

УДК 681.324:621.325

А.А. Коваленко

Харківський національний університет радіоелектроніки, Харків

ДИНАМІЧНЕ КЕРУВАННЯ ПАРАМЕТРАМИ ПРОТОКОЛУ TCP VEGAS

Проведений аналіз впливу коефіцієнтів α та β протоколу TCP Vegas на його ефективність при різних мережесих сценаріях. Розглянуті випадки існування декількох з'єднань. Наведені графіки залежності ефективності протоколу TCP Vegas при різних значеннях цих коефіцієнтів.

протокол TCP Vegas, трафік, ефективність, алгоритм, модель, мережа, плаваюче вікно, запобігання перевантаженням, пакет

Вступ

Протокол TCP Vegas, на відміну від протоколу TCP Reno, у якого спостерігається більший приріст швидкості за одиницю часу, показує кращі результати ефективності та меншу кількість повторних передач пакетів. Однак, для TCP Vegas, все ще існує декілька значних перешкод, які не дозволяють йому використовуватися в сучасних високошвидкісних мережах, у тому числі військового призначення, наприклад: нерівноправність розподілення доступних ресурсів мережі, яке виникає при одночасному існуванні у каналі потоків TCP Vegas та фонового трафіка (наприклад, з'єднань сучасним протоколом TCP Reno); проблеми, які виникають при динамічній зміні маршруту з'єднання; можливість виникнення постійного перевантаження та невідповідність швидкості потоку даних параметрам мережі. У даній статті досліджуються перелічені вище проблеми та пропонується модифікація механізму керування перевантаженнями протоколу TCP Vegas, яка дозволяє у значній мірі вирішувати більшість з них. Ці рішення не залежать а ні від значень важливих параметрів протоколу, а ні від схем керування буферами маршрутизаторів.

Протокол TCP Vegas використовує принципово іншу схему керування перевантаженнями [1], ніж протокол TCP Reno, досягаючи більшої ефективності, що обумовлено значно меншою кількістю повторних передач пакетів та відсутністю впливу на нього з'єднань з відносно більшою затримкою розповсюдження сигналу в з'єднанні T. Однак у потоків протоколом TCP Vegas не спостерігається рівномірного розподілення смуги пропускання з потоками протоколом TCP Reno [2].

Завдяки розділенню протоколу TCP Vegas на різні фази та описанню їх впливу на ефективність, було показано, що заявлений приріст ефективності протоколу TCP Vegas досягається в основному за рахунок нової реалізації фаз повільного старту та відновлення від перевантажень, у той час як механізм запобігання перевантаженням має мінімальний вплив на ефективність, але є ключовим у питаннях справедливості розподілення доступних ресурсів мережі [3]. У разі використання маршрутизаторів з дисципліною обслуговування черги RED замість аналогічних, але реалізуючих, алгоритм відкидання задньої частини черги (Droptail), рівномірність розподілення доступних ресурсів мережі між протоколами TCP Vegas та TCP Reno у деякій мірі покращу-

ється [4, 5]. При будь-якому сценарії існує неминучий компроміс між справедливістю та ефективністю цих протоколів: якщо імовірність виключення пакета із черги алгоритмом RED встановити досить великою, ефективність протоколу TCP Vegas, зокрема, збільшиться, однак загальна ефективність знизиться. У даній статті запропоновані зміни лише для механізму запобігання перевантаженням протоколу TCP Vegas з метою вирішення проблем, пов'язаних з його ефективністю.

Джерело [6] розглядає проблеми, які пов'язані з використанням протоколом TCP Vegas оцінки параметра T для прийняття рішень.

Зокрема, розглянуті проблеми динамічної зміни маршруту з'єднання, постійного перевантаження та невідповідності швидкості потоку параметрам мережі. Запропоновано вирішення першої з проблем та показано можливе вирішення проблеми постійного перевантаження у з'єднанні.

