

# الباب الاول

## 1.1 المقدمة

مشكلة الازدحام [1] هي مشكلة كبيرة شغلت الكثير من الباحثين لعقود من الزمن، حيث أن أي حل لهذه المشكلة في حينه كان يحمل في طياته مشاكل صغيرة تتحول الى مشاكل كبيرة بعد فترة قصيرة وتكون شبكة الانترنت قد تطورت في هذه الفترة الزمنية وظهرت خدمات جديدة وتتحول المشاكل الصغيرة السابقة الى مشاكل كبيرة بالنسبة للخدمات الجديدة ولحل هذه المشكلة الكبيره يتم التفكير في حلول جديدة لتصبح فيما بعد مشاكل كبيره وهكذا [2]. اذا مشكلة الازدحام في الموجهات عويصة وذلك عند إتباع هذه الطريقة في الحل، وذلك لسبب واحد هو انه يتم تصميم الخوارزمية الجديدة لحل مشكلة محددة من مشاكل الخوارزمية القديمة، مثال لذلك تم تصميم خوارزمية الاسقاط المبكر العشوائي RED [3] لحل مشكلة التزامن الذي تعاني منه خوارزمية الاسقاط من آخر الصف Drop-Tail [4] عن طريق الاسقاط العشوائي والذي لتقوم به (اي الاسقاط العشوائي) تحتاج الى صف طويل لضمان عدم تكرار الاسقاط من نفس الاتصال، هذا الصف الطويل كان هو الاساس لمشكلة جديدة هي مشكلة التأخير في الصفوف. اذا تصميم خوارزمية جديدة لحل مشكلة خوارزمية قديمة من غير التفكير في ان هذا الحل هو الحل الشامل لكل مشاكل الخوارزمية القديمة ادى الى ان يتسبب الحل الجديد نفسه في مشكله جديده.

## 2.1 تطور خوارزميات التحكم بالصفوف

وجود صفوف في الموجهات أمر حيوي جداً وذلك لأن طبيعة شبكات تبديل الحزم Packet Switching Network هي الإرسال الانفجاري والذي يعتمد بروتوكول التحكم بالنقل TCP [5].

[6] الذي يمثل الناقل الأكبر للبيانات على شبكة الانترنت [7]، تمثل الذواكر الصفوف التي يتم تخزين الحزم بها وذلك عند ورودها من المرسل في صورة دفقات Bursts [8-10] ليتم فيما بعد توجيهها الى الخرج بالسرعة المناسبة، اذا لم يوجد صف وفي حالة عدم تماثل سرعة الدخل والخرج فإن الحزم الواردة في صورة دفعه سوف لن تمر جميعها، بمعنى أن الحزم الواردة سوف يسقط جزء كبير منها وذلك في حالة ان تكون سرعة خط الدخل أكبر من سرعة خط الخرج، تم اعتماد طريقة القادم اولاً يخدم أولاً FCFS في صفوف الموجهات، هذه الطريقة سهلة وبسيطة وموجودة قبل ومستخدمه منذ زمن بعيد لخدمة الصفوف، اذا للإجابة على السؤال من أين جاءت هذه الخوارزمية والتي فيما بعد سوف تسبب الكثير من المشاكل يقال انها جاءت بصورة طبيعية لأنها الخوارزمية الايسر لخدمة الصفوف، عملت هذه الخوارزمية بكفاءة عالية في خدمة صفوف الموجهات لفترة طويلة من غير اي عيوب فيها وذلك لعدة أسباب منها صغر الشبكة وقلة عدد المستخدمين، مثل مشكلة التزامن وهي قيام الخوارزمية بالاسقاط من كل الاتصالات الواردة في أزمان متقاربة يؤدي هذا الاسقاط الى ان تقوم كل الاتصالات والتي تعرضت لاسقاط بعض حزمها بتقليل معدل ارسالها والذي يتسبب بالاجمال في عدم الاستفادة التامه من معدل الإرسال في خط الخرج نسبة لتراجع بروتوكول التحكم بالنقل TCP عن الإرسال.

### 3.1 الحاجة الى خوارزمية للتحكم بالصف في الموجهات

تحتاج شبكة الانترنت بشده الى وجود الصفوف على الموجهات وذلك لامتصاص الارسال الزائد وارساله فيما بعد [11-14], ولكن مع توسع الشبكة وزيادة عدد المستخدمين زاد اطوال الصفوف وبالتالي زادت الحاجة الى وجود خوارزميه ما تعمل على التحكم بهذا الصف, ولكن مع تنوع الشبكات والخدمات التي تقدمها كان لايد من ايجاد الخوارزمية المناسبه, تختلف وجهات النظر المطلوبه من الشبكة وخدماتها بين مقدمي خدمات الانترنت والمستخدمين, حيث ان مقدمي خدمات الانترنت يهتمون بالاستفاده من سعة القناه بصوره تامه, بالاضافه الى ذلك تقليل التزامن والذي يؤدي الى عدم الاستفاده من سعة القناه ايضا اما العدالة فهم مهتمون بها بقدر اهتمامهم بعدم تحول زبائنهم الى مقدمين لخدمات الانترنت آخرين, اما المستخدمون فأنهم يهتمون بنوعيه الخدمه المقدمه لهم ومن اهم المؤثرات على نوع الخدمه المقدمه الى الزبائن هي التأخير الذي تتعرض له الحزم وبالتالي التأخير المفروض على نوع الخدمه, في هذه الفقره سوف يتم التحدث عن هذه النقاط الاربعه لتوضيح تأثيرها على أداء شبكة الانترنت.

#### 1.3.1 العدالة

على مر السنوات السابقة كان هناك عدة تعريفات للعدالة [15, 16] ولكن بصورة عامة هي مدى قدرة الخوارزمية على توزيع مصادر الشبكة على الاتصالات المارة بها بصورة عادلة [17, 18] وهذا التعريف هو تعريف متشدد لأن حاجة الاتصالات غير متساوية, هناك تعريف آخر ينص على إعطاء كل إتصال الحد الادنى من الموارد والتي تمكنه من العمل بصورة جيدة, تقوم بعض الخوارزميات لتوفير العدالة للاتصالات بحجز جزء من الصف لكل اتصال على حده, انواع اخري

من الخوارزميات تقوم بالاسقاط العشوائي من مجموعة الاتصالات حيث ان عدد الحزم المسقطه الاكبر سيكون من الاتصالات ذات الاستهلاك الاكبر [19] ولكن الحصول على كل الامتيازات غير ممكن تقريبا [20].

### 2.3.1 الاستفادة من عرض النطاق

نصت دراسة سابقة [4] على أنه للإستفادة من عرض النطاق الاستفادة القصوى لابد من أن يكون حجم ذاكرة الموجة مساوى لحاصل ضرب عرض النطاق Bandwidth في تأخير القناة RTT ويسمى حاصل الضرب هذا بحاصل ضرب عرض النطاق في التأخير اختصاراً BDP، في حالة الموجهات الكبيرة والتي يمر عبرها عدد كبير من الاتصالات يكون الناتج ضخماً جداً ويؤدي بالتالي إلى أن يكون حجم ذاكرة الموجه كبيرة جداً، في دراسته لاحقة [21] تم إثبات أن هذا الأمر صحيح فقط في حالة كون عدد الاتصالات قليل جداً (حسب الدرسته ثمانية اتصالات) ولكن في حالة زيادة عدد الاتصالات فإنه يمكن تقليل حجم ذاكرة الموجه بصورة كبيرة، يتم حساب طول الصف في هذه الدرسته [21] بواسطة المعادلة رقم (1-1) :

$$\frac{RTT * C}{\sqrt{n}} \quad (1-1)$$

حيث أن n هي عدد الاتصالات التي تمر عبر الموجه وذلك للحصول على نفس القدر تقريباً من الاستفادة من عرض النطاق التي نصت عليها الدرسته السابقة، و المعامل c هو عرض النطاق [4]. خوارزميات الجدولة وخوارزميات إدارة الصف تعتمد على طول الصف وذلك للتخلص من مشكلة عدم الاستفادة من عرض النطاق بجعل طول صف الموجه كبير جداً للحصول على أكبر قدر ممكن من الحزم ومنع الاسقاط.

### 3.3.1 التأخير

هناك اسباب مختلفه للتأخير مثل التأخير الناتج من إنتشار الموجه الكهرومغناطيسية للإشارة من المرسل الى المستقبل وهو تأخير ثابت لا يمكن التحكم به، اما التأخير المقصود والذي يهتم به البحث فهو تأخير الحزمه بعد دخولها الى صف الموجه والى حين ان يقرر الموجه ارسالها فى اتجاه الخرج، وهو تأخير يمكن التحكم به عن طريق طول الصف او الارسال من غير ترتيب للرسائل حسب دخولها، ولم يكن هناك عدد كبير من الحزم لتنتج تأخير يمكن ملاحظته حتى، بل كان التأخير مجرد عامل يتم مراعاته عند تصميم اي خوارزمية من النوعين السابقين، ولكن مع الزيادة الكبيره في عدد المستخدمين فى الاعوام الاخيره ومع انخفاض اسعار الذواكر تم إضافة الكثير منها الى الموجهات لتصبح الصفوف باطوال كبيره ساهمت فى تحويل التأخير الذى كان عاملاً يتم مراعاته الى مشكلة كبيرة جداً يجب التفكير الجاد فى حلها.

يؤثر التأخير فى نوعي البروتوكولات بروتوكول حزم المستخدم UDP [22] وبروتوكول التحكم بالنقل TCP [23, 24], تأثير التأخير على بروتوكول التحكم بالنقل TCP يتمثل فى ان هذا البروتوكول يعتمد على مؤقت زمني ينتهي اذا لم يستلم حزمه اقرار ACK على الحزمة التى ارسلها سابقاً والتأخير الموجود على الصف يضاف الى التأخير الثابت وهو تأخير الانتشار والذي قد يكون اقل من زمن المؤقت الزمني، اما تأثير التأخير على بروتوكول حزم المستخدم UDP فهو ظاهر جداً حيث ان بروتوكول حزم المستخدم UDP هو بروتوكول نقل لتطبيقات الزمن الحقيقي من صوت وصورة وهذه التطبيقات لا تتحمل أي تأخير إضافي حيث يسبب هذا التأخير الاضافي فى عدم تمكن الخدمة من الاستمرار.

### 4.3.1 التزامن

يعتمد بروتوكول التحكم بالنقل TCP على خوارزمية التحكم بالازدحام [25] والتي تنص على ان يبدأ الارسال بحزمه واحده في عملية الارسال الاولى ثم يبدأ التضاعف الاسي الى حزمتين ثم اربعة حزم ثم ثمانية وهكذا, الى ان نصل عتبه معينه تبدأ بعدها زياده خطيه, عند ازدحام اي موجه في مسار الاتصال فإنه يقوم باسقاط حزمه واحده او عدة حزم من هذا الاتصال, عملية الاسقاط هذه تؤثر في الاتصال المعني حيث يقوم هذه الاتصال وعند سقوط هذه الحزمه بانقاص معدل ارساله الى النصف وعند تكرار الاسقاط يتكرر انقاص معدل الارسال في كل مره الى النصف, بالرجوع الى موضوع التزامن فإنه يعني ان يقوم الموجه وعند ازدحامه بالاسقاط من كل الاتصالات التي تمر عليه وبالتالي تقوم كل هذه الاتصالات بانقاص معدل ارسالها الى النصف [26] وبالتالي يقل معدل الارسال للقناة التي يتحكم بها الموجه وهي خاصيه غير جيده حيث انه من المفيد الاستفاده من كل الحيز المتاح على تلك القناة, وبالتالي يجب منع تزامن الاسقاط من كل الاتصالات التي تمر اي لابد ان يتم الاسقاط من اتصالات محده في كل مره وعدم الاسقاط من كل الاتصالات حتى لا تقل الاستفادة من عرض النطاق.

### 4.1 التأخير وتوسع الشبكه

كما هو معلوم فإن شبكة الانترنت لم تكن بهذا الحجم التي هي عليه اليوم بل كانت شبكه صغيره, توسعت شبكة الانترنت بصوره كبيره في السنوات الاخيره وذلك لتنوع الخدمات التي تقدمها للمستخدمين, لم يكن التوسع في شبكة الانترنت في عدد المستخدمين فقط بل كان ايضا التوسع على

مساحة الارض كلها, اي ان كل الدول والقارات اصبحت في شبكه واحده اسمها شبكة الانترنت, ادى توسع شبكة الانترنت لتشمل كل الدول الى زيادة التأخير, عموما لا يمكن التحكم بتأخير الانتشار حيث انه تأخير الاشارة الكهرومغناطيسيه من المرسل الى المستقبل, لم يكن تأخير الانتشار ذو اهميه عندما كانت شبكه الانترنت صغيره ولكن مع زيادة عدد المستخدمين ظهر نوع جديد من التأخير وهو التأخير الذي تفرضه الموجهات على الحزم لحين ارسالها نتيجة للازدحام, وذلك كله بسبب زيادة عدد المستخدمين, بما ان تأخير الانتشار لا يمكن التحكم به بالتالي كان لابد من الاتجاه نحو النوع الاخر من التأخير وهو تأخير الصفوف او تأخير الازدحام على الموجهات وذلك للتعامل معه بغرض تقليله الى ادنى حد ممكن وذلك للسماح للتطبيقات العمل بصورة جيده.

## 5.1 مشكلة تضخم الصفوف

تضخم الصفوف Bufferbloat [27-29] هو زيادة طول الصف بحيث يمنع تماما سقوط الحزم, بالرغم من ان وجود الصف مطلوب بشده في شبكات تبديل الحزم حيث ان وجود الصف يساعد الشبكة على امتصاص الارسال الزائد ثم ارساله فيما بعد ولكن زيادة طول الصف عن الحد المعقول يؤدي الى ظهور مشكله جديده هي مشكلة التأخير [30], في السنوات الاخيره ادى التطور الصناعي الى انخفاض اسعار الدواكر وبالتالي تم اضافة الكثير من هذه الدواكر الى الموجهات لزيادة طول الصف وذلك لمنع الحزم من السقوط [31] باعتبار ان سقوط الحزم هو الذي يؤدي الى تدهور عمل الشبكة ولكن ادى ذلك الى نتيجة عكسيه حيث ادى منع السقوط الى تراكم الحزم في هذه الصفوف وبالتالي زيادة التأخير بصورة كبيره [32], يمكن حل مشكلة التأخير الناتج عن تضخم الصفوف

ويكل بساطه عن طريق تقليل طول الصف الذي تتم اضافته للموجه, ولكن معرفة طول الصف المناسب علي الموجه عملية غير سهله تتمثل في معرفة مجموعه من المحددات والعوامل التي تحكم عمل الموجه. وبما ان نفس الموجه المصنع من جهة محدهه يمكن ان يعمل بصورة جيده في مكان ما ولكنه لا يعمل بنفس الكفاءة في مكان آخر وذلك نسبة لاختلاف بيئة عمل كل منهما على كل منهما في تلك البيئه, بالتالي كان لابد من إيجاد طريقه ما تتمكن من معرفة الطول المناسب للصف ليعمل عليه الموجه وتكون هذه الطريقه معزوله عن كمية الذاكره [33] المضافه الى الموجه كصف.

