Poglavje 8

Modeli generiranja prometa v računalniških omrežjih

Kakršenkoli model računalniškega omrežja, po katerem se pretakajo paketi oziroma zahteve, mora vsebovati tudi takšne sestavne dele, ki so sposobni te pakete oziroma zahteve generirati[1]. Imenujemo jih za *modele generatorjev prometa* ali *modele izvornih točk omrežja*, ki delujejo po nekemu vnaprej določenemu *matematičnemu modelu generiranja prometa*.

Modeli generatorjev prometa so primarno[2] zadolženi za izračune zaporedja časovnih točk, v katerih se zgodijo porajanja posameznih zahtev oziroma paketov. Tako je na prvi pogled primarni rezultat modela generatorja prometa zaporedje izračunanih časovnih točk, v katerih se porodijo posamezne zahteve po izrazu

$$[t_1, t_2, ..., t_n]\,. \tag{8.1}$$

Pri tem čas t_i predstavlja časovno točko porajanja oziroma rojstva i-te zahteve z_i $(i = 1, ..., n)$. Modeli generatorjev prometa v resnici ne izračunavajo absolutnih časovnih točk porajanj po izrazu (8.1), temveč zgolj čase med porajanji ali rojstvi posameznih zahtev[3] (angl. *interarrival time*). Tako je v resnici primarni rezultat modela generatorja prometa zaporedje izračunanih medporajalnih oziroma medrojstnih časov po izrazu

$$[\Delta t_1, \Delta t_2, ..., \Delta t_n]\,, \tag{8.2}$$

pri čemer Δt_1 predstavlja čas, ki preteče od začetka izvajanja simulacije do rojstva prve zahteve z_1, Δt_i pa čas, ki preteče med rojstvoma zahtev z_{i-1} in z_i $(i = 2, ..., n)$. Tako lahko na osnovi izrazov (8.1) in (8.2) zapišemo izraz

$$t_1 = \Delta t_1, t_i = t_{i-1} + \Delta t_i, i = 2, ..., n. \tag{8.3}$$

[1] Razlikovanje med paketi in zahtevami je odvisno od OSI/ISO nivoja, na katerem izvajamo analizo prometa v omrežju.

[2] Sekundarna funkcija generatorja je določitev tipa oziroma velikosti zahteve (paketa).

[3] Čase med porajanji ali rojstvi posameznih zahtev imenujemo tudi za medporajalne oziroma medrojstne čase zahtev.

Za večino programskih simulacijskih okolij velja, da se čas Δt_{i+1} med porojevanjema zahtev z_i in z_{i+1} izračuna v časovni točki t_i, ko se je porodila zahteva z_i.

Modele generatorjev prometa glede na značilnosti izračunanih medporajalnih časov med posameznimi zahtevami oziroma paketi delimo v sledeči dve skupini:

- *deterministični generatorji zahtev*: skozi simulacijo dinamike omrežja generirajo konstantne oziroma enake čase med porajanji sosednjih zahtev; v kontekstu izraza (8.2), tako za deterministično porajane medporajalne čase velja izraz
$$\forall i : \Delta t_i = \Delta t, i = 1, ..., n, \tag{8.4}$$
pri čemer Δt predstavlja konstantno vrednost oziroma konstanten časovni interval;

- *nedeterministični generatorji zahtev*: skozi simulacijo delovanja omrežja generirajo spremenljive - različne čase med porajanji sosednjih zahtev; spremenljivost ali variabilnost teh časov običajno formaliziramo z ustrezajočo verjetnostno porazdelitvijo; glavna razloga za spremenljivost časov med porajanji posameznih zahtev sta nedeterminističnost interakcije uporabnika z izvorno točko omrežja (npr. uporabe spletnih storitev) ter variabilnost časa procesiranja odgovora na zahtevo v računalniških strežnih sistemih;

Že v poglavju o teoriji strežbe smo govorili o karakterističnih porazdelitvah *medprihodnih časov* v strežno enoto. Zakonitosti medprihodnih časov zahtev v strežne enote mnogokrat opišemo s karakteristikami medporojevalnih časov zahtev. Slednje lahko storimo zaradi tega, ker je bodisi model generatorja prometa neposredno povezan s strežno enoto (ji predaja zahteve), ali pa je vhodno breme zahtev proti strežni enoti posredovano s strani neke druge strežne enote, katere strežni proces ima porazdelitveno karakteristiko strežbe, ki je enaka modelu porajalnega procesa.