Однак пошук оптимальних значень параметрів для механізму RED та запропонованої модифікації механізму запобігання перевантаженням протоколу TCP Vegas усе ще залишаються відкритими питаннями. Ще не було запропоновано механізмів керування перевантаженнями протоколу TCP Vegas, які дозволяли б підвищувати його ефективність у гетерогенних мережевих середовищах.

Таким чином, задача створення й дослідження нових методів та алгоритмів фази запобігання перевантаженням протоколу TCP Vegas, які дозволяють йому добиватися рівномірності розподілення мережевих ресурсів у гетерогенних мережевих сценаріях, є **актуальною**.

Метою даної статті є дослідження впливу параметрів протоколу TCP Vegas на його ефективність та розробка нового алгоритму динамічного керування параметрами протоколу з метою підвищення його ефективності.

Результати теоретичних досліджень

Алгоритм запобігання перевантаженням. Протокол TCP керує швидкістю відправки пакетів, базуючись на оцінці доступного розміру смуги пропускання з'єднання. Серед усіх нововведень, запропонованих протоколом TCP Vegas, найбільш вагомий нюанс полягає в новій, відносно протоколу TCP Reno, схемі оцінювання доступного розміру смуги пропускання.

Протокол TCP Vegas динамічно змінює розмір вікна пакетів, що передаються, базуючись на вимірюваннях параметра T , тоді коли протокол TCP Reno продовжує збільшувати розмір свого плаваючого вікна до втрати пакета, яку він розглядає як ознаку перевантаження. TCP Vegas використовує значно більш складну схему оцінки доступного з'єднанню розміру смуги пропускання, для чого обчислює різницю між очікуваною (теоретично мо-

жливою у даний момент часу) та реальною швидкостями відправки пакетів у мережу. Ідея полягає в тому, що коли очікувана та реальна швидкості практично рівні, мережа не перевантажена.

Іншими словами, під час перевантаження, реальна швидкість буде значно меншою, ніж очікувана. Таким чином, механізм керування перевантаженнями протоколу TCP Vegas функціонує таким чином:

– виконується обчислення очікуваної швидкості відправки пакетів джерелом у мережу:

$$V_o = w(t)/T_b, \quad (1)$$

де V_o – очікувана швидкість, $w(t)$ – теперішній розмір плаваючого вікна, а T_b – мінімальне значення T даного з'єднання;

– обчислюється реальна швидкість відправки пакетів джерелом у мережу як

$$V_p = w(t)/T_p, \quad (2)$$

де V_p – реальна швидкість, а T_p – реальне значення параметра T ;

– для кожного пакета підтвердження, що приймається, джерело виконує оцінку кількості пакетів, які знаходяться в черзі маршрутизатора, як різницю

$$\delta = (V_o - V_p) \cdot T_b \quad (3)$$

– знаючи значення параметра δ , розмір плаваючого вікна $swnd$ корегується таким чином:

$$w(t) = \begin{cases} w(t-1)+1, & \text{якщо } \delta < \alpha; \\ w(t-1)-1, & \text{якщо } \delta > \beta; \\ w(t-1), & \text{в інших випадках,} \end{cases} \quad (4)$$

де α – згладжуючий коефіцієнт у процедурі обчислення тривалості циклу таймера повторного передавання пакета даних; β – коефіцієнт дисперсії затримки розповсюдження сигналу на заданому маршруті.

Із алгоритму, що наведений вище, видно, що коли очікувана та реальна швидкості близькі одна до одної, з'єднання не використовує увесь доступний розмір смуги пропускання, а отже, необхідно збільшувати швидкість відправки даних. Коли ж реальна швидкість значно нижче очікуваної, то мережа перевантажена, і з'єднання повинне знижувати швидкість.

