

تحسين أداء بروتوكول التحكم بالنقل TCP المستخدم في الوصلات الفضائية من خلال تعديل بروتوكول TCP-peach+ ليصبح بروتوكول نقل خاص بالوصلة الفضائية فقط (satellite transport protocol) STP

الدكتور صادق علي*

الدكتور علي أحمد**

سهاد حلوم***

(تاريخ الإيداع 11 / 6 / 2014. قُبل للنشر في 6 / 11 / 2014)

□ ملخص □

يتضمن البحث اقتراح تعديل على بروتوكول TCP-peach+ من أجل تحسين أداء بروتوكول الـ TCP عبر الوصلات الفضائية ، حيث تم اقتراح البروتوكول المعدل بالاعتماد على تقنية تجزئة مسار الاتصال بين المصدر والهدف والمسماة (SPLITTING) و ذلك من أجل الحصول على بروتوكول معدل من نوع (Satellite transport protocol) STP خاص بالجزء الفضائي فقط ،من خلال اقتراح خوارزمية جديدة لهذا البروتوكول الموجود ضمن طبقة النقل الفضائي (Satellite protocol layer) STL ، وقد ركزنا بالبحث على المقارنة ما بين البروتوكول المقترح و بروتوكول TCP-Peach+ .

الكلمات المفتاحية: طبقات البروتوكولات الفضائية- TCP-Peach+ - بروتوكول النقل الفضائي- الاتصالات الفضائية

* أستاذ مساعد-قسم هندسة الاتصالات و الإلكترونيات-كلية الهندسة الميكانيكية و الكهربائية-جامعة تشرين-اللاذقية-سورية.
** أستاذ مساعد-قسم هندسة الاتصالات و الإلكترونيات-كلية الهندسة الميكانيكية و الكهربائية-جامعة تشرين-اللاذقية-سورية.
*** طالب دكتوراه- قسم هندسة الاتصالات و الإلكترونيات-كلية الهندسة الميكانيكية و الكهربائية-جامعة تشرين-اللاذقية-سورية.

Improving the Performance of TCP over Satellite Links by Modifying the Protocol TCP-Peach+ to Become Satellite Transport Protocol (STP)

Dr.Sadek Ali*
Dr.Ali Ahmad**
Sohad Halloum***

(Received 11 / 6 / 2014. Accepted 6 / 11 / 2014)

□ ABSTRACT □

This research includes an amendment to a TCP-Peach+ protocol to improve the performance of the TCP over satellite links, Where the protocol has been proposed based on splitting the connecting path between the source and target, in order to obtain an adjusted protocol of the kind of Satellite transport protocol (STP) specialized in the space segment only, and that by proposing a new algorithm for this protocol which is located under the satellite transport layer (STL). We have focused in our research on a comparison between the proposed protocol and TCP-Peach+ protocol.

Keywords: Satellite Protocol Stack(SPS); TCP-Peach+; TCP over Satellite Links; Satellite Communications; Performance Enhancement Proxies(PEPs).

* Associate Professor, Department of Communication Engineering; Faculty of Mechanical & Electrical Engineering; University of Tishreen; Lattakia, Syria.

** Associate Professor, Department of Communication Engineering; Faculty of Mechanical & Electrical Engineering; University of Tishreen; Lattakia, Syria.

*** PH.D Student, Department of Communication Engineering; Faculty of Mechanical & Electrical Engineering; University of Tishreen; Lattakia, Syria.

مقدمة:

إنّ لكل من الشبكة الأرضية و الوصلة الفضائية خصائص متباينة تماماً تؤثر بشكل مختلف في أداء بروتوكول التحكم بالنقل من نوع end-to-end حيث تتصف الشبكة الأرضية بأنها تمتلك:

1. زمن استجابة RTT (round trip time) قصير نسبياً و ذلك ينعكس إيجاباً على سرعة عملية استرداد حجم نافذة بعد كل تخفيض لها نتيجة أخطاء الإرسال.
2. BER (bit error rate) منخفض أي إنّ هنالك أخطاء وصلة أقل بينما تكون أخطاء الازدحام بالشبكة أكثر شيوعاً نتيجة تشعب هذه الشبكة.

بينما تتصف الشبكة الفضائية بما يأتي:

1. زمن استجابة RTT طويل نسبياً وذلك ينعكس سلباً على سرعة عملية استرداد حجم نافذة بعد كل تخفيض لها نتيجة أخطاء الإرسال مما يؤثر في أداء كامل مسار الاتصال بما فيها الشبكة الأرضية .
2. BER مرتفع أي إنّ أخطاء الوصلة شائعة الحدوث بينما يكون الأخطاء الناتجة عن الازدحام معدومة نتيجة استخدام بروتوكولات التحكم بالوصول للقناة الفضائية والتي تقوم بتنظيم الوصول للقناة الفضائية و بحل مشكلة التصادم على مستوى طبقة وصلة المعطيات ،بالإضافة إلى معرفة المسار المحدد للوصلة الفضائية أي عدم وجود تشعب بمسارها (uplink-downlink) .

وبالتالي عند الجمع بين الشبكتين سوف نحصل على شبكة ذات BER مرتفع و RTT طويل وشبكة مزدحمة يكون لها تأثير سلبي كبير في أي بروتوكول تحكم بالنقل من نوع end-to-end. لذلك فإن عملية الفصل ما بين الشبكتين على مستوى طبقة النقل سوف يحقق أداء أفضل لمسار الاتصال ككل ،من خلال فصل بروتوكولات التحكم بالنقل لكل شبكة على حدة من أجل الاستفادة من ميزات كل شبكة لتحسين أداء بروتوكول التحكم بالنقل .

أهمية البحث و أهدافه:

تكمن أهمية هذا البحث من اقتراح تعديل على بروتوكول TCP-peach+ من أجل تحسين أداء بروتوكول الـ TCP عبر الوصلات الفضائية ، والتخلص من نقاط الضعف الموجودة في أداء البرتوكول TCP-PEACH+ و الذي طور من أجل الاتصالات الفضائية و المتمثلة بالنقاط التالية: [1,2]

• التناقص التدريجي لحجم نافذة الازدحام (congestion window) في حال تكرار حدوث أخطاء الإرسال حيث نلاحظ من خلال خوارزمية الاسترداد السريع أن المرسل يقوم بتخفيض قيمة نافذة الازدحام المستخدمة إلى النصف عند حدوث كل ضياع و تكرار هذه العملية ضمن نفس النافذة يؤدي إلى تذبذب قيمة حجم نافذة الإرسال و من ثمّ عدم استقرار الأداء لقناة الاتصال.

• البرتوكول TCP-PEACH+ هو بروتوكول END-TO-END أي عملية التحكم بالإرسال بكل ما يتضمنه من آليات تقادي الازدحام و إعادة الإرسال للمعلومات التي لم تصل بشكل صحيح وعمليات الاسترداد السريع لمعدل الإرسال بعد حدوث أخطاء في الشبكة تتم في المرسل و هذه العمليات تؤثر في كامل المسار من المرسل إلى المستقبل والتي تكون مقبولة في الشبكات الأرضية ذات معدل الـ BER و التأخير الزمني الصغيرين .

