ارزیابی تأخیر کدگشایی در روش کدگذاری پرپچوال

ساناز محمدی [۱]، کارشناسی ارشد علوم کامپیوتر، پیمان پهلوانی [۲]، استادیار

۱- دانشکده علوم رایانه و فناوری اطلاعات - دانشگاه تحصیلات تکمیلی علوم پایه - زنجان - ایران – mohammadi.s@iasbs.ac.ir
۲- دانشکده علوم رایانه و فناوری اطلاعات - دانشگاه تحصیلات تکمیلی علوم پایه - زنجان - ایران – pahlevani@iasbs.ac.ir

چکیده: کدگذاری پرپچوال روش کدگذاری تنک است که ضرایب به صورت ساختار یافته برای عملیات کدگذاری استفاده می‌شود. نشان داده شده است که این روش پیچیدگی محاسباتی روش کدگذای خطی تصادفی را کاهش می‌دهد. هدف از این مقاله بیان یک مدل ریاضی برای نشان دادن عملکرد کدگذاری پرپچوال است و نشان دادن این مطلب که در کدگذاری پرپچوال در کانال‌های دارای خطا بسته‌های وابسته‌ی خطی ارسالی به شدت به پارامتر عرض (ω) بستگی دارد. پارامتر عرض به تعداد ضرایب غیر صفر پشت سر هم که در هر بسته‌ی کد شده بعد از عنصر محور می‌آید گفته می‌شود. سپس یک مدل تحلیلی ریاضی برای تعداد بسته‌های ارسال شده ارائه می‌شود که مدل ارائه شده تعداد بسته‌ها را تا دور دوم پیشبینی می‌کند. در نهایت یک توزیع احتمال کدگشایی بسته‌ها در دور kام را بدست می‌آوریم و آنرا از طریق شبیه سازی اعتبار سنجی می‌کنیم. نتایج نشان می‌دهند که برای احتمال خطای کوچک و (ω) کم، مقدار سربار حتی می‌تواند به عددی نزدیک ۷۰٪ برسد. برای کاهش سربار فرستنده باید مقدار (ω) به صورت درست انتخاب شود و انتخاب درست (ω) به شدت به احتمال خطای کانال وابسته است. همچنین برای $\omega = ۵$ و اندازه‌ی نسل برابر با $g = ۲۵۶$ و احتمال پایین خطا در کانال ارتباطی، گره مقصد به طور میانگین ۷۰٪ بسته‌ی اضافی دریافت می‌کند. با افزایش (ω)، سربار کمتر می‌شود و برای $\omega \geq ۶۰$ این مقدار قابل چشم پوشی است. همچنین نشان دادیم که روش ارائه شده به دلیل کاهش ۳۷/۴۴ درصدی میانگین تأخیر کدگشایی، بهبود مناسبی در کارایی سیستم ایجاد می‌کند.

واژه‌های کلیدی: شبکه، کدگذاری کانال، RLNC، پرپچوال، کدگشایی.

Evaluation of Decoding Delay in the Perpetual Coding

Sanaz Mohammadi, MSc Student [1], Peyman Pahlevani, Assistant Professor[2]

1- Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran, Email: mohammadi.s@iasbs.ac.ir
2- Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan, Iran, Email: pahlevani@iasbs.ac.ir

Abstract: Perpetual coding is a sparse coding method in which coefficients are used in a structured way for coding operations. It has been shown that this method reduces the computational complexity of the random linear network coding method. The purpose of this paper is to articulate a mathematical model to illustrate the performance of perpetual coding, and to show that in a perpetual coding with the presence of an erasure channel the number of linear dependent packet transmissions is highly dependent on a parameter called the width (ω) which represents the number of consecutive non-zero coding coefficients present in each coded packet after a pivot element. Then, a mathematical analytical model for the number of transmitted packets is provided, which predicts the number of packets until the second round. Finally, we obtain the probability of the packet's decoding in round k and validate it by simulation. The results show that for low error probability and small (ω) on the link, overhead value can even reach a number close to 70%. To reduce the transmitter overhead, the (ω) value must be selected correctly, and the correct choice of (ω) is highly dependent on the probability of channel error. Also, for $\omega = 5$, generation size $g = 256$ and low erasure probability on the link, a destination can receive up to 70% overhead on average. Moreover, by increasing the width, the overhead decreases and for $\omega \geq 60$ it becomes negligible. We show that our mechanism reduces the delay by 37.44 percent.

Keywords: Network, Channel Coding, RLNC, Perpetual, Decoding.

نام نویسنده مسئول: پیمان پهلوانی
نشانی نویسنده مسئول: ایران - زنجان - گاوازنگ- دانشگاه تحصیلات تکمیلی علوم پایه - دانشکده علوم رایانه و فناوری اطلاعات و کامپیوتر

۱- مقدمه

در پیاده‌سازی عملی شبکه‌های بی‌سیم داده‌های منتقل شده تحت تأثیر ویژگی‌های کانال مانند اختلال، تداخل و کاهش توان سیگنال قرار می‌گیرند. در نتیجه، داده‌ی دریافت شده توسط گیرنده ممکن است با داده‌ی ارسال شده متفاوت باشد و داده دچار خطا شود. کدگذاری کانال مجموعه‌ای از روش‌ها در سیستم‌های ارتباط دیجیتال برای کنترل خطا است. کدگذاری شبکه[1] روشی نوینی است که مزایای زیادی در کاربردهای مختلفی مانند ذخیره‌سازی و ارتباطات کامپیوتری دارد. در روش NC هر یک از گره‌ها می‌توانند بسته‌ها را کدگذاری کنند، برخلاف روش‌های دیگر (مانند LT [۱] و رپتور[۲]) که در آن‌ها تنها گره‌های مبدا و مقصد عملیات کدگذاری و کدگشایی را انجام می‌دهند. درکارهای اولیه که توسط آلسوید و همکارانش انجام شد، نشان داده شد که در ارتباط یک به چند با استفاده از کدگذاری شبکه می‌توان از ظرفیت کانال به صورت بیشینه استفاده کرد[۳].

لی و همکارانش [۴] نشان دادند که در ارتباط یک به چند کدگذاری شبکه‌ی خطی برای دستیابی به بیشینه‌ی ظرفیت کانال کافی است. کدگذاری خطی تصادفی شبکه[2] برای نخستین بار توسط مدارد و همکارانش [۵] در سال ۲۰۰۶ ارائه شد. در این روش برای تولید یک بسته‌ی کد شده باید ضرایب به صورت تصادفی از یک فضای متناهی[3] انتخاب شوند و سپس به صورت خطی با یکدیگر ترکیب شوند. انتخاب تصادفی ضرایب، این امکان را فراهم می‌کند که تعداد بسیار زیادی بسته کد شده تولید شود. بسته‌ی اولیه را می‌توان با هر ترکیب کافی از داده‌های کد شده، کدگشایی کرد. یکی دیگر از ویژگی‌های RLNC قابلیت این روش در ارسال داده با هر نرخ ارسال است. با توجه به این ویژگی‌ها، پیاده سازی RLNC ساده است و به عنوان روشی مناسب برای شبکه‌های بی‌سیم به کار می‌رود. یکی دیگر از مواردی که در ارتباط بی‌سیم اهمیت می‌یابد حفظ انرژی گره‌ها است زیرا بیشترگره‌ها در ارتباطات بی‌سیم با استفاده از باطری تغذیه می‌شوند. در نتیجه هرچه تعداد بسته‌های ارسالی کاهش یابد، انرژی کمتری در گره‌ها استفاده می شود. RLNC در ارسال بسته‌ها از تمام بسته‌ها برای کدگذاری استفاده می‌کند. بنابراین این روش می‌تواند با خطاهای احتمالی رخ داده در کانال مقابله کند و نیاز گره‌ها به ارسال مجدد بسته‌ها را کاهش دهد.