У фазі повільного старту, вхід до якої здійснюється одразу після успішного встановлення з'єднання, протокол TCP Vegas залишається до досягнення деякого порогового значення γ (зазвичай рівного 1). Доки виконується умова $\delta < \gamma$, здійснюється збільшення розміру плаваючого вікна на один пакет за кожну другу годину передавання пакета. Отже, під час фази повільного старту розмір плаваючого вікна протоколу TCP Vegas збільшується експоненційно, хоча і з меншою швидкістю, ніж у протоколу TCP Reno. Коли або розмір плаваючого вікна $swnd$ досягає порогового значення виходу з фази повільного старту, або починає виконуватися умова $\delta > \gamma$, протокол TCP Vegas входить у фазу запобігання перевантаженням.

У даній фазі протокол TCP Vegas оперує вже з двома пороговими значеннями – α та β (звичайно рівним 1 та 3, відповідно) і змінює розмір плаваючого вікна відповідно до умови (4).

Втрата пакета може виявлятися за допомогою однієї з двох подій: закінченням таймера або одержанням трьох пакетів підтверджень, які підряд дублюються. У першому випадку порогове значення виходу з фази повільного старту встановлюється в половинне значення теперішнього розміру плаваючого вікна, а розмір плаваючого вікна – в одиницю (2 в деяких реалізаціях), і потім протокол TCP Vegas знов входить у фазу повільного старту. У другому випадку, коли джерело TCP Vegas одержує три пакети підтвердження, які дублюються, він виконує швидке повторне передавання та швидке відновлення, як і протокол TCP Reno. Фактично, у протоколі TCP Vegas реалізований покращений механізм швидкого повторного передавання, який базується на більш точному внутрішньому таймері [2]. Після фази швидкого повторного передавання протокол TCP Vegas встановлює розмір свого плаваючого вікна в 3/4 його теперішнього значення та знов запускає фазу запобігання перевантаженням.

Незважаючи на те, що протокол TCP Vegas має низку переваг порівняно з сучасними реалізаціями протоколу TCP, як краща ефективність, менші флуктуації розміру плаваючого вікна та значно менша кількість повторних передач пакетів, також існують недоліки, які стримують його впровадження в сучасні мережі.

Особливості розподілення доступних ресурсів мережі. Протокол TCP Vegas використовує досить стриманий алгоритм для прийняття рішень стосовно збільшення/зменшення розміру плаваючого вікна. На відміну від нього, TCP Reno, з метою повного використання доступного розміру смуги пропускання каналу, продовжує збільшення розміру плаваючого вікна до першої втрати пакета. У той час як TCP Vegas намагається підтримувати у чергах невелику кількість своїх пакетів, TCP Reno та подібні йому реалізації вимагають значно більшої кількості буферного простору для зберігання своїх пакетів, що призводить до захоплення більшої частини смуги пропускання каналу.

Коли потоки даних протоколами TCP Vegas та TCP Reno проходять крізь один канал, який є для них вузьким містом, Reno захоплює значно більшу частину буфера маршрутизатора. TCP Vegas сприймає це як сигнал перевантаження та зменшує розмір свого плаваючого вікна, що призводить до нерівномірного розподілу смуги пропускання (для протоколу TCP Vegas). Більше того, коли розмір буфера маршрутизатора стає великим, середній розмір вікон з'єднань TCP Reno також буде великим, у той час, коли розмір вікна з'єднань протоколом TCP Vegas

буде залишатися незмінним, що обумовлено значенням константи β . Це означає, що більший розмір буфера маршрутизатора негативно відбивається на рівномірності розподілення доступних ресурсів мережі між з'єднаннями протоколами TCP Reno та TCP Vegas.

Однією зі спроб подолання проблем справедливості протоколу TCP Vegas є протокол TCP Vegas⁺ [4]. Він має два режими – помірний та агресивний. У помірному режимі він поводить себе повністю ідентично протоколу TCP Vegas, але може переходити в агресивний режим для збільшення розміру плаваючого вікна ідентично протоколу TCP Reno, при виявленні в каналі з'єднань протоколом TCP Reno. Однак TCP Vegas⁺ вважає, що збільшення значення T завжди відбувається тільки внаслідок виникнення фоновому трафіка у каналі, виключаючи всілякі інші фактори. Окрім того, ефективність протоколу TCP Vegas⁺ залежить від вибору оптимального значення параметра, який відповідає за переключення між звичайним та агресивним режимами його роботи.