وللتوضيح : فإن خطأ الإرسال والذي يسبب عدم وصول البيانات من المرسل إلى المستقبل بشكل صحيح يمكن

أن ينتج عن سببين أساسيين :

1. الازدحام بالشبكة (حجم المعلومات المرسله عبر الشبكة أكبر من سعة هذه الشبكة مما يسبب ضياع البيانات و عدم وصولها بشكل صحيح)
 2. خطأ الوصلة (LINK ERROR) و المتعلق بشكل أساس بال BER .
- يعتمد بروتوكول TCP-peach+ في آلية عمله على اكتشاف خطأ الإرسال بناءً على ما يسمى رسالة الإقرار بالوصول المكررة duplicated ACK فعند وصول ثلاث رسائل من هذا النوع تقوم بإعادة إرسال الجزء الضائع و تدخل في طور الاسترداد السريع. لكن الخوارزمية المستخدمة في مرحلة الإصلاح السريع تبدأ بإنقاص نافذة الازدحام إلى نصف القيمة التي كان عليها قبل حدوث الخطأ وذلك دون معرفة منشأ هذا الخطأ (ناتج عن الازدحام أم عن خطأ الوصلة)، وبما أن خطأ الوصلة لا يتطلب تخفيض معدل الإرسال و سيكون التخفيض السابق غير ضروري للحمل مما يؤدي إلى انخفاض أداء قناة الاتصال والذي يكون في الأساس منخفض في الوصلة الفضائية أكثر مما هو عليه في الوصلة الأرضية.

ومن خلال دراسة بروتوكولات التحكم بالوصول للقناة الفضائية فإن تلك البروتوكولات تضمن عدم حدوث ازدحام بالقناة الفضائية من خلال تنظيم عملية الوصول للقناة و حل مشكلة التصادم للرمز المرسله على مستوى طبقة المعطيات. مما سبق نستنتج أن أخطاء الإرسال الناتجة عن الوصلة الفضائية تنتج فقط عن خطأ الوصلة الذي لا يتطلب تخفيضاً لحجم نافذة الإرسال، ومن ثم عند استخدام بروتوكول تحكم بالنقل من نوع end-to-end فإن آلية تصحيح أخطاء الإرسال المستخدمة ضمن هذا البروتوكول على مستوى طبقة النقل سوف تؤثر سلباً في أداء الوصلة الفضائية من خلال التخفيض غير الضروري لحجم نافذة الإرسال بالنسبة إلى الوصلة الفضائية والتي تستغرق وقتاً طويلاً نسبياً لمحاولة استرداد حجم النافذة لوضعه الطبيعي نتيجة ال RTT الطويل للوصلة الفضائية .

• يتطلب تطبيق بروتوكول TCP-peach+ إرسال NIL segment ذات أفضلية منخفضة عن أجزاء البيانات النظامية الأخرى و بما أنه بروتوكول end-to-end فإن هذا يستوجب تحديد تلك الأفضليات في كل العقد الموجودة في مسار الاتصال بين المرسل و المستقبل. كما يؤدي إرسال NIL segments إلى حمل زائد (overhead) على الشبكة و يكون التأثير الأكبر على الوصلة الفضائية التي غالباً ما تكون تكلفتها متناسبة مع حجم المعلومات المرسله. لذلك تم اقتراح حل من خلال فصل عمليات التحكم بالإرسال و عدم استخدام بروتوكولات END-TO-END حيث يتم تطبيق بروتوكولات تحكم بالإرسال في الشبكات الأرضية بشكل منفصل عن بروتوكول التحكم بالإرسال للجزء الفضائي و بالتالي فإن كل شبكة تكون مسؤولة عن أخطاء الإرسال ضمنها فقط و عدم إشغال الشبكات الأخرى بهذا الخطأ [5,6].

طرائق البحث ومواده:

اقترح حل فصل عمليات التحكم بالإرسال من خلال عدم استخدام بروتوكولات END-TO-END ، وذلك للحصول على بروتوكولات تحكم بالإرسال للجزء الفضائي بشكل منفصل عن بروتوكول التحكم بالإرسال في الشبكات الأرضية و بالتالي كل شبكة تكون مسؤولة عن أخطاء الإرسال التي تحدث ضمنها فقط وعدم إشغال الشبكات الأخرى بهذا الخطأ ، تم الاعتماد على التحليل الرياضي لكل من خوارزمية TCP-peach+ المعروف أداؤها [1,2] ، والخوارزمية المعدلة المقترحة باستخدام برنامج ال Matlab ، ومن ثم مقارنة أداء البروتوكول المقترح مع أداء بروتوكول

البروتوكول بالاعتماد على TCP-peach+ المستخدم في الاتصالات الفضائية والحصول على النتائج انطلاقاً من هذه المقارنة، تم اقتراح البروتوكول بالاعتماد على :

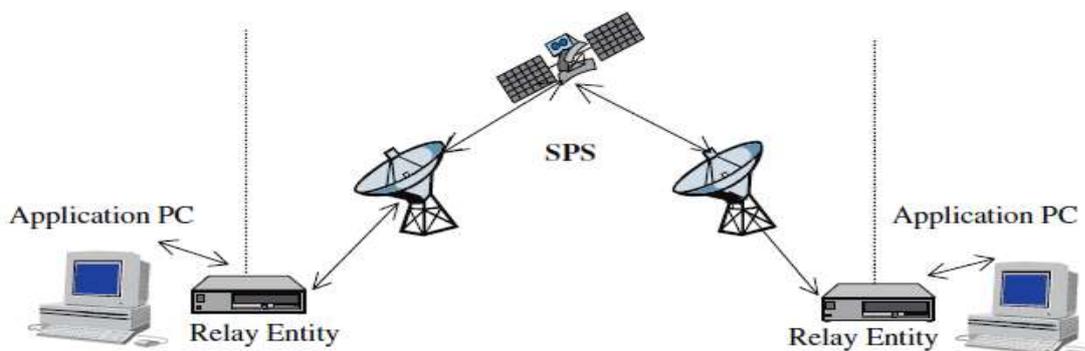
دراسة طرق تحسين أداء المزودات (PEP_s) (performance enhancement proxies) [3,4] :

يفضل عند إجراء أي تعديل أو تطوير لبروتوكول معين ألا يتطلب هذا التعديل تغييراً أو تعديلاً في النهايات الطرفية ومن ثم فإن معظم التعديلات التي طبقت على بروتوكول التحكم بالنقل TCP المطبق بطريقة END-TO-END من أجل تحسين أدائه عبر الوصلة الفضائية كانت تتطلب تعديلاً في طبقة النقل للمحطات الطرفية و أحياناً في عقد الاتصال مثل (TCP-PEACH, TCP-PEACH+). ومن أجل التخلص من هذا التعديل في المحطات الطرفية أو في عقد الاتصال كافة، تم التركيز على تحسين أداء بروتوكول التحكم بالنقل من خلال العمل فقط على الجزء الفضائي للشبكة. لتحقيق ذلك طرحت مجموعة من المخططات لبناء الشبكة مع الموازنة بين تحسين أداء بروتوكولات طبقة النقل و تعقيد النظام، سميت تلك التعديلات بـ PEP_s والتي تتضمن طرح حلول لبروتوكولات نقل خاصة بالوصلة الفضائية بشكل منفصل عن الجزء الأرضي.