V pričujočem poglavju predstavimo nekaj tipičnih modelov generiranja prometa zahtev oziroma paketov (angl. *traffic generation model*, *packet generation model*), specifičnih za računalniška omrežja. V nadaljevanju poglavja bomo govorili le o *medporojevalnih časih* zahtev, saj so le ti neposredno povezani z modeli generatorjev prometa. Ključni parametri modelov generatorjev prometa so sledeči:

- *intenzivnost porajanja zahtev* λ: izrazimo jo s številom porajanj zahtev v opazovanem časovnem intervalu; predpostavili bomo, da je λ skozi čas konstantna in se skozi čas torej ne spreminja;

- *model porazdelitve medporojevalnih časov zahtev*: modele porazdelitev lahko glede na naravo medporojevalnih časov razdelimo na deterministične (D) in naključno pogojene (npr. z eksponentno porazdelitvijo M);

- *tipi zahtev* oziroma paketov: s tipi zahtev oziroma paketov imamo v mislih različne vrste zahtev ali paketov in s tem posredno njihove dolžine, od katere je lahko odvisen čas njihovega generiranja ali njihove kasnejše obdelave; na to mesto sodijo tudi eventuelne prioritete, ki jih dodelimo zahtevam oziroma paketom; v pričujočem poglavju bomo zaradi splošnosti predpostavljali, da so tipi zahtev oziroma paketov enaki;

8.1 Vzorčni primeri modelov porazdelitev medrojstnih časov zahtev

V nadaljevanju predstavimo nekaj vzorčnih primerov modelov generiranja zahtev oziroma paketov v računalniških omrežjih, povzetih po viru [30].

8.1.1 Determinističen model porajanja zahtev

V primeru tovrstnega modela porajanja prometa lahko smatramo, da je intenzivnost porajanja zahtev konstantna (ponazorimo jo z oznako λ), konstantni pa so tudi časi med porajanji zahtev. Takšen model lahko ponazorimo s prvim parametrom notacije strežne enote, ki je v tem primeru D, ki definira deterministično oziroma izrojeno porazdelitev medprihodnih časov. Izračun medrojstnega časa v tem primeru izvedemo na osnovi izraza

$$t_{interbirth} = \frac{1}{\lambda}, \tag{8.5}$$

pri čemer $t_{interbirth}$ predstavlja čas med rojstvoma dveh zaporednih zahtev, ki je za vse pare sosednjih zahtev enak.

8.1.2 Poissonov model porajanja zahtev

Poissonov model porajanja prometa se uveljavi pred dobrim stoletjem zaradi potreb modeliranja porajanja prometa v klasični telefoniji, zato ga imenujemo tudi za tradicionalen model porajanja prometa. Uporaben je za modeliranje večjega števila med seboj neodvisnih virov porajanja zahtev (npr. telefonskih klicev ali večih izvorov omrežnih paketov) [31].

Kot smo navedli že v poglavju o teoriji strežbe v Poissonovem modelu porajanja prometa verjetnost določenega števila porojenih zahtev na časovni interval dobimo po Poissonovi verjetnostni porazdelitvi podani z izrazom

$$P(X(t) = k) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}, \tag{8.6}$$

kjer λ predstavlja intenzivnost porajanja zahtev v časovnem intervalu $[0, t]$, k pa število porojenih zahtev. Omenjeno funkcijo smo predstavili že v poglavju o teoriji strežbe.