روش RLNC از متداول‌ترین روش‌ها در زمینه‌ی کدگذاری شبکه است. با وجود مزیت‌هایی که استفاده از این روش دارد پیچیدگی محاسبات کدگذاری و کدگشایی در این روش بالا است و اجازه‌ی استفاده از این روش را درمحیط‌هایی که محاسبات بالا برای آن‌ها محدودیت به حساب می‌آید، نمی‌دهد. از ویژگی‌های مثبت این روش توانایی مقابله با خطا، تأخیر کم، توپولوژی شبکه‌ی منعطف وظرفیت قابل تغییر است. اما برای پیاده‌سازی عمل RLNC باید بار محاسبات تحمیل‌شده به سیستم‌کاهش یابد. روش‌های متنوع برای کاهش بار محاسبات ارائه شده است.

یکی از این روش‌ها، روش کدگذاری پرپچوال می‌باشد که نخستین بار توسط هاید و همکارانش باهدف کاهش پیچیدگی عملیات کدگشایی معرفی شد[۶]. در این روش با استفاده‌ی ترکیبی از ویژگی های مثبت RLNC و ادغام آن با روش‌های دیگر مانند روش تنک سعی شده است به پیچیدگی محاسبات روش RLNC غلبه شود. روش پرپچوال سرباری مشابه با روش RLNC دارد ولی توانسته است تا با تکنیک‌های مورد استفاده، توان عملیات را در فرآیندهای کدگذاری و کدگشایی بهبود دهد. روش تنک به عنوان یک رویکرد جایگزین، در[۹] و[۱۰] و[۱۱] معرفی شده است.

در این روش ضرایب کدگذاری به طور پراکنده انتخاب می شوند. بیشترین ضرایب در ماتریس کدگذاری صفر است. حال اگر بسته‌ی کد شده دقیقاً شامل ω ضریب کدگذاری غیر صفر باشد $\omega - sparse$ نامیده می‌شود. فیضی و همکارانش، در [۱۳] یک الگوریتم کدگشایی با پیچیدگی زمان خطی برای روش SNC معرفی کرده است. همچنین نشان داده شده است که SNC می تواند به طور قابل توجهی میزان سربار را به وسیله‌ی پیوست ضرایب به بسته‌های کدشده کاهش دهد[۱۰]. زارعی و همکارانش در [۱۲] نشان دادند که SNC از نظر تأخیر از RLNC پیشی می‌گیرد. علاوه بر این، SNC هیچ راه حلی برای موضوع به ترتیب رسیدن بسته‌ها که معیار مهمی برای برنامه‌های جریانی است، ارائه نمی‌دهد. تحقیقات اخیر طرح پنجره کشویی RLNC را برای دستیابی به تأخیر کم در تحویل به ترتیب بسته‌ها ایجاد با این حال، این طرح نیز به بازخوردگیری دارد [۱۴، ۱۵].

روش دیگری که به منظور رفع مشکلات ناشی از کدگذاری RLNC مطرح شده است روش fulcrum است[۱۶]. هدف اصلی این نوع کدگذاری کم کردن سربار ارسال ناشی از فرستادن بسته‌های اضافی همراه با بسته‌های کدشده و سربار ارسال بسته‌های وابسته خطی است. مزیت دیگر این روش ساده کردن عملیات در گره‌های میانی با هدف کاهش پیچیدگی کدگذاری می باشد. یکی دیگر از اهداف روش fulcrum سازگار کردن کدگذاری شبکه با انواع ساختارهای موجود است. برای دستیابی به این هدف کدگذاری و کدگشایی در روش fulcrum در دو مرحله داخلی و خارجی انجام می شود. در قسمت داخلی می‌توان از روش های مختلف کدگذاری استفاده کرد ولی وظیفه بخش خارجی کد کردن داده‌ها به روشی است که سازگار با انواع ساختارهای شبکه با شد. نتیجه این امر بالا رفتن کارایی سیستم به دلیل آزادی انتخاب کدگذاری در قسمت داخلی و سازگاری آن با بیشتر ساختارهای موجود می باشد. کدگذاری داخلی و خارجی تنها در فرستنده و گیرنده اعمال می‌شود و در گره‌های میانی عملیات کدگذاری مجدد به ساده‌ترین شکل ممکن اجرا می شود. مشکل اصلی این روش پیچیدگی عملیات کدگذاری و کدگشایی می‌باشد.

کدگذاری در روش پرپچوال به صورت تنک و غیر یکنواخت است که امکان کدگشایی سریع را فراهم می‌کند و از طرفی با توجه به ویژگی طول ثابت بسته‌ها امکان کدگذاری مجدد در گره‌های میانی را

می‌دهد. در روش پرپچوال یک عنصر محور و یک طول مشخص در نظر گرفته می شود و عنا صر بعد از محور به صورت پ شت سر هم تا طول مشخص غیر صفر در نظر گرفته می‌شوند و مقدار این عناصر به صورت تصـادفی از فضـای متناهی مورد نظر گرفته می‌شـود. در هر گام تنها بسـته‌های گره مبدأ که در این پنجره‌ی مشـخص جای می‌گیرند برای کدگذاری انتخاب می شوند و بقیه عنا صر بردار کدگذاری صفر ا ست. با توجه به این‌که در بردار کدگذاری طول ثابت اسـت و تنها عنصـر محور اهمیت دارد، برای ار سال بردارکدگذاری می‌توان از روش‌هایی ا ستفاده کرد و سربار ا ضافی را تا حد زیادی کاهش داد. پس از دریافت بسته‌ها در کدگشـا، کدگشـایی با اسـتفاده از بردار کدگذاری انجام می‌گیرد. اسـاس کار کدگشـایی در اینجا نیز اسـتفاده از ویژگی‌های تنک بودن ماتریس نهایی اسـت. به این صـورت که ماتریس نهایی با اسـتفاده از روش‌های حذف گاوس سـاده شـده و حل دسـتگاه معادلات متناظر با بسته‌های کد شده ساده می‌شود.

پس از ارا ئه‌ی روش پرپچوال، در[۷] مدل تحلیلی برای کد های پرپچوال در کانال‌های دارای خطا ارائه داده‌ایم. در مدل ارائه شده مانند روش اصلی هر بردار کدگذاری توسط یک عنصر محور و ω عنصر پس از آن تشکیل می‌شود. مقدار عنصر محور همیشه یک است و مقدار ω عنصر بعدی به صورت تصادفی ازمیدان متناهی Fq انتخاب می شود. عنصـر محور را می‌توان به دو روش تصـادفی یا پشـت سـر هم انتخاب کرد. در این مدل‌سازی روش پشت سر هم در نظر گرفته شده است.