## 6.1 بيان المشكله

توسعت شبكة الانترنت بصورة كبيره جدا في السنوات الاخيره من حيث المساحه الجغرافيه او من حيث عدد المستخدمين, ادى هذا التوسع الى ظهور مشكله التأخير, يتمثل التأخير في تأخر وصول البيانات الى الهدف النهائي وهذا التأخير يعني ان البيانات غير ذات اي فائده بعد استلامها, بصورة عملية يؤثر هذا التأخير في بروتوكولي نقل البيانات بروتوكول التحكم بالنقل TCP و بروتوكول حزم المستخدم UDP حيث ان بروتوكول التحكم بالنقل TCP يعتمد على مؤقت زمني قيمته 200ms و بروتوكول حزم المستخدم UDP المخصص لنقل بيانات الزمن الحقيقي من صوت وصوره والتي ترتبط بحدود زمني معينه للوصول مرتبطه بحواس الانسان من سمع وبصر, هذه الخدمات ذات تاخير يتراوح ما بين 150ms و 250ms [34], كانت شبكة الانترنت قبل ان تتوسع محصوره في مساحه جغرافيه محدوده وبالتالي فأن التأخير الذي يحدث في الصفوف يعوض من التأخير المسموح به وهو 200ms لبروتوكول التحكم بالنقل TCP و 150ms-250ms لبروتوكول حزم المستخدم

UDP ولكن مع توسع شبكة الانترنت في المساحة الجغرافيه فأن التأخير الكلي المسموح به للبروتوكولين اصبحت تستهلك منه قيمه كبيره في الوصول الى الهدف النهائي اي ان قيمة التأخير RTT اصبحت كبيره واصبح هناك هامش تأخير قليل مسموح به في صفوف الموجهات, العامل الاخر في زيادة التأخير على شبكة الانترنت هو زيادة عدد المستخدمين, حيث ان شبكة الانترنت وخدماتها اصبحت تقريبا بحجم شبكة الهاتف والتي هي بحجم العالم اجمع, عدد المستخدمين الكبير هذا ادى الى تراكم الحزم في صفوف الموجهات في انتظار الارسال, حيث ان التراكم ينتج من ان قدرة خطوط نقل البيانات اقل من كمية البيانات التي يتناقلها المستخدمون, بعد تزواج هذين العاملين وهما مساحة شبكة الانترنت الجغرافيه الكبيره وعدد المستخدمين الكبير نتج عنهما قيمة تأخير كبيره اصبحت تهدد عمل الانترنت ومستقبلها بصوره جديده.

على مر السنوات والعقود السابقه كانت هناك مشاكل تواجه حركة مرور ونقل البيانات, هذه المشاكل هي مشكلة التزامن والتي تواجه بروتوكول التحكم بالنقل TCP ومشكلة العدالة التي تواجه بروتوكولي نقل البيانات بروتوكول التحكم بالنقل TCP و بروتوكول حزم المستخدم UDP, تم تطوير خوارزميات عديده لحل هذه المشاكل, اتبعت هذه الخوارزميات الطريقه التقليديه وهي حل المشكله باعتبار ان هناك تزامن او عدم عداله تواجه الحزم قبل ان تصل الى الهدف النهائي. وبما ان التأخير اصبح هو المشكله الاساسيه التي يجب على خوارزميات التحكم بالازدحام الاهتمام به, وبعد ان تزايد التأخير على شبكة الانترنت كان لابد من ايجاد خوارزميه يكون التأخير هو محور الاهتمام الاساسي لها, باعتبار انه العامل الاشد تأثيرا على شبكة الانترنت اليوم.

## 7.1 إهتمام البحث

يعتبر البحث ان المشكله الاساسيه في شبكة الانترنت هي التأخير وخاصة في الاتصالات البعيده, لذلك الافضل لشبكة الانترنت والمستخدمين على حد سواء ضمان عدم تراكم الحزم في صفوف الموجهات, لذلك يهتم البحث بمشكلة التأخير والذي اكدت الدراسات [27] على انه المشكله الحاليه وانه ايضا سيتفاقم في السنوات المقبله بصوره اكبر, اسقاط الحزم بكثافه هو الحل الوحيد لضمان عدم تراكم الحزم ولكنه يؤدي الى تدني مستوى الخدمه المقدمه للمستخدم بالرغم من ذلك ليس هناك حل آخر يمكن اللجوء اليه فكمية البيانات المرسله عبر الشبكة اكبر من قدرة الشبكة ومن قدرة الموجهات على الاستقبال والمعالجه والتوجيه والارسال.

## 8.1 طريقة الحل

يتكون التأخير من قيمتين هما تأخير انتشار الاشاره من المرسل الى المستقبل وهي قيمه تضاعفت نسبة للتوسع الجغرافي لشبكة الانترنت وهذه القيمه لا يمكن التحكم بها والقيمه الاخرى هي التأخير المفروض على الحزم في صفوف الموجهات نتيجة لزيادة عدد المستخدمين وهذ القيمه يمكن التعامل معها والتحكم بها بهدف تقليل قيمة التأخير الكلي, اذا الخيار الوحيد لتقليل التأخير هو التحكم بالتأخير في صفوف الموجهات, من الخوارزميات التي سلكت هذا الطريق خوارزمية التحكم بتأخير الصف Codel [35] والتي لديها عوائق جديه تمنع تطبيقها بصوره عمليه في موجهات اليوم, اذا سيكون اهتمام هذا البحث هو ايجاد خوارزميه جديده تسلك مسلك خوارزمية التحكم بتأخير الصف Codel في وضع التأخير موضع الاهتمام, اي تقليل التأخير الى ادنى حد ممكن بحيث ان دخول

الحزمه يعني ارسالها او اسقاطها فورا, العامل الاخر الذي يجب الاهتمام به هو امكانية التطبيق العملي لهذه الخوارزمية على الموجهات والتي ليس لديها القدره على تطبيق خوارزمية التحكم بتأخير الصف Code1 نسبة لحاجتها لقدرة معالجه عاليه تشغل معالج الموجه من دوره الاساسي في توجيه الحزم الوارده.

## 9.1 حدود البحث

بافتراض ان الشبكة توسعت جغرافيا وبالتالي زاد التأخير الزمني RTT لجزء من الاتصالات ولم تزيد هذه القيمه لكل الاتصالات, بالتالي فان تطبيق خوارزميه تعتمد تماما على قياس التأخير للتحكم في مرور البيانات يمكن ان يكون غير صحيح لكل الاتصالات فبعض الاتصالات العالميه تستهلك زمن كبير من التأخير الكلي في الوصول الى الهدف النهائي وبعض الاتصالات الاخرى محليه تستطيع ان تتحمل بعض التأخير في الموجهات وبالتالي فان خوارزمية قياس الزمن المقترحه وعند تطبيقها يمكن ان تضر بالاتصالات المحليه حيث ستقوم باسقاط عدد كبير من حزمها والتي كان من الممكن ان تتحمل تأخير اكبر من التأخير المفروض بواسطة الخوارزمية الجديده. بالتالي فانه يمكن استعمال الخوارزمية الجديده في الموجهات المفصلية باعتبار انها تقوم بتوجيه الاتصالات البعيده.

## 10.1 بنية الرسالة

تحتوي هذه الرسالة على ستة فصول مقسمة على النحو التالي

الباب الثاني : يحتوي هذا الباب الدراسات السابقة في موضوع خوارزميات التحكم بالازدحام بصورة عامه ابتداء من الخوارزمية الاولى التي عملت في مجال التحكم بالازدحام وهي خوارزمية الاسقاط من آخر الصف Drop-Tail, تم تقسيم الخوارزميات العامله في مجال التحكم بالازدحام الى خوارزميات الجدوله وخوارزميات ادارة الصفوف وتوضيح الفروقات بينها, تم اعادة تقسيم الخوارزميات الى خوارزميات تعتمد على طول الصف وخوارزميات اخرى تعمل على قياس التأخير.

الباب الثالث : تم تخصيص الباب الثالث للتحدث عن الخوارزمية الاكثر انتشارا من بين خوارزميات التحكم بتأخير الصف وهي خوارزمية التحكم بتأخير الصف Code1, حيث يشتمل الباب علي بيان طريقة عملها والمشاكل والمعوقات التي ادت الى عدم تطبيقها عمليا.

الباب الرابع : يحتوي هذه الباب على تحليل لاداء الخوارزمية المقترحه في هذا البحث وهي خوارزمية التقدير الذاتي لطول الصف auto وهي الخوارزمية التي تعتمد على قياس طول الصف المناسب للعمل عليه, يتكون هذا الباب من ثلاثة اقسام القسم الاول يتحدث عن قدرة الخوارزمية على تقدير عدد الاتصالات الحيه الماره في الموجه, والقسم الثاني يتحدث عن قدرة الخوارزمية على تقدير طول الصف المناسب للعمل عليه من غير فقدان الاستفاده من عرض النطاق throughput والقسم الثالث يوضح تأثير عدد الاتصالات على طول الصف في خوارزمية التقدير الذاتي لطول الصف auto وايضا تأثير العدالة بين الاتصالات في التقدير الذاتي لطول الصف auto بعدد الاتصالات.

الباب الخامس : يحتوي هذه الباب على نتائج المقارنات العملية, حيث تم استخدام المحاكى ns-2 [36] على مجموعه من الخوارزميات التي تعتمد على طول الصف والخوارزميات التي تعتمد على قياس التأخير ومنها خوارزمية التقدير الذاتي لطول الصف, تم اجراء مجموعه من التجارب لمعرفة اداء كل من هذه الخوارزميات وتحديد الخوارزمية الاكثر كفاءة في خدمة شبكة الانترنت.

الباب السادس : يحتوي هذه الباب على خاتمه للرساله, ويقوم ايضا بتغطية محدودية خوارزمية التقدير الذاتي لطول الصف auto في اداء عملها على شبكة الانترنت, يشتمل الباب ايضا على امكانية تطوير الخوارزمية واتجاهات البحث في هذه المجال, بالاضافه الى التوصيات.

## 11.1 الخلاصه

تناول هذا الباب مقدمه لهذا البحث وهي الحاجه الى خوارزمية للتحكم بالازدحام في الموجهات, ووضح ان هناك عدة معاملات يمكن عن طريقها التحكم بالازدحام وهي العدالة, الاستفادة من عرض النطاق, التأخير والتزامن, تناول الباب ايضا مسيبيات توسع شبكة الانترنت ورد ذلك الى زيادة عدد المستخدمين والتوسع الفيزيائي في الشبكه, بعد ذلك حددت مشكلة البحث الاساسيه وهي إمتلاء الصفوف او زيادة الصفوف عن الحد الطبيعي وهو الاداء المستقر لشبكة الانترنت الى زيادة التأخير الكبيره والتي سميت فيما بعد بتضخم الصفوف, بعد ذلك تم تحديد عامل التأخير لحل مشكلة تضخم الصفوف.

## الباب الثاني

### 1.2 الاعمال السابقة

يتناول هذا الباب الخوارزميات المستخدمة في مجال التحكم بالازدحام، سنتناول اولاً بروتوكول التحكم بالنقل TCP وطريقة عمله، هذا البروتوكول يستخدم لنقل حوالي 90% من البيانات عبر شبكة الانترنت. يؤدي الازدحام إلى تراكم الحزم في صف الموجه، هذا التراكم يؤثر على التطبيقات المعتمدة على البروتوكولين بروتوكول حزم المستخدم UDP وبروتوكول التحكم بالنقل TCP، بما أن بروتوكول التحكم بالنقل TCP يعتمد أساساً على سقوط الحزم وذلك لعدة أسباب أولها ان سقوط الحزم يؤدي الى ان يقوم بروتوكول التحكم بالنقل TCP بالحصول على الحيز المتاح له من القناة وذلك بصورة عادلة وايضاً يستطيع بروتوكول التحكم بالنقل TCP ان يستعيد عافيته بسرعة بعد سقوط حزمه واكتشافه لهذا لسقوط حيث انه يلجأ في هذه الحالة الى خوارزمية الارسال السريع لاستعادة الارسال إما نتائج تأخير الحزم فهي اسوأ على بروتوكول التحكم بالنقل TCP حيث أن التأخير يجعل بروتوكول التحكم بالنقل TCP يفقد معدل الارسال السابق ورجوعه الى خوارزمية البداية البطيئة Slow Start [25]، وبما أن التأخير اسوأ على بروتوكول التحكم بالنقل TCP فهو اسوأ على كل التطبيقات القائمة عليه، أما تأثير التأخير على تطبيقات بروتوكول حزم المستخدم UDP فهو تأثير كبير مباشر حيث أن هذا البروتوكول هو بروتوكول مخصص لنقل بيانات الزمن الحقيقي، وبالتالي فإن تراكم الحزم في ذاكرة الموجه يؤدي إلى تأخيرها وهذا التأخير يؤدي إلى تأثير على كل تطبيقات الزمن الحقيقي مثل نقل الصوت في الزمن الحقيقي حيث ان تأخير مقداره 150ms هو تأخير مقبول وتأخير مقداره 250ms هو تأخير تعتبر معه الخدمة غير جيدة، التأخير

عند تضخم الصفوف قد يصل إلى أضعاف هذه القيم وبالتالي إستحالة تقديم هذه الخدمات بصورة مقبولة.

## 2.2 خوارزمية التحكم بالازدحام

تحتوى خوارزمية التحكم بالازدحام [25] التي تحكم عمل بروتوكول التحكم بالنقل TCP على جزئيتين مهمتين هما

- خوارزمية البداية البطيئة

- خوارزمية سرعة اعادة الارسال وخوارزمية الاستعادة السريعة

### 1.2.2 خوارزمية البداية البطيئة

باستعمال خوارزمية البداية البطيئة فان طرف بروتوكول التحكم بالنقل TCP المرسل يبدأ بنافذة ازدحام تحتوى على حزمه واحدة ثم يبدأ فى زيادة نافذة الازدحام بصورة أسية [25]، عندما تصل نافذة الازدحام للعتبه threshold فإنه يوقف الزيادة الاسية لنافذة الازدحام ويتحول إلى الزيادة الخطية للتأكد من انه يستخدم كل الحيز المتاح من عرض النطاق. يمكن إستعادة الحزمة المفقودة فقط بواسطة خوارزمية سرعة اعادة الارسال او خوارزمية البداية البطيئة بعد إنتهاء فترة إعادة الارسال Retransmission Timeout.