Динамічне змінення маршруту з'єднання. У протоколі TCP Vegas параметр T_b означає найменшу затримку розповсюдження сигналу у з'єднанні та використовується для кількісного виміру очікуваної швидкості. При зміні маршруту параметр T з'єднання може змінитися. На TCP Vegas цей факт не впливає в значній мірі, окрім того випадку, коли новий маршрут має менше значення T , оскільки значення параметра T_b оновиться. Однак, якщо новий маршрут буде мати більше значення T , з'єднання буде нездатним визначити причину такого збільшення – початок перевантаження або зміна маршруту. Протокол TCP Vegas вважає, що таке збільшення T обумовлено перевантаженням і, отже, зменшує розмір плаваючого вікна.

Це є суворо протилежною дією, як треба було діяти в такій ситуації: при збільшенні затримки добуток розміру смуги пропускання з'єднання на затримку у ньому ($b \cdot d$), що є оцінкою кількості пакетів, які знаходяться в мережі, збільшується. Для будь-якого з'єднання вираз $w(t) \cdot b \cdot d$ є оцінкою кількості пакетів, які знаходяться в чергах на маршруті з'єднання. Оскільки завданням протоколу TCP Vegas є підтримка кількості пакетів у чергах між значеннями α та β , то він повинен збільшувати розмір плаваючого вікна для підтримки незмінної кількості пакетів у чергах при збільшенні затримки.

Джерелом [6] запропонована модифікація до протоколу TCP Vegas, яка дозволяє йому краще протидіяти проблемі динамічної зміни маршруту з'єднання. Ця модифікація як сигнал зміни маршруту використовує тривале збільшення параметра T . Для перших K пакетів механізм аналогічний TCP Vegas. Після цього джерело спостерігає за величи-

ною T для кожних N пакетів і якщо мінімальне T значно більше значення T_b на протязі L наступних разів, то джерело встановлює значення T_b у мінімальне зі значень T за N послідовних пакетів, а значення $cwnd$ – відповідно до нового значення T_b . Ідея полягає у тому, що оскільки збільшення параметра T провокує джерело до зменшення величини $cwnd$, то збільшення T здебільшого відбувається внаслідок збільшення затримки у новому маршруті, а не з причини перевантаження. Однак у даному механізмі існують ще два параметри – δ та γ , які пов'язані зі значенням T_b , а пошук оптимальних значень для параметрів K , N , L , δ та γ залишається нерозв'язаним завданням.

Згідно з [3], механізм керування перевантаженнями протоколу TCP Vegas є несправедливим до більш довгоживучих з'єднань. Коли з'єднання встановлене на непереважній ділянці мережі, значення T_b буде близьким до мінімально можливого. Якщо пізніше починається перевантаження, вимірюване значення T збільшується, а відношення T_b/T зменшується. Розглянемо нове з'єднання, яке відчиняється після того, як у мережі почалося перевантаження. Значення $T_{b\ i+1}$ даного з'єднання буде більш ніж $T_{b\ i}$, а, отже, відношення $T_{b\ i+1}/T_{i+1}$ буде більше, відношення T_b/T_i , оскільки значення $T_{b\ i}$ та $T_{b\ i+1}$ практично рівні. Умова для початку зменшення розміру плаваючого вікна має такий вигляд [3]:

$$cwnd > \beta / \left(1 - \frac{T_b}{T_p} \right). \quad (5)$$

Звідси видно, що для більш давніх з'єднань критичне значення параметра $cwnd$ буде меншим, ніж для більш нових з'єднань. Як наслідок – більш «молоді» з'єднання можуть досягати більших значень ефективності.