این مقاله در ادامه و جهت تکمیل مقاله‌ای که در کارگاه بین المللی ارتباطات دسترسی چندگانه [۷] ارائه شده آمده است. در مقاله‌ی قبلی فقط مدل تحلیلی برای ار سال داده تا دور دوم ارائه داده شده ا ست. در این مقاله تعداد دورهایی که باید داده‌ها ارسـال شـود تا گیرنده بتواند تمام داده‌ها را کدگشـایی کند را به صـورت تحلیلی بدسـت می‌آوریم. برای این کار ابتدا نشـان می‌دهیم که عملکرد روش پرپچوال به پارامتر عر ضی وابسته ا ست. سپس یک مدل تحلیلی ارائه می‌دهیم که تعداد ارسـال‌ها را تا دور دوم محاسـبه می‌کند. این مدل تحلیلی زمانی که مقدار خطا در شـبکه کم اسـت و ارسـال ها به دورهای بعدی کشـیده نمی شود، منا سب ا ست. در انتها مدل قبلی را گ سترش داده و به یک مدل تحلیلی برای توزیع احتمال کدگشایی بسته‌ها در دورهای مختلف پرداخته می‌شود و تأثیر خطای شبکه در این خصوص بررسی می‌شود. همچنین با اسـتفاده از شـبیه سـازی مدل‌های تحلیلی خود را اعتبار سنجی می‌کنیم. این مدل نشان می‌دهد کارایی روش پرپچوال به شدت به عرض بردار کدگذاری بسـتگی دارد. علاوه بر این، بررسـی‌ها نشـان می‌دهد که اگر احتمال خطا در کانال ۰/۲ = ε باشـد، با فرض اندازه‌ی نسـل برابر با ۲۵۶ = g و ۵ = ω، ۷۰٪ بسـته‌های دریافتی وابسـته‌ی خطی هسـتند. با افزایش اندازه‌ی ω این میزان به مقدار زیادی کاهش می‌یابد تا جایی که برای ۶۰ ≤ ω قابل اغماض می‌شود.

ساختار این مقاله به صورت زیر است: در بخش ۲ به معرفی کدهای پرپچوال و اهداف این روش می‌پردازیم. در بخش ۳ مدلی برای تحلیل روش پرپچوال و توزیع احتمال کدگشـایی بسـته‌ها بسـته‌ها ارائه می‌دهیم. نتایج ریاضی در بخش ۴ آورده شده است. در نهایت در بخش ۵ نتیجه‌گیری ارائه می‌شود.

۲- تقسیمات کدگذاری پرپچوال

در این مقاله فرض کرده‌ایم که داده از یک گره مبدا به یک گره مقصد ارسال می‌شود. داده‌ی اصلی را به نسل‌هایی[۴] با اندازه‌ی g تقسیم کرده‌ایم. هر نسل از تعدادی بسته تشکیل شده است که ترکیب خطی این بسـته‌ها با هم بسـته‌های کد شـده را ایجاد می‌کنند. گره مبدا بسته‌های کد شده را ایجاد کرده و تا زمانی که گره مقصد کل نسل را کدگ شایی کند به ار سال آنها ادامه می‌دهد. وقتی گره مق صد به تعداد کافی بسته‌ی مستقل[۵] دریافت کند می‌تواند یک نسل را کدگشایی کند. سپس گره مقصد یک تأیید[۶] به گره مبدا می‌فر ستد تا گره مبدا ار سال بسته را متوقف کند. ممکن است بسته‌های ارسال شده توسط گره مبدا به دلیل امکان خطا در کانال از بین بروند. احتمال از بین رفتن بسته‌ها در کانال بین گره مبدا و مقصد برابر ω است

ابتدا فرض می‌کنیم کانال بین گره مق صد و مبدا بدون خطا ا ست یعنی تأیید گره مقصد حتماً بلافاصله بعد از تولید به گره مبدا می‌رسد.

ضـرایب کدگذاری برای سـاخت بسـته‌ی کد شـده از یک فضـای متناهی Fq با اندازه‌ی q انتخاب می‌شـوند. به مجموعه‌ی ضـرایب کدگذاری که در یک بسته‌ی کد شده ا ستفاده شده‌اند بردار کدگذاری گفته می‌شـود. یک بردار کدگذاری در روش پرپچوال توسـط یک المان محور که ضریبی با مقدار ۱ است و ω ضریب که به صورت تصادفی از Fq انتخاب شده‌اند ساخته می‌شود. بقیه‌ی ضرایب صفر در نظر گرفته می‌شـود. ω را عرض کدگذاری پرپچوال می‌نامیم. این پارامتر در گره مبدا در نظر گرفته می‌شود و در طول فرایند ارسال ثابت می‌ماند.

برای ار سال بسته‌های کد شده در پرپچوال دو روش مختلف وجود دارد: (۱) روش تصـادفی: المان‌های محور به صـورت تصـادفی انتخاب می‌شـود و سـپس با توجه به المان محور بردار کدگذاری تولید شـده و ارسـال می‌شـود. گره مبدا بعد از دریافت تأیید از گره مقصـد ارسـال بسته‌ها را متوقف می‌کند. (۲) روش دنباله‌ای: برای یک نسل با اندازه‌ی g، یک ماتریس با g بردار کدگذاری در گره مبدا سـاخته می‌شـود. المان‌های محور این بردار به صـورت دنباله‌ای از اولین سـطر تا آخرین سـطر انتخاب می‌شـود. شـکل ۱ ماتریس کدگذاری را برای ۸ = g و ۳ = ω و $\alpha_{i,j} \in Fq$ نشـان می‌دهد. به دلیل وجود خطا در کانال فر ستنده ممکن ا ست چندین بار ار سال را به صورت نوبت گرد شی[۷] انجام دهد تا زمانی که گیرنده بتواند کل نسل را شناسایی کند. در هر دور مقادیر ماتریس کدگذاری با مقادیر جدید جایگزین می‌شـوند. فرسـتنده با دریافت تأیید از جانب گیرنده عملیات ارسـال را متوقف می‌کند. در این مقاله ما تنها روش دنباله‌ای را در انتخاب المان محوری در نظر گرفته‌ایم.

برای درک بهتر مفهوم بسته‌های وابسته‌ی خطی، پارامتر سربار را به صورت زیر تعریف می‌کنیم:

$$O = \frac{R_d}{R_i} \qquad (۱)$$

در این رابطه R_d تعداد بسته‌های وابسته‌ی خطی و R_i تعداد بسته‌های مستقل خطی درگیرنده هستند.

1	α	α	α				
	1	α	α	α			
		1	α	α	α		
			1	α	α	α	
				1	α	α	α
α					1	α	α
α	α					1	α
α	α	α					1

شکل ۱: بردارهای کدگذاری پرپچوال برای $g=8$ و $\omega=3$. عناصر α به طور تصادفی از Fq انتخاب شده‌اند.

۳- مدل تحلیلی

در این قسمت ابتدا مدلی ارائه می‌دهیم که ابعاد زیر مجموعه‌ی تعریف شده توسط بردارهای کدگذاری دریافت شده را معرفی می‌کند. با استفاده از این مدل ما تاثیر ω بر روی سربار کدهای پرپچوال را تحلیل می‌کنیم و برای تعداد بسته‌های ارسال شده در کدهای پرپچوال در کانال دارای خطا تحلیل ارائه می‌دهیم. در نهایت توزیع احتمال کدگشایی بسته‌ها را در دور kام برای کانال دارای خطا ارائه می‌دهیم و تاثیر پارامتر ω را تحلیل می‌نماییم.