Izračun medrojstnega časa v tem primeru izvedemo na osnovi izraza

$$t_{interbirth} = -\frac{1}{\lambda} * ln(1-p), \tag{8.7}$$

pri čemer p predstavlja naključno izbrano vrednost z intervala $[0, 1)$.

Zaporedje medrojstnih časov po Poissonovem modelu je med seboj neodvisno, ker je porajalni proces brez pomnjenja (angl. *memoryless*) in naključno porazdeljeno po eksponentni porazdelitvi. Ker je proces Poissonovega porojevanja zahtev brez pomnjenja, je takšen model slab v domeni modeliranja nihajočega oziroma neenakomernega (angl. *bursty*) ali pa soodvisnega porajanja zahtev. V naslednjem razdelku si ogledamo model, ki to zmožnost poseduje.

8.1.3 Dolgorepi model porajanja zahtev

Z vse večjim številom spletnih uporabniških storitev je v zadnjem desetletju prišlo do vidnih sprememb pri porajanju prometa s paketi v računalniških omrežjih. Medrojstni časi paketov naj bi po izvedenih meritvah in statističnih analizah v današnjem času vse bolj gravitirali proti tako imenovani porazdelitvi *„dolgih repov“* (angl. *long tail distribution of interarrival times*, *long tail traffic model*, *heavy tailed traffic model*), za katero se kot eden od najustreznejših modelov porazdelitve uporablja *Paretova porazdelitev* medrojstnih časov. Slednja upošteva medsebojno sodvisnost medrojstnih časov vsebinsko povezanih paketov gledano preko daljšega časovnega obdobja. Za dolgorepo ali Paretovo porazdelitev je karakteristično, da v primerjavi z eksponentno generira manj vrednosti medrojstnih časov z veliko verjetnostjo in generira več različnih vrednosti medrojstnih časov z manjšo verjetnostjo.

Če je naključna spremenljivka X porazdeljena po Paretovi porazdelitvi, zbirno funkcijo verjetnosti (CDF) spremenljivke X ali verjetnost, da naključna spremenljivka X zavzame večjo vrednost od x, zapišemo z izrazom [32]

$$P(X > x) = (\frac{x}{x_m})^{-\alpha} \ for \ all \ x \geq x_m, \tag{8.8}$$

pri čemer naključna spremenljivka X predstavlja medrojstni čas, α pozitivno število in x_m minimalno možno vrednost naključne spremenljivke X. Parameter α imenujemo za *parameter oblike*, x_m pa za *lokacijski parameter*. Medrojstne čase po Paretovi porazdelitvi izračunavamo po izrazu

$$t_{interbirth} = \frac{x_m}{(1-p)^{\frac{1}{\alpha}}}, \tag{8.9}$$

pri čemer p zopet predstavlja naključno izbrano število z intervala $[0, 1)$. Parameter oblike določa kako ukrivljena je krivulja porazdelitve. Manjši je parameter oblike, daljši je rep in obratno. Ob veljavnosti pogoja iz izraza

$$\alpha \leq 1 \tag{8.10}$$

je rep neskončno dolg, ob neveljavnosti tega izraza pa imamo kratek rep - torej velike verjetnosti za krajše medprojstne čase podobne vrednosti lokacijskega parametra in majhne verjetnosti za daljše medrojstne čase na kratkem repu [33].

Osnovni problem dolgorepega modela porajanja zahtev izhaja iz odsotnosti parametra λ, ki ne nastopa neposredno v izrazu (8.9) za izračun medrojstnih časov. Tako velja, da moramo pred uporabo omenjenega modela pridobiti vrednosti parametrov x_m in α. Slednja pridobimo na osnovi meritev porajanja prometa zahtev in iskanja parametrov x_m in α, pri katerih Paretova porazdelitev najbolje „pokrije" dejansko porazdelitev medrojstnih časov.

