a návod na použití.

Uvedení do problematiky SIP

Protokol SIP je textový internetový protokol pro přenos signalizace v internetové telefonii. Defaultně používá port 5060. Vlastní přenos hovoru je uskutečněn pomocí protokolu RTP. Protokol vychází z protokolu HTTP.

Požadavky

Jednotlivé požadavky jsou odeslány na server (v rámci projektu server Asterisk).

Základní požadavky protokolu SIP:

1. INVITE
2. ACK
3. BYE
4. CANCEL
5. OPTIONS
6. REGISTER
7. NOTIFY
8. PUBLISH
9. MESSAGE

V rámci projektu jsou implementovány tyto požadavky:

1. REGISTER – registrace uživatele na serveru SIP
2. MESSAGE – zaslání soukromé zprávy uživateli

Odpovědi

Na každý odeslaný požadavek server odpoví příslušným kódem, který uživatele informuje o průběhu zpracování jeho požadavku.

Typy odpovědí:

1. 1xx – dočasná odpověď
2. 2xx – kladné vyřízení požadavku
3. 3xx – přesměrování
4. 4xx – chyba na straně klienta
5. 5xx – chyba na straně serveru
6. 6xx – globální chyba

Registrace

Klient zahajuje registraci odesláním požadavku REGISTER.

Příklad registrace z ukázkového .pcap souboru:

```
REGISTER sip:10.0.0.1 SIP/2.0
CSeq: 53 REGISTER
Via: SIP/2.0/UDP 10.0.0.3:5060;branch=z9hG4bK5e6a98a7-286f-e511-81c6-080027426d5a;rport
User-Agent: Ekiga/4.0.1
From: <sip:bob@10.0.0.1>;tag=8a4d98a7-286f-e511-81c6-080027426d5a
Call-ID: 344998a7-286f-e511-81c6-080027426d5a@test2-VB
To: <sip:bob@10.0.0.1>
Contact: <sip:bob@10.0.0.3:5060>;q=1
Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,SUBSCRIBE,NOTIFY,REFER,MESSAGE,INFO,PING,PRACK
Expires: 3600
Content-Length: 0
Max-Forwards: 70
```

Na to server zareaguje odpovědí s kódem 401 a žádá po klientovi ověření. Rovněž zasílá potřebné údaje jako je realm a nonce.

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.0.0.3:5060;branch=z9hG4bK5e6a98a7-286f-e511-81c6-080027426d5a;received=10.0.0.3;rport=5060
From: <sip:bob@10.0.0.1>;tag=8a4d98a7-286f-e511-81c6-080027426d5a
To: <sip:bob@10.0.0.1>;tag=as4db23355
Call-ID: 344998a7-286f-e511-81c6-080027426d5a@test2-VB
CSeq: 53 REGISTER
Server: Asterisk PBX 11.7.0~dfsg-1ubuntu1
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH
Supported: replaces, timer
WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="149b7c26"
Content-Length: 0
```

V tuto chvíli klient musí spočítat hash a odeslat serveru potvrzení s přiloženým kódem hash.

Vzorec pro výpočet hash hodnoty je následující:

```
HA1 = MD5("myusername:realm:password")
HA2 = MD5("REGISTER:sip:sip.example.com")
response = MD5(HA1+ : + nonce + : +HA2);
```

Následně klient odešle potvrzující požadavek obsahující hash hodnotu v poli response.

```
REGISTER sip:10.0.0.1 SIP/2.0
CSeq: 54 REGISTER
Via: SIP/2.0/UDP 10.0.0.3:5060;branch=z9hG4bKb02199a7-286f-e511-81c6-080027426d5a;rport
User-Agent: Ekiga/4.0.1
Authorization: Digest username="bob", realm="asterisk", nonce="149b7c26", uri="sip:10.0.0.1", algorithm=MD5, response="593f84f5e2294740e8602c8a79065bb8"
From: <sip:bob@10.0.0.1>;tag=8a4d98a7-286f-e511-81c6-080027426d5a
Call-ID: 344998a7-286f-e511-81c6-080027426d5a@test2-VB
To: <sip:bob@10.0.0.1>
Contact: <sip:bob@10.0.0.3:5060>;q=1
Allow: INVITE,ACK,OPTIONS,BYE,CANCEL,SUBSCRIBE,NOTIFY,REFER,MESSAGE,INFO,PING,PRACK
Expires: 3600
Content-Length: 0
Max-Forwards: 70
```

V případě správného výpočtu hodnoty hash server zašle potvrzující zprávu s kódem 200 a uživatel je přihlášen na serveru po dobu, která je uvedena v položce Expires.

Zásílání zpráv

Klienti si zprávy zasílají mezi sebou přímo. V tuto chvíli komunikace neprobíhá přes server.