يتمثل الحل المطروح بما يأتي:

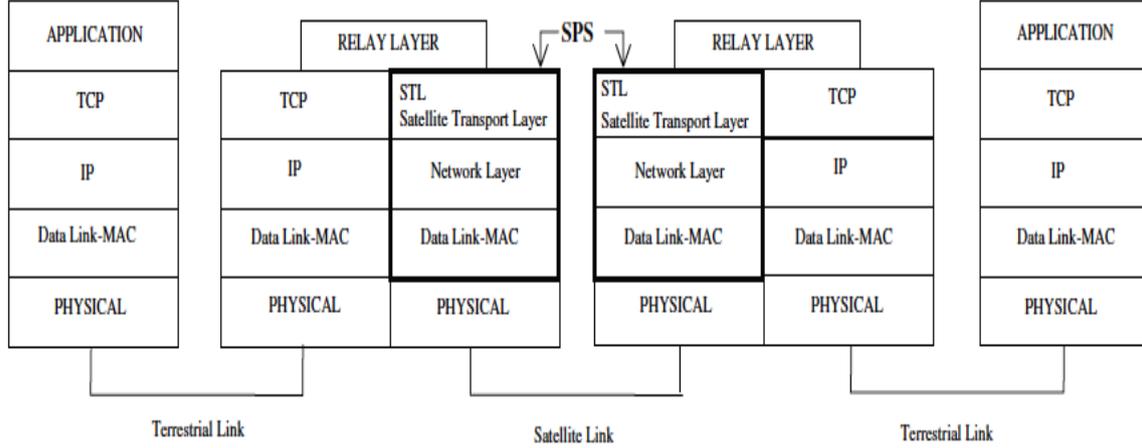
• الفصل ما بين الجزء الفضائي و الجزء الأرضي للاتصال splitting :

إن الهدف الرئيس من فصل الجزء الفضائي للاتصال عن باقي الشبكة هو توفير المرونة اللازمة لتصميم و تطبيق بنية طبقات تتضمن بروتوكولات مخصصة فقط للجزء الفضائي ذي الخصوصية المختلفة عن الاتصال الأرضي. تتمثل عملية الفصل بتجزئة اتصال TCP الذي يعمل بألية END-TO-END إلى ثلاثة اتصالات منفصلة: اثنان منها لربط طرفي الاتصال (المرسل والمستقبل) مع البوابتين الفضائيتين و الثالث للربط ما بين البوابتين عبر الجزء الفضائي [12]. لذلك تم تصميم بنية جديدة للشبكة بحيث يحجب الجزء الفضائي عن باقي أجزاء الشبكة المتمثلة بالنهايات الطرفية التي تستمر باستخدام طبقات TCP النموذجية للاتصال بالبوابة الفضائية، بينما تستخدم البوابات الفضائية بروتوكولات مصممة بشكل خاص للوصلة الفضائية للاتصال فيما بينها، و البنية المطروحة للشبكة هي Satellite protocol stack (SPS) و الموضحة بالشكل (1) [3,4] :



الشكل (1) عملية الفصل بين الجزء الفضائي والأجزاء الأرضية للاتصال

تعتبر بنية الشبكة الموضحة بالشكل (1) عن جمع ما بين الشبكة الأرضية و الجزء الفضائي من خلال استخدام ما يسمى بالمبدلات (RELAY ENTITY) و هي عبارة عن جهاز بسيط يوصل مباشرة إلى جهاز حاسب شخصي PC أو يمكن أن يكون عبارة عن كرت يركب ضمن الحاسب مع تنصيب البرنامج اللازم لتشغيله[4].و بالتالي يمكن تمثيل بنية SPS بالشكل(2)[4] :



الشكل(2) بنية SPS

تضمن طبقة التحويل (relay layer) الاتصال بين طبقة النقل الفضائي STL و طبقة الـ TCP للجزء الأرضي. تضم طبقة النقل الفضائي STL بروتوكول نقل فضائي STP (Satellite transport protocol) تطرح هذه البنية الجديدة إمكانية تصميم بروتوكولات نقل خاصة بالجزء الفضائي فقط بشكل مناسب للوصلة الفضائية متخلصة من آلية العمل بطريقة END-TO-END .

النتائج والمناقشة:

بناءً على ما سبق يمكن اقتراح بروتوكول تحكم بالإرسال الفضائي STP من خلال ما يأتي :

- (1) الاعتماد على حل الفصل ما بين الجزء الأرضي و الجزء الفضائي لتصبح بنية الشبكة المدروسة كما هو مبين في الشكلين (1,2) ليتم التركيز على بنية SPS والبروتوكولات المستخدمة لتحسين أداء القناة الفضائية.
- (2) تقديم آلية جديدة لتحديد حجم نافذة الازدحام العظمى maxcwnd (عدد TCP-segments الموجودة في الوصلة الفضائية بين المرسل و المستقبل خلال RTT واحدة): يتم تحديد الحجم الأعظمى لمجموع الـ TCP_segments المرسله خلال RTT واحدة بحيث تكون أصغر أو تساوي حجم الـ buffer في المستقبل (نافذة المستقبل) [7,8]:

$$\text{Segments_size} \leq \text{rwd} \quad (1)$$

يقوم المستقبل بإخبار المرسل بهذا الحجم مقدراً بالبايت عن طريق كلمة بطول 16 bit محجوزة ضمن الـ TCP_segment وبافتراض حيز 1 BYTE للحمل الزائد (overhead) يصبح حجم نافذة المستقبل كما يأتي:

$$\text{rwd} = 2^{16} - 1 = 65.535 \text{ Byte} \approx 64 \text{ KB}$$

$$\text{Segments_size} \leq 64 \text{ KB}$$

تمثل القيمة 64KB الحجم الأعظمي للـ TCP_segments المرسله خلال RTT واحدة. يؤمن الحجم هذا أداءً جيداً بالنسبة إلى الاتصالات الأرضية ولكن بسبب خصوصية الوصلة الفضائية من حيث التأخير الزمني الكبير (RTT=0.55 sec) فإن هذا ينتج ما يسمى بـ (Bandwidth –delay product) والذي يعبر عن حجم الـ TCP_segments التي يمكن أن تشغل الوصلة الفضائية بين المرسل و المستقبل خلال RTT واحدة ويمكن أن تسمى (pipe line) [8]. فمن أجل معدلات بت مرتفعة فإن القيمة السابقة (64 KB) لا تعبر عن الحجم الفعلي بشكل سليم وفعال فعلى سبيل المثال:

عند وجود وصلة فضائية تعمل بمعدل بت R=30Mbps أي ما يساوي 3.75*10⁶ Byte/S والذي يساوي حوالي 3662 KBps يقوم المرسل باستخدام هذا المعدل ويكون حجم المعلومات الأعظمي التي يمكن إرسالها خلال RTT=0.55sec تساوي: [8]

$$BDP=R_b * R_{TT} \quad (2)$$

$$BDP=3662 * 0.55=2014 \text{ KB}$$

نلاحظ الفرق الكبير ما بين هذه القيمة (2014KB) و القيمة السابقة (64KB) و هذا ما يجعل أداء القناة الفضائية منخفض بسبب عدم استخدام القناة بالشكل الفعال.