### 2.2.2 خوارزمية سرعة إعادة الإرسال وخوارزمية الاستعادة السريعة

يقوم المستقبل بارسال حزمه اقرار للحزمه ذات الرقم التسلسلي الاعلى والتي استلمها سابقاً [37]، فإنه عندما يستقبل حزمه خارج الترتيب يقوم فى هذه الحالة بتوليد مجموعة حزم إقرار للحزمه التي

تحمل الرقم التسلسلي الاعلى, تسمى هذه الحزم بحزم الاقرار المكرره وعندما يستقبل المستقبل ثلاثة حزم اقرار مكررة, فإن خوارزمية سرعة إعادة الإرسال تستنتج ان هناك حزمه قد فقدت, وتفترض ان الحزمه المفقودة تبدأ برقم متسلسل يقع مباشرةً بعد الرقم التسلسلي الذي تم إقراره, ثم تقوم بإرسال تلك الحزمه بالاضافة الى ان العتبه ونافذة الازدحام يتم إنقاصها الى نصف نافذة الازدحام قبل الدخول فى حالة اعاده الارسال بواسطة خوارزمية سرعة إعادة الإرسال. تحكم خوارزمية الاستعادة السريعة الطريقة التى يتم بها تعديل نافذة الازدحام والعتبه بعد انتهاء خوارزمية إعادة الإرسال من عملها وبالتالي فإن المرسل يقلل معدل ارساله ويتحول الارسال للزيادة الخطيه بدلاً من الزيادة الاسية لنافذة الارسال.

### 3.2 المعاملان طول وتأخير الصف

تتحدث هذه الفقرة عن معاملين مهمين تعتمد عليهما كل خوارزميات التحكم بالازدحام هما طول الصف وتأخير الصف, حيث ان هناك مجموعه كبيره من الخوارزميات تعتمد على طول الصف وايضا مجموعه كبيره اخرى من الخوارزميات تعتمد على تأخير الصف.

#### 1.3.2 طول الصف

طول الصف هو سعة ذاكرة الموجه من الحزم, وطول الصف يمكن أن يكون لا متناهيا نظريا مع خوارزميات مثل خوارزمية الاسقاط من آخر الصف Drop-Tail ويمكن ان يكون محدوداً مع خوارزميات مثل خوارزمية الخنق CHOKE [38], خوارزمية الصفوف العادل FRED وخوارزمية الاسقاط المبكر العشوائي RED تم إعتقاد طول الصف للتحكم بالازدحام كان للأسباب التاليه: أولاً

ضرورة وجود الصف جاءت من طبيعة عمل شبكة الانترنت حيث تعمل الشبكة على مبدأ تبديل الحزم Packet Switching لذا زودت شبكة الانترنت بالصفوف على الموجهات لضمان تخزين الحزم وارسالها، حيث ان البروتوكول المسيطر هو بروتوكول التحكم بالنقل TCP وطبيعة عمله تعتمد ارسال البيانات فى صورة دفعات كبيره متقطعة من الحزم، وهذه الدفعات الكبيره تحتاج بالضرورة الى ذاكرة فى الموجه وذلك لتخزينها وضمان عدم سقوطها وارسالها فيما بعد.

كانت الخوارزمية المستخدمه فى الشبكات هي خوارزمية الاسقاط من آخر الصف Drop-Tail مع طول صف فى الموجه محدود، جاءت هذه المحدودية نسبة لقله عدد المستخدمين و لارتفاع تكلفة الذاكر وبالتالى كانت سعة الذاكرة قليلة وأدى ذلك لقصر طول الصف فى خوارزمية الاسقاط من آخر الصف Drop-Tail، مع زيادة عدد المستخدمين وزيادة حجم الذاكر على الموجهات ظهرت بعض المشاكل مع هذه الخوارزمية وهي تزامن الاسقاط وعدم العادلة بين الاتصالات التى تمر عبر الموجه. للتغلب على مشاكل تزامن الاسقاط وعدم العادلة تم إستنباط خوارزميات اخرى جديدة اقترحت هذه الخوارزميات للتغلب على هاتين المشكلتين بواسطة الاسقاط العشوائى للرسائل، الاسقاط العشوائى يعتمد على إختيار حزمه بصورة عشوائية للاسقاط ولاختيار مثل هذه الحزمه لابد ان يكون هناك حزم اخرى للاختيار من بينها حتم كل ذلك وجود صف، وهذا الصف لابد ان يكون طويل جداً فى حالة الحاجة لضمان العادلة وايضاً لابد ان يكون طويل لضمان التخلص من مشكلة تزامن الاسقاط.

### 1.1.3.2 خوارزميات التحكم بالازدحام التي تعتمد على طول الصف

مع بداية عمل شبكة الانترنت لم يكن هناك ازدحام في الشبكة وذلك لصغر حجمها، وبالتالي كان التركيز على صياغة البروتوكولات وتحسينها وكيفية عمل التطبيقات على هذه البروتوكولات بالإضافة الى زيادة عدد البروتوكولات وتنوعها. إكتفى المصممين الاوائل لشبكة الانترنت بوضع ذاكرة في صورة صف وحيد للموجه وهي الخوارزمية المعروفة حالياً بخوارزمية الاسقاط من آخر الصف Drop-Tail هذه الخوارزمية عبارة عن طريقة بسيطة لعمل صف حيث تدخل الحزم من الخلف وتخرج من الامام ويتم خدمة الحزم الاول فالاول بطريقة القادم اولا يخدم اولا (FCFO) لم تؤدي هذه الطريقة لحدوث اي مشاكل نسبة لصغر حجم الشبكة بالإضافة لقصر طول الصف نفسه حيث ان هذا الصف عبارة عن ذاكرة ولم تكن الذاكر رخيصة ليتم إضافة اكبر حجم منها للموجه، وبالتالي تسبب الامرين السابقين في عدم حدوث مشاكل للموجه بل كانت المشاكل تنتج من قصر طول الصف بسبب زيادة السرعات على الشبكات لبعض الخطوط بالإضافة لندرة الذاكر المكونة لهذه الصفوف. بعد ذلك بسنوات زادت كمية الذاكر على الموجهات بسبب انخفاض اسعارها وبالتالي اصبحت الصفوف طويلة جداً، والناظر اليها للوهلة الاولى يظن ان ذلك سيؤدي لعدم حدوث مشاكل بسبب عدم سقوط الحزم، حيث ان سقوط الحزم من الموجهات كانت هي المشكلة الاساسية التي تواجه بروتوكولات نقل البيانات على الشبكة، بعد ذلك تبين ان سقوط الحزم في الشبكة يعتبر أمر حيوي لبروتوكولات (سقوط حزمه من بروتوكول التحكم بالنقل TCP سيؤدي إلى إعادة ارسالها بواسطة خوارزمية سرعة إعادة الارسال اما اذا تأخرت ولم يعرف مصيرها فسيعاد ارسالها ولكن بعد إنتهاء زمن وصولها وهناك فرق كبير بين الحالتين حيث ان الحالة الاولى طريقة ترميمية [39] لا

تستغرق زمن طويل اما الطريقه الثانية تستغرق زمن طويل جداً يحاول دائماً بروتوكول التحكم بالنقل TCP تجنبه في إعادة الإرسال) نقل البيانات وذلك ليستمر عملها بصورة فعالة.

### 2.3.2 التأخير على الصف

معامل التأخير على الصف، بعد معاناة شبكة الانترنت لسنوات من مشاكل في الأداء تم التنبيه الى ان هذه المشاكل ناتجة عن التأخير الكبير الذي تتعرض له الحزم في الصفوف والتي تحولت الى صفوف طويلة تمنع الحزم من السقوط، ولم يكن في الاعتبار أن سقوط الحزم هو ظاهرة طبيعية وإيجابية لعمل شبكة الانترنت.

المقصود بالتأخير هنا زمن بقاء الحزمه في صف الموجه، اي زمن بقائها في الصف في إنتظار ان يتم ارسالها الى وجهتها، هذا الزمن قد يطول وقد يقصر اعتماداً على عدد الحزم التي يتم استلامها وعدد الحزم التي يتم ارسالها من منطقة عنق الزجاجة حيث تتميز منطقة عنق الزجاجة بسرعة دخل اعلى من سرعة خرج الحزم، في منطقة عنق الزجاجة لا بد من وجود صف لتخزين الحزم الواردة بسرعة كبيرة ومحاولة ارسالها الى الخرج و ضمان عدم سقوط اي حزمه من هذه الحزم الواردة. وبالتالي يمكن إنشاء صف طويل جداً لضمان استلام كل الحزم الواردة وعدم سقوط اي منها ثم محاولة ارسالها فيما بعد للخرج حسب السرعة المتاحة. قد يبدو ان هذا الامر جيد اي عدم اسقاط اي حزمه وارده ولكن عند المستقبل النهائي لهذه الحزم هناك وقت محدد لاستلام الحزم وبعد انتهاء هذا الزمن تكون هذه الحزم غير ذات أهمية والاسواء من ذلك هو ان هذه الحزم تتسبب في ضياع حيز من التخزين [40] والارسال في الشبكة كان يمكن ان يستفاد منه لخدمه حزم اخرى. من هذه النقطة جاءت فكرة اعتماد التأخير كعامل حاسم للتحكم بالازدحام في بعض الخوارزميات الجديدة.

يقوم الموجه فى هذه الطريقة الجديدة بالتشديد على ان يكون تأخير اي حزمه فى الصف اقل مايمكن وذلك لضمان وصول هذه الحزمه فى الزمن المناسب الى المستقبل النهائي وإلا سيتم اسقاطها.

## 4.2 خوارزميات التحكم بالازدحام

تم تقسيم خوارزميات التحكم بالازدحام على الموجه الى نوعين هما خوارزميات الجدوله و خوارزميات إدارة الصف ولكن بعد إكتشاف ظاهرة تضخم الصف Bufferbloat يمكن إعادة النظر فى هذا التقسيم، حيث يمكن ضم كل الخوارزميات فى النوعين اعلاه فى نوع واحد شامل يعتمد على طول الصف، اما النوع الثاني فهو النوع الذى يحاول حل مشكلة تضخم الذاكرة والذى يعتمد على التأخير فى الصف، اي تأخير كل حزمه بذاكرة الموجه وبالتالي يكون التقسيم الجديد هو الخوارزميات المعتمدة على طول الصف والخوارزميات المعتمدة على التأخير فى الصف.

### 1.4.2 الخوارزميات التي تعتمد على طول الصف

خوارزميات طول الصف هي الخوارزميات التى لاتقوم بالاسقاط من الصف الا اذا تجاوز طول معين ومن هذه الخوارزميات التي تعتمد بصورة واضحة على طول الصف للإسقاط خوارزمية الاسقاط من آخر الصف Drop-Tail حيث لن يتم اسقاط اي حزمه الا اذا إمتلأ الصف تماماً والخوارزمية الاخرى هي خوارزمية الاسقط البكر العشوائي RED حيث يتم الاسقاط اذا تجاوز طول الصف حد معين، يتم الاسقاط فى هذه الخوارزمية بطريقتين المرة الأولى اذا تجاوز طول الصف العتبه الدنيا  $min_{th}$  حيث يتم الاسقاط بصورة عشوائية والمرة الثانية يتم الاسقاط اذا تجاوز طول الصف العتبه العليا  $max_{th}$  وذلك بصورة الزامية، هناك خوارزميات اخرى تعتمد على طول الصف

للتحكم بالازدحام منها خوارزمية الاسقاط المبكر العشوائي العادل FRED وخوارزمية الخنق CHOKE حيث أن هذه الخوارزميات هي إشتقاق من خوارزمية الاسقاط المبكر العشوائي RED وتعتمد على نفس الحدين للإسقاط العشوائي و الاسقاط الالزامي ولكن مع بعض الإختلافات. في هذا الجزء يتم توضيح ان التقسيم الجديد هو تقسيم صحيح يمكن إعماده حيث سنأتي بعدد من الأمثلة لتوضيح ان كل هذه الخوارزميات تعتمد على طول الصف.

#### 1.1.4.2 خوارزميات الجدولة

هي الخوارزميات التي صممت لحل مشكلة عدم العدالة، وهي خوارزميات تتعامل مع كل اتصال منفرد مار عبر القناة وتحتاج الى قدر كبير من الذاكرة يتناسب مع عدد الاتصالات الماره عبر القناة وتحتاج ايضاً الى قدر كبير من السرعة في المعالجة لأنها خوارزميات معقدة.

#### 1.1.1.4.2 خوارزمية الصفوف العادلة

تتطلب هذه الخوارزمية [41, 42] تقسيم كل ذاكرة الحزم الى صفوف منفصله لكل مدخل على الموجه وكل واحد من هذه الصفوف المنفصله يحتوى على حزم لإتصال واحد فقط، بعد ذلك يتم وضع الحزم من كل صف منفصل على خط الخرج بواسطة مجدول يأخذ حزمه من كل صف منفصل بصورة دائرية تعتمد هذه الخوارزمية بصورة واضحة على طول الصف، حيث لا بد من وجود صف طويل كفاية لتخزين عدد متساوي من الحزم لكل إتصال يمر بالموجه في هذه اللحظة، وبما ان عدد الاتصالات التي تدخل غير معروفه بالتالي فان طول الصف لا بد ان يكون كبيراً والا فان اتصالاتا ما سوف يدخل ولن يجد له حيزاً في الصف.

#### 2.1.1.4.2 خوارزمية عدم التسجيل العادلة

يتم تقسيم الموجهات في هذه الطريقة إلى قسمين هما موجهات الاساسيه Core router وموجهات الاطراف Edge router [43]. موجهات الأطراف تحتفظ ببيانات لكل إتصال يمر عليها ثم تقوم بتقدير معدل الوصول لكل إتصال بصورة منفردة، تقوم بعد ذلك بوضع هذه التقديرات (تقديرات معدلات الوصول للاتصالات) في ترويسة الحزم وتمررها الى الموجهات الاساسيه، الموجهات الاساسيه لا تحتفظ ببيانات لكل إتصال ولكن عند حدوث إزدحام تقوم بإسقاط الحزم بناءً على تقديرات معدلات الوصول الموجودة في ترويسة الحزم والممررة من موجهات الاطراف. بما ان موجهات الاطراف تحتفظ ببيانات لكل إتصال يمر عليها بالتالي لابد من وجود صف طويل وبالتالي هذه الخوارزمية هي خوارزميه معتمده على طول الصف بصورة واضحه.