```
MESSAGE sip:alice@10.0.0.2 SIP/2.0
CSeq: 57 MESSAGE
Via: SIP/2.0/UDP 10.0.0.3:5060;branch=z9hG4bKbe0007b6-286f-e511-81c6-080027426d5a;rport
User-Agent: Ekiga/4.0.1
From: <sip:test2@10.0.0.3>
Call-ID: 1aeb06b6-286f-e511-81c6-080027426d5a@test2-VB
To: <sip:alice@10.0.0.2>
Expires: 5000
Content-Length: 9
Content-Type: text/plain;charset=UTF-8
Max-Forwards: 70
```

Hi there!

Pokud zprávu obdržel správný uživatel, informuje klienta o úspěšném doručení zprávy:

```
SIP/2.0 202 Accepted
CSeq: 57 MESSAGE
Via: SIP/2.0/UDP 10.0.0.3:5060;branch=z9hG4bKbe0007b6-286f-e511-81c6-080027426d5a;rport=5060;received=10.0.0.3
User-Agent: Ekiga/4.0.1
From: <sip:test2@10.0.0.3>
Call-ID: 1aeb06b6-286f-e511-81c6-080027426d5a@test2-VB
To: <sip:alice@10.0.0.2>
Contact: <sip:alice@10.0.0.2>
Content-Length: 0
```

Návrh a implementace

Celá aplikace je implementována v jazyce C++. Pro realizaci klienta byl použit návrhový vzor Singleton.

Jako první aplikace provede zpracování vstupních souborů a naplní datové struktury. Dále zjišťuje dostupný interface, na kterém bude klient vysílat.

Následně zavolá funkci *bind()* na port 5061. V případě, že port není volný, aplikace zkouší dostupné porty až po port 5080. Pokud žádný port v tomto rozmezí není volný, aplikace skončí s chybovým hlášením.

Klient odešle požadavek REGISTER a čeká na odpověď od serveru. Pokud server odpověděl kódem 5xx, klient každých 5 vteřin zkouší registraci znovu. V případě úspěšné registrace, klient si uloží hodnotu z pole Expires a těsně před vypršením odešle nový požadavek REGISTER, aby udržel klienta přihlášeného.

Klient naslouchá a zpracovává přijaté zprávy. Pokud klient obdrží jiný požadavek než MESSAGE, tak jej ignoruje. V případě požadavku MESSAGE klient ověří, zda je příjemcem zprávy. Pokud ano, odešle potvrzující zprávu 202 Accepted. V opačném případě odešle zprávu 403 Forbidden.

Po obdržení signálu SIGTERM, SIGQUIT nebo SIGINT klient odešle serveru požadavek na odhlášení ze serveru. Tato zpráva je stejná jako zpráva pro registraci, s tím rozdílem, že hodnota u pole Expires je 0. Klient čeká 1 vteřinu, pokud do této doby neproběhne potvrzení odhlášení ze serveru, klient se korektně ukončí.

Veškerá komunikace je vypisována na stdout v následujícím formátu:

<sekvenční_číslo> <název_požadavku/číslo_a_název_odpovědi> <zdrojová_sip_adresa> <cílová_sip_adresa>

Návod na použití

Jako první se program musí přeložit pomocí přiloženého souboru Makefile. Stačí do konzole napsat příkaz *make*. Následně se aplikace spouští pomocí následujícího příkazu:

./sipklient -p profile.txt [-m messages.txt]

Parametr *-p* je povinný. Obsahuje soubor, který obsahuje informace o uživateli a serveru. Formát souboru *profile.txt* je následující:

```
server=SERVER
username=USERNAME
password=PASSWORD
expires=EXPIRES
```

Dále může obsahovat volitelný parametr *-m* se souborem obsahující zprávy k odeslání. Formát souboru *messages.txt* je následující:

jmeno_uzivatele@adresa_uzivatele zprava

Závěr

Aplikace byla otestována a vyvíjena na virtuálním stroji ISA2015. Testování probíhalo vytvořením 3 virtuálních strojů, které byly propojeny pomocí lokální sítě. Uživatelé si posílali zprávy v rámci sítě. Dále byla komunikace otestována pomocí klienta Ekiga. Program splňuje zadaná kritéria.

Literatura

[1] Ukázková komunikace SIP. [online]. 2015 [cit. 2015-11-21]. Dostupné z: <http://www.stud.fit.vutbr.cz/~xfrank08/sip-message-example.pcap>

[2] SIP: Session Initiation Protocol. [online]. 2002 [cit. 2015-11-21]. Dostupné z: <http://tools.ietf.org/html/rfc3261>

[3] Reversing SIP digest authentication in JavaScript. *kapejod.org*. [online]. 2013 [cit. 2015-11-21]. Dostupné z: <http://www.kapejod.org/en/2013/02/reversing-sip-digest-authentication-in-javascript-node-js>