لذلك نقترح الآلية الجديدة لتحديد حجم maxcwnd للمرسل مقدرة بعدد TCP_segments من خلال العلاقة:

$$\text{maxcwnd}=R_b * RTT / \text{SEGMENT SIZE} \quad (3)$$

حيث: segment size تعبر عن حجم TCP_segment واحد. مما يجعلها دائماً مرتبطة بمعدل الإرسال و زمن RTT .

(3) اقتراح بروتوكول نقل فضائي STP ضمن طبقة النقل الفضائي STL وذلك بإجراء تعديلات على بروتوكول TCP-PECAH+ المستخدم بآلية END-TO-END ليصبح بروتوكول نقل فضائي عبر الوصلة الفضائية بين البوابتين الفضائيتين ليعمل وفق خصوصية القناة الفضائية .

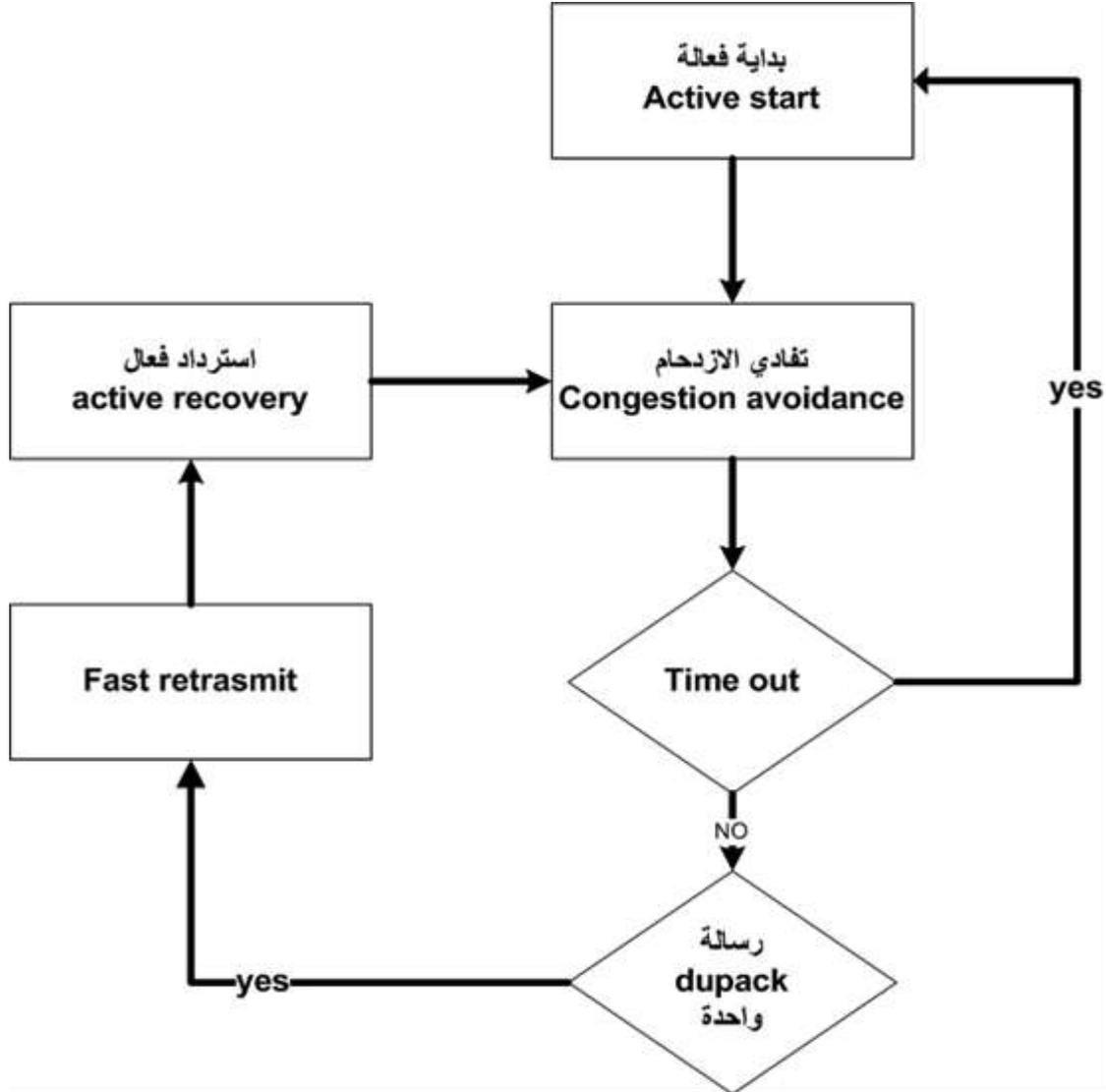
خوارزمية بروتوكول النقل الفضائي المقترحة:

يوضح الشكل (3) الخوارزمية المقترحة ، نلاحظ أنه تم اقتراح خوارزميتين جديدتين بحيث تتوافقان مع آلية التحكم الجديدة المقترحة وهما:

- 1) خوارزمية البداية الفعالة (active start) بدلاً من jump start الموجودة في الـ TCP-peach+.
- 2) خوارزمية الاسترداد الفعال (active recovery) بدلاً من quick recovery الموجودة في الـ TCP-peach+ .

(3) يتم الدخول في مرحلتي إعادة الإرسال السريع و الاسترداد الفعال بعد وصول رسالة dupack واحدة بدلاً من ثلاث كما هو مطبق في TCP-PEACH+ لأن بروتوكول النقل المقترح يطبق فقط على الوصلة الفضائية ذات المسار الواحد و المعروف للمحطة المرسله و المحطة المستقبلة بالإضافة إلى أن جميع أجزاء البيانات ذات أفضلية واحدة و بالتالي فإن الأجزاء المرسله (TCP_segments) سوف تصل بالترتيب نفسه إلى الهدف .وإن استخدام رسالة

dupack واحدة سوف تسرع عملية اكتشاف الخطأ و إصلاحه بشكل سريع بحيث يقل احتمال حدوث أخطاء أخرى لاحقة قبل اكتشاف الخطأ السابق.