#### 3.1.1.4.2 خوارزمية الاسقاط المبكر العشوائي العادلة FRED

تعمل هذه الخوارزمية بطريقة مشابهه لطريقة عمل خوارزمية الاسقاط المبكر العشوائي RED ولكن مع بعض الاضافات تقوم خوارزمية FRED [44] بإنشاء معاملين جديدين هما  $\max_q$  ,  $\min_q$  حيث يحددان العدد الاصغر والاكبر لعدد الحزم الذي يسمح بتخزينه لكل اتصال. تقوم خوارزمية FRED ايضا بإنشاء المعامل العام avgcq وهو عبارة تقدير لمتوسط عدد الحزم في ذاكرة الموجه لكل اتصال. الاتصالات التي لها عدد اقل من avgcq من الحزم في ذاكرة الموجه مفضله اكثر من الاتصالات التي لها عدد حزم اكبر من avgcq، خوارزمية FRED ايضا تحدد الحزم المخزنة حالياً لكل إتصال qlen. وأخيراً هناك أيضاً معامل يسمى عدد الضربات strike لكل إتصال على حده، والذي يحدد عدد المرات التي يفشل فيها الاتصال المعنى في الاستجابة لاشارة الازدحام، حيث تقوم

FRED بمعاقبه الاتصالات ذات عدد الضربات strike الاكبر, من الواضح ان هذه الخوارزمية لن تعمل بصورة جيدة اذا لم يتوفر لديها صف طويل حيث انها تعمل على ان يكون لكل اتصال صف منفصل وبالتالي صف طويل جدا ومتناسب مع عدد الاتصالات, يأتي هذا التعقيد بالاضافة الى طول الصف لرغبة الخوارزمية FRED في خدمة الاتصالات بصورة عادله تماما.

#### 2.1.4.2 خوارزميات ادارة الصفوف

صممت هذه الخوارزميات لحل مشكلة التزامن والتعقيد التي تعاني منها خوارزميات الجدوله تمتاز هذه الخوارزميات بالبساطة حيث انها تتعامل مع مجموعة الاتصالات المارة عبر القناة ولا تتعامل مع كل اتصال منفردا، هذه الميزة جعلتها ممكنة التطبيق [45]، تعتمد هذه الخوارزميات على المبدأ العشوائي للاسقاط والذي يعتمد بدوره على وجود عدد متناسب من الحزم لكل اتصال بالصف لتقرر الخوارزمية هل تقوم بالاسقاط من هذا الاتصال ام تقوم بعدم الاسقاط والذي يحدد بدوره نسبة مشاركة هذا الاتصال في القناة.

#### 1.2.1.4.2 خوارزمية الاسقاط المبكر العشوائي RED

في خوارزمية الاسقاط المبكر العشوائي RED [3] يتم تحديد حد أدنى لا يمكن قبله إسقاط الحزم وحد أعلى يمكن بعده إسقاط الحزم، تقوم هذه الخوارزمية باسقاط الحزم التي تصل عشوائياً عند حدوث الازدحام، ويزيد احتمال الاسقاط مع زيادة الازدحام، وبالتالي تحديد مستويين للاسقاط داخل الصف يخبر عن اعتماد الخوارزمية على الطول بصورة تامة، بالرغم من ان هناك من يوصي بتعديل هذين الحدين بصورة آليه [46] إلا ان تحديد الطول يعني ان الخوارزمية ولتعمل بصورة جيدة

لابد من ضبط الطول بشكل محدد لكل حمل من الاحمال يمر بالموجه. تعقيد هذه الخوارزمية وصعوبة تحديد عتبة الاسقاط ومنع الاسقاط المناسبه ادى ببعض الباحثين لينصح بعدم العمل بها [47] بالاضافه الى دراسات اخرى نصحت بتعديل العتبات بصورة آليه [46, 48-50], وبالتالي فان هذه الخوارزمية تعتمد بصورة واضحه على طول الصف لتعمل بصورة مناسبه.

#### 2.2.1.4.2 خوارزمية الاسقاط المبكر العشوائي المستقره SRED

صممت هذه الخوارزمية [51] لتحديد الاتصالات النشطة الاكثر استهلاكاً للنطاق ومعاقبها, ذلك بتطبيق بعض القيود على حركة ارسالها للبيانات. هذه الخوارزمية شبيهه بخوارزمية الاسقاط المبكر العشوائي RED حيث أنها تقوم باسقاط الحزم بصورة عشوائية تعتمد على الحمل وذلك عندما يتعرض صف الموجه للازدحام، تعمل هذه الطريقة كالاتي: تنشئ قائمة تسمى قائمة الاموات الاحياء Zombie list تحتوى هذه القائمة على كل الحزم فى الموجه بالاضافة الى الحزم التى تم ارسالها من الموجه سابقاً وحيث ان طول هذه القائمة ثابت فإنه يحتوى على كل الحزم فى ذاكرة الموجه بالاضافة الى عدد محدد من الحزم المرسله سابقاً حيث تقوم بمقارنة الحزمه الواصلة حديثاً بحزمه مأخوذة عشوائياً من قائمة الاموات تطابق الحزمتين (يعنى نجاح المقارنة) يؤدي هذا الى زيادة عداد القائمة بالاضافة الى وضع زمن وصول الحزمه فى المؤقت الزمني الخاص بالقائمة اما عدم نجاح المقارنة يحدث عندما لا تتطابق الحزمتين، فى هذه الحالة يتم وضع الحزمه الواصلة حديثاً بدلاً من الحزمه القديمة فى القائمة بإحتمال P أو عدم وضعها بإحتمال  $1-P$ ، بما انها تعتمد على الاسقاط العشوائي فالاسقاط العشوائي يحتاج الى صف طويل ليعمل بصورة جيده فهي بالتالي تعتمد على طول الصف في الموجه.

#### 3.2.1.4.2 خوارزمية الخنق العادل CHOKE

صممت هذه الخوارزمية [38] بغرض ضمان العدالة بين الاتصالات حيث يتم مقارنة الحزم التي تصل حديثاً مع حزمه مأخوذة بصورة عشوائية من الصف فاذا كانت الحزمتين تنتميان لنفس الاتصال فانه يتم اسقاطهما معاً، واذا كانت الحزمتين مختلفتان فان الحزمه التي تصل حديثاً يتم اسقاطها باحتمال محدد. تعتمد هذه الخوارزمية بصورة كبيرة على طول الصف ايضا لانها تعتمد على الاسقاط العشوائي، لتعمل هذه الخوارزمية بصورة جيدة لابد من تخزين عدد كبير من الحزم [52, 53] وذلك لضمان عدالة تامة بين الاتصالات، اذا كان طول الصف غير كاف او قصير يؤدي ذلك الى عدم عمل الخوارزمية بصورة جيدة ولن تستطيع تحديد الاتصالات الاكثر استهلاكاً لعرض النطاق وبالتالي معاقبتها.

#### 2.4.2 الخوارزميات التي تعتمد على التأخير في الصف

خوارزميات تأخير الصف هي الخوارزميات التي لا تقوم بالإسقاط من الصف الا اذا تجاوز التأخير الناتج عن حد معين وبالتالي هذه الخوارزميات لا تضع اي إعتبار لطول الصف للإسقاط، صممت هذه الخوارزميات اصلاً للتخلص من مشكلة تضخم ذواكر الموجهات وبما أن تضخم الذاكرة هو في الأصل عبارة عن إزدحام فإن الخوارزميات التي تعمل على حل هذه المشكلة تعتبر من خوارزميات فك الإزدحام. من هذه الخوارزميات خوارزمية التحكم بتأخير الصف Code1 حيث تقوم هذه الخوارزمية بحساب تأخير الحزمة الواردة إلى ذاكرة الموجة وتقوم باسقاطها إذا تجاوز تأخيرها 5ms (ثابت تستعمله الخوارزمية) داخل صف الموجة. في الفقرات التاليه سوف نتحدث عن الخوارزميات

التي تعتمد على التأخير اي تأخير الحزمه من زمن دخولها الى صف الموجه وحتى زمن خروجها من صف الموجه الى الخارج.

#### 1.2.4.2 خوارزمية الاسقاط من آخر الصف ذات الضبط الآلي ADT

تعتبر خوارزمية ADT من اقدم الخوارزميات التي اهتمت بمعالجة مشكلة زيادة التأخير, ظهرت هذه الخوارزمية قبل عملية استكشاف وظهور مشكلة تضخم الصفوف Bufferbloat, الفكرة الاساسيه لهذه الخوارزمية هي السماح فقط للحزم التي تحافظ على عرض نطاق عنق الزجاجة في وضع جيد بالدخول الى صف الموجه واسقاط بقيه الحزم, تعتبر هذه الدراسة ADT [54] ان الغرض الاساسي هو تقليل التأخير ثم بعد ذلك المحافظة على عرض النطاق ولكن في الحقيقة الغرض الاساسي هو المحافظة على عرض النطاق ثم تقليل التأخير في المرحلة التاليه, ذلك واضح من طريقة عمل الخوارزمية, تعمل خوارزمية الاسقاط من آخر الصف ذات الضبط الآلي ADT في خطوتين هما: الاولى تقدير متوسط الاستفاده من عرض النطاق, والثاني هو موائمة طول صف الموجه حسب متوسط الاستفاده من عرض النطاق والذي يقاس دوريا.

تتميز هذه الخوارزمية بان لها أداء متماثل مع أداء خوارزمية الصفوف القصيره التي تتبني عليها الخوارزمية التي تم تطويرها في هذا البحث خوارزمية التقدير الذاتي لطول الصف auto, ولكن خوارزمية الاسقاط من آخر الصف ذات الضبط الآلي ADT تتميز بان لها أداء معقد جدا ولا يمكن تطبيقها عمليا نسبه لتعقيد هذه الخوارزمية والموضحه في الشكل رقم (2-1).

ON EVERY PACKET ARRIVAL:

IF (now – LastUpdate) > SamplePeriod

. CURTHR :=  $\frac{\text{NmbBEnq}(\text{now}) - \text{NmbBEnq}(\text{LastUpdate})}{\text{now} - \text{LastUpdate}}$

. THR :=  $\rho \text{ CURTHR} + (1 - \rho) \text{ THR}$

. IF ( $\frac{\text{THR}}{\text{ServiceRate}} < u$ )

. qADT := qADT · c

. OTHERWISE

. qADT := qADT  $\frac{\text{qADT}}{c}$

. END

. qADT := min(qADT , SizeOfBuffer)

. LastUpdate := now

END

الشكل 1-2 الشفرة الاولى لخوارزمية ADT

p عبارة عن معامل توسيط ياخذ القيمة بين (0 - 1) يستخدم لحساب متوسط عرض النطاق.

c يكون بقيمه اكبر من الواحد وهو عبارة عن معامل دالة التحكم MIMD للتحكم بالمعامل qADT

SamplePeriod عبارة عن فتره زمنيه لاخذ العينات.

u عبارة عن عرض النطاق المطلوب

now زمن اخذ العينه الحالي

LastUpdate آخر زمن تعديل

CURTHR عرض النطاق الحالي

NmbBEnq(t) يحدد عدد البايت الداخلة الى الصف في الفتره (t-0)

THR عرض النطاق في زمن اخذ العينه السابق

$q_{ADT}$  عبارته عن معامل يحدد عدد الحزم او يحدد طول الصف

ينحصر تعقيد هذه الخوارزمية في نقطتين هما:

1. عدد عمليات الجمع والطرح والضرب والقسمه والمقارنه التي توجد في هذه الخوارزمية والتي

لا بد من اجرائها عند دخول اي حزمه الى الموجه, تؤدي هذه الخوارزمية الى جهد كبير على

معالج الموجه وهي بالتالي عملية غير ممكنه لانه لا يمكن اشغال معالج الموجه المشغول

اصلا بعملية اخرى هي عملية التوجيه وهي العملية الاساسيه للموجه بالطبع.

2. المعاملات تحتاج قيم مختلفه تختلف باختلاف القناة التي يتحكم فيها الموجه, هذه

المعاملات هي اولا معامل الفتره الزمنية لاختذ العينه sample period, والمعامل  $c$  نصت

الدراسه [54] على ان يكون الثابت  $c$  بقيم اكبر من واحد, فعندما يكون عرض النطاق

الحالي اصغر من عرض النطاق  $u$  المطلوب, يكون المعامل  $c$  اكبر من الواحد ليتم ضربه

في المعامل  $q_{ADT}$ , يؤدي هذا الى ان تزيد قيمة عدد الحزم  $q_{ADT}$  ويؤدي هذه التقليل الى

تقليل معدل الارسال. اما عندما يكون عرض النطاق الحالي اصغر من عرض النطاق

المطلوب فسيكون اصغر من الواحد, يتم قسمة المعامل  $q_{ADT}$  على المعامل  $c$  يؤدي هذا

الى ان تقل قيمة عدد الحزم  $q_{ADT}$  يؤدي هذا الى زيادة معدل الاسقاط.

#### 2.2.4.2 خوارزمية التحكم بتأخير الصف Codel

خوارزمية التحكم بتأخير الصف Codel [35] تحتوى على ثلاثة مكونات اساسية تميزها عن

خوارزميات AQM السابقة. هذه الخوارزمية لا تعتمد على طول الصف, متوسط طول الصف,

عتبه طول الصف, معدل الارسال, الاستفاده القصوى من عرض النطاق, معدل الاسقاط او كمية

الحزم الموجودة بالصف. تستخدم هذه الخوارزمية الطول الاقصر للصف كتقدير جيد ودقيق للصف العامل Standing queue بعد ذلك يمكن استعمال متغير واحد لتحديد الطول الاقصر المناسب هل هو ادنى او اعلى من القيمة الهدف للتأخير Target للصف العامل Standing queue. أخيراً بدلاً من حساب طول الصف بالبايت او بعدد الحزم فإنه يتم حساب زمن يسمى زمن زيارة الحزمة عبر الصف.

استعمال القيمة الاقل لطول الصف ذو فوائد كثيرة، حيث أنه يمكن تقليل عمل الخوارزمية وحصر عمل الموجه فقط عند خروج وارسال الحزمه من الموجه Dequeued وهذا يعنى ان كل عمل خوارزمية التحكم بتأخير الصف Code1 يتم عندما تكون الحزم في حالة الخروج النهائي من الموجه. الشئ الوحيد الذي تفعله هذه الخوارزمية هو تسجيل زمن وصول الحزمه الى الموجه، واما اذا كان الصف ممتلئاً عند ورود الحزمه فإنه في هذه الحالة يتم اسقاطها. تفترض خوارزمية التحكم بتأخير الصف Code1 ان الصف العامل ذو التأخير Target مقبول ولايتم اسقاط اي حزمه من صف الموجه وكذلك عندما يكون طول الصف يساوي حزمه واحدة فقط. تعتمد طريقة خوارزمية التحكم بتأخير الصف Code1 فى تحديد التأخير الدائم بواسطة مراقبة التأخير الاقصى والذي تختبره الحزم فى الصف، ولضمان ان القيمة الاقصرية لن تصبح قيمة غير ذات فائدة يلزم التأكد من القيمة الاقصرية داخل الفترة الزمنية الاقرب، عندما يتجاوز التأخير الهدف Target لفته زمنية محددة فإن الحزمه يتم اسقاطها وبعد ذلك يتم حساب زمن الاسقاط التالى. زمن الاسقاط التالى ينقص بصورة عكسية مع مربع عدد الحزم المسقطة وذلك منذ دخول منطقة الاسقاط، عندما يصبح تأخير الصف ادنى من القيمة الهدف Target فإن الخوارزمية توقف الاسقاط. ليس هناك اسقاط اذا كان صف

الموجه أقل أو يساوي من حزمه واحده, هناك ايضا بعض الضمانات لعدم إعادة الدخول في منطقة الإسقاط بعد فترة قصيرة من الخروج من هذه المنطقة [30].

