Vliv velikosti vysílacího okna metody *Selective Repeat*

Jakub Adamec – adamej14@fel.cvut.cz
Daniel Petránek – petrada6@fel.cvut.cz

1. Princip

Protokol selektivního opakování (Selective Repeat) je protokol zpětné vazby s automatickým opakováním pro zajištění spolehlivé komunikace v síti. Je velmi podobný protokolu Go-Back-N, avšak s rozdílem, že se po ztrátě nějakého paketu posílá pouze daný paket a ne celé okno. Tímto způsobem zajišťujeme spolehlivou komunikaci na nespolehlivých linkách.

Odesílatel posílá několik paketů určených velikostí okna. Pakety si očísluje od jedné do n, kde n je velikost okna. Jakmile odešle celé okno (n paketů), čeká na potvrzení (ACK) od příjemce. Příjemce musí mít stejně veliké okno. Jestliže paket dorazí k příjemci, odesílá ihned ACK s číslem paketu. Zároveň si posouvá svoje okno. Tohle provádí pro celé okno. Jakmile odesílatel přijme ACK se správným číslováním paketů, posouvá si svoje okno a pokračuje v posílání.

1.1 Možné chyby při komunikaci

Jestliže paket či ACK nedorazí, okno se neposouvá. Pokud nedorazí i-tý paket, okno příjemce by se zastavilo na i-té pozici, přičemž pakety $i + 1$ až n si bude muset někde uložit. Pokud však nedorazí ACK, odesílatel okno neposouvá a čeká na timeout. Po uplynutí času timeout se chybějící paket znovu odešle.

2. Velikost okna

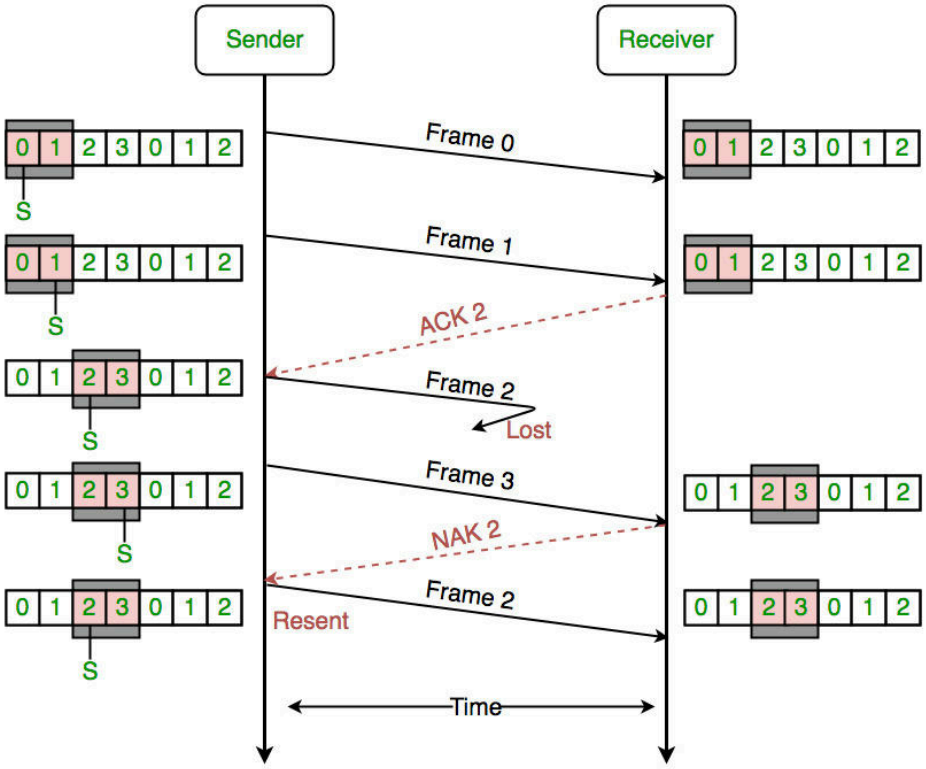
Maximální velikost okna by neměla přesáhnout polovinu rozsahu indexování paketů. Například pokud sender bude číslovat pakety pomocí čísel 0-5 (receiver stejně), a velikost okna by byla 4, mohou nastat situace, kdy sender bude posílat nějaké pakety znovu, protože mu nepřišla potvrzení, a receiver by mohl považovat tyto pakety za nové.

2.1 Příklad

Sender má okno, ve kterém ukládá pakety na poslání a indexuje je čísly 0, 1, 2, 3. Jakmile by poslal 4. paket, znovu ho indexuje číslem 0. Receiver má také okno, ve kterém čísluje pakety 0, 1, 2, 3. Jakmile mu přijde paket s indexem 0, celé okno se posouvá a po čísle 3 se indexuje opět od 0.

V tomto příkladu se posílají 2 pakety najednou (tedy okno spojení má velikost 2). Sender pošle 2 pakety a receiver pošle jedno potvrzení o přijetí celého okna. Při odesílání dalšího okna se ztratí první paket okna, což dá receiver najevo tím, že zašle negativní potvrzení (NAK) s číslem paketu, který je vadný, v tomto případě paket, kterému vypršel čas na doručení. Sender tuto chybu napraví znovuodesláním chybějícího paketu. Až po validním přijetí se sliding window receivera posune dále.

Praha, léto 2024



Obrázek 1: Příklad funkce sliding window

2.2 Minimální velikost okna pro dané zpoždění

Velikost okna by neměla být příliš velká, protože by si receiver musel rezervovat velké množství paměti. Zároveň by ale neměla být příliš malá, protože by se rychlost posílání příliš nelišila od metody *Stop & Wait*.