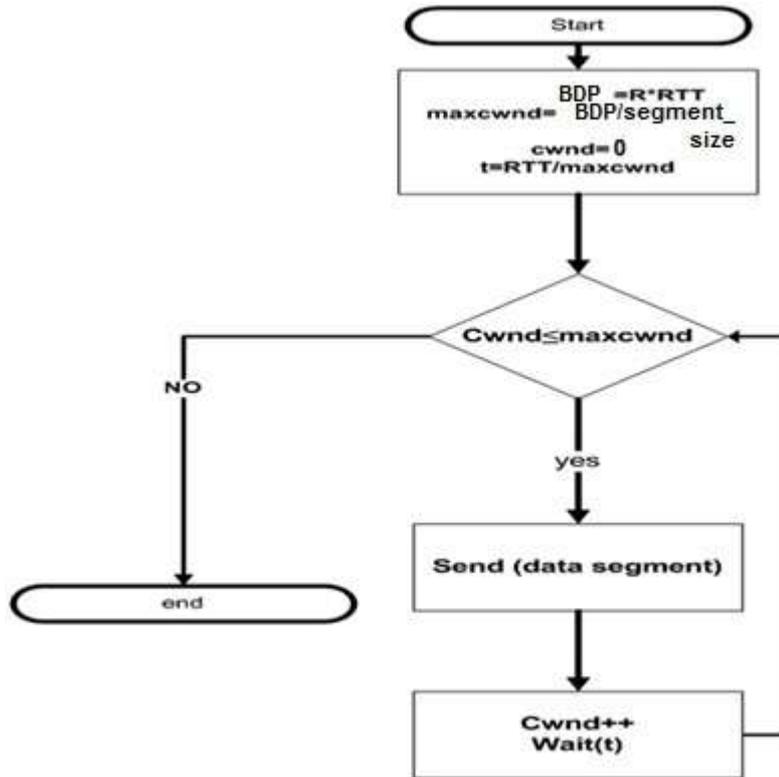


الشكل (3) خوارزمية بروتوكول النقل الفضائي المقترحة

A. خوارزمية البداية الفعالة (active start): تمثل هذه الخوارزمية بالشكل (4) ، يمكن إيضاح التباين بين هذه الخوارزمية وخوارزمية Jump start المستخدمة في TCP-Peach+ بالشكل الآتي:

خوارزمية البداية الفعالة (active start) المقترحة	خوارزمية Jump start المستخدمة في TCP-Peach+ [1,2]
Active start() $BDP=R*RTT;$ $maxcwnd= BDP/segment_size;$ $t=RTT/maxcwnd;$	Jump_start() $cwnd=1;$ $t=RTT/maxcwnd;$ Send(TCP_segment);

<pre> cwnd=0; while(cwnd≤maxcwnd) send(TCP_segment); cwnd++; wait(t); end; end; </pre>	<pre> For i=1 to (maxcwnd-1) wait(t); send(NIL_segment); end; end; </pre>
--	---



الشكل (4) خوارزمية البداية الفعالة (active start)

من خلال المقارنة بين الخوارزميتين نجد أن خوارزمية jump start تعتمد على حجم buffer في المستقبل (rwd) و تقوم بإرسال TCP-segment واحد و [(maxcwnd-1)NIL segments] خلال RTT واحد مما يزيد من التحميل الإضافي overhead على كامل مسار الاتصال end-to-end. أما في خوارزمية البداية الفعالة المقترحة active start فقد تم استخدام الآلية المقترحة سابقاً لتحديد maxcwnd من خلال معدل الإرسال و الـ RTT حيث تقوم الخوارزمية بتحديد الحجم الأعظمي لـ TCP_segments الذي يمكن إرسالهم من خلال نافذة الإرسال بالمتحول BDP ومن ثم تحديد maxcwnd والتي تعبر عن عدد TCP-segments التي سترسل خلال RTT واحدة و الناتج عن تقسيم المتحول BDP على حجم TCP_segment. تتابع عملية الإرسال لحين الوصول لحجم النافذة الأعظمي حيث يتم بعدها دخول مرحلة تقادي الازدحام. نلاحظ أن كل الأجزاء المرسله هي TCP_segments أي عدم وجود حمل زائد (over head) على عكس خوارزمية jump start.

.B مرحلة تفادي الازدحام :

```

Congestion avoidance()
While(ACK- ARRIVAL)
send(TCP-segment);
end;
end;

```

نلاحظ أنه في مرحلة تفادي الازدحام يتم إرسال TCP_segment بشكل متزامن مع وصول رسائل ACK للأجزاء المرسله بحيث نحافظ على حجم النافذة الأعظمي maxcwnd .

.C مرحلة الاسترداد الفعال (Active recovery) :

دلالة بعض البارامترات المستخدمة في الجزء البرمجي كما يلي: [1,2]
 High ACK : رسالة ذات الرقم التسلسلي الأكبر المستقبل بلحظة ما بشكل تراكمي (مرتب).
 High data : يعبر عن جزء البيانات ذي الرقم التسلسلي الأكبر و المرسل قبل الدخول في مرحلة الاسترداد السريع.

Duplicate ACK : رسالة ACK لـ TCP_segment حيث يكون رقمه التسلسلي (التراكمي) يساوي القيمة الحالية لـ High ACK وينقل أيضاً معلومات حول تأكيد وصول أجزاء بيانات جديدة عن طريق خيار SACK المستخدم في هذه الخوارزمية و الذي يسهم في تحديث سجل الإصابات (SCORBOARD) في المرسل.
 Partial ACK : رسالة ACK جزئية تقوم بزيادة قيمة High ACK ولكنها لا تعبر عن إقرار توصيل لكل TCP_segments المرسله قبل الدخول في مرحلة الاسترداد السريع أي لا تعبر عن وصول الـ High data .

Recover ACK : رسالة ACK للجزء High data .

NdupACK : عدد رسائل ACK المكررة و التي تستوجب إعادة الإرسال و الدخول في طور الاسترداد السريع (تكون القيمة هي 3 في TCP-peach+ و 1 في البروتوكول المقترح كما هو موضح سابقاً).
 maxburst : يعبر عن العدد الأعظمي للـ TCP-segments التي يمكن أن ترسل بعد استقبال رسالة ACK حتى لو سمحت نافذة الإرسال بأكثر من ذلك.

amountacked : عدد أجزاء الـ TCP_segments التي أقر وصولها من خلال استقبال الـ ACK.

adps : عدد الـ TCP_segments المسموح إرسالها عبر نافذة الإرسال.

nps : عدد الـ TCP_segments التي ترسل خلال مرحلة الاسترداد السريع .

pipe : عدد الـ TCP_segments المرسله والمتواجدة في الشبكة و التي لم يتم البت بأمرها بعد ،أي لم تصل

بشأنها أي رسالة تأكيد وصول أو عدم تأكيد وصول (dupACK) إلى المرسل.

يمكن إيضاح التباين بين هذه الخوارزمية المقترحة (active recovery) وخوارزمية الاسترداد السريع

(quick recovery) المستخدمة في TCP-Peach+ بالشكل الآتي:

<p>خوارزمية الاسترداد الفعال المقترحة</p> <p>Active recovery</p>	<p>خوارزمية الاسترداد السريع المستخدمة في TCP-Peach+ [1,2]</p> <p>quick recovery</p>
<pre> Active recovery() adps=0; END=0; Pipe=cwnd; while(END=0) if(ACK_ARRIVAL) if(Duplicate ACK) pipe=pipe-1; update scoreboard; end; else if(Partial ACK) pipe=pipe- amountacked; update High ACK; update scoreboard; end; else if(Recover ACK) update High ACK; clear scoreboard; END=1; end; adps=cwnd-pipe; nps=min(maxburst,adps); if (nps>0) send nps missing TCP-segments and/or new TCP-segments; pipe=pipe+nps; end; end; end; end; </pre>	<pre> Quick recovery() cwnd=cwnd/2; adps=0; END=0; Pipe=2*cwnd; t=(RTT/2)/cwnd; send(cwnd)NIL segment at each interval t; while(END=0) if(ACK_ARRIVAL) if(Duplicate ACK) pipe=pipe-1; update scoreboard; end; else if(Partial ACK) pipe=pipe- amountacked; update High ACK; update scoreboard; end; else if(NIL ACK) cwnd=cwnd+1; update scoreboard; end; else if(Recover ACK) update High ACK; clear scoreboard; END=1; end; adps=cwnd-pipe; nps=min(maxburst,adps); </pre>