Умовою початку збільшення розміру плаваючого вікна є такий вираз:

$$cwnd < \alpha / \left(1 - \frac{T_b}{T_p} \right). \quad (6)$$

Оскільки значення виразу, який знаходиться в правій частині, більше для «молодих» з'єднань, у них більша ймовірність збільшення розміру плаваючого вікна, що призводить до нерівномірного розподілення доступної смуги пропускання.

Досить близький зміст має проблема постійного перевантаження, коли з'єднання можуть невірно оцінювати величини затримок, то вводять мережу в постійно перевантажений стан. Нехай у перевантаженій мережі відчиняється з'єднання. Внаслідок деякої затримки постановки пакетів у чергу, яка обумовлена наявністю пакетів інших з'єднань, пакети нового з'єднання мають значення T значно більші, ніж дійсна затримка на даному маршруті. Як результат такого неточного оцінювання параметра

T_b – розмір вікна буде переоцінений. Такий сценарій буде в точності повторюватися для кожного нового з'єднання, що призведе до неперервного перевантаження. Метод, що аналогічний вищеописаному, для випадку динамічної зміни маршруту з'єднання у поєднанні з використанням маршрутизаторів з дисципліною RED може бути одним з можливих рішень для проблеми постійного перевантаження [6]. У джерелі [7] запропонована деяка модифікація протоколу TCP Vegas з дисципліною активного керування чергою, яка подібна механізму випадкового експоненційного маркування пакетів (REM), яка також дозволяє вирішувати проблему виникнення постійного перевантаження в мережі.

Модифікація алгоритму. У протоколі TCP Vegas величини α та β є константами, а завданням протоколу є таке керування розміром плаваючого вікна, коли кількість пакетів, що знаходяться в буферах на маршруті з'єднання, буде знаходитися між значеннями α та β . Керуючись цим принципом, у новому алгоритмі організована динамічна зміна значень параметрів α та β . При відчиненні з'єднання параметри α та β встановлені у свої значення за замовчуванням, а потім починають динамічно змінюватися відповідно до завантаження мережі. Фази повільного старту та відновлення від перевантажень залишились незмінними (як у протоколі TCP Vegas), змінам піддався лише механізм запобігання перевантаженням.

Для збереження значення дійсної швидкості, яка була досягнута за попередній час передавання пакета, у алгоритм введена лише одна нова змінна, оскільки протокол TCP Vegas виконує обчислення дійсної швидкості за кожен час передавання пакета.

Модифікований алгоритм фази запобігання перевантаженням показаний на рис. 1.

Практичні результати

Моделювання проводилося при використанні The Network Simulator, ns-2 [8]. Топологія мережі аналогічна такій у [9] та наведена на рис. 2.

Модель мережі складається з трьох джерел трафіка, потоки даних від яких проходять через канал (між вузлами $n1$ та $n2$), який є вузьким місцем маршруту з'єднання, що показано на рис. 2. Канали між вузлами $n0 - n1$, $n3 - n1$, $n6 - n1$, $n2 - n4$, $n2 - n5$ і $n2 - n7$ мають пропускну здатність 1 Гб/с та затримку 1 мс. Вузол $n0$ відправляє дані до вузла $n4$, вузол $n3$ – вузла $n5$, а між вузлами $n6$ та $n7$ генерується фоновий трафік. Канал між вузлами $n1$ та $n2$ має ємність, що дорівнює 100 Мб/с, затримку 10 мс, та є вузьким місцем усіх з'єднань. Вузли $n1$ та $n2$ є маршрутизаторами з дисципліною відкидання задньої частини черги. Для аналізу ефективності цих двох протоколів TCP використовуються залежності розміру плаваючого вікна від часу та швидкості відправлення пакетів у мережу кожним зі з'єднань від часу.