۳-۱- تأثیر پارامتر عرضی در کدگشایی

فرض کنید از کدهای پرپچوال برای ارسال یک نسل با g بسته استفاده کرده‌ایم. از آنجایی‌که اندازه‌ی نسل برابر با g است اندازه‌ی ω کمتر از g است.

$$0 \le \omega \le g-1 \qquad (۲)$$

کلاس C_i را به صورت مجموعه‌ای از بسته‌های کد شده با المان محور i تعریف می‌کنیم. از آنجایی‌که g محور مختلف داریم هر بسته‌ی کد شده متعلق به یکی از این g کلاس است. در کلاس C_i ما بسته‌های کد شده‌ای داریم که در آنها ω ضریب تصادفی بعد از i امین محور می‌آیند.

مدل سازی پرپچوال را می‌توان با در نظر گرفتن نیاز گیرنده به جمع‌آوری بسته های مستقل خطی از همه کلاس ها انجام داد. $\dim(c_i)$ را به عنوان اندازه‌ی زیر مجموعه‌ی کلاس C_i تعریف می‌کنیم. وقتی گره مقصد $\omega+1$ بسته‌ی کد شده‌ی مستقل خطی برای یک کلاس دریافت می‌کند می‌تواند آن کلاس را کدگشایی کند. بنابراین بُعد یک کلاس که با تعداد بردارهای کدگذاری شده در آن کلاس تعریف می‌شود نمی‌تواند بیشتر از $\omega+1$ باشد

$$0 \le dim \le \omega+1 \quad \forall i \in \{0.\ldots.g-1\} \qquad (۳)$$

$$0 \le dim\left(\bigcup_{i=0}^{g-1} C_i\right) \le g \qquad (۴)$$

برای این که بتوان یک نسل را کدگشایی کرد، گیرنده باید از همه‌ی کلاس‌ها بسته‌های مستقل خطی جمع آوری کند. برای دو کلاس پشت سر هم C_i و $C_{i+1 \mod g}$ گیرنده باید مقصد $\omega+2$ بسته‌ی کد شده دریافت کند تا بتواند هر دو کلاس را کدگشایی کند.

$$0 \le \dim\left(C_{i \mod g} \cup C_{i+1 \mod g}\right) \le \min(\omega+2.g) \qquad (۵)$$

می‌توان رابطه‌ی قبلی را به کلاس‌های $C_i, C_{i+1 \mod g}, \ldots C_{i+k \mod g}$ تعمیم داد.

$$0 \le dim\left(\bigcup_{j=i \mod g}^{i+k \mod g} C_j\right) \le \min(\omega+k+1.g) \qquad (۶)$$

$$\forall_i \in \{0.\ldots.g-1\}.\ \forall k \in \{1.\ldots.g-2\}$$

زیر مجموعه‌ای از k بسته‌ی کد شده که توسط k بردار کدگذاری آخر ماتریس کدگذاری تولید شدند را در نظر بگیرید. در این‌جا نشان می‌دهیم که وقتی فرستنده k بسته‌ای که به k کلاس موجود در مستطیل تعلق دارند را ارسال می‌کند و $\omega+1$ بسته گم می‌شوند، فرستنده باید حداقل یک بسته متعلق به k کلاس موجود در مستطیل را ارسال کند تا گیرنده بتواند کدگشایی را تمام کند..

در دور اول ارسال، فرستنده بسته‌های کد شده با المان محوری ۰ تا g-1 را تولید و با استفاده از ماتریس کدگذاری اول آنها را ارسال می‌کند. به دلیل خطا در کانال فرض می کنیم $\omega+1$ بسته داخل مستطیل گم می‌شوند این مستطیل از المان محوری k-gام آغاز می شود و تا المان محوری g-1 ام ادامه می‌یابد و $k \ge \omega+1$. علاوه بر این فرض می‌کنیم اولین بسته‌ی گم شده دارای المان محوری k-g است. فرض می‌کنیم بسته‌های دارای المان محوری ۱ تا g-k-1 به صورت موفق در فرستنده دریافت شده‌اند.

$$dim\left(\bigcup_{i=g-k}^{g-1} C_i\right) = k-\omega-1 \qquad (۷)$$

بنابراین از طرف دیگر از k بسته‌ی داخل مستطیل (g-k-1) بسته به صورت موفق دریافت شده‌اند.

$$dim\left(\bigcup_{i=0}^{g-k-1} C_i\right) = \omega+g-k \qquad (۸)$$

به دلیل از بین رفتن بسته‌ها در دور اول ارسال، دور دوم با ساختن ماتریس ضرایب جدید انجام می‌شود. از آنجا که فرض کردیم g-k-1 $> \omega$ ، بنابراین g-k-1 بسته‌ی کد شده‌ی اول دور دوم با فرض درست رسیدن به مقصد ω عدد DOF به معادله‌ی ۷ اضافه می کنند.

$$dim\left(\bigcup_{i=0}^{g-k-1} C_i\right) = \omega + g - k \qquad (۹)$$

بنابراین با توجه به رابطه‌ی۶، $C_0,,,,C_{g-k-1}$ کد گشایی می‌شوند. با ترکیب رابطه ی ۸ و ۹ تعداد کل DOF های دریافتی از رابطه‌ی زیر بدست می‌آیند.

$$dim\left(\bigcup_{i=0}^{g-1} C_i\right) = g - 1 \qquad (۱۰)$$

با توجه به رابطه‌ی ۱۰ گیرنده فقط یک DOF نیاز دارد تا بتواند کل نسل را کدگشایی کند. با توجه به این که کل بسته‌های موجود در بیرون مستطیل کدگشایی شده‌اند، فرستنده باید حداقل یک بسته از داخل مستطیل دریافت کند.

موقعیت $\omega+1$ بسته‌ی گم شده داخل مستطیل تأثیر بسزایی در تعداد ارسال‌ها دارد. اگر $k=\omega+1$ با شد، تمام بسته‌های کد شده‌ی موجود در مستطیل از بین رفته‌اند. این بسته‌ها متعلق به کلاس‌های $C_{g-\omega-1}$ تا C_{g-1} هستند. بنابراین در دور دوم ارسال فرستنده باید حداقل یک بسته از داخل مستطیل به علاوه‌ی $g-k-1$ ارسال انجام دهد. بنابراین در دور دوم ارسال فرستنده باید ارسال‌هایی از کلاس‌های $C_0,,,,C_{g-\omega-1}$ داشته باشد که در کل $g-\omega$ بسته ارسال می‌شود. برای حالتی که $k=\omega+2$ ، مستطیل به کلاس های $C_{g-\omega-2}$ تا C_{g-1} اشاره می‌کند. برای ارسال حداقل یک بسته از بسته‌های متعلق به کلاس‌های داخل مستطیل، گره مبدا باید از کلاس‌های C_0 تا $C_{g-\omega-2}$ بسته ارسال کند که در نهایت $g-\omega-1$ بسته‌ی کد شده ارسال می‌شود. اگر بخواهیم این نتایج را تعمیم بدهیم، اگر مستطیل به کلاس های C_{g-k} تا C_{g-1} اشاره کند، فرستنده باید $g-k+1$ بسته‌ی کد شده قبل از ارسال یک بسته‌ی متعلق به مستطیل ارسال کند.