	<pre> if (nps>0) send nps missing segments and/or new segments; pipe=pipe+nps; end; end; end; end; </pre>
--	--

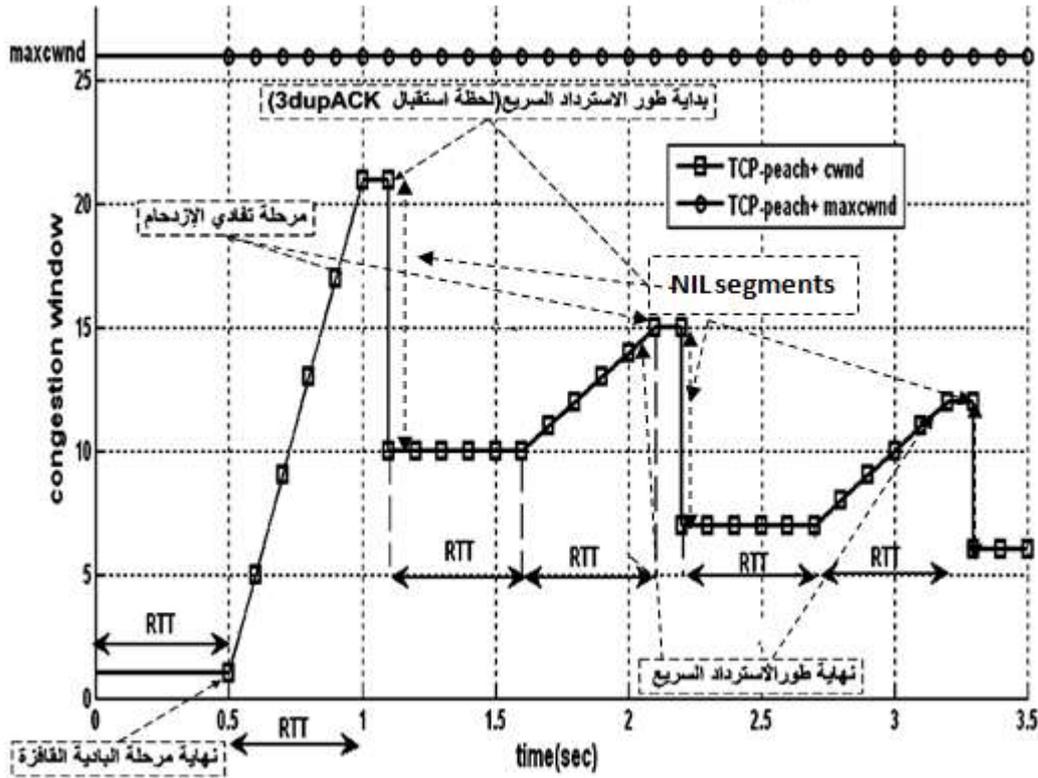
نلاحظ أن الفرق بين الخوارزميتين يكمن في أن خوارزمية الاسترداد الفعال المقترحة تعتبر أن رسالة الـ dupACK نتيجة خطأ الوصلة لذلك لا تقوم بتصنيف نافذة الإرسال و إنما فقط تكفي بإعادة الإرسال للجزء الضائع عن طريق مرحلة إعادة الإرسال السريع التي تسبقها على ثم تقوم بمراقبة سجل الإصابات scoreboard من أجل تحديد فيما إذا حدث ضياع جديد و لا تخرج من هذا الطور إلا عند وصول رسالة ACK لأخر جزء أعيد إرساله. بينما تقوم خوارزمية الاسترداد السريع بتصنيف نافذة الإرسال. كما نلاحظ أيضاً عدم وجود حمل زائد overhead على الشبكة على عكس خوارزمية الاسترداد السريع التي تقوم بإرسال NIL segment بمقدار حجم النافذة الجديدة بعد التصنيف.

طالما هناك وصول لرسائل ACKs إلى المرسل فمن المتوقع حدوث إحدى الأحداث الآتية:

- (1) حالة وصول Duplicate ACK: يدل هذا على وصول TCP_segment بشكل صحيح إلى الهدف لذلك سوف ينقص pipe بمقدار واحد. تتوفر نتيجة تطبيق خيار SACK معلومات حول الأجزاء المستقبلية مما يتيح تحديث معلومات سجل الإصابات الموجود في المرسل و التي تبين فيما إذا كان هناك جزء جديد قد فُقد.
- (2) حالة وصول Partial ACK: يعبر هذا عن أن TCP_segment المعاد إرساله أو المزيد من الـ TCP_segments المتلاحقة قد وصلت إلى المستقبل و يقتضي ذلك تخفيض قيمة pipe بمقدار حجم الأجزاء التي غادرت الشبكة و تحديث سجل الإصابات و HighACK، لا تعبر Partial ACK عن آخر جزء قد أعيد إرساله لأنه كما هو واضح في الخوارزمية يمكن أن تصل رسائل dupACK جديدة خلال طور الاسترداد السريع تعبر عن ضياع أجزاء جديدة.
- (3) حالة وصول Recover ACK: عندها تكون كل الأجزاء المعاد إرسالها قد غادرت الشبكة لذلك يتم إنهاء طور الاسترداد و الدخول في طور تقادي الازدحام.
- (4) بعد كل حالة من الحالات السابقة تقوم الخوارزمية بحساب عدد TCP_segments التي يمكن إرسالها من خلال المتحول adps. في حال كانت قيمة adps موجبة يكون عندها حجم نافذة الازدحام أكبر من حجم الـ TCP_segments الموجودة في الشبكة لذلك يقوم المرسل بفحص سجل الإصابات من أجل إرسال أجزاء جديدة. إذا كان هناك ضياعات لأكثر من جزء خلال نفس نافذة الإرسال، يتم إعادة إرسال هذه الأجزاء أولاً قبل أي أجزاء جديدة.

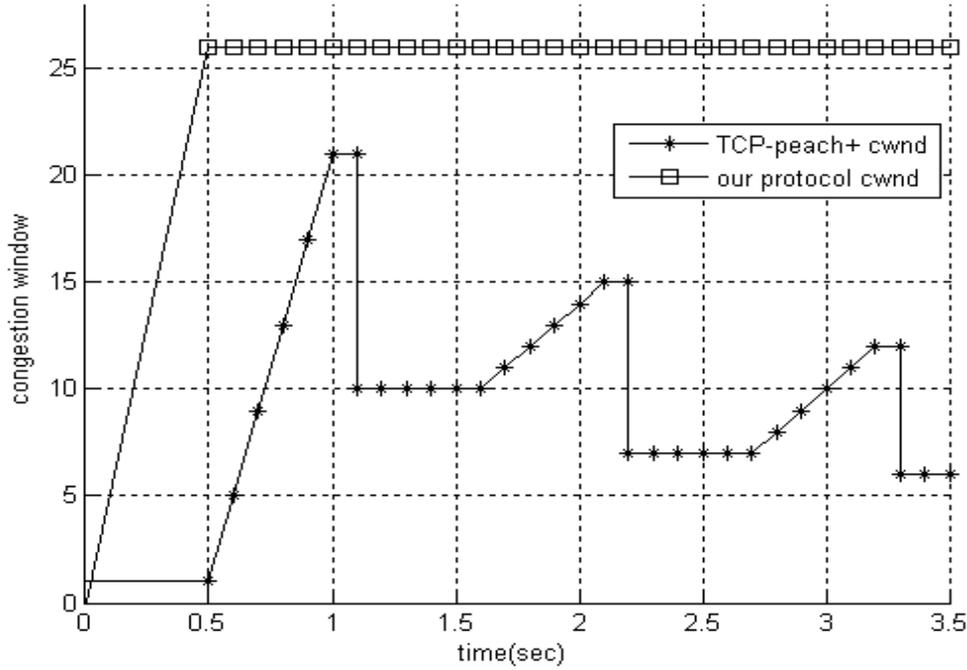
بالنتيجة إن حجم نافذة الإرسال خلال مرحلة الاسترداد الفعال يبقى ثابتاً ومحافظةً على القيمة التي كانت قبل الدخول إلى هذه المرحلة مما ينعكس على ثبات أداء القناة الفضائية بشكل إيجابي ويؤدي إلى التخلص من التذبذب الذي كان يحدث نتيجة تطبيق خوارزمية الاسترداد السريع .

