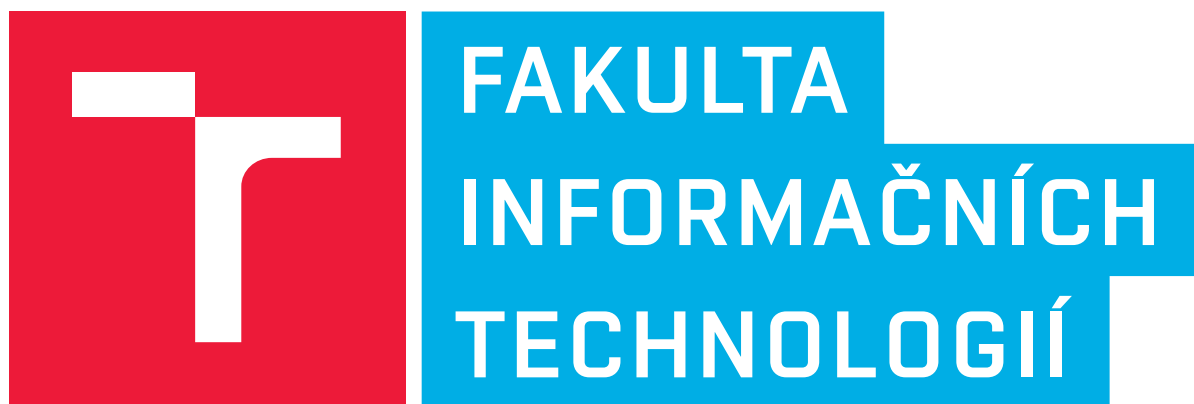


Vysoké Učení Technické v Brně



Síťové aplikace a správa sítí

Jednoduchý SNMP manager

22. listopadu 2015

Autor: Peter Malina, xmlin26@stud.fit.vutbr.cz

Obsah

1. Úvod
2. Prehľad protokolu SNMP
 - a. Štruktúra paketu
 - b. Štruktúra bloku paketu
3. Návrh aplikácie
 - a. Celkový pohľad
 - b. Klient
 - c. Entita
 - d. Paket
 - e. Kontajner pre informácie
4. Implementácia aplikácie
 - a. Odosielanie a prijímanie paketov
 - b. Spracúvanie paketov
5. Návod na použitie
6. Záver
7. Literatúra

1) Úvod

Tento manuál slúži ako základný prehľad implementácie a použitia jednoduchého SNMP klienta.

Vysvetľuje akým spôsobom medzi sebou strany komunikujú a čo je v komunikácii dôležité.

Dokument tiež vysvetľuje architektúru aplikácie a prečo bola takto aplikácia navrhnutá. Táto implementácia sa týka SNMP verzie 1.

2) Prehľad protokolu SNMP

Protokol SNMP slúži ako jednoduchý manažérsky protokol siete. Pre prenos informácií sa používa typicky protokol UDP a port 161. SNMP podporuje rôzne dátové typy. Typy potrebné pre implementáciu aplikácie sú v súbore types.h . Pri komunikácii cez SNMP, sa klient dotazuje serveru, teda server vždy odpovedá na požiadavky klienta.

2a) Štruktúra paketu

SNMP paket a jeho bloky majú vždy presne určený typ a dĺžku. Typ sa nachádza na prvom bajte, dĺžka na ďalších. Implementácia tohoto klienta však predpokladá, že dĺžka paketu nie je viac ako 255 bajtov, teda, že sa číslo dĺžky zmestí na jeden bajt. Po hlavičkovej časti (typ a dĺžka) nasleduje samotná hodnota bloku.

Celý paket je teda zapúzdzrený v hlavičke, ktorá určuje jeho typ a celkovú veľkosť za hlavičkou. Paket SNMP ktorý je použitý, má takto ďalej zapúzdzrenú verziu, a community string. Za nimi nasleduje PDU, ktorý znova zapúzdzruje request ID, chyby a sekvenciu premenných.

2b) Štruktúra bloku paketu

Každý blok paketu sa skladá z typu, dĺžky a hodnoty. SNMP paket sa skladá z takýchto blokov, pričom aj samotný paket je blok. Každý blok sa dá na základe informácií o type a dĺžke korektne deserializovať.

