



УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ИСПИТНИ РАД

Кандидат: Мићо Делић

Број индекса: ra241/2018

Предмет: Међурачунарске комуникације и рачунарске мреже 1

Тема рада: FTP проху

Ментор рада: Проф. Илија Башичевић

Нови Сад, Децембар 2020.

SADRŽAJ

1.	Задатак	1
2.	Концепт решења	2
3.	Опис решења	3
4.	Тестирање	5
5.	Закључак	6
6.	Литература	7

СПИСАК СЛИКА

1. СДЛ дијаграм 3
2. МСЦ дијаграм 4

1. Задатак

Задатак испитног рада је једноставан модел ФТП проксија. Као клијентска апликација се користи неки од интернет претраживача, а као сервер се користи готова апликација FTP3serverlite. Прокси треба да на себе презима сав саобраћај између клијента и сервера. За имплементацију је потребно користити језгро комуникационе програмске подршке и

2. Концепт решења

С обзиром да је коришћено језгро за комуникациону подршку, морамо дефинисати ресешење као систем аутомата са коначним бројем стања. Конкретан проблем је решен помоћу једног аутомата који се састоји од три стања.

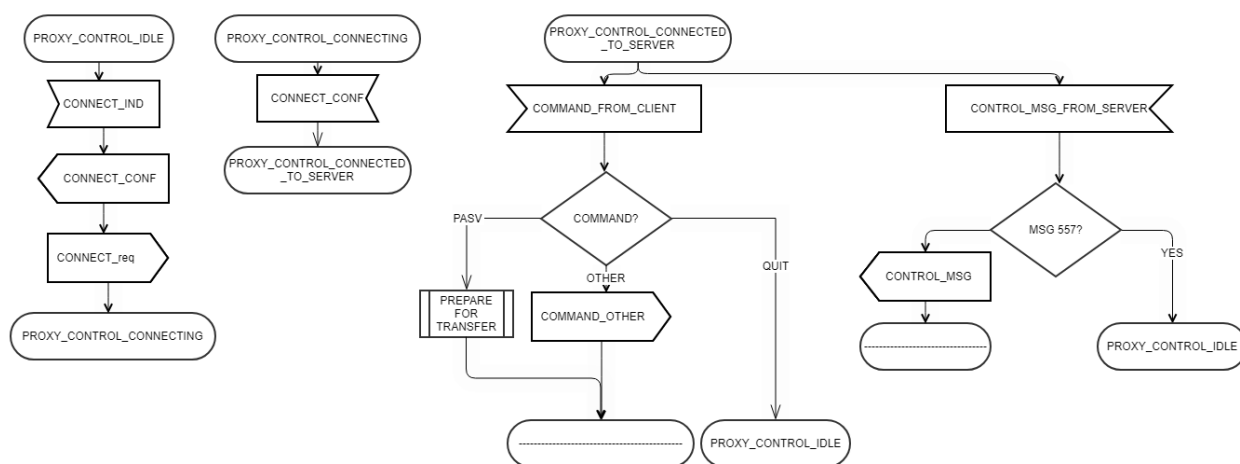
Прво и друго стање служе за успостављање конекција клијент → прокси и

прокси → сервер. Док треће стање је стање у којем се покрећу четири програмске нити. Прва и друга програмска нит служе за трансфер команди од стране клијента и контролних порука од стране сервера. Трећа и четврта раде идентично, само за трансфер података уместо команди.

3. Опис решења

Као што је написано у концепту, решење ће се састојати од једног аутомата ProxyControlAuto. Он ће бити реализован као класа ProxyControlAuto која ће наследити класу FiniteStateMachine која је део језгра за комуникациону подршку. Стања, функције прелаза, поруке које их изазивају, као и сам начин комуникације ће даље бити објашњени и графички приказани помоћу SDL (Specification and Description Language) и MSC (Message Sequence Chart) дијаграма.

3.1 SDL



Слика 1: SDL дијаграм

Као што је раније наведено, и као што се види на слици, аутомат се састоји из три стања:

Proxy_control_idle: је почетно стање аутомата, у њему аутомат чека и ослускује да ли корисник жели да се конектује на њега, уколико дође до конекције, аутомат конекцију прихвата, и шаље захтев да би се сам прокси конектовао на сервер, у том тренутку прелази у друго стање