۳-۲- تعداد ارسال‌ها تا دور دوم

۱. فرستنده بیش تر از ۲ دور ارسال انجام نمی‌دهد. این فرض برای سیستم‌های عملی که در آنها احتمال خطا کمتر از ۰/۵ درصد است می‌تواند فرض درستی باشد.
۲. اندازه‌ی نسل g بالا است و احتمال خطای ε کوچک است.
۳. پارامتر ω به اندازه‌ی کافی بزرگ است به گونه‌ای که با در نظر گرفتن ۲ دور احتمال کدگشایی بالا است.

در این بخش رویداد IN را به این صورت تعریف می‌کنیم:

$$P(IN)\binom{k-1}{\square}\epsilon^{\square+1}(1-\epsilon)^{k-(\square+1)} \qquad (۱۱)$$

بسته‌ی اول متعلق به مستطیل با اندازه‌ی k از بین رفته است و ω بسته‌ی دیگر داخل مستطیل نیز از بین رفته اند بنابراین $k-(\omega+1)$ بسته به صورت درست دریافت شده اند. احتمال این رویداد با رابطه‌ی زیر محاسبه می‌شود:

اگر i بسته خارج از مستطیل از بین رفته باشد، فرستنده باید حداقل $\omega+i$ بسته ارسال کند تا گیرنده بتواند کدگشایی را انجام دهد.

رویداد OUT را به صورت زیر تعریف می‌کنیم:

از بین رفتن i بسته خارج از مستطیل و احتمال این رویداد از رابطه‌ی زیر محاسبه می شود:

$$P(OUT)=\binom{g-1}{i}\varepsilon^{i}(1-\varepsilon)^{(g-k-i)} \qquad (۱۲)$$

همان گونه که قبلاً اشاره شد، فرستنده باید حداقل $g-k+1$ بسته‌ی کد شده را قبل از ارسال یک بسته‌ی متعلق به مستطیل ارسال کند. تعداد کل ارسال‌ها وقتی i بسته بیرون از مستطیل و $\omega+1$ بسته داخل مستطیل از دست بروند برابر با $\max(g-k+\omega+1)$ است. برای جبران این اتفاق، فرستنده باید $\frac{\max(g-k+1+\omega+i)}{(1-\varepsilon)}$ بسته ارسال کند تا بسته‌های گم شده جبران شود. از آنجا که ما فقط ۲ دور را در نظر گرفته‌ایم، در دور دوم فرستنده حداکثر g بسته می‌تواند ارسال کند. بنابراین تعداد بسته‌های ارسال شده $\min(\frac{(g-k+1+\omega+i)}{(1-\varepsilon)},g)$ است.

با در نظر گرفتن روابط ۱۱ و ۱۲ تعداد ارسال‌های پیش بینی شده در دور دوم را می‌توان توسط رابطه زیر تقریب زد.

$$T_{2=}\sum_{k=\omega+1}^{g}\sum_{i=0}^{g-k}\min\left(\frac{\max(g-k+1.\omega+i)}{(1-\varepsilon)},g\right)\binom{k-1}{w}\varepsilon^{\omega+1}(1-\varepsilon)^{k-(\omega+1)}\binom{g-k}{i}\varepsilon^{i}(1-\varepsilon)^{(g-k-i)} \qquad (۱۳)$$

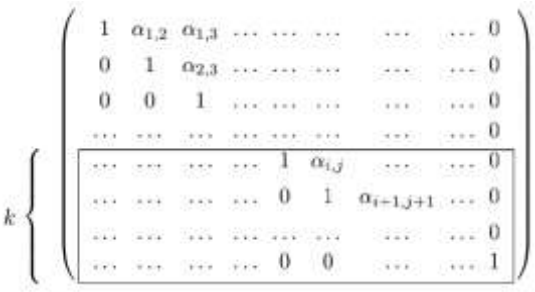
از آنجایی که تعداد بسته های ارسال شده در دور اول $T_1=g$ است تعداد کل ارسال‌ها در ۲ دور برابر است با:

$$T_{1.2}\approx T_2+g \qquad (۱۴)$$

با اضافه کردن تعداد ارسال‌ها در حالتی که تعداد بسته‌های گم شده کمتر از $\omega+1$ است به رابطه‌ی زیر می‌رسیم :

$$T_T \approx T_{1.2}+\sum_{i=0}^{\omega}min\left(\frac{i}{1-\varepsilon}.g\right)\binom{g}{i}\varepsilon^{i}(1-\varepsilon)^{g-i} \qquad (۱۵)$$

معادله‌ی ۱۵ تقریبی برای تعداد کل بسته‌های کد شده‌ی ارسال شده‌ی TT در دو دور ارسال است. این رابطه تابعی از اندازه‌ی نسل g و عرض ω در کدگذاری پرپچوال و احتمال خطای کانال ε با فرض‌های بالا است. پارامتر ω تاثیر فراوانی در تعداد ارسال‌های کدگذاری پرپچوال دارد. اگر ω کوچک باشد، از بین رفتن بسته‌ها در دور اول منجر به ارسال بسته‌های وابسته‌ی خطی در دور دوم می‌شود.



شکل ۲: k ردیف پایینی شکل، نشان دهنده‌ی k بردار کدگذاری زمانی که $\omega+1$ بسته از بین رفته است.

۳-۳- تعداد دورهای ارسال شده تا کدگشایی

در این بخش توزیع احتمال کدگشایی بسته‌ها در دور kام را شرح می‌دهیم:

فرض کنید از کدهای پرپچوال برای ارسال یک نسل با g بسته استفاده کرده‌ایم قصد داریم توزیع احتمال کدگشایی بسته‌ها در دور k ام را محاسبه نماییم. برای کدگشایی بسته‌ها در هر دور باید در سمت گیرنده حداقل به تعداد g یعنی به تعداد اندازه‌ی نسل بسته دریافت شده باشد. در صورتی که این اتفاق رخ ندهد آن دور برای کدگشایی بسته‌ها مطلوب واقع نمی‌شود و به دور بعدی می‌رود در نتیجه اگر کدگشایی بسته‌ها در دور kام صورت گرفته است این اتفاق در حالی رخ داده است که در تمامی $k-1$ دور قبلی گیرنده به تعداد g بسته دریافت نکرده است و به محض دریافت بسته به تعداد اندازه‌ی نسل در دور kام عمل کدگشایی انجام شده است.

اگر احتمال اینکه تا دور $k-1$ فقط به تعداد i بسته گیرنده دریافت کرده باشد را $p(i)$ بنامیم در این صورت داریم:

انتخاب i بسته از کل gk بسته‌ی ارسالی: $\binom{gk}{i}$.

احتمال رسیدن i بسته‌ی انتخابی: $(1-\varepsilon)^i$.

احتمال با خطا مواجه شدن و در نتیجه نرسیدن همه‌ی بسته‌های غیر از i بسته‌ی انتخابی: ε^{gk-i}.

احتمال خطای شبکه برابر است با ε.