## 5.2 خلاصة

في هذا الباب تناولنا الخوارزميات التي تعمل في مجال التحكم بالازدحام والتي تم تقسيمها سابقا الي قسمين حسب إمكانية معالجة كل اتصال على حده او معالجة كل الاتصالات الواردة بصورة عامه, حيث سميت الاولى بالخوارزميات العادله وسميت الخوارزميات من النوع الثاني بالخوارزميات الخفيفه stateless او خوارزميات الجدوله وخلصنا الى انه يمكن اعادة عملية التقسيم هذه الى خوارزميات تعتمد على طول الصف وهذا النوع يشتمل على النوعين اعلاه وخوارزميات تعتمد على التأخير على الصف وهذا النوع من الخوارزميات هي موضوع هذا البحث, سيتناول الباب القادم احد اهم الخوارزميات التي تعمل في مجال التحكم بالازدحام وهي خوارزمية التحكم بتأخير الصف Code1, والتي تعتمد على التأخير في الصف وذلك كتمهيد لدراسة مشاكل هذا النوع من الخوارزميات ومقارنتها فيما بعد بالخوارزمية المقترحه في هذا البحث وهي خوارزمية التقدير الذاتي لطول الصف .auto

## الباب الثالث

### 1.3 خوارزمية التحكم بتأخير الصف Codel

خوارزمية التحكم بتأخير الصف Codel [35] حاولت حل مشكلة الازدحام على الموجبات عن طريق مراقبة التأخير ولكن هناك خوارزميات اخرى تحاول ان تجعل الصف قصيرا بغرض تقليل التأخير [55, 56], تحاول خوارزمية التحكم بتأخير الصف Codel حل مشكلة الازدحام الشديد والمسبب لمشكلة تضخم الذاكر Bufferbloat [27, 29, 57], وتعتبر هذه الخوارزمية ان العامل الحاسم والاساسي فى إسقاط الحزم هو التأخير الملاحظ منذ دخول الحزمة وحتى خروجها، لذلك صنفت هذه الخوارزمية ضمن الخوارزميات التى تعتمد حساب التأخير للتحكم بالازدحام، صممت هذه الخوارزمية بالاساس للتخلص من مشكلة إمتلاء وتضخم صفوف الموجبات وحيث ان تضخم الصفوف وإمتلاءها هو نوع من أنواع الازدحام لذلك صنفت هذه الخوارزمية مع خوارزميات التحكم بالازدحام.

تستخدم هذه الخوارزمية الطول الاقصر للصف كتقدير جيد ودقيق للتأخير على الصف الحالى Standing Queue بعد ذلك يمكن استعمال متغير واحد لتحديد الطول الاقصر المناسب هل هو اسفل او اعلى من القيمة الهدف Target للتأخير الحالى. بدلاً من حساب طول الصف فإنه يتم حساب زمن يسمى زمن زيارة الحزمة, تعتمد خوارزمية التحكم بتأخير الصف Codel فى تحديد التأخير الدائم بواسطة مراجعة التأخير الاقصر والذى تختبره الحزم فى الصف ولضمان ان القيمة الصغرى لن تصبح قيمة غير ذات فائدة يلزم التأكد من القيمة الصغرى داخل الفترة الزمنية الاقرب،

عندما يتجاوز تأخير الصف الزمن الهدف Target لفترة زمنية محددة Interval فإن الحزمة يتم إسقاطها وبعد ذلك يتم حساب زمن الإسقاط التالي.

### 2.3 العلاقة بين الازدحام و تضخم الصفوف

الازدحام هو ورود عدد كبير من الحزم من مدخل الموجه لا يستطيع مخرج الموجه ان يقوم بارسالها وتظهر في هذه الحالة الحاجة الى خوارزمية تقوم بفك حالة الازدحام حيث كانت اول خوارزمية الاسقاط من آخر الصف Drop\_Tail تقوم باسقاط اخر حزمه لاتجد لها مكانا في صف الموجه, لا تعتبر هذه الطريقة خوارزميه بل هي حل تلقائي, فعندما لا يكون هناك مكان لتخزين الحزم سيتم إسقاطها, ولكن عند دراسة مشكلة الإزدحام بالمقارنة مع مشكلة تضخم الصفوف تعتبر هذه الطريقة خوارزمية حيث أنه في مشكلة تضخم الصفوف لا يتم الإسقاط وهذه هي المشكلة الأساسية وذلك لطول الصف في الموجه، كل الخوارزميات التي تعمل في مجال التحكم بالازدحام تقوم باسقاط الحزم من الصف وذلك بغرض محدد مرتبط بالخوارزمية نفسها، مثلاً في خوارزمية الاسقاط المبكر العشوائي RED لابد ان يكون هناك طول محدد للصف تقوم الخوارزمية باسقاط الحزم عند هذا الحد وذلك للتخلص من عدة مشاكل كانت مرتبطة بخوارزمية الاسقاط من آخر الصف Drop\_Tail وهي مشكلة تزامن الاسقاط وعدم العدالة بين الاتصالات, ليس بين المشاكل التي يتم اسقاط الحزم من أجلها مشكلة تضخم صف الموجه، خوارزمية اخرى هي خوارزمية FRED تقوم باسقاط الحزم من الموجه للتخلص من مشكلة التزامن بالإضافة إلى ضمان العدالة بين الاتصالات التي تمر عبر الموجه، خوارزمية Choke ايضاً تقوم بضمان العدالة بين الاتصالات حيث يتم الاسقاط عشوائياً ولكنها تختلف عن خوارزمية FRED في انها تنتمي إلى عائلة خوارزميات ادارة

الصف الفعالة AQM. بدأت مشكلة تضخم الصف بالموجهات تظهر حديثا, حيث لاحظ بعض المهندسين ان هناك تأخير كبير جدا تتعرض له الحزم, يتجاوز هذا التأخير احيانا حدود الواحد ثانيه [28] حيث انه لم يكن مثل هذا التأخير موجودا من قبل في شبكات الحاسوب, بعد التدقيق والبحث تم اكتشاف ان هذا التأخير ينتج عن زيادة طول الصفوف بالموجهات الحديثه, حيث لاحظ المهندسون انه ليس هناك اي حزم مسقطه من الشبكة بالصوره المناسبه مع مقدار الحمل حيث يمكن ان ينتج عن الحمل غالبا عند ازدحام الشبكات سقوط الحزم وبالتالي فانه اذا لم يكن هناك اي اسقاط للرسائل فأين تذهب هذه الحزم, تم ملاحظة ان الحزم لا تسقط وبالتالي فهي مخزنه داخل صفوف الموجهات والتي اصبحت ذات اطوال كبيره وتسع لعدد كبير من الحزم, تنتظر هذه الحزم داخل هذه الصفوف الطويله دورها في الارسال, اما بالنسبه لمستقبل هذه الحزم والذي فقد الامل في وصولها بعد التأخير الكبير فانه سوف يطلب اعاده ارسالها, بالرجوع الى موضوع الصفوف الطويله الموجوده داخل الموجهات, تم التوصل الى ان كل الموجهات المصنعه حديثا تحوي ذواكر كبيره تكون نواه لتشكل صفوف طويله, سميت زيادة الصفوف هذه والتي تسببت في زيادة التأخير بمشكلة تضخم الصفوف Bufferbloat, اما فيما يخص العلاقه بينهما مما سبق ومن تعريف الازدحام يتأكد لنا ان تضخم الصفوف هو نوع من انواع الازدحام وجب البحث عن حل والسعي للتخلص منه, في كل الخوارزميات المتعلقة بايجاد حل لمشكلة الازدحام كان يتم البحث عن العامل الاساسي المسبب للازدحام وذلك للتخلص منه حيث انه في فترة ما من فترات تطوير خوارزميات فك الازدحام تقرر ان العامل الحاسم لفك الازدحام هو التحكم بطول الصف, تم تطبيق هذا الحل في خوارزمية الاسقاط المبكر العشوائي RED, في هذه الخوارزمية يتم الاسقاط من الصف بعد تجاوز عدد الحزم طول

محدد من الطول الكلي للصف, تعتبر الخوارزميات التي تهتم بطول الصف واعتباره في عمليات فك الازدحام من انجح الخوارزميات التي لعبت دورا كبيرا في التخلص من الازدحام الى درجه معقوله ولكن صعوبة التحكم بمعاملات خوارزمية الاسقاط المبكر العشوائي RED [49] جعل مجتمع الشبكات يبتعد عنها, بعد اكتشاف مشكلة تضخم الصفوف تم اللجوء الى معامل جديد للتحكم في الازدحام والتخلص منه, في هذه المره كان العامل المسبب للازدحام ليس هو العامل الذي تم اعتماده سابقا للتخلص من الازدحام وهو طول الصف, حيث كان لابد من ايجاد عامل آخر للتحكم, توجه الباحثين مباشرة نحو العامل الاساسي في هذه المره وهو الزمن الذي تمكثه الحزمه داخل الصف, حيث ينتج تضخم صفوف الموجهات ازمان تأخير كبيره للرسائل عليه, وبالتالي للتخلص من هذا الازدحام كان لابد من التخلص من تاخير الحزم على الصف, يمكن ان يكون الطريق الاقصر للتخلص من مشكلة تضخم الصفوف هو التخلص من الصفوف نفسها او تقليل طول الصفوف الى اقل حد ممكن ولكن الصف مهم جدا وذلك لامتصاص الارسال الانفجاري الذي تتميز به شبكة الانترنت, اذا كان حل مشكلة الازدحام الناتج عن تضخم الصفوف هو التحكم بتأخير الحزم داخل صف الموجه.

تتشترك كل الخوارزميات التي تعمل في مجال التحكم بالازدحام في الاسقاط من الموجه لاجراض واهداف متعددة قد تختلف في بعض هذه الاهداف وقد تتوافق ولكن ليس من بين هذه الاهداف اسقاط الحزم بغرض تقليل التأخير وذلك للتغلب على مشكلة تضخم الصفوف بالموجهات وذلك في الخوارزميات السابقه.

### 3.3 الاسقاط فى خوارزمية التحكم بتأخير الصف Codel

يتم الاسقاط فى خوارزمية التحكم بتأخير الصف Codel لغرض وحيد هو تقليل تأخير الحزم بالصف الى حد معين وذلك لضبط التأخير, الخوارزمية التى تقوم بهذا العمل هي خوارزمية التحكم بتأخير الصف Codel حيث صممت كمحاولة للتخلص من مشكلة تضخم صف الموجه او زيادة طول صف الموجه.

### 1.3.3 الاسقاط وبروتوكول التحكم بالنقل TCP

الاسقاط يعنى ضياع الحزم ولكن بالرغم من انه امر غير جيد ولكنه أمر حيوي جداً لخدمة تطبيقات بروتوكول التحكم بالنقل TCP والذي يعتمد على الاسقاط لتقليل معدل ارساله، فهذا البروتوكول يقوم بالارسال وزيادة معدل ارساله مع مرور الوقت ولا يقلل معدل ارساله الا اذا تعرض لاسقاط حزمه واحدة او أكثر، وبالتالي اذا لم يتعرض للاسقاط فهو قادر على زيادة ارساله الى ما لانهاية وقادر ايضاً على ان يملأ كل الصفوف بكل الموجهات التى بمساره، نخلص الى ان الاسقاط حيوي وذلك حتى نسمح بمشاركة عرض النطاق بين الاتصالات المختلفة.

### 4.3 التأخير فى خوارزمية التحكم بتأخير الصف Codel

خوارزمية التحكم بتأخير الصف Codel هي من الخوارزميات المعتمدة على حساب التأخير وذلك للتحكم الازدحام وحل المشكلة المرتبطة بالازدحام وهي مشكلة تضخم صفوف الموجهات. التأخير المشار اليه هنا هو التأخير المحسوب منذ دخول الحزمة الى صف الموجه وحتى خروجها منه ليتم

ارسالها [58]. تقوم هذه الخوارزمية بحساب قيمة التأخير هذه للتحكم بطول الصف والذي يسبب مشكلة تضخم الصف.

### 1.4.3 تأثير التأخير على بروتوكول التحكم بالنقل TCP وبروتوكول حزم المستخدم

#### UDP

يمكن أن يكون التأخير الناتج من طول الصف أكثر سوءاً من سقوط الحزم الناتج عن قصر الصف لعدة أسباب هي:

أولاً بروتوكول التحكم بالنقل TCP يعتمد على مؤقت، وظيفة هذه المؤقت تنحصر في أنه عند إرسال أي حزمه ينتظر بروتوكول التحكم بالنقل TCP لفترة زمنية لا تتجاوز 200ms وذلك حتى تصل حزمه الإقرار ACK. وإذا لم تصل حزمه الإقرار في هذا الزمن المحدد فإن المؤقت ستنتهي قيمته ويقوم بروتوكول التحكم بالنقل TCP بإعادة إرسال الحزمه مرة أخرى، هذه التقنية اساسية في بروتوكول التحكم بالنقل TCP وبما انه بروتوكول موثوق فإنه بواسطة هذه التقنية يستطيع التأكد من ان كل الحزم المرسله قد وصلت بسلام، حيث أنه عندما كانت الصفوف قصيره تكون الحزمة قد فقدت حقيقة "إما تم إسقاطها أو فقدت نتيجة لتخريبها" أما بعد زيادة طول الصفوف فإن الحزمة تكون موجودة في الصف وإعادة إرسالها لا يحل المشكلة بل يزيدا تعقيداً، تزداد المشكلة تعقيداً لأن الصف الممتلئ اصلاً يزداد طولاً وبالتالي يزداد تأخيره نسبة لزيادة عدد الحزم المعاد ارسالها.

ثانياً ليس لبروتوكول حزم المستخدم UDP وهو البروتوكول المعتمد عليه في نقل خدمات الزمن الحقيقي اي مؤقت زمني يخشى من إنتهائه ولكن خدمات الزمن الحقيقي تعتمد على تأخير أصغر من المؤقت الزمني لبروتوكول التحكم بالنقل TCP حيث أن خدمات الزمن الحقيقي لا تتحمل تأخير

أكبر من 250ms وهي أكبر قيمة للتأخير لتكون الخدمة مقبولة, قيم التأخير الكبيرة الناتجة عن طول الصفوف والتي قد تصل الى ثانية كاملة أكبر بكثير من أن تتحملها خدمات الزمن الحقيقي القائمة على بروتوكول حزم المستخدم UDP.

