

Semestrální práce z KIV/PT

Standardní zadání – simulace posílání dat počítačovou sítí

Jakub Vítek, A16B0165P, viteja@students.zcu.cz
Jitka Poubová, A16B0110P, jpoubova@students.zcu.cz

10. 12. 2017

Obsah

1	Zadání	3
1.1	Simulace posílání dat počítačovou sítí	3
1.1.1	Vstupy	3
1.1.2	Simulace	3
2	Analýza úlohy	4
3	Návrh programu	5
3.1	Implementace	5
3.2	UML diagramy	5
4	Uživatelská dokumentace	8
5	Závěr	10

1 Zadání

1.1 Simulace posílání dat počítačovou sítí

Firma KIV-Internet vlastní rozsáhlou počítačovou síť (v řádech tisíců uzlů). Počítačová síť slouží k přenosu dat mezi jednotlivými stroji a jejich přístupu k internetu. Síť využívá různé technologie od wifi po optické spoje. Kvalita a rychlost jednotlivých uzlů je dána typem spoje. V současné době se provoz po síti řídí protokolem OSPF, který ale není úplně optimálním řešením. Cílem Vaší práce je proto navrhnout protokol pro optimalizaci datových toků sítě.

Síť lze reprezentovat jako mesh topologii pomocí grafu. Získáme tím graf s ohodnocením jednotlivých hran (dvojice čísel), které nám určuje maximální propustnost a stabilitu daného spoje. Maximální propustnost udává, kolik můžeme spojem poslat Mbit/s (1s = nejmenší krok simulace propustnosti sítě). Tato hodnota bude v rozsahu 1 – 10000 Mbit/s. Druhá hodnota vyjadřuje spolehlivost a bude v rozsahu 0 - 1. Hodnota 1 udává maximální spolehlivost, kdy nedochází k žádné ztrátě packetů, jakákoliv hodnota menší než 1 udává, kdy začne docházet ke ztrátě 50% při zatížení. Například 0.9 znamená, že při vytížení linky nad 90% ztrátě 50% všech dat, které v dané vteřině přenášíte, tj. budou se muset odeslat znovu. Pravděpodobnost ztráty bude také 50% (realizujte generátorem náhodných čísel).

Na uzlech sítě jsou „chytré“ routery se síťovým smart-stackem (implementujete Vy), který dokáže uchovávat až 100Mb dat pro případ, že do uzlu přiteče více dat, než můžete aktuálně odeslat. Router má inteligentní routovací systém, který se může měnit v čase (implementujete Vy). Routery mohou ovlivnit jak cestu, tak počet posílaných dat sousednímu uzlu. V případě přetečení stacku v cílovém uzlu dojde ke ztrátě celého balíku dat a musí se poslat znovu.

1.1.1 Vstupy

Graf bude popsán souborem, který bude mít následující strukturu na každém řádku: ID_UZLU_1 – ID_UZLU_2 – max_propustnost – chybovost Pozor, řádky nemusí jít za sebou v pořadí uzlů! Výsledný graf bude vždy souvislý.

1.1.2 Simulace

Pro simulaci bude použit generovaný seznam požadavků ve tvaru ČAS – ID_ZDROJ – ID_CÍL – VELIKOST_DAT

2 Analýza úlohy

Na počítačovou síť lze nahlížet jako na graf. Ten se dá reprezentovat buď pomocí incidenční matice nebo pomocí seznamu – spojového či ArrayListu. Incidenční matice má avšak vyšší paměťovou náročnost oproti seznamu. Dále také existuje více možností, jak realizovat samotnou simulaci. V úvahu připadaly algoritmy jako je Dijkstrův algoritmus (pro vyhledání nejkratší cesty mezi dvěma routery) či Ford-Fulkersonův algoritmus (pro vyhledání maximálního toku).