Špeciálnym typom bloku v SNMP pakete je typ generickej hodnoty. Teda dopredu nevieme, aký je typ tohoto bloku. Tento typ sa používa pri prenášaní premenných v PDU. Ostatné bloky majú presne určený typ.

3) Návrh aplikácie

Aplikácia bola navrhnutá na základe nadobudnutých znalostí, ktoré boli popísané v sekcii 2) . Bol použitý objektový návrh, ktorý sa snaží čo najbližšie reprezentovať SNMP paket aj jeho bloky.

3a) Celkový pohľad

Architektúra klienta je rozdelená do troch hlavných blokov: klient samotný, paket a jeho bloky, kontajner pre informácie interface-och. Klient využíva triedu paketu ako aj triedu kontajneru.

3b) Klient

Klientovou hlavnou úlohou je vytvorenie spojenia k agentovi, odosielanie informácií, prijímanie informácií a ich spracovávanie pomocou kontajneru pre informácie. Klient sa stará o chod programu. Aplikácia beží, kým nie je zachytený interrupt.

3c) Entita

Entitou bol nazvaný každý blok SNMP paketu. Ide o abstraktnú triedu, ktorá poskytuje interface pre serializáciu a deserializáciu. Týmto spôsobom je jednoduchá a oddelená implementácia každého rôzneho typu SNMP packetu. Táto trieda bola zavedená, ako prostredník pre serializáciu rôznych blokov paketu.

3d) Paket

Paket je základnou triedou, ktorá uchováva celú štruktúru SNMP paketu. Táto trieda iba zapúzdruje všetky Entity, ktoré SNMP paket obsahuje, pričom dáva možnosť jednoduchej serializácie a deserializácie.

3e) Kontajner pre informácie

Kontajner slúži ako jednoduchý prostriedok pre uchovávanie dôležitých informácií, ktoré sú poskytnuté agentom klientovi. Jeho rozhranie dovoľuje spracovávať akékoľvek SNMP pakety, pričom kontajner vždy uchováva časť paketu s premennými, pričom ukladá iba ich hodnoty.

4) Implementácia aplikácie

Implementácia aplikácie sa odrážala od jej objektového návrhu. Boli použité techniky objektovo orientovaného programovania, pričom implementácia dbá na štýl kódu, prezentovaný spoločnosťou Google. Všetky jeho body však nemusia byť dodržané.

Hlavnou implementačnou časťou je časť spracovania paketov (SNMPPacket.cpp), ktorá obsahuje implementácie tried, ktoré dodržia rozhranie Entity. Boli implementované typy Integer, OctetString, ObjectIdentifier a Value. Pre tieto typy je teda možné serializovať a deserializovať dáta.

4a) Odosielanie a prijímanie paketov

I/O paketov rieši Klient a to dvoma metódami: ReceiveGetPacket a SendGetPacket. Pre odosielanie je tiež použitá factory metóda CreateGetPacket, ktorá nastaví v pakete všetky základné entity. Dáta sú prenášané vo vektoroch alebo listoch. Po implementácii je však jasné, že list nie je najlepší typ kontajneru pre účel prijímania paketov.

4b) Spracúvanie paketov

Po prijatí paketu, je následné spracovaný. Vektor bajtov je ďalej posunutý do deserializeru paketu, ktorý sa postará o naplnenie entít v pakete správnymi informáciami. V prípade chybného vstupu je vrátená chyba, ktorá nastala počas deserializácie. Keďže by chyby takéhoto typu nemali nastať, predpokladá sa, že došlo ku chybe v sieti a klient sa ukončí. V opačnom prípade budú všetky entity správne naplnené.

5) Návod na použitie

Príklad spustenia aplikácie: `./isaSnmplfLog [-i interval] -c community_string AGENT`

Interval - počet milisekúnd medzi dotazmi na agenta

Agent - ip adresa alebo hostname agenta

6) Záver

Klient po spustení vytvorí spojenie s agentom a začne sa dotazovať na ifTable pomocou správy get-next-request. Takto pokračuje, až kým vrátený objekt patrí do ifEntry objektov. Program je ukončiteľný pomocou systémového interruptu.

7) Literatúra

[rfc](#)