### 5.3 طريقة عمل خوارزمية التحكم بتأخير الصف Codel

عند ورود اي حزمه الى صف خوارزمية التحكم بتأخير الصف Codel تقوم بتسجيل زمن دخول هذه الحزمه الى الصف ثم بعد ذلك تسجل الخوارزمية زمن خروجها ثم تقوم بحساب الفرق بين هذين الزمنين ويسمى هذا الزمن المحسوب بزمن الزيارة Sojourn time اذا كان زمن الزيارة اقل من زمن محدد (5ms) يتم تمرير الحزمة ولن يتم اسقاطها اما اذا كان زمن الزيارة اكبر من هذه القيمة يتم اسقاط هذه الحزمة باحتمال (اي قد يتم اسقاطها او قد تمر من غير اسقاط) هناك حالتين تكون فيهما الخوارزمية هما حالة الاسقاط وهي مرحلة تدخل فيها الخوارزمية بعد ان يتجاوز التأخير القيمة المحددة (5ms) وذلك لفترة محددة تسمى بالقيمة الاسمية للتأخير وهي عبارة عن (100ms) والحالة الثانية وهي حالة عدم الاسقاط وهي أيضاً مرحلة تدخل فيها الخوارزمية وذلك عند إكتشاف ان الحزمة التي يتم اختبارها لديها زمن تأخير أقل من (5ms) وهي القيمة المحددة لضبط التأخير على الموجه. والقيمة المحددة (5ms) هي قيمة مختارة بعناية تامة، تجعل هذه القيمة طول الصف مساوي للقدرة التخزينية للصف في 5ms [59].

### 1.5.3 مؤثرات إعتداد التأخير في الصف على الاداء

هناك عدة عوامل أساسية يجب مراعاتها عند مراقبة أداء أي خوارزمية تعمل في مجال التحكم بالازدحام هذه العوامل هي في الأصل العوامل التي من أجلها تم تصميم خوارزميات التحكم بالازدحام. في الحالات السابقة من خوارزميات التحكم بالازدحام يتم تصميم الخوارزمية بصورة عامة ثم مراقبة تأثيرها على المعاملات الأساسية وهي العدالة، معدل الإسقاط، الاستفادة من عرض النطاق، أما في هذه الحالة والتي سندرسها في هذا الباب فهي دراسة خوارزمية تحكم لا تعمل بصورة عامة كما في السابق ولكن بصورة محددة أي أن تصميمها يهتم بمعامل واحد من المعاملات السابقة ومراقبة تأثير هذه الخوارزمية على المعاملات الأخرى في هذه الحالة وكما هو متوقع فإن المعامل أو المشكلة التي ستهم بها هذه الخوارزمية ستكون هي الجانب المؤثر من هذه الخوارزمية أما العوامل الأخرى فسوف تصبح ثانوية في نظر الخوارزمية وسيتم مراقبتها للتأكد من أن الخوارزمية لن تؤثر فيها بصورة سلبية.

#### 1.1.5.3 العدالة

هناك عدة تعريفات للعدالة ولكن بصورة عامة هي مدى قدرة الخوارزمية على توزيع مصادر الشبكة على الاتصالات المارة بها بصورة عادلة وهذا التعريف متشدد لأن حاجة الاتصالات غير متساوية، هناك تعريف آخر يقول بإعطاء كل إتصال الحد الأدنى من الموارد والتي تمكنه من العمل بصورة جيدة.

خوارزمية التحكم بتأخير الصف Code1 تقوم بتحديد حد أقصى للتأخير على صفها وهو 5ms وذلك يعني أن أي حزمه يتجاوز تأخيرها 5ms وهي موجودة في الصف ستقوم الخوارزمية بإسقاطها

فوراً وستدخل بعد ذلك الخوارزمية فى مرحلة او فترة الاسقاط حيث تقوم باسقاط الحزم باحتمال الى ان تخرج من هذه المرحلة او الفترة. فكرة الاسقاط باحتمال عشوائي هي اصلاً مأخوذة عن خوارزميات ادارة الصفوف حيث تم اقتراح الاسقاط باحتمال وذلك للتخلص من التزامن وضمان الحد الادنى من العدالة, ادراج الاسقاط باحتمال فى خوارزمية التحكم بتأخير الصف Codel وهي احدى خوارزميات التحكم بالتأخير يؤدي الى ضمان الحد الادنى من العدالة بين الاتصالات وسيكون تأثير الاسقاط باحتمال عشوائي على هذه الخوارزمية مثل تأثيره على خوارزميات ادارة الصفوف مثل خوارزمية الاسقاط المبكر العشوائي RED، كان تأثير الاسقاط باحتمال على خوارزمية الاسقاط المبكر العشوائي RED هو ضمان الحد الادنى من العدالة.

### 2.1.5.3 معدل الاسقاط

معدل الاسقاط هو عدد الحزم التى يتم اسقاطها من الصف مقارنة مع عدد الحزم التى يتم تمريرها، معدل الاسقاط مرتبط ارتباط وثيق مع طول الصف، فيمكن تقليل معدل الاسقاط الى أقل حد ممكن وذلك بزيادة طول الصف، ينطبق هذا الامر على خوارزمية الاسقاط من آخر الصف وكل خوارزميات الجدولة، أما فى حالة خوارزميات إدارة الصف فإن زيادة طول الصف لا تؤثر فى معدل الاسقاط بصورة مباشرة، ولكن عملياً يميل المصممون فى خوارزميات إدارة الصف الى زيادة طول الصف وذلك بزيادة حدود الاسقاط العليا والدنيا، خوارزمية التحكم بتأخير الصف Codel تتميز بانها الخوارزمية ذات معدل الاسقاط الاعلى وذلك لانها تركز بصورة اساسيه على تقليل التأخير الى حد معين، تقليل التأخير يعني لخوارزمية التحكم بتأخير الصف Codel صف قصير، اما الصف القصير فيعني الاسقاط المكثف من الصف للمحافظة على التأخير اقل ما يمكن.

### 3.1.5.3 الاستفادة من عرض النطاق

لم تضع خوارزمية التحكم بتأخير الصف Codel الاستفادة من سعة القناة موضع الاهتمام الكبير حيث ان اهتمامها الاكبر هو تقليل زمن التأخير المحسوب علي الموجه, من اسباب اضافة الصفوف الى الموجهات الاستفادة من عرض النطاق نسبة لطبيعة الارسال الانفجاري, التشدد الذي تسلكه خوارزمية التحكم بتأخير الصف Codel في اسقاط الحزم بعد تجاوزها حدا معيناً من الزمن داخل الصف يجعلها تتهاون في مجال آخر هو المحافظة على الاستفادة من عرض النطاق, تنتهج خوارزمية التحكم بتأخير الصف Codel الاسقاط المكثف من الصف وذلك بغرض وحيد هو التخلص من مشكلة التأخير او تضخم الصفوف Bufferbloat.

### 6.3 مشاكل خوارزمية التحكم بتأخير الصف Codel

في هذه الفقرة سنسرد المشاكل الاساسيه التي تواجهها خوارزمية التحكم بتأخير الصف Codel في الواقع العملي, المشكله الاولى هي تأثير العدالة بين الاتصالات في خوارزمية التحكم بتأخير الصف Codel بعدد الاتصالات, المشكله الثانيه هي سرعة خط الدخل الي موجه خوارزمية التحكم بتأخير الصف Codel, عند زيادة سرعة خط الدخل الي سرعات كبيره جدا سوف يتحول موجه خوارزمية التحكم بتأخير الصف Codel الي العمل كأنه موجه خوارزمية الاسقاط من آخر الصف Drop\_Tail اي انه سوف يقوم باسقاط كل الحزم الوارده من آخر الصف ولن تتمكن الخوارزمية من العمل بالصوره الطبيعیه المخطط لها, اما المشكله الثالثه فهي حساب زمن الزياره, حيث ان هذه

العملية الحسابية تستهلك زمن المعالج بصورة كبيرة، فسوف ستكون هذه المشكلة هي العائق الاساسي الذي سوف يقف امام امكانية تطبيق هذه الخوارزمية.

### 1.6.3 تأثير العدالة بين الاتصالات في خوارزمية التحكم بتأخير الصف Codel

#### بعدد الاتصالات

تقوم خوارزمية التحكم بتأخير الصف Codel باسقاط اي حزمة تتجاوز فترة تخزينها 5ms فهي تقوم بتسجيل زمن دخول الحزمة الى الصف ثم تقوم بتسجيل زمن خروج الحزمة من الصف فاذا كان الزمن اكبر من 5ms تقوم باسقاط هذه الحزمة واذا كان الزمن المحسوب اقل من 5ms تقوم خوارزمية التحكم بتأخير الصف Codel بالدخول في حالة تسمى مرحلة الاسقاط، عندما تكون الخوارزمية في هذه الحالة تقوم باسقاط اي حزمة وذلك باحتمال عشوائي، وجود حزمة ذات تأخير أكبر من 5ms هو اشارة على ان التأخير اصبح كبيراً وبالتالي ايضاً طول الصف اصبح كبيراً، تستمر خوارزمية التحكم بتأخير الصف Codel بالاسقاط وهي داخل مرحلة الاسقاط الى ان تأتي حزمة أخرى ذات تأخير أقل من 5ms وهذه الحزمة هي إشارة الى أن التأخير اصبح صغيراً وبالتالي طول الصف أصبح قصيراً فتقوم الخوارزمية بايقاف الاسقاط .

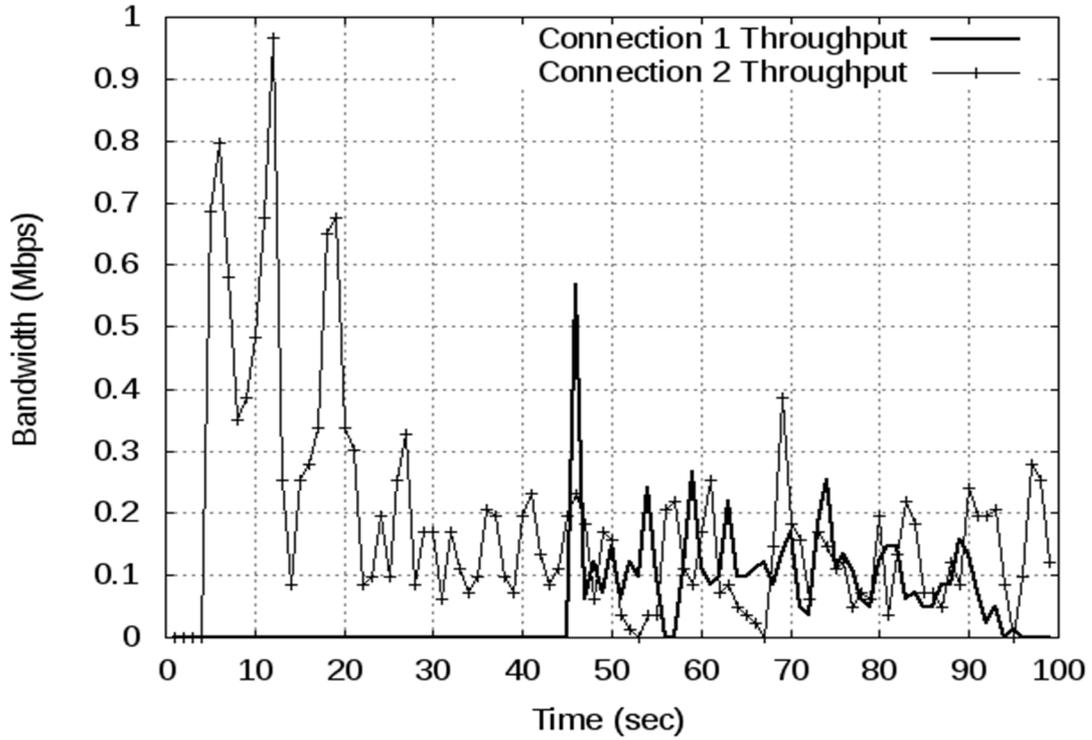
يرتبط معدل الاسقاط في خوارزمية التحكم بتأخير الصف Codel، هناك قيمتان للتأخير يمكن تمييزهما، الاولى التأخير الكلى وهو التأخير الذى تتعرض اي حزمة منذ خروجها وحتى وصولها وعودة حزمة الاقرار الخاصة بها وذلك فى بروتوكول التحكم بالنقل TCP او بالنسبة لبروتوكول حزم المستخدم UDP هو التأخير الذى تتعرض له الحزمة حتى وصولها هدفها والتأخير الثاني هو

التأخير الذي تحسبه خوارزمية التحكم بتأخير الصف Codel وهي قيمة تأخير صغيرة تساوي 5ms تلزم خوارزمية التحكم بتأخير الصف Codel كل الحزم بعدم تجاوزها وإذا تم تجاوزها فإن الخوارزمية تقوم في هذه الحالة بالاسقاط. اذاً قيمة التأخير الاولى هي القيمة الاساسية وهي القيمة التي تؤثر على الاتصالات الفعلية القائمة على الشبكة وقيمة التأخير الثانية هي قيمة محلية خاصة بخوارزمية التحكم بتأخير الصف Codel.

معدل الاسقاط في خوارزمية التحكم بتأخير الصف Codel مرتبط بقيمة التأخير المحلى وهي 5ms وطول الصف وليس لديه ارتباط بالتأخير الكلى المسموح به للاتصال المعني، كمثال اذا كان التأخير المسموح به هو 150ms لاتصال محدد والتأخير الكلى للقناة التي يمر بها الاتصال هو 50ms، فذلك يعنى ان تأخير هذا الاتصال بقيمة أكبر من 5ms (بافتراض ان هناك موجه وحيد في مسار هذا الاتصال) لن يؤثر على أداء هذا الاتصال ولكن خوارزمية التحكم بتأخير الصف Codel سوف تقوم باسقاط حزم هذا الاتصال من الصف وذلك عندما يتجاوز طول الصف 5ms دون مراعاة ان هذا الاتصال يستطيع ان يتحمل تأخير أكبر من 5ms.