8.2 Model požrešnega izvora prometa

Uporabo porazdelitev iz predhodnjega razdelka si bomo ogledali na *modelu požrešnega izvora prometa* (angl. *greedy source traffic generation model*). Slednji predstavlja model generatorja prometa, ki zahteve ves čas oddaja s tendenco doseganja neke maksimalne možne intenzivnosti. Ob dokončnem porajanju zahteve z_{i-1} (njeni oddaji na komunikacijski kanal) je potrebno novo zahtevo z_i najprej zgenerirati, nato pa jo odposlati po odhodnem kanalu. Ko je odpošiljanje izvedeno, se gre v generiranje nove zahteve z_{i+1} [34]. Tako na strani modela generatorja zahtev ne prihaja do polnjenja odhodne čakalne vrste.

Porazdelitev medrojstnih časov je tako neposredno odvisna od časov generiranj zahtev t_{g_i} in časov odpošiljanj zahtev t_{s_i}, pri čemer indeks i označuje i-to zahtevo. V primeru, da se i-ta zahteva odpošlje (ali porodi) v časovni točki t_i, tako medrojstni čas med i-to in $i+1$ zahtevo izračunamo po izrazu

$$\Delta t_{i+1} = t_{g_{i+1}} + t_{s_{i+1}}, \tag{8.11}$$

absolutni rojstni čas $i+1$ zahteve pa po izrazu

$$t_{i+1} = t_i + \Delta t_{i+1}. \tag{8.12}$$

Porazdelitev medrojstnih časov je v modelu požrešnega izvora prometa tako odvisna od porazdelitve časov t_{g_i} in $t_{s_i}, i = 1, ..., n$. V primeru, da sta slednja za vse zahteve enaka, govorimo o deterministični porazdelitvi, v primeru pa da nista, moramo raziskati značilnosti obeh časov in v nadaljevanju za njuno okarakterizacijo izbirati med eksponentno, Paretovo ali celo neko tretjo ustreznejšo porazdelitvijo.

Model požrešnega izvora prometa uporabljamo za določitev največje propustnosti omrežja in je ena od osnovnih možnih izbir načina generiranja prometa v vseh programskih simulatorjih generatorjev omrežnega prometa. Najenostavnejši med njimi so `iperf`, `bwping`, `Mausezahn`, `Harpoon` itd.

8.3 Vloga generatorjev psevdo naključnih števil

Kot smo videli v prejšnjih razdelkih, je izračunavanje medrojstnih časov zahtev pogojenih z neko porazdelitvijo v večini primerov pogojeno z uporabo naključnih števil. Za njihovo generiranje so v programskih orodjih zadolženi *generatorji psevdo naključnih števil*[4].

V večini primerov generatorji psevdo naključnih števil temeljijo na iterativnih enačbah, ki jih v splošnem lahko zapišemo z izrazom

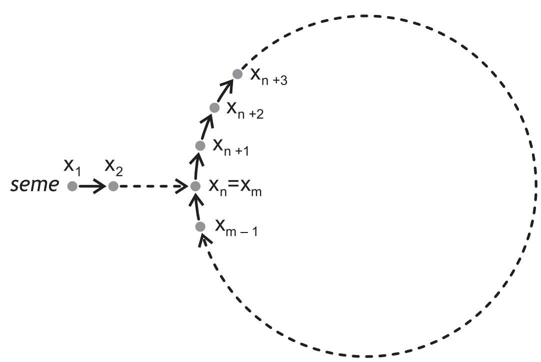
$$x_n = f\left(x_{n-1}, x_{n-2}, \ldots, x_0\right), \tag{8.13}$$

pri čemer x_n predstavlja n-to izračunano psevdo naključno število. Primer enostavne tovrstne funkcije je prikazan v izrazu

$$x_n = (5x_{n-1} + 1) \ \mathbf{mod}\ 16, \tag{8.14}$$

kjer je novo izračunano psevdo naključno število odvisno samo od predhodno izračunanega psevdo naključnega števila. Iz izraza je razvidno, da je vrnjeni rezultat venomer celo število med 0 in 15, kar pomeni, da je generator zmožen tvorbe 16 različnih števil in da se začnejo psevdo naključna števila hitro ponavljati s periodo ponovitev 16. V primeru, da se izračun začne s številom 3 ($x_0 = 3$), je generirano zaporedje psevdo naključnih števil 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3, 0, ... Iz zaporedja je razvidno, da se po 16 korakih začnejo števila ponavljati.