و به رابطه‌ی ۱۶ می‌رسیم.

$$P(i) = \binom{gk}{i}(1-\varepsilon)^i \ \varepsilon^{gk-i} \qquad (۱۶)$$

از طرفی در دور kام به دلیل اینکه به تعداد نسل بسته به گیرنده رسیده است عملیات کدگشایی آغاز میگردد. بنابراین $p(j)$ را احتمال اینکه در دور kام بیشتر از i بسته دریافت شده باشد در نظر می‌گیریم و در نتیجه داریم:

$$P(j) = \sum_{j=g-i}^{g+1}\binom{g}{j}(1-\varepsilon)^j \ \varepsilon^{g-j} \qquad (۱۷)$$

انتخاب j بسته از g بسته‌ی ارسالی: $\binom{g}{j}$.

احتمال رسیدن j بسته‌ی انتخابی: $(1-\varepsilon)^j$.

احتمال با خطا مواجه شدن و در نتیجه نرسیدن همه‌ی بسته‌های غیر از j بسته‌ی انتخابی: ε^{g-i} .

احتمال خطای شبکه برابر است با ε.

در اینجا $j = g - i$ و داریم:

بنابراین توزیع احتمال کدگشایی بسته‌ها در دور k ام برابر است با حاصل ضرب دو رابطه‌ی فوق:

$$p(x = k) = p(i) * p(j)$$

با جایگذاری رابطه‌ی ۱۶ و ۱۷ داریم:

$$p = (x = k) = \sum_{j=g-i}^{g+1}\binom{g}{j}(1-\varepsilon)^j \varepsilon^{g-j} \sum_{i=0}^{g-1}\binom{gk}{i}(1-\varepsilon)^i \varepsilon^{gk-i} * \qquad (۱۹)$$

۴- نتایج ریاضی

در این بخش، در ابتدا سربار روش پرپچوال با روشRLNC مقایسه می‌شود. سپس نشان می‌دهیم ω پارامتری بنیادین در سربار کدهای پرپچوال است. در نهایت تحلیل ریاضی توزیع احتمال کدگشایی بسته‌ها ارائه می‌شود.

در بخش آخر تحلیل‌های ریاضی ارائه شده با استفاده از شبیه سازی‌ها مقایسه می شوند. نتایج شبیه سازی‌ها نشان می‌دهند تحلیل‌های ریاضی ارائه شده درست هستند.

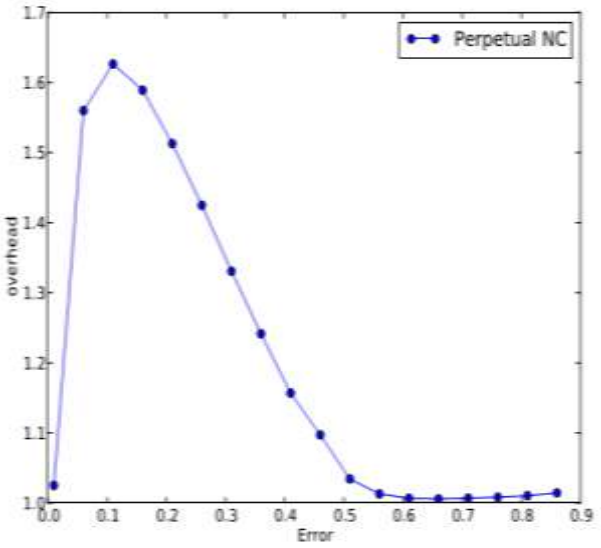
برای نمایش کدهای پرپچوال، یک شبیه‌سازی انجام داده و با استفاده از کتابخانه‌ی KODO [۸], روش RLNC را با روش پرپچوال مقایسه کردیم. کتابخانه KODO یک کتابخانه‌ی استاندارد برای شبیه سازی و پیاده سازی NC است. شکل ۳ سربار کدهای پرپچوال را برای ۲ دور با $g = ۲۵۶$, $\omega = ۵$ نشان می‌دهد. برای احتمال خطای کوچک، سربار کدهای پرپچوال تقریباً ۰/۷است. یعنی به ازای هر DOF دریافت شده سیستم باید ۱/۷ بسته به صورت میانگین ارسال کند.

شکل ۴ تعداد کل ارسال‌ها برای کانال‌هایی با خطاهای مختلف را در سه روش پرپچوال و RLNC و تنک برای $g = ۲۵۶$, $\omega = ۵$ نشان می‌دهد. در این مثال فرض شده است فرستنده می‌تواند در چندین دور ارسال انجام دهد. همان‌گونه که در شکل ۴ مشاهده می‌کنید، حتی با احتمال خطای کوچک، تعداد بسته‌های ارسالی با استفاده از روش پرپچوال نزدیک ۵۰۰ ارسال است یعنی ۵۰٪ بیشتر از ارسال‌های انجام شده توسط روش RLNC و تقریبا ۲۵ درصد بیشتر از روش تنک.

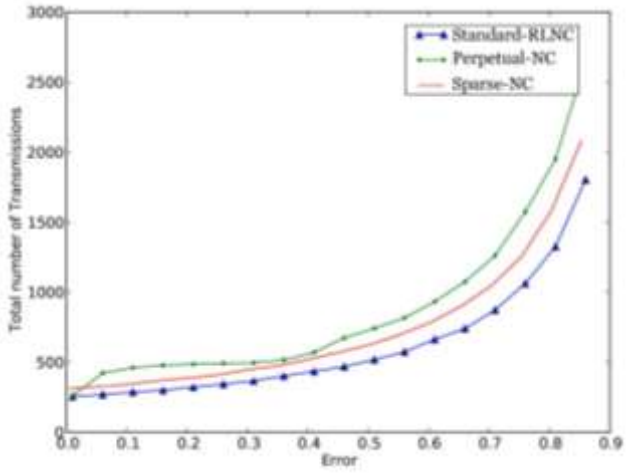
شکل ۵ نتایج بدست آمده از ارسال بسته‌ها توسط روش پرپچوال را برای احتمال خطای ۰/۲ و $g = ۲۵۶$ و مقادیر مختلف ω نشان می‌دهد. شکل نشان می‌دهد برای مقادیر $\omega \le ۴۰$، پارامتر ω تأثیر بسیار زیادی بر روی تعداد کل ارسال‌ها دارد. علاوه بر این می‌توان دید

برای $\omega \geq$ ۶۰ تاثیر ω بر روی تعداد ارسال‌ها محدود می‌شود. پارامتر ω تاثیر روی روش RLNC ندارد و در روش تنک فقط بر روی مقادیر کوچک ω تاثیر دارد.

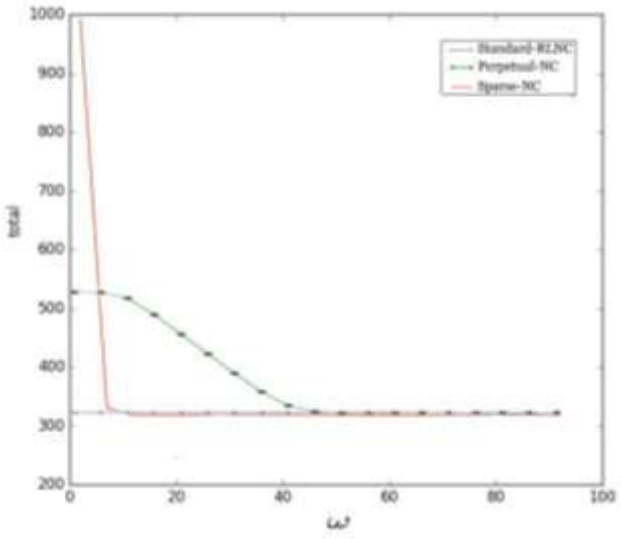
شکل ۶ شبیه‌سازی ۱۵ دور ارسال با استفاده از کدهای پرپچوال را با نتایج ریاضی بدست آمده از بخش قبل مقایسه می‌کند. نتایج شبیه‌سازی نشان می‌دهد که توزیع احتمال کدگشایی بسته‌ها در k دور ارسال بدست آمده توسط تحلیل ریاضی تقریب خوبی از نتایج شبیه‌سازی است. این نتایج برای احتمال خطاهای ۰/۲ و ۰/۴ و ۰/۸و و g = ۱۰۰ و ω = ۲۰ بدست آمده است.