Také je nutné se rozhodnout, zda-li data budou posílána jednou cestou, nebo se rozdělí a například polovina dat se pošle jednou cestou a druhá polovina jinou. Při použití druhého způsobu (více cest) by se využilo větší množství hran a byla by větší šance, že data dorazí do cílového routeru dříve. Dále bylo nutno zvážit, kolik dat se bude posílat – pošleme vždy jen bezpečné množství (na hranici spolehlivosti), abychom měli jistotu, že data dorazí do cíle v pořádku, nebo budeme riskovat a využijeme vždy hranu maximálně, i přes hranici spolehlivosti? Pokud se bude posílat přes hranici spolehlivosti, je větší riziko, že se data budou muset poslat znovu.

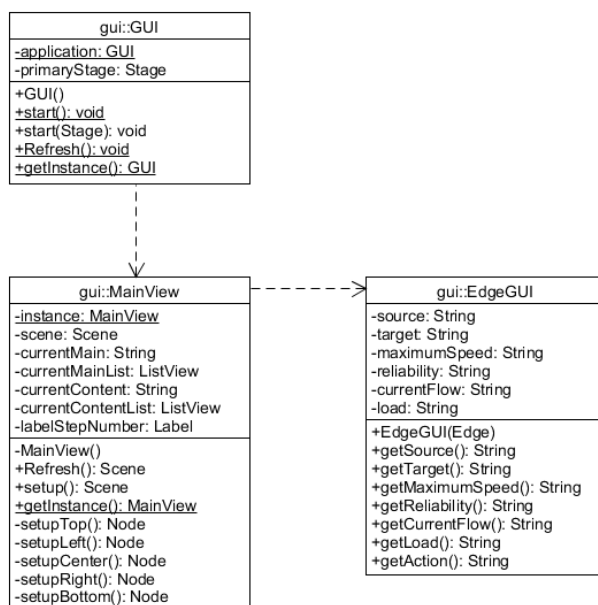