هناك ارتباط كبير بين الاسقاط باحتمال وطول الصف بالموجه والعدالة، حيث ان الاسقاط العشوائي تم اقتراحه لضمان ان الاسقاط سوف يكون موزعا بعداله بين كل الاتصالات والمارة حالياً بالموجه، وليكون الاسقاط العشوائي عادلاً لابد ان يكون هناك العدد الكافي من الحزم على الصف والتي تمثل كل الاتصالات الحية وان يكون لكل اتصال على الصف عدد من الحزم متناسب مع حجم مشاركته في القناة. ليتمكن الاسقاط العشوائي من ضمان العدالة بين الاتصالات، لابد ان يكون الصف طويلاً كفاية وذلك لتتمكن كل الاتصالات الحية من تمثيل نفسها في الصف، اذاً فإن تأثير خوارزمية

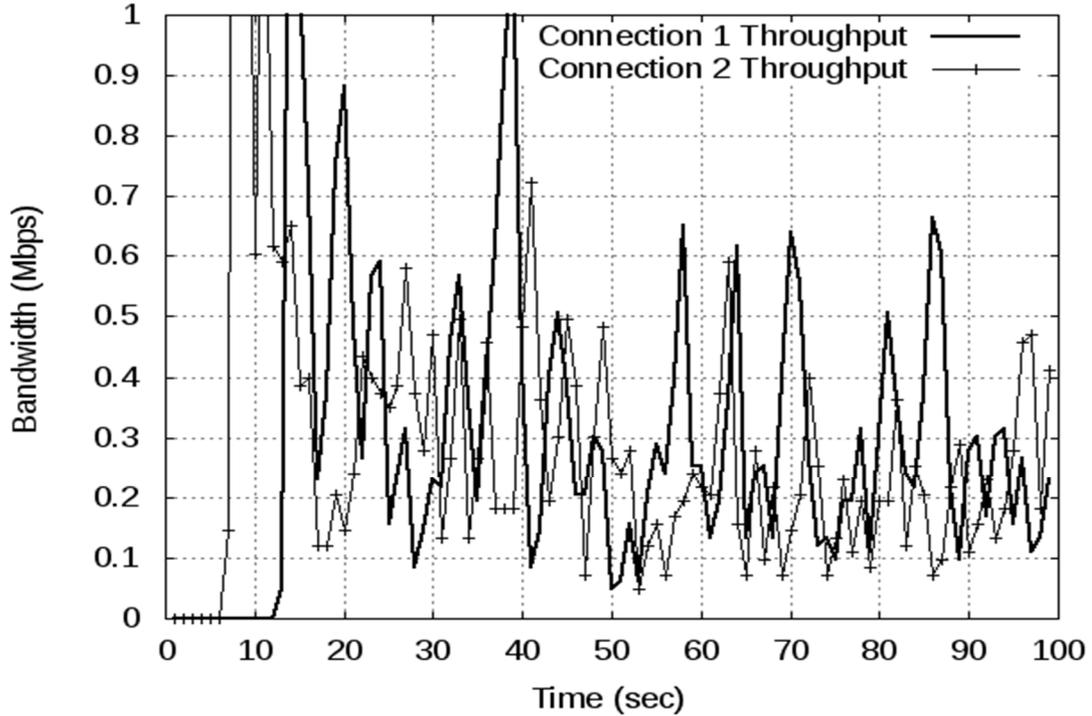
التحكم بتأخير الصف Codel على العدالة سوف تكون متناسباً مع عدد الاتصالات، اي ان خوارزمية التحكم بتأخير الصف Codel سوف تميل الى العدالة عندما يكون عدد الاتصالات صغيراً، وذلك بسبب ان خوارزمية التحكم بتأخير الصف Codel تحد من طول الصف، أما اذا كان عدد الاتصالات كبيراً فإن خوارزمية التحكم بتأخير الصف Codel لن تستطيع ان تضمن العدالة وذلك لأنها ذات صف قصير وبالتالي فإن عدد الاتصالات الكبير لن يجد المكان المناسب على الصف لتمثيل كل اتصال وبالتالي فإن الخوارزمية تقوم بالاسقاط العشوائي من الاتصالات الحية الممثلة بحزم على الصف وهي اتصالات اقل من عدد الاتصالات الحية الفعلية وبالتالي فإن الاتصالات التي تتضرر وتقوم بتعديل معدل ارسالها ليست كل الاتصالات دائماً، بل جزء من الاتصالات الكلية المارة حالياً بالموجه وهذا يؤدي الى ان تنجو بعض الاتصالات من عملية الاسقاط التي تقوم بها الخوارزمية ويؤدي ذلك الى عدم ضمان العدالة، وبالتالي فإن خوارزمية التحكم بتأخير الصف Codel لن تضمن العدالة في هذه الحالة، يوضح الشكل رقم (3-1) ذلك حيث تم زيادة عدد الاتصالات المارة الى 200 اتصال وعرض نطاق عنق الزجاجة 10Mbps وتم اختيار اتصاليين بصوره عشوائيه من مجموع عدد الاتصالات يتوضح جليا من الشكل ان خوارزمية التحكم بتأخير الصف Codel لا تستطيع ضمان العدالة عند زيادة عدد الاتصالات.



الشكل 3-1 زيادة عدد الاتصالات يؤدي الى عدم ضمان العدالة في Codel .

التفسير لهذا الامر هو ان الاسقاط العشوائي يحتاج الى صف طويل لضمان ان يتم الاسقاط من كل الاتصالات بصورة عادله, وبما ان عرض نطاق عنق الزجاجة في التجربه هو 10Mb\س فان طول الصف يكون حسب قانون حاصل ضرب عرض النطاق في التأخير PDB تقريبا 200 حزمه وهو عدد قليل مقارنة مع عدد الاتصالات البالغ 200 اتصال, في الشكل رقم (3-2) تم تغيير عدد الاتصالات الى 30 اتصال فقط مع تثبيت كل المعاملات الاخرى, في هذه التجربه استطاعت خوارزمية التحكم بتأخير الصف Codel ان تضمن العدالة وذلك لان عدد الاتصالات اصبح صغيرا مقارنة مع طول الصف وهو 200 حزمه حيث استطاعت الخوارزمية وبواسطة الصف الطويل ان تحتفظ بعدد من الحزم لكل اتصال حي وبالتالي تقوم بالاسقاط بصورة عادله من هذه الاتصالات

المارة، واي اتصال يتعرض لاسقاط حزمه او اكثر يقوم بالتراجع وتقليل معدل ارساله وبالتالي فان العدالة تشمل جميع الاتصالات.



الشكل 2-3 تقليل عدد الاتصالات يؤدي الى ضمان قدر من العدالة في Codel

### 2.6.3 سرعة الدخل

اكبر مشكلة قد تواجه خوارزمية التحكم بتأخير الصف Codel هو سرعة الدخل حيث ان لخوارزمية التحكم بتأخير الصف Codel ميزة أساسية وعن طريقها تحافظ على قيمة التأخير القليلة والتي صممت أساساً من أجلها. هذه الميزة هي القيمة الابتدائية التي تنتظرها خوارزمية التحكم بتأخير الصف Codel قبل ان تبدأ بالاحساس ان هناك تأخير على صف الموجه وهي عبارة عن قيمة زمنية لا تتجاوز 5ms وهذه تترجم الى عدد من الحزم كالاتي، تقوم الخوارزمية بتسجيل زمن دخول

الحزمة وفيما بعد تسجل زمن خروجها من الموجه ويسمى هذا الفرق بزمن الزيارة وهو الزمن الذي تمكنه الحزمة في الصف بالموجه، اذا كان زمن الزيارة أكبر من الزمن 5ms (الزمن الهدف) فإن الخوارزمية تحس بأن هناك تأخير ثم تبدأ بالتصرف حيال ذلك، يتحول الزمن 5ms في الصف الى عدد حزم متناسب مع سرعة خط الدخل الى الموجه Access link فاذا كانت سرعة الخط 2Mbps والتأخير على خط الدخل 20ms فإن عرض النطاق لخط الدخل 3 حزم  $(2Mbps * 20ms / 1500 * 8)$  اي ان هناك ثلاثة حزم تنتشر على امتداد خط الدخل، (او بكلمات اخرى هناك حزمه واحدة تدخل الى صف الموجه كل 7ms تقريباً  $(20ms/3)$ ،، واذا كان الاتصال وحيداً فإن ذلك لن يؤثر على عمل خوارزمية التحكم بتأخير الصف Codel باعتبار ان سعة الصف تساوي 20 حزمه، فاذا تم زيادة عدد الاتصالات الى 10 اتصالات وهذه الاتصالات ترسل حزمها بصورة متزامنة فإن صف الموجه سوف يستقبل 10 حزم لكل 7ms وهذا ايضاً لن يؤثر على عمل خوارزمية التحكم بتأخير الصف Codel بمعنى ان ورود 10 حزم (10 حزم كل 7ms) في آن واحد لن يؤدي الى ان يمتلئ صف خوارزمية التحكم بتأخير الصف Codel ويقوم بالاسقاط من آخر الصف ومن جهه اخرى فإن الخوارزمية سوف تكتشف ان هناك بواذر إزدحام وذلك بعد ان تكتشف ان زمن الزيارة أكبر من 5ms ولكن هذا لا يعني ان تقوم الخوارزمية بالاسقاط مباشرة فهي لن تقوم بالاسقاط الا بعد 100ms (ثابت في خوارزمية التحكم بتأخير الصف Codel) وان يكون زمن الزيارة مازال اكبر من 5ms, مثال آخر لسرعة خط الدخل في هذه الحالة نفترض ان سرعة خط الدخل هي 300Mbps، وذلك يعني ان هناك 500 حزمه تنتشر على خط الدخل كله  $(300Mbps * 20ms / 1500 * 8)$ ، ويعنى ايضاً ان هناك أكثر من حزمه واحدة تدخل الى صف

الموجه (بالضبط  $20\text{ms}/500=0.4$ ) لكل  $0.4\text{ms}$  كمثال اذا كان هناك 10 اتصالات متزامنة يؤدي ذلك الى دخول 4 حزم كل  $1\text{ms}$  وذلك لن يؤثر على عمل خوارزمية التحكم بتأخير الصف Codel ولكن اذا كان عدد الاتصالات اكبر من 10 اتصالات كمثال اذا كان عدد الاتصالات 30 اتصال فإن ذلك يعنى دخول  $0.4\text{ms} * 30 = 12$  حزمه كل  $1\text{ms}$  و 60 حزمه كل  $5\text{ms}$  واذا كان طول الصف الكلى لخوارزمية التحكم بتأخير الصف Codel 20 حزمه فإن الصف فى هذه الحالة سيمتلئ تماماً ويقوم باسقاط الحزم من آخر الصف قبل ان تكتشف خوارزمية التحكم بتأخير الصف Codel ان هناك بادرة إزدحام، استخدمنا كلمة بادرة ازدحام لان خوارزمية التحكم بتأخير الصف Codel عندما تحس بالازدحام وذلك عند زيادة زمن الزيارة للحزمه عن  $5\text{ms}$  لاتقوم بالاسقاط مباشرة وتعتبر ذلك إشارة او بادرة للازدحام وتقوم بالاسقاط بعد  $100\text{ms}$  وذلك اذا استمر زمن الزيارة اكبر من  $5\text{ms}$ .

### 3.6.3 حساب زمن الزياره

تقوم خوارزمية التحكم بتأخير الصف Codel بحساب زمن الزياره لضمان قيمة تأخير معقوله في صف الموجه, ولكن لحساب هذه القيمة يقوم الموجه بحساب زمن الدخول وزمن الخروج لاي حزمه ليتمكن من حساب زمن الزياره الذي يساوي زمن الخروج مطروحا منه زمن الدخول, بعد حساب زمن الزياره يستطيع الموجه ان يتخذ قرار باسقاط الحزمه او عدم اسقاطها, تكمن المشكله في ان العملية الحسابيه لزمن الزياره تستهلك معالج الموجه Codel حيث انه يحسب لكل الحزم الداخله للموجه [38], بالرغم من ان خوارزمية التحكم بتأخير الصف Codel تمكنت من ضمان قيمة تأخير جيده

لعمل البروتوكولات الا انها لا يمكن تطبيقها بصورة عملية في الموجهات الحقيقيه وذلك بسبب العمليات الحسابيه التي تقوم بها الخوارزميه وذلك لحساب زمن الزياره.

### 7.3 خلاصة

في هذا الباب تناولنا خوارزميه التحكم بتأخير الصف Codel وهي من الخوارزميات المهمه العامله في مجال التحكم بالازدحام والتي تعتمد في عملها على حساب التأخير في الصف, حيث تم التعرف علي طريقه عملها بالتفصيل, تناولنا تأثير هذه الخوارزميه على العدالة بين الاتصالات وعلى معدل الاستفاده من عرض النطاق, وتأثيرها ايضا على معدل الاسقاط, وخلصنا الى ان هناك عدة مشاكل لا يمكن تجاوزها عند محاوله تطبيق هذه الخوارزميه بصورة عملية وهذه المشاكل هي مشكله العدالة بين الاتصالات, مشكله سرعة خط الدخل acces link والمشكله الكبيره وهي مشكله حساب زمن الزياره لكل حزمه, حيث انها السبب الاساسي في منع هذه الخوارزميه من التطبيق العملي في الموجهات الحاليه, والذي ايضا بسببه قمنا بتطوير الخوارزميه الجديده خوارزميه التقدير الذاتي لطول الصف auto, الباب القادم سيتناول خوارزميه التقدير الذاتي لطول الصف بغرض تقليل التأخير وهذه الخوارزميه هي موضوع هذه البحت.