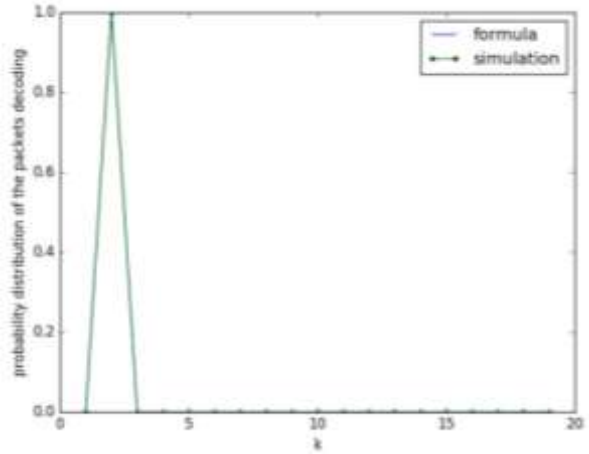
شکل ۳: سربار ارسال روش پرپچوال برای g = ۲۵۶ و ω = ۵



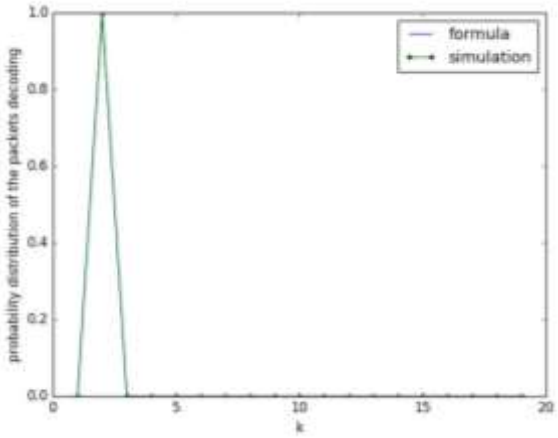
شکل ۴: مقایسه‌ی تعداد ارسال در سه روش پرپچوال و RLNC و تنک برای ω = ۵ و g = ۲۵۶



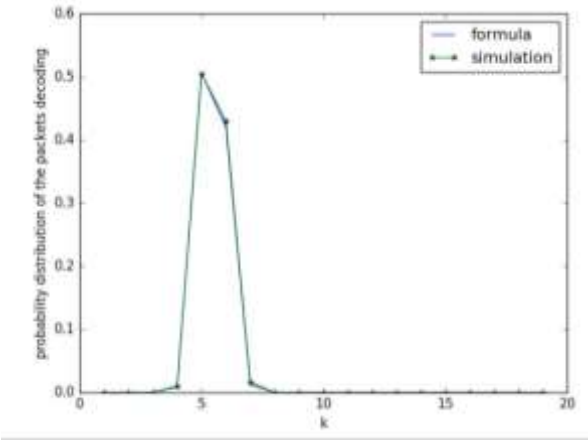
شکل ۵: مقایسه‌ی تعداد ارسال در سه روش پرپچوال، RLNC و تنک برای ω های مختلف



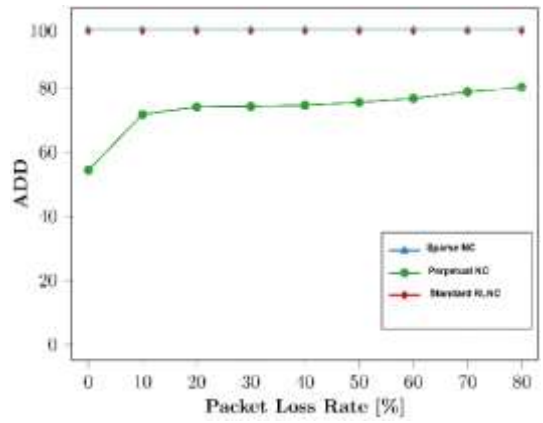
شکل ۶-الف: توزیع احتمال کدگشایی در روش پرپچوال برای ω=۲۰ و g = ۱۰۰ و خطای ۰/۲.



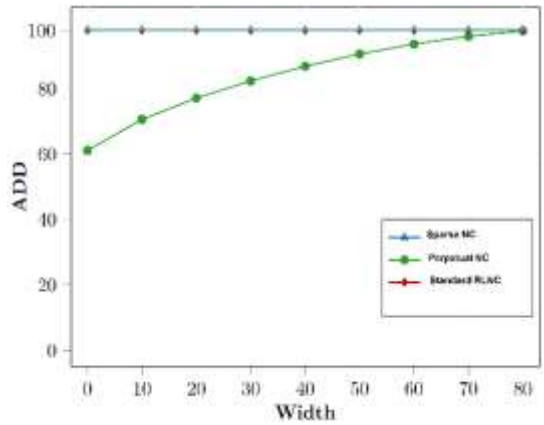
شکل ۶-ب: توزیع احتمال کدگشایی در روش پرپچوال برای ω=۲۰ و g = ۱۰۰ و خطای ۰/۴.



شکل ۶-ج: توزیع احتمال کدگشایی در روش پرپچوال برای ω = ۲۰ و ۱۰۰= g و خطای ۰/۸.



شکل ۷-الف: میانگین تاخیر کدگشایی در سه روش پرپچوال و RLNC و تنک برای ω = ۵ و g = ۲۵۶



شکل۷-ب: میانگین تاخیر کدگشایی در سه روش پرپچوال، RLNC و تنک برای ω های مختلف و g = ۲۵۶

هنگامی که گره مبدا یک بسته‌ی کد شده را که شامل بسته‌ی جدیدی است به سمت گره مقصد ارسال می‌کند بسته در اصطلاح بازدید شده است اما هنوز کدگشایی نشده است. میزان تاخیر بین بازدید و کدگشایی بسته تاخیر کدگشایی برای بسته‌های رسیده در سمت گیرنده، تاخیر کدگشایی نامیده می‌شود و مقدار میانگین تاخیر برای تمامی بسته‌ها میانگین تاخیر کدگشایی (ADD)[۸] نامیده می‌شود.