Število x_0, s katerim se začne izračun psevdo naključnega generatorja, se imenuje *seme* (angl. *seed*). Praviloma ima uporabnik generatorja možnost načina določanja semena, a se v večini primerov za seme uporablja vnaprej določena konstantna privzeta vrednost, zato je rezultat naključnega generatorja ob zagonu vedno enak. V nekaterih primerih je za seme privzeta vrednost hitro spreminjajoče se sistemske spremenljivke gostitelja (npr. njegove realne ure). Primer takšne spremenljivke so stotinke ali tisočinke sekunde. Na sliki 8.1 je prikazan splošni cikel generatorja psevdo naključnih števil.



Slika 8.1: Cikel generatorja psevdo naključnih števil [4].

Seme je lahko tudi število, ki ne pripada množici števil iz izračunanega cikla psevdo naključnih vrednosti. Dober naključni generator mora biti računsko enostaven, da za izračun ne porabimo preveč procesne moči. Perioda ali dolžina cikla generatorja mora biti čimvečja, sicer se začnejo generirana števila prehitro ponavljati. Sosednja števila v zaporedju morajo biti statistično neodvisna in enakomerno porazdeljena. V literaturi lahko najdemo več primerov generatorjev, ki pa so po večini vezani na dolžino besede računalnika. Primeri takšnih generatorjev so podani v izrazih

$$x_n = 7^5 x_{n-1} \ \mathbf{mod} \ \left(2^{31} - 1\right), \tag{8.15}$$

$$x_n = \left(2^{16} + 3\right) x_{n-1} \ \mathbf{mod} \ 2^{31}, \tag{8.16}$$

$$x_n = \left(2^{8} + 3\right) x_{n-1} \ \mathbf{mod} \ 2^{15}, \tag{8.17}$$

$$x_n = 2^{13} x_{n-1} \ \mathbf{mod} \ 2^{35}. \tag{8.18}$$

Dobri generatorji psevdo naključnih števil morajo biti tako hitro izračunljivi, imeti morajo čim večjo zalogo vrednosti, izračunane vrednosti pa morajo biti statistično neodvisne in čimbolj enakomerno porazdeljene. Posebno moramo biti pozorni, kakšno je začetno seme generatorja in ali je to ob zagonu aplikacije venomer enako. Če je temu tako, bomo ob vsakokratnem zagonu simulacije modela venomer dobivali enake simulacijske rezultate.

Literatura

[1] N. C. Hock, *Queuing Modelling Fundamentals.* John Wiley & Sons, Chichester, Anglija, 1996.

[2] M. Anu, "Introduction to modeling and simulation," in *Proceedings of the 29th conference on Winter simulation* (S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, eds.), pp. 7–13, 1997.

[3] L. Kleinrock and R. Gail, *Queuing systems, problems and solutions.* John Wiley & Sons, New York, ZDA, 1996.

[4] N. Zimic and M. Mraz, *Temelji zmogljivosti računalniških sistemov.* Založba FE in FRI, Ljubljana, Slovenija, 2006.

[5] R. Jamnik, *Verjetnostni račun in statistika.* Društvo matematikov, fizikov in astronomov socialistične republike Slovenije, Zveza organizacij za tehnično kulturo Slovenije, Ljubljana, Slovenija, 1986.

[6] K. S. Trivedi, *Probability and Statistics with Reliability, Queueing and Computer Science Applications.* John Wiley & Sons Inc., New York, ZDA, 2002.

[7] H. Stöcker, *Matematični priročnik z osnovami računalništva.* Tehnična založba Slovenije, Ljubljana, Slovenija, 2006.

[8] J. Virant, *Modeliranje in simuliranje računalniških sistemov.* Didakta, Radovljica, Slovenija, 1991.