3 Návrh programu

3.1 Implementace

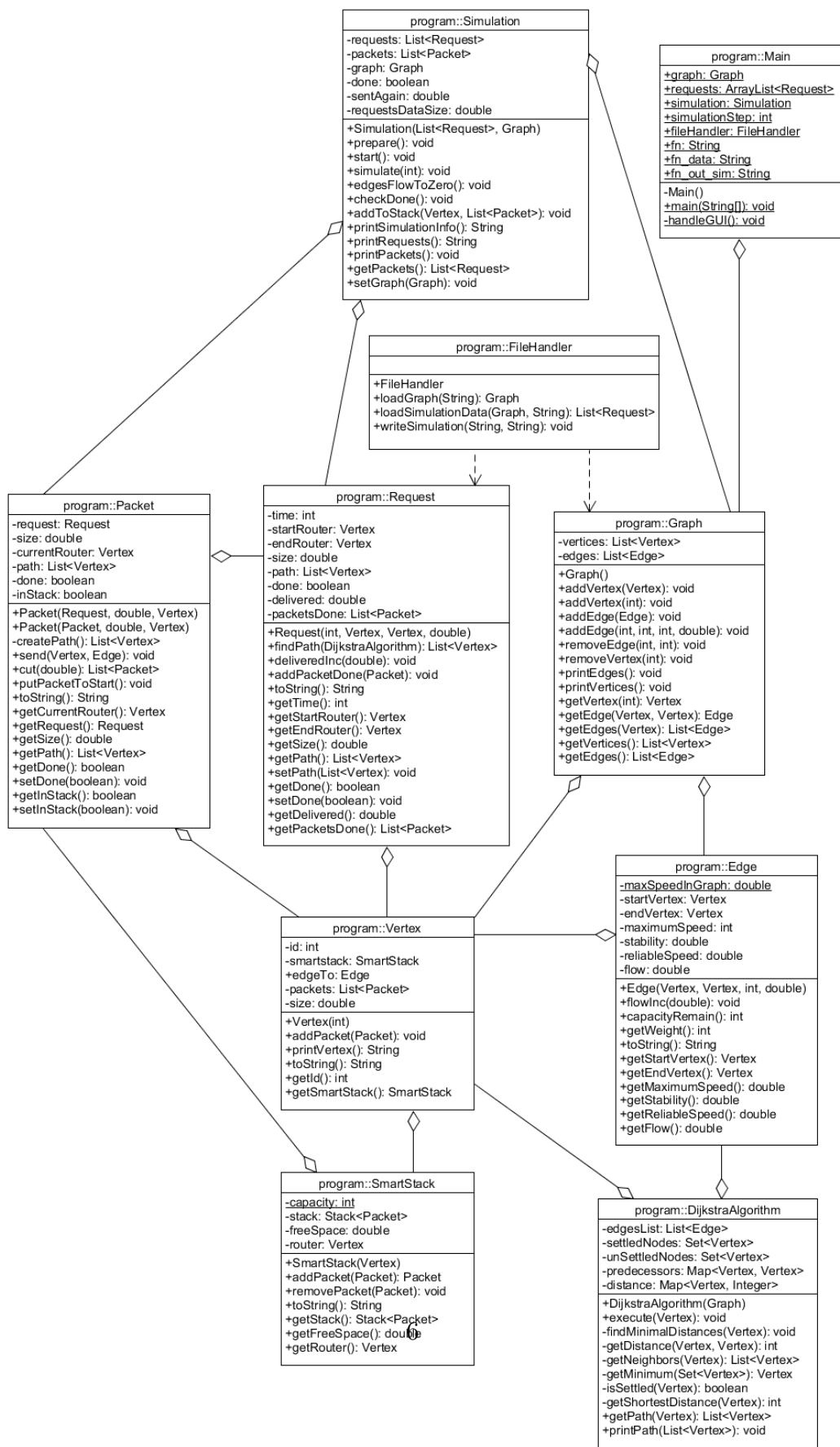
Pro reprezentaci grafu (počítačové sítě) jsme zvolili dva seznamy (ArrayListy) – graf má k dispozici seznam vrcholů a seznam hran. Co se týče samotné simulace a její přípravy, vybrali jsme si Dijkstrův algoritmus, který každému požadavku najde nejlepší cestu ze zdrojového routeru do cílového. Každý požadavek se tedy bude posílat pouze po této jedné cestě. Dále jsme se rozhodli, že nebudeme riskovat 50% šanci, že se 50 % balíku ztratí, tudíž jsme hranu využívali vždy jen na hranici její spolehlivosti a tuto hranici nikdy nepřesáhli. Nemuseli jsme tedy řešit, jestli se polovina balíku ztratí nebo ne.