Mějme zpoždění $R_{\mathbb{T}}$, které lze spočítat jako:

$$R_{\mathbb{T}} = \text{čas přijetí odesílatelem} - \text{čas odeslání}$$

Zpoždění signálu (propagation delay, P) je tedy:

$$P = \frac{R_{\mathbb{T}}}{2}$$

Zpoždění přenosu (transmission delay, T) je doba, kterou trvá odeslání všech bitů paketů v rámci jednoho okna při dané šířce pásma:

$$T = \frac{N}{B}$$

kde N je velikost okna v bitech a B je šířka pásma v bitech za sekundu (bps).

Abychom maximalizovali využití šířky pásma, chceme, aby doba odeslání všech dat v rámci jednoho okna T byla rovna zpoždění signálu P:

$$T = P$$

$$\frac{N}{B} = P$$

$$N = B \cdot P$$

Toto je výsledná velikost okna v bitech pro dané zpoždění.

Praha, léto 2024

3. Ukázka testování pomocí NetDerperu

Nastavení testu

Pro testování vlivu velikosti vysílacího okna na datovou propustnost budeme používat program NetDerper. Simulujeme různé hodnoty zpoždění a velikostí oken, přičemž sledujeme dosaženou datovou propustnost.

Testovací nastavení

Testovali jsme odesílání souboru velikosti 255 kB. Přenos bez chyb potřebuje 258 paketů (255 s daty souboru a další tři na název a velikost, ukončení komunikace a hash). Nastavení pro komunikaci bylo následující: timeout u odesílatele byl nastaven na 0,2 s, míra chyb (error rate) a ztrátovost paketů (drop rate) v obou směrech byla nastavena na 5 %.

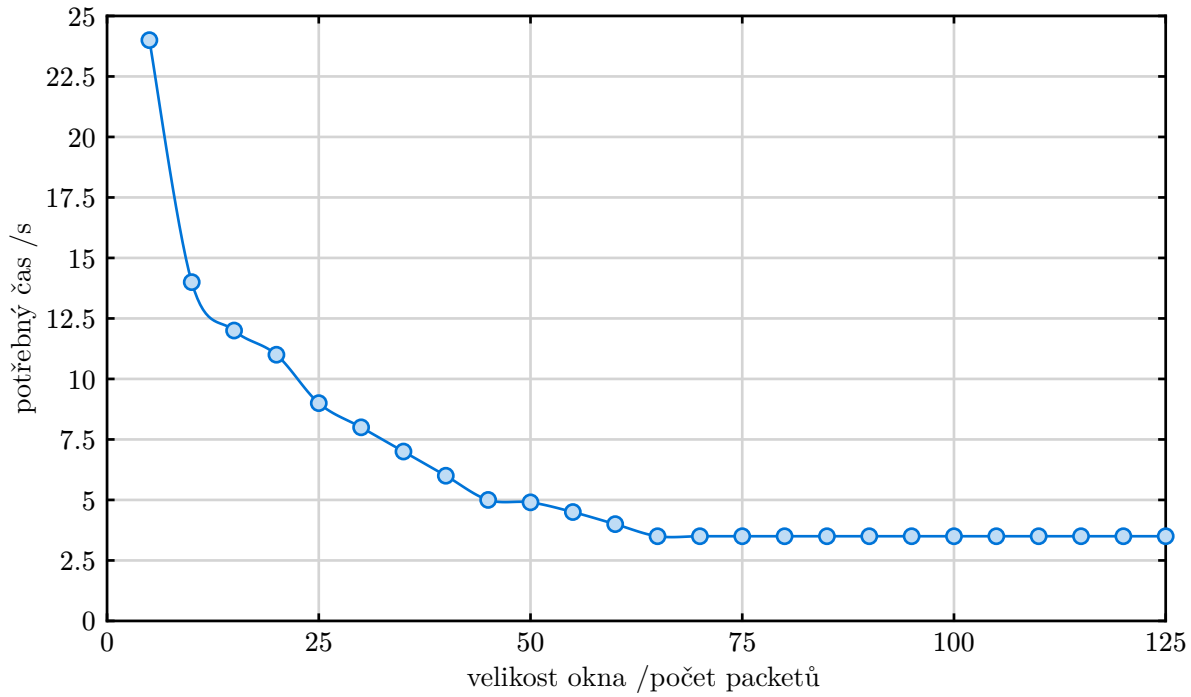
Zpoždění simulované programem NetDerper bylo následující:

- Posílání dat: **Mean**: 100.0 ms, **StdDev**: 15.0 ms
- Posílání potvrzení (ACK): **Mean**: 101.0 ms, **StdDev**: 16.0 ms

Poznámka: **Mean** určuje základní hodnotu zpoždění (střední hodnotu, protože jednotlivé pakety se liší) v milisekundách, **StdDev** značí maximální odchylku od střední hodnoty v milisekundách.

Metodologie

Velikost vysílacího okna jsme měnili od 5 do 125, zvyšovali jsme po 5. U každé velikosti jsme odebrali dva časy, a od velikosti okna 65 jsme začali odebírat čtyři časy, které jsme následně zprůměrovali. Přibližně od velikosti okna 70 už zvětšování okna nemělo žádný efekt na dosaženou datovou propustnost.



Obrázek 2: Závislost rychlosti přenosu na velikosti sliding window

Závěr

Na základě výsledků testování jsme zjistili, že po dosažení velikosti vysílacího okna kolem 70 paketů již další zvětšování okna nemělo žádný vliv na zlepšení datové propustnosti. Tento závěr je důležitý pro efektivní nastavení síťových protokolů v prostředích s podobnými parametry zpoždění a chybovosti.

Praha, léto 2024