در شکل ۷-الف رابطه‌ی بین ADD و نرخ‌های مختلف تلفات بسته برای ω مختلف نشان داده شده است. در روش پرپچوال با افزایش میزان تلفات بسته، احتمال از دست دادن بسته با محورهای متوالی افزایش می‌یابد و این موضوع باعث افزایش ADD می‌شود و همانطور که مشاهده می‌شود رابطه‌ی بین ADD و نرخ تلفات بسته، رابطه‌ی مستقیم است و با افزایش نرخ تلفات، میانگین تأخیرکدگشایی نیز افزایش پیدا می‌کند. برای نرخ تلفات پایین با ا ستفاده از روش پرپچوال کاهش قابل ملاحظه‌ای در میانگین تاخیر کدگشایی بسته‌ها مشاهده به نسبت سایر روش‌ها مشاهده می‌شود. و از آنجاکه یکی از مهمترین معیارها در روش کدگذاری به‌خصوص برای سیستم‌های بلادرنگ[۹]، میزان تأخیر کدگشایی بسته‌ها می‌باشد. از این‌رو روش ارائه شده به دلیل کاهش ۳۷/۴۴ درصدی میانگین تأخیر کدگشایی، بهبود مناسبی در کارایی این نوع سیستم ها ایجاد می‌کند. در شکل ۷-ب رابطه‌ی ADD و ω برای نرخ تلفات مختلف بررسی شده است. این نمودار به صورت مقایسه‌ای عملکرد روش‌های روش پرپچوال و تنک و همچنین روش RLNC را به خوبی نمایش می‌دهد. مشاهده می‌شود که در روش بر پرپچوال، کاهش قابل توجهی در میانگین تاخیر کدگشایی به نسبت سایر روش‌ها داریم. با افزایش اندازه‌ی پنجره میانگین تاخیر کدگشایی بسته ها در روش پرپچوال به سمت مقدار بدست آمده از روش‌های تنک و RLNC همگرا می‌شود.

۵- نتیجه

در این مقاله مدل ریاضی‌ای ارائه داده‌ایم که تعداد بسته‌های ارسال شده در کدگذاری پرپچوال را در کانال دارای خطا نشان می‌دهد. با استفاده از این مدل تاثیر ω بر روی سربار ارسال کدهای پرپچوال مورد بررسی قرار گرفته است. در این مقاله نشان می‌دهیم که در کانال‌های دارای خطا، استفاده از روش پرپچوال با ω منجر به دریافت تعداد زیادی بسته‌ی وابسته‌ی خطی می‌شود. برای احتمال خطای کوچک و $\omega < 10$ مقدار سربار حتی می‌تواند به عددی نزدیک ۷۰٪ برسد. برای کاهش سربار فرستنده باید مقدار ω را به صورت درست انتخاب کند. انتخاب درست ω به شدت به احتمال خطای کانال وابسته است.

در این مقاله تقریبی از رابطه‌ی بین ω و تعداد ارسال‌ها را معرفی کردیم و سپس به ارائه‌ی مدل تحلیلی برای توزیع احتمال کدگشایی بسته‌ها در دورهای مختلف پرداختیم و تأثیر خطای شبکه را در این خصوص بررسی کردیم.

همچنین به دلیل اینکه تاخیر کدگشایی بسته‌ها خصوصا در سیستم‌های بلادرنگ اهمیت بالایی دارد در این روش به محاسبه‌ی تاخیر کدگشایی بسته‌ها بر ا ساس سه روش پرپچوال، RLNC و تنک پرداختیم و مشاهده شد با روش پرپچوال به کاهش ۳۷/۴۴ درصدی در میانگین تاخیر کدگشایی دست می‌یابیم.

شبیه سازی‌های عددی با استفاده از کتابخانه‌ی KODO ارائه شده است. نتایج شبیه‌سازی تأثیر مقادیر مختلف ω بر روی تعداد بسته‌های ارسالی و سربار را برجسته می‌کند. این شبیه‌سازی‌ها نشان می‌دهد که تحلیل ریاضی ارائه شده توسط ما تقریب خوبی از تعداد بسته‌های ارسال شده توسط کدگذاری پرپچوال است.

لازم به ذکر است تنها انتخاب المان محوری به صورت دنباله‌ای (سیستماتیک) را در نظر گرفتیم که به معنی انتخاب المان‌های محوری به صورت پشت سر هم است.

مراجع

[1] M.Luby, “Lt codes”, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002, pp. 271 – 280.

[2] A. Shokrollahi, “Raptor codes”, IEEE Transactions on Information Theory, Vol.52, NO.6, June 2006, pp. 2551–2567.

[3] R.Ahlswede, N.Cai, S.Y.Li, R.W Yeung, “Network information flow”, IEEE Transactions on Information Theory, Vol. 46, NO. 4, Jul 2000, pp. 1204 – 1216.

[4] S.Y.R.Li, R.W.Yeung, N.Cai, “Linear network coding”, IEEE Transactions on Information Theory Vol.52, NO.2, Feb 2003, pp. 371–381.

[5] S.Katti, H.Rahul, W.Hu, D.Katabi, M.Médard, J.Crowcroft. “XORs in the air: Practical wireless network coding”. In ACM SIGCOMM computer communication review, volume 36, ACM, 2006, pages 243–254.

[6] H. Janus, M.V.Pedersen, F.Fitzek, M.Medard,, “A perpetual code for network coding”, 2014 IEEE 79th Vehicular Technology Conference (VTC Spring), May 2014, pp. 1– 6.

[7] P.Pahlevani, S.Crisóstomo, D.E.Lucani. “An analytical model for perpetual network codes in packet erasure channels”. In International Workshop on Multiple Access Communications, Springer, 2016, pages 126–135.

[8] M.Pedersen, J.Heide, F.H.Fitzek, “KODO: An open and research oriented network coding library”, Springer, 2011, pp. 145–152.

[9] H.Sehat, P.Pahlevani, “On the probability of partial decoding in sparse network coding”, arXiv preprint arXiv:1907.12051, 2019

[10] S.Danilo, W.Zeng, F.Kschischang, “Sparse network coding with overlapping classes”, workshop on network coding,Theory, and Applications,2009, pp. 74-79

[11] H. Janus, M.V.Pedersen, F.Fitzek, M.Medard, “On code parameters and coding vector representation for practical rlnc”. In: 2011 IEEE nternational Conference on Communications (ICC), 2011, pp 1-5

[12] A. Zarei, P. Pahlevani and M. Davoodi, “On the Partial Decoding Delay of Sparse Network Coding”, IEEE Communications Letters, vol. 22, no. 8, 2018, pp. 1668-1671.

[13] S.Feizi, DE. Lucani, M.Medard “Tunable sparse network coding” ,22th International Zurich Seminar on Communications (IZS), Eidgen Eidgenossische Technische Hochschule Zurich, 2012

[14] M. Karzand, D. Leith, J.Cloud, “Design of FEC for low delay in 5G”, IEEE Journal on Selected Areas in Communications, 2017, pp.1783-1793

[15] F. Gabriel, S. Wunderlich, S. Pandi, F. H. P. Fitzek and M. Reisslein, “Caterpillar RLNC With Feedback (CRLNC-FB): Reducing Delay in Selective Repeat ARQ Through Coding”, IEEE Access, vol. 6 2018, pp. 44787-44802.

[16] V. Nguyen, E. Tasdemir, G. T. Nguyen, D. E. Lucani, F. H. P. Fitzek and M. Reisslein, "DSEP Fulcrum: Dynamic Sparsity and Expansion Packets for Fulcrum Network Coding," in IEEE Access, vol. 8, 2020, pp. 78293-78314.

زیرنویس‌ها

۱ Network Coding

۲ Random Linear Network Coding

۳ Finite Field

۴ Generation

۵ DOF

۶ Acknowledgement

۷ Round Robin

۸ Average Decoding Delay

۹ Real Time