Pro celý program jsme poté vytvořili okenní aplikaci, kde uživatel snadno vidí, co se v simulaci právě děje. Může také před simulací vymazat hranu či router (simulace výpadku v síti).

3.2 UML diagramy



Obrázek 3.1: UML diagram balíku GUI



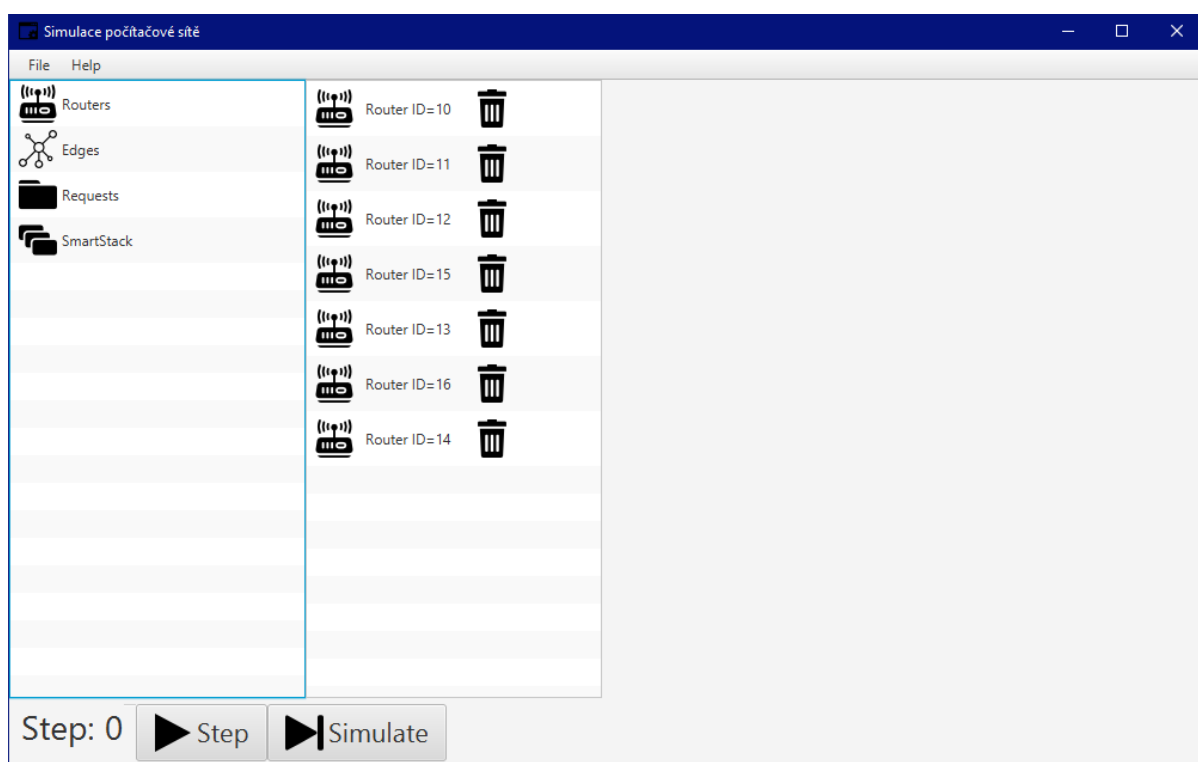
Obrázek 3.2: UML diagram balíku Program

4 Uživatelská dokumentace

Ve složce *data* jsou uloženy soubory, ze kterých se načítá. Jsou tam soubory "data.txt", ve kterém jsou uložena data pro danou počítačovou síť (graf) a "simulace.txt", ve kterém jsou uloženy požadavky, které se mají posílat přes síť. Pokud nechcete použít tyto soubory, přepište jejich obsah svými daty (jak mají soubory vypadat je popsáno výše v sekci Zadání).

Poté spusťte program – rozklikněte soubor "app.jar", který se nachází v adresáři celého programu. Otevře se okno s aplikací (vizte obrázek 4.1). Po levé straně je menu, kde si můžete vybrat, co chcete sledovat – zda-li se chcete podívat na routery, hrany, požadavky či na smartstacky jednotlivých routerů. Po kliknutí na vaší volbu se vedle tohoto menu zobrazí další menu s položkami výběru. Pokud jste si vybrali hrany, zobrazí se rovnou tabulka všech hran, pokud jste vybrali například smartstacky, musíte si poté kliknout na jeden z nich a až poté se zobrazí tabulka, co je právě v daném routeru uloženo. Pro spuštění celé simulace klikněte na tlačítko "Simulate", čímž se provede celá simulace. Pokud si chcete projít simulaci po jednotlivých krocích, stiskněte tlačítko "Step". Jestliže jste zvolili hrany a vidíte jejich tabulku, při klikání na tlačítko "Step" hned vidíte, jak se průtok v jednotlivých hranách mění. Nebo si můžete vybrat jakýkoliv smartstack a zjistit, jaké balíky dat jsou v něm právě uloženy.

Pro ukončení aplikace vyberte z menu "File" a poté "Exit", nebo jen zavřete okno aplikace, tím se celý program ukončí. Po proběhnutí celé simulace můžete nahlédnout do složky *out*, která se nachází v adresáři s celým programem a v ní je soubor "simulationInfo.txt". V tomto souboru jsou uloženy informace o celé simulaci – její průběh, kudy byla data přenášena a ztrátovost celé simulace.



Obrázek 4.1: Aplikace po spuštění

5 Závěr

Zvolili jsme řešení, které bylo spíše snadné na implementaci, než že by bylo efektivní. Bylo by lepší například data posílat více cestami, než jim jen spočítat jednu a tu poté následně využívat. Co se týče posílání dat na hranici spolehlivosti, myslíme si, že zde záleží na konkrétních datech. Pokud by všechny hrany v grafu měly spolehlivost mezi 0,9 a 1, naše řešení je vhodné a v pořádku. Avšak pokud by hodnoty spolehlivosti u hran byly například mezi 0,3 a 0,6, naše řešení by se nedalo použít, časově by bylo náročnější a nebylo by to efektivní.