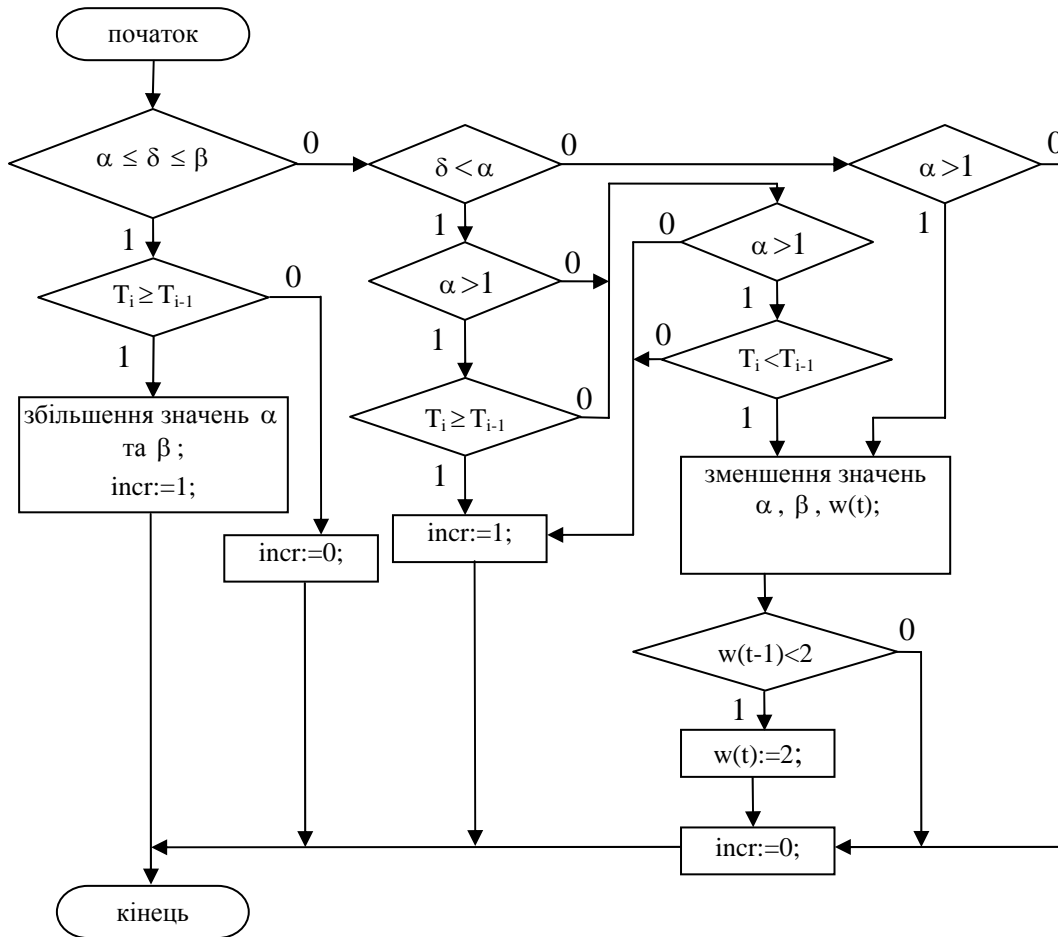


Рис. 1. Алгоритм фази запобігання перевантаженням протоколу TCP Vegas з динамічною зміною параметрів α та β

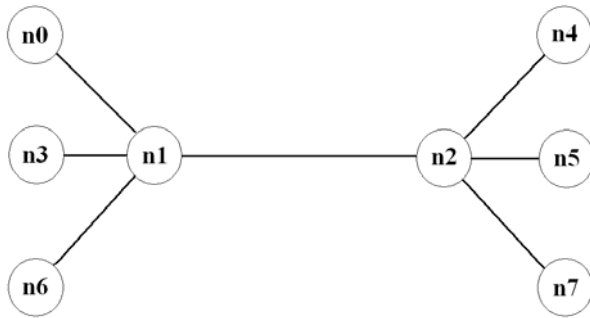


Рис. 2. Структура мережі, що моделюється

У першому експерименті моделювався випадок з джерелами трафіку протоколами TCP Vegas, TCP Reno та фоновим UDP трафіком. Результати для розміру плаваючого вікна та швидкості наведені на рис. 3 та 4, відповідно.