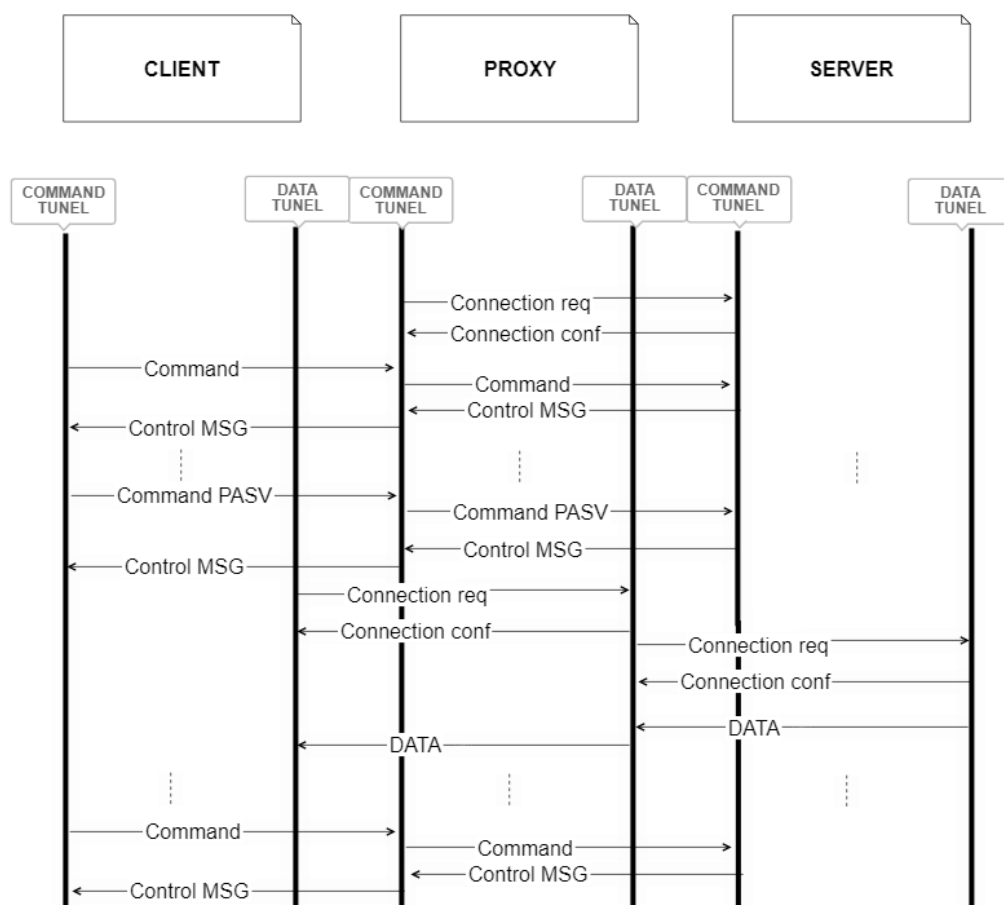
Proxy_control_connecting: у другом стању, прокси чека да сервер прихвати конекцију, и после тога одма прелази у нове стање

Proxy_control_connected: последње стање, у овом стању се покрећу две програмске нити које служе за командни канал, једна нит служи за хватање команди са клијента и прослеђивање њих на сервер, друга нит служи за хватање контролних порука са сервера које се прослеђују клијенту.

Уколико клијент проследи команду PASV, то значи да жели да добава податке са сервера. Покрећу се две нове програмске нити, које раде исто као прве две, само сто служе за транспортни тунел на другим портovima (њих одреди сервер и пошаље у командној поруци после PASV). После завршеног транспорта те програмске нити се гасе. Транспортни слој је урађен помоћу нових програмских нити да би контролни слој наставио нормално да ради.

Последња провера у аутомату је провера да ли је досла контролна порука 557, која заправо ради само на почетку, и представља обавештење да тај корисник није регистрован на серверу.

3.1 MSC



Слика 2: MSC dijagram

4. Тестирање

Тестирање је вршено ручно, због недостатка времена да се изврши помоћу CppUnit библиотеке која служи за јединично тестирање програмског кода.

Функционалности које су имплементирани су тестиране део по део. Прогресија кода је ишла следећим редоследом. Прво је направљена главна класа, у њој дефинисане методе, стања, поља... После тога су имплементирани функције преноса, и тестирани прелази из једног у другог стање и позивање свих функција прелаза. Први ствар када су убачене програмске нити, коришћене су само прве две за контролну везу, док је веза за трансфер ишла директно са клијента на сервер без посредства проксија. Када је утврђено да прве две програмске нити немају видљивих проблема, додате су последње две које су тестиране на исти начин.

Треба напоменути да је прокси тестиран са фајловима величине до 255 бајтова.

5. Закључак

Крајњи резултат је тестиран коришћењем Mozilla firefox-а као клијент, успоставља се конекција и скидају се фајлови са сервера.

Утврђено је да прокси коректно прослеђује команде, поруке, податке.

6. Литература

- [1] *Priručnik radnog okruženja za pisanje protokola, Verzija 0.2*, Univerzitet u Novom Sadu, Fakultet Tehničkih Nauka, 2007
- [2] RFC 959
- [3] RFC 1919