[9] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, *Fundamentals of queueing theory.* John Wiley & Sons, Hoboken, ZDA, 2018.

[10] J. L. Peterson, *Petri Net Theory and the Modeling of Systems.* Prentice Hall Inc., Englewood Cliffs, ZDA, 1981.

[11] J. Bordon, M. Moškon, N. Zimic, and M. Mraz, "Semi-quantitative Modelling of Gene Regulatory Processes with Unknown Parameter Values Using Fuzzy Logic and Petri Nets," *Fundamenta Informaticae*, vol. 160, no. 1–2, pp. 81–100, 2018.

[12] J. Virant, *Logične osnove odločanja in pomnjenja v računalniških sistemih.* Založba FE in FRI, Ljubljana, Slovenija, 1996.

[13] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of The IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[14] W. G. Schneeweiss, *Petri Nets for Reliability Modeling.* LiLoLe Verlag, 1999.

[15] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks.* Prentice Hall Inc., Boston, ZDA, 2011.

[16] N. Jensen and L. Kristensen, *Coloured Petri Nets.* Springer,, 1998.

[17] K. Jensen, "A Brief Introduction to Coloured Petri Nets," in *Proceedings of the Third International Workshop on Tools and Algorithms for Construction and Analysis of Systems (TACAS '97)*, 1997.

[18] K. Jensen, *Coloured Petri Nets Basic concepts.* Springer, 1997.

[19] D. Božić, *Analiza in zgled uporabe programskega orodja CPNTools za postavljanje modelov dinamičnih sistemov.* Diplomsko delo FRI-UL, 2012.

[20] M. Dolenc, *Verifikacija komunikacijskih protokolov na osnovi barvnih Petrijevih mrež.* Diplomsko delo FRI-UL, 2015.

[21] "Matematična ocena latence proizvajalca Rugged." `https://w3.siemens.com/mcms/industrial-communication/en/rugged-communication/Documents/AN8.pdf/`, 2015.

[22] "Matematična ocena latence proizvajalca O3b Networks." `http://www.o3bnetworks.com/media/40980/white%20paper_latency%20matters.pdf/`, 2015.

[23] "Primer on Latency and Bandwidth." `https://www.oreilly.com/library/view/high-performance-browser/9781449344757/ch01.html`, 2018.

[24] "Hop Count Definition." `https://www.techopedia.com/definition/26127/hop-count`, 2019.

[25] "Network traffic monitoring." `https://www.techopedia.com/definition/29977/network-traffic-monitoring`, 2018.

[26] "Internet dveh hitrosti." `http://webfoundation.org/2015/10/net-neutrality-fails-to-load-web-foundation-response-to-todays-eu-vote//`, 2016.

[27] "About Cacti." `http://www.cacti.net//`, 2016.

[28] P. Antončič, *Monitoriranje računalniških omrežij.* Diplomsko delo FRI-UL, 2012.

[29] N. Bricman, *Zasnova fizičnega omrežja za potrebe novega ponudnika internetnih storitev.* Diplomsko delo FRI-UL, 2018.

[30] J. Zhang, J. Tang, X. Zhang, W. Ouyang, and D. Wang, "A survey of network traffic generation," in *Third International Conference on Cyberspace Technology*, 2015.

[31] "J.Virtamo: Poisson process." `https://www.netlab.tkk.fi/opetus/s383143/kalvot/E_poisson.pdf`, 2018.

[32] A. Pehlivanli, "Software implementations of QOS scheduling algorithms for high speed networks," Master's thesis, Middle East Technical University, 2015.

[33] T. Zupančič, *Analiza vpliva medprihodnih časov zahtev na njihove čakalne čase v procesu strežbe.* Diplomsko delo FRI-UL, 2020.

[34] A. Banerjea and S. Keshav, "Queueing Delays in Rate Controlled ATM Networks," in *Proceedings of IEEE INFOCOM '93 The Conference on Computer Communications*, 1993.