Спостерігається характерна нерівномірність розподілення доступних ресурсів мережі по відношенню до протоколу TCP Vegas, що відповідає [9], та, як наслідок, значне зниження швидкості з'єднань протоколом TCP Vegas в присутності з'єднань TCP Reno в одному каналі.

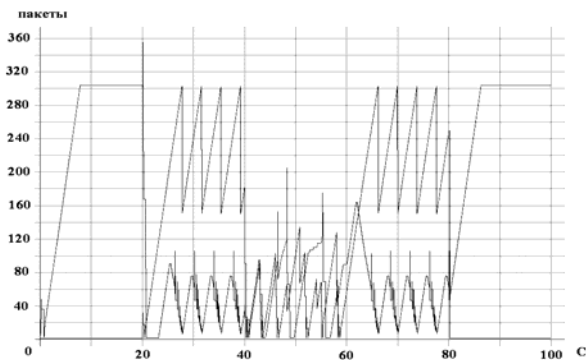


Рис. 3. Залежність розміру вікна від часу для протоколу TCP Vegas

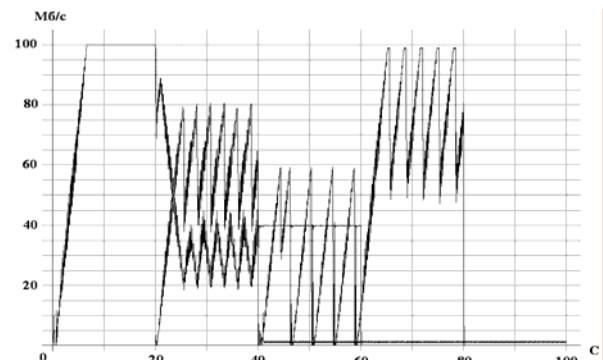


Рис. 4. Залежність швидкості від часу для протоколу TCP Vegas

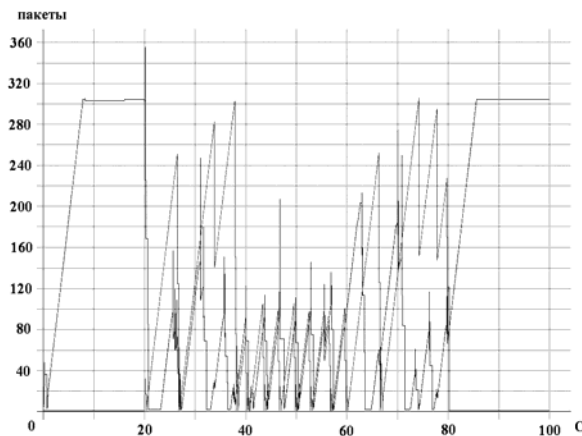


Рис. 5. Залежність розміру вікна від часу для протоколу TCP Vegas з динамічною зміною параметрів

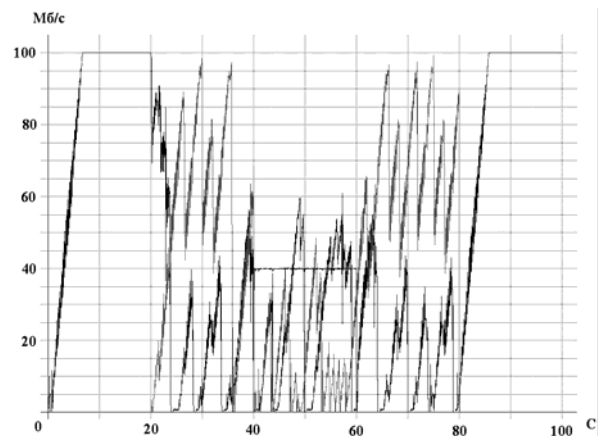


Рис. 6. Залежність швидкості від часу для протоколу TCP Vegas з динамічною зміною параметрів

Протокол TCP Vegas використовувався з його звичайними параметрами ($\alpha = 1$, $\beta = 3$) [1]. З'єднання протоколом TCP Vegas активне в інтервалі часу 0-100 секунд, з'єднання TCP Reno – 20-80 секунд, а фоновий трафік інтенсивністю 40 Мб/с існує в інтервалі 40-60 секунд.