بينما نلاحظ تذبذب وتناقص تدريجي لحجم نافذة الإرسال في بروتوكول TCP-PEACH+ عند تكرار حدوث ضياع للأجزاء المرسله كما هو مبين بالشكل(5) باعتبار $\text{maxcwnd} = 26$ segments [1,2] :



الشكل(5) حجم نافذة الازدحام في بروتوكول TCP-peach+ عند حدوث ضياعات متكررة للـ TCP_segments المرسله

يبين الشكل (6) مقارنة حجم نافذة الازدحام للبروتوكول المقترح مع حجمها في بروتوكول TCP-peach+ عند حدوث ضياعات متكررة للـ TCP_segments المرسله :



الشكل (6) مقارنة حجم نافذة الازدحام للبروتوكول المقترح مع حجمها في بروتوكول TCP-peach+ عند حدوث ضياعات متكررة للـ TCP_segments المرسل

نلاحظ أن البروتوكول المقترح سوف يصل إلى maxcwnd في نهاية مرحلة البداية الفعالة (أي أسرع من البداية السريعة المستخدمة في TCP-peach+ بمقدار RTT) و من ثم يحافظ على هذه القيمة طيلة فترة الإرسال حتى في حال حدوث ضياعات للـ TCP_segments المرسل كما هو مبين سابقاً من خلال عمل مرحلة الاسترداد الفعال بينما يكون هناك تذبذب بحجم النافذة للبروتوكول TCP-PEACH+ عند ضياع الـ TCP_segments بشكل متكرر، مع الملاحظة أن حجم نافذة الازدحام في البروتوكول المقترح هي فقط لقناة الاتصال الفضائية، أما بالنسبة إلى TCP-Peach+ فهي لكامل الوصلة من المرسل إلى المستقبل (للشبكة الأرضية و الفضائية). لذلك عند استخدام البروتوكول المقترح يمكن تطبيق أحد البروتوكولات المقترحة للشبكات الأرضية بشكل مستقل عن الوصلة الفضائية مثل TCP-NEWRENO الذي أثبت أداء عالي في الشبكات الأرضية نتيجة التأخير الزمني المنخفض وبالتالي زيادة سرعة استرداد معدل الإرسال الطبيعي [1,11].

إن عدم استقرار نافذة الإرسال ينعكس على أداء القناة بشكل كبير و الذي يتم توضيحه من خلال العلاقة المبينة أدناه والتي توضح أداء قناة الاتصال عند استخدام البروتوكول SR (selective repeat) للإرسال: [9,10]

$$\eta_c = \frac{D(1 - P_f)cwnd}{L + R_b T_{RTT}} \quad cwnd < 1 + \frac{R_b T_{RTT}}{L} \quad (4)$$

إذ إن:

cwnd: حجم نافذة الازدحام (Congestion window).

D(bits): عدد بتات المعلومات في كل TCP-segment.

L(bits)=D+H: حجم TCP_segment، H: عدد بتات الترويسة.

$$P_f = 1 - (1 - BER)^L \text{ :احتمال الخطأ في TCP_segment}$$

R_b (b/s) :معدل البت.

T_{RTT} (s) :الزمن اللازم للإرسال من المرسل إلى المستقبل واستقبال الرد من المستقبل إلى المرسل(زمن

الاستجابة).

$(R_b T_{RTT} / L)$ تعبر عن عدد TCP_segments التي يمكن أن تنقل بين المرسل و المستقبل بالاتجاهين.

بمقارنة أداء القناة للبروتوكولين من خلال العلاقة(4) و افتراض القيم الآتية:

$$D=1000 \text{ byte} - H=448 \text{ byte} - L=1448 \text{ byte} - R_b=2 \text{ Mb/s} - T_{RTT}=550 \text{ ms} - P_f=1\%$$

بحساب maxcwnd للبروتوكول المقترح من قبلنا و التي تحسب كما شرحنا سابقاً خلال مرحلة البداية الفعالة

بالشكل الآتي:

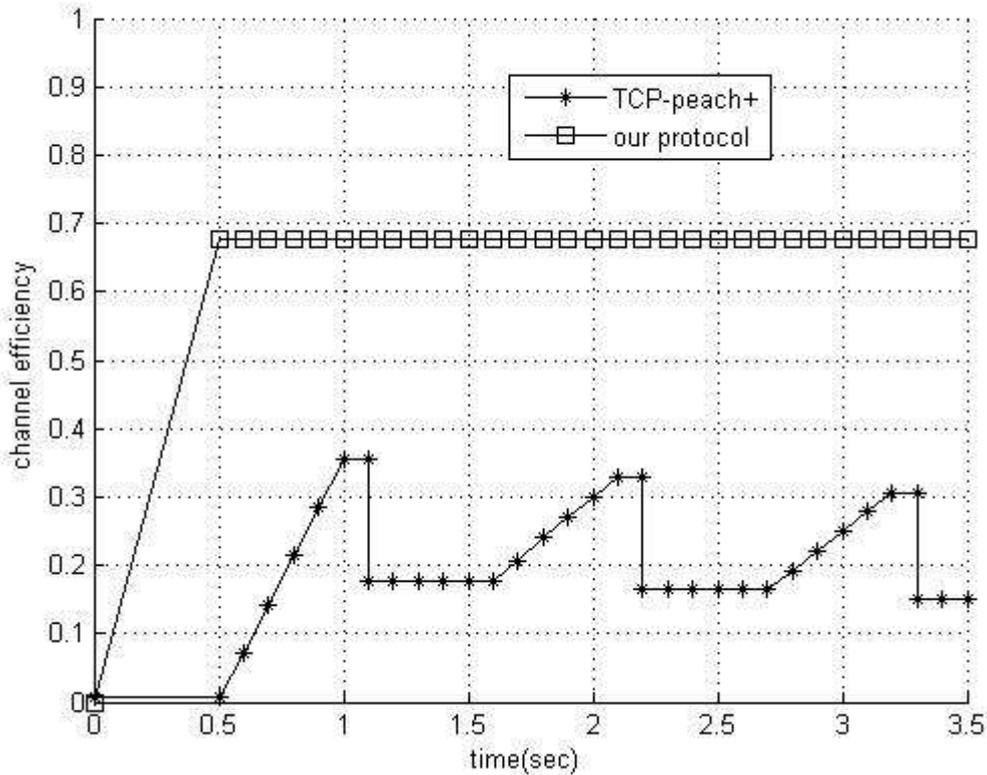
$$BDP = R_b \times RTT = 2 \times 10^6 \times 550 \times 10^{-3} = 1100000 \text{ bits}$$

$$\text{maxcwnd} = \text{maxdata} / L = 95 \text{ data-segment}$$

بالاستفادة من العلاقة(4) والقيم المفروضة و باعتبار حجم البفر (buffer size) الخاص بطبقة النقل للبوابات