У наступному сценарії промодельований випадок, аналогічний попередньому, але за винятком того, що замість протоколу TCP Vegas застосований описаний вище удосконалений алгоритм. Результати показані на рис. 5 та 6.

У другому експерименті спостерігається збільшення розміру вікна протоколу TCP Vegas з динамічною зміною параметрів у випадку гетерогенного середовища та, як наслідок, ефективності, що обумовлено динамічною зміною параметрів α та β .

Висновки

Були розглянуті питання роботи протоколів TCP Vegas та його версії з динамічною зміною параметрів у гетерогенному мережевому сценарії.

Наведені результати моделювань гетерогенного мережевого середовища, де присутні з'єднання протоколами TCP Vegas та TCP Reno з однаковими затримками розповсюдження сигналів у з'єднаннях сумісно з фоновим UDP трафіком. Вони показують низьку сумісність звичайних реалізацій протоколів TCP Vegas та TCP Reno. Протокол TCP Vegas має більшу схильність до переважань при виникненні фонового трафіка та суттєво меншу агресивність (або динамічну швидкість зміни розміру плаваючого вікна), ніж TCP Reno.

Модифікована версія протоколу TCP Vegas дозволяє досягати більш рівномірного розподілення доступної смуги пропускання каналу з іншими потоками TCP.

Протокол TCP Vegas з динамічною зміною параметрів показав більший рівень агресивності, що відбивається у збільшеній спроможності захоплення доступних ресурсів каналу передачі даних, по від-

ношенню до співіснуючого трафіка та, як наслідок, більш рівномірне розподілення мережевих ресурсів у гетерогенному середовищі.

Список літератури

1. Brakmo L.S., O'Malley S.W., Peterson L.L. TCP Vegas: New techniques for congestion detection and avoidance // Proc. of ACM SIGCOMM'94. – P. 24-35.
2. Коваленко А.А., Завизиступ Ю.Ю., Партыка С.А. Проблемы производительности протокола TCP в гетерогенных сетях и методы ее улучшения // Вестник ХНТУ. – 2005. – № 1 (21). – С. 305-311.
3. Коваленко А.А., Завизиступ Ю. Ю., Партыка С.А. TCP: проблемы производительности и влияние на нее различных факторов // Автоматизированные системы управления и приборы автоматики. – 2005. – № 133. – С. 113-119.
4. Hasegawa, Kurata K., Murata M. Analysis and Improvement of Fairness between TCP Reno and Vegas for Deployment of TCP Vegas to the Internet // Proc. of the IEEE International Conference on Network Protocols (ICNP 2000).
5. Raghavendra, Kinicki R.R. A Simulation Performance study of TCP Vegas and Random Early Detection // Proc. of IPCCC'99. – P. 169-176.
6. Ильина И.В., Можяев А.А., Коваленко А.А. Многоуровневая модель телекоммуникационного трафика системы глобальной навигации // Системи управління, навігації та зв'язку. – 2007. – Вип. 2. – С. 76-82.
7. Steven Low, Larry Peterson, Limin Wang. Understanding TCP Vegas: A Duality Model // Proc of ACM SIGMETRICS 2001. – P. 226-235.
8. The Network Simulator (NS-2). – [Електрон. ресурс]. – Режим доступу: <http://www.isi.edu/nsnam/ns>.
9. Завизиступ Ю.Ю., Коваленко А.А., Партыка С.А., Бабич А.В. Проблемы управления перегрузками в протоколе TCP // Вестник ХНТУ. – 2006. – № 1 (24). – С. 173-177.

Надійшла до редколегії 30.03.2007

Рецензент: канд. фіз.-мат. наук, ст. наук. співр. О.О. Можяєв, Харківський університет Повітряних Сил ім. І. Кожедуба, Харків.