الفضائية تساوي قيمة BDP يمكن أن نحصل على مقارنة أداء البروتوكول المقترح مع أداء بروتوكول

TCP-peach+ عند حدوث ضياعات للشكل(7):



الشكل(7)

مقارنة أداء البروتوكول المقترح مع أداء بروتوكول TCP-peach+ عند حدوث ضياعات متكررة للـ TCP_segments المرسل

نلاحظ من الشكل السابق أن أداء البروتوكول المقترح أفضل بكثير من بروتوكول الـ TCP-PEACH+ حيث لا يعاني من تذبذب عند حدوث ضياعات متكررة للـ TCP_segments المرسل، نتيجة استخدام بروتوكول فعال للوصول إلى القناة الفضائية وبالتالي ربط أداء القناة الفضائية بشكل أساسي بمعدل خطأ البت BER والذي يمكن تحسينه من خلال العمل على الطبقات الدنيا.

الاستنتاجات والتوصيات:

نتيجة للدراسة التي قمنا بها في هذا البحث تم الاعتماد على تقنية PEPs من أجل فصل الجزء الفضائي عن باقي الشبكة و بالتالي توافر المرونة بتقديم بروتوكول نقل خاص بالقناة الفضائية بشكل منفصل عن بروتوكول التحكم بالإرسال المستخدم في الاتصالات الأرضية وتوصلنا إلى ما يأتي :

- تغيير آلية عمل بروتوكولات التحكم بالإرسال من end-to-end إلى آلية جديدة تقوم بتطبيق بروتوكولات التحكم بالإرسال في الشبكات الأرضية بشكل منفصل عن بروتوكول التحكم بالإرسال للجزء الفضائي من خلال استخدام محولات (RELAY)، لتصبح كل شبكة مسؤولة عن أخطاء الإرسال التي تحدث ضمنها فقط وعدم إشغال الشبكات الأخرى بهذا الخطأ وخصوصاً الوصلة الفضائية ذات الخصوصية المذكورة سابقاً والتي يكون فيها احتمال ضياع الـ TCP_segment الناتج عن خطأ الوصلة أكبر بكثير من احتمال ضياع الـ TCP_segment نتيجة الازدحام (أي ليست بحاجة إلى تخفيض نافذة الازدحام إلى النصف عند كل ضياع الـ TCP_segment مرسل). من جهة أخرى فإن تطبيق أحد البروتوكولات المقترحة للشبكات الأرضية (التي يكون فيها احتمال ضياع الـ TCP_segment نتيجة الازدحام أكبر بكثير من احتمال ضياع الـ TCP_segment الناتج عن خطأ الوصلة) بشكل منفصل عن الوصلة الفضائية مثل TCP-NEWRENO قد أثبت أداء عالي نتيجة التأخير الزمني المنخفض في الشبكات الأرضية وبالتالي تسريع عملية استرداد معدل الإرسال الطبيعي بعد تخفيضه نتيجة الازدحام في الشبكة الأرضية.

- اقتراح خوارزمية جديدة لبروتوكول النقل الفضائي STP الموجود ضمن طبقة النقل الفضائي STL . وجدنا بتطبيق البروتوكول الجديد على القناة الفضائية أن أداء هذه القناة أصبح ثابت حيث تم التخلص من التذبذب حتى عند تكرار حدوث ضياعات لـ TCP_segments المرسل و الناتج عن استخدام البروتوكول TCP-PEACH+ ، وربط أداء القناة الفضائية بشكل أساسي بالـ BER.

- لا يتطلب تطبيق البروتوكول المقترح سوى تعديلات في المحطات الأرضية فقط، على عكس البروتوكول TCP-PEACH+ الذي يعمل بتقنية end-to-end إذ يتطلب تعديل بالطرفيات المرسل و المستقبل بالإضافة إلى تعديل في كل عقد الاتصال المتواجدة في كل المسارات المتوقعة بين المرسل و المستقبل، ذلك لضمان العمل بنظام الأفضليات المختلفة للباكيتات المرسل كما وضحنا سابقاً.

- التخلص من التحميل الإضافي (overhead) على الشبكة الأرضية و الفضائية والناتج عن إرسال NIL segments الناتجان عن استخدام بروتوكول TCP-PEACH+.

المراجع:

- 1-Satoshi,Utsum;Salahuddin,Muhammad;Slaim,Zabir;Norio,Shiratori.*TCP-charry:new approach for TCP congestion control over satellite IP networks*. computer communication,31,2008,2541-2561.
- 2-cheny,ha;yaw,chen;cheny Chan. *Fast retransmitted and fast recovery schemes of transport protocols: A survey and taxonomy*. computer network,52,2008,1308-1327.
- 3-Elizabet,Rendon;Morales,Joryemata;Junajo,Alins;Josel,Manoz;Oscar,Esparza. *cross-Layer architecture for TCP splitting in the return channel over satellite networks*.IEEE,2009.
- 4-Carlo, Caini; Rosario, Fivrincieli ; Mario,Marchese; Tomaso,Decola; Michel,luglio; Cesare,Roseti; Nedo,Celandroni; Francesco,potrottri. *Transport layer protocols and architectures for satellite networks* .international journal of satellite communications and networking,2006.
- 5-Carlo,Caini ;Rosario Firrincieli ; Daniele,Lacamera. *PEPsal:a performance enhancing proxy desiged for TCP SATELLITE CONNECTIONS* .IEEE,2006.
- 6-N,Thanthry;S,Shingvi ;R,Pendse. *Aircraft data networks and performance enhancement proxies*.IEEE,2006.
- 7-Suranjit,Paul;Md.safiul,Bashar;Md,Sanaullah,Suzan;Chandan,Kumar.*performance analysis of TCP over long-delay path and algorithm for improvement*.4th international conference on electrical and computer engineering ICECE ,2006,19-20 DECEMBER ,Dhaka,Bangladesh.
- 8-DENNIS Roddy ,*Satellite Communication* ,third edition,2001,586 .
- 9-T,decola;T,pecorella;L,ronga.*communication and networking over satellites :SatNEXexperimental activities and testbeds* .International journal of satellite communications 27,2009, 1-33.
- 10-G.MARAL ,*VSAT NETWORKS* ,Second Edition ,England, 2003,271 .
- 11-Liu Feng,Liu Hengna,Zhao Ham.*TCP_ATCA: improve transmissions control algorithm in satellite network* .Chines journal of aeronautics .21,2008, 155-161.
- 12-www.Viasat.com/support .*compact XPEP TCP/IP accelerator,eXterme performance enhancing proxy*.