## الباب الرابع

### 1.4 خوارزمية التقدير الذاتي لطول الصف

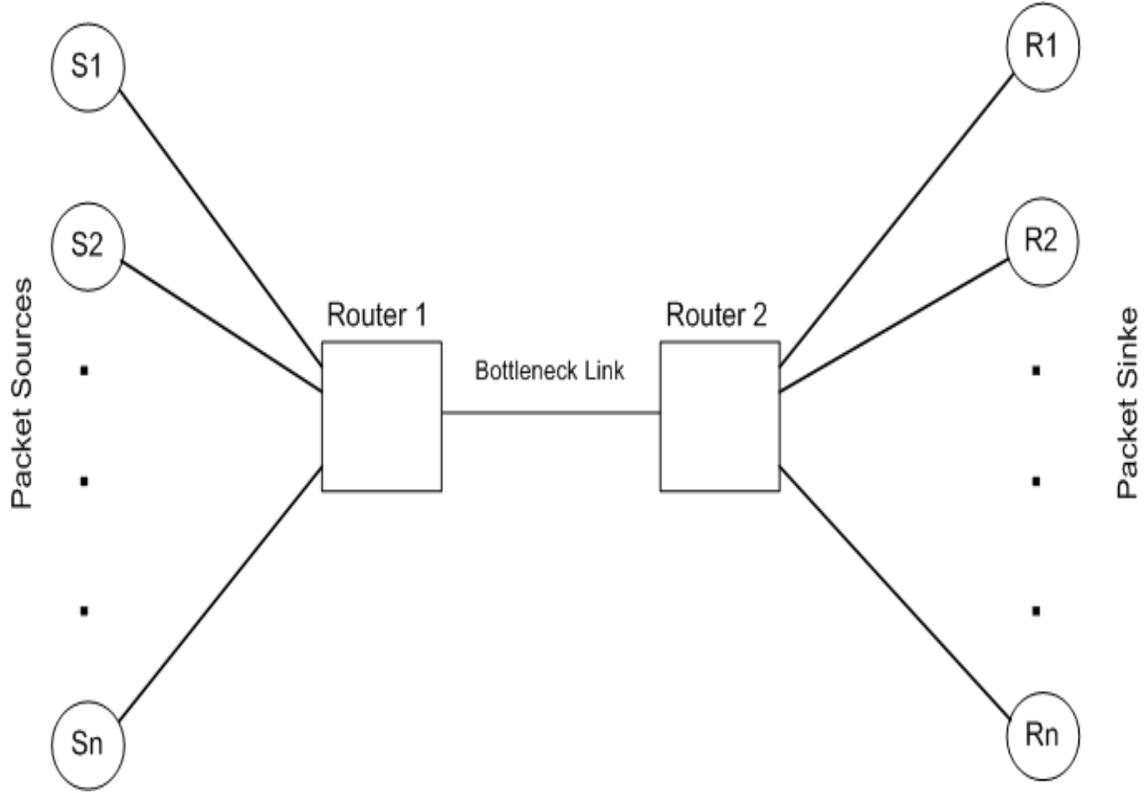
تعتمد خوارزمية الصفوف القصيرة على المعادلة (1-4) لتقدير طول الصف المناسب للعمل عليه, B هو طول الصف والمعامل RTT هو زمن وصول الحزمة وعودة الاقرار عليها و Bandwidth هو عرض نطاق عنق الزجاجة و n هو عدد الاتصالات التي تمر في الخط, تم وضع RTT وهي القيمة المتوسطة لزمن ارسال الحزمه وعودة الاقرار عليها, اما عرض نطاق Bandwidth فهو قيمة ثابتة عباره عن عرض نطاق عنق الزجاجة, المعامل الذي يصعب تقديره هو عدد الاتصالات n التي تتشارك في القناة وبالتالي لابد من إيجاد طريقة ما تمكن من تقدير عدد الاتصالات, سوف يتم تقدير عدد الاتصالات بصورة احتمالية من الحزم الوارده, هذه الفكرة مأخوذة من خوارزمية سابقة هي الدراسة SRED [51] والتي تستخدمها لتحديد الاتصالات الاكثر استخداماً للقناة ومعاقبته.

$$B = \frac{RTT * Bandwidth}{\sqrt{n}} \quad (1-4)$$

اخترت هذه الطريقة وهي تقدير عدد الاتصالات بصورة احتماليةً من الحزم الواردة وذلك لان السبب الذي جعل خوارزمية التحكم بتأخير الصف CodeI لا يمكن تطبيقها عمليا هو انها تطلب من الموجه ان يقوم بحساب زمن دخول وخروج اي حزمه ليحسب منه فيما بعد زمن الزياره, وهي عملية غير منطقيه وخاصه في الموجهات الكبيره والتي تتعامل مع عدد كبير من الحزم [38].

#### 1.1.4 بيئة الشبكة

الشكل رقم (4-1) يوضح تصميم الشبكة التي تم إجراء تجربته عليها, حيث تحتوي على موجهين متصلين عبر خط سعته 10Mbps وذو تأخير إنتشار يبلغ 50ms, خطوط إتصال المرسلين والمستقبلين مع الموجهين ذات عرض نطاق مختلف حسب نوع التجربة التي يتم اجراءها وذات تأخير ثابت يبلغ 20ms, المرسل S1 يقوم بالارسال الى المستقبل R1, المرسل S2 يقوم بالارسال الى المستقبل R2 وهكذا, يرسل المستقبلين حزم إقرار ACK [60] كما هو مطلوب من بروتوكول التحكم بالنقل TCP [25, 26] الى المرسلين, نافذة الازدحام لكل جهاز مرسل علي الشبكة هو  $16 * bdp$  اي عرض النطاق مضروب في الثابت 16 ليكون حجم النافذة كبير كفايه وذلك حتى لا يؤثر في اداء الجهاز المرسل, طول الحزمة هو الطول المعياري 1460 بايت , يحتوي كل موجه على خوارزمية التقدير الذاتي لطول الصف auto سيتم اجراء عدة تجارب على هذه الشبكة البسيطة وذلك بغرض معرفة قدرة الموجهات التي تحتوي هذه الخوارزمية على التحكم بالتأخير والمحافظة على عرض النطاق, عدد المرسلين 100 مرسل وعدد المستقبلين 100 مستقبل يعمل كل من المرسلين والمستقبلين على بروتوكول التحكم بالنقل TCP وتطبيق بروتوكول نقل الملفات FTP, اختير بروتوكول نقل الملفات FTP وذلك بسبب ان هذا البروتوكول يستخدم لنقل الملفات كبيرة الحجم, هذه الملفات الكبيره تستخدم لاشغال القناة المستخدمه في التجارب الى اقصى حد ممكن, وذلك لدراسة الخوارزمية ومعرفة استجاباتها تجاه الازدحام المسبب للتأخير, الملحق (1.8) يحوي البرنامج المكتوب بلغة TCL والذي يوضح شكل ومعاملات الشبكة التي اجريت عليها التجربة.



الشكل 1-4 شبكة تحتوي على موجهين وبينهما رابط عنق الزجاجة

## 2.4 طريقة عمل خوارزمية التقدير الذاتي لطول لصف auto

تقوم خوارزمية التقدير الذاتي لطول الصف بتقدير طول الصف المناسب للعمل عليه عن طريق المعادلة رقم (1-4) والمستعملة في خوارزمية الصفوف القصيرة حيث ان خوارزمية الصفوف القصيرة هي الخوارزمية المناسبة للعمل في عالم اليوم المتمسم بالتأخير المتسبب في مشكلة تضخم الصفوف Bufferbloat ولكن خوارزمية الصفوف القصيرة ينقصها عامل حاسم وهو تقدير حجم الصف المناسب بصورة آلية وحيث ان طول الصف في خوارزمية الصفوف القصيرة يتم تقديره من عدة معاملات هي عرض النطاق Bandwidth، والمعامل RTT وعدد الاتصالات n حيث ان

عرض النطاق Bandwidth هو معامل ثابت و RTT هو معامل متغير ولكن يتم أخذ قيمة ثابتة له في هذه الحالة وهي 100ms وذلك لمقارنة هذه الخوارزمية الجديدة مع خوارزمية التحكم بتأخير الصف CodeI اما المعامل الذي يجب تقديره هو عدد الاتصالات  $n$  التي تمر حالياً في القناة.

#### 1.2.4 تقدير عدد الاتصالات التي تمر بالموجة

يمكن للموجة ان يعرف عدد الاتصالات التي تمر به ولكن ذلك يتطلب منه جهد وذاكرة كبيرين يتمثل الجهد الكبير الذي يبذله الموجه في عدد دورات المعالج التي يحتاجها لتصنيف الحزم, حيث يحتاج الموجه لتصنيف الحزم بغرض معرفة عنوان المرسل وعنوان المستقبل وهذا عمل كبير سيكون من الصعب على الموجه القيام به بالاضافة الى عمله الاساسي وهو استقبال عدد كبير من الحزم وتوجيهها ويحتاج الموجه ايضاً الى ذاكرة كبيرة لتصنيف الحزم بهذه الطريقة, قبل ان تحسب خوارزمية التقدير الذاتي لطول الصف auto طول الصف تقوم اولا بتقدير عدد الاتصالات  $n$  الماره عبر الموجه, ثم بعد ذلك تقوم بحساب طول الصف من المعادله (4-1), يهتم هذه الجزء من الباب بتحليل ودراسة عملية تقدير عدد الاتصالات الحيه التي تمر عبر الموجه, توجد طريقه بسيطه لمعرفة عدد الاتصالات  $n$  بها يمكن تقدير عدد الاتصالات  $n$  احتمالياً من الحزم التي تمر بالموجه حيث انه وفي دراسة سابقة [51] طورت لمعرفة عدد الاتصالات التي تستهلك القناة اكثر من غيرها تقوم فكرة هذه الخوارزمية على إنشاء قائمة تحتوى الحزم الموجودة في الصف ويمكن ان تكون اكبر من ذلك لتحتوى الحزم الموجودة في الصف (الاحياء) وبعض الحزم التي تم ارسالها (الاموات) تسمى هذه القائمة باسم Zombie list قائمة الاموات الاحياء وذلك بسبب ان الحزم التي تم ارسالها

عبارة عن حزم ميتة (اموات) ولكن يحتفظ بها للقيام ببعض الاعمال التي تقوم بها هذه الخوارزمية وهي اولا ملء قائمة الاموات الاحياء ثم بعد ذلك مقارنة كل حزمه وارده مع حزمه من القائمة بصورة عشوائية فاذا تطابقت هاتان الرسالتان يسمى هذا (Hit) واذا لم يحدث تطابق يسمى هذا (Miss) المعادله (2-4) ويمكن في هذه الحالة ان تحل الحزمة الجديدة محل الحزمة القديمة بصورة احتمالية ايضاً وبما ان أي إتصال يمر بالموجه يمكن ان يكون له حزمه واحدة او عدة حزم في الصف، فإنه يمكن تقدير عدد الاتصالات من حالات التطابق التي تحدث، لايجاد عدد الاتصالات من حالات التطابق نفترض ان لدينا  $f(t)$  عبارة عن تقدير لحالات التطابق حول زمن وصول حزمه محدد للصف من المعادلة (3-4) حيث ان

$$\text{Hit}(t) = 0 \quad \text{if no hit}$$

$$\text{Hit}(t) = 1 \quad \text{if hit} \quad (2-4)$$

$$f(t) = (1 - \alpha) f(t - 1) + \alpha \text{Hit}(t) \quad (3-4)$$

اما  $\alpha$  فهو عبارة عن قيمه ثابتته  $0 < \alpha < 1$ , من المعادلة (4-4) يمكن إيجاد عدد الاتصالات التي تمر بالموجه, إثبات  $f(t)$  في الملحق (1.7) [61].

$$\text{Number of connections } (n) = 1/f(t) \quad (4-4)$$

يمكن استعمال عدد الاتصالات من المعادلة (4-4) مباشرة في المعادلة (1-4) والتي تستخدم في تقدير طول الصف المناسب.

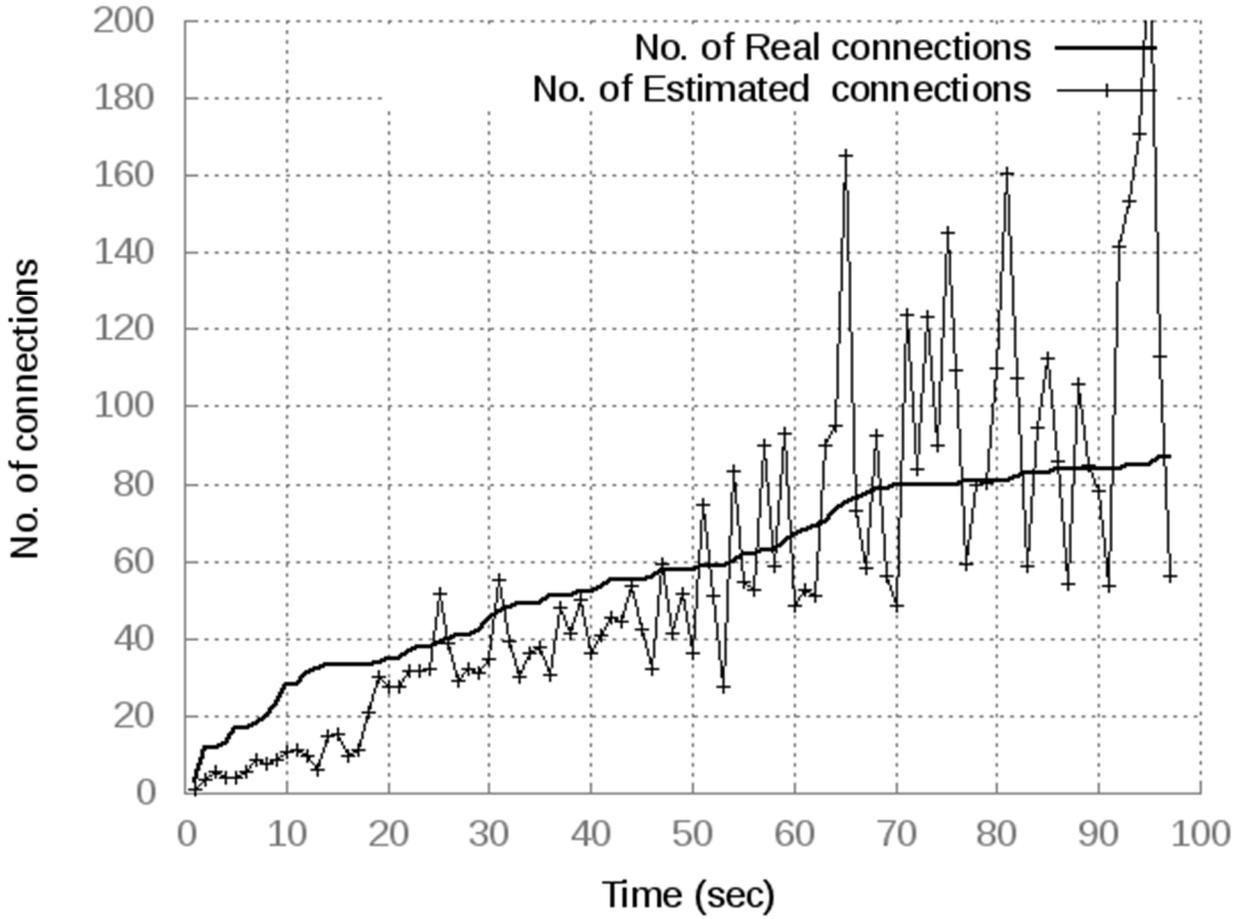
#### 1.1.2.4 تقدير عدد الاتصالات على المحاكى ns-2

تم برمجة خوارزمية التقدير الذاتي لطول الصف auto على المحاكى ns-2 (الملحق 3.7), حيث سيكون عمل الخوارزمية الاساسي هو تقدير طول الصف الذي سيعمل عليه الموجه ولكن قبل تقدير طول الصف في شفرة البرنامج توجد داله صغيره تقوم بتقدير عدد الاتصالات الحيه التي تمر بالموجه سنقوم في هذه الفقره بدراسة هذه الداله لمعرفة دقة عملها في تقدير عدد الاتصالات والذي تعتمد عليه خوارزمية التقدير الذاتي لطول الصف auto لتقدير طول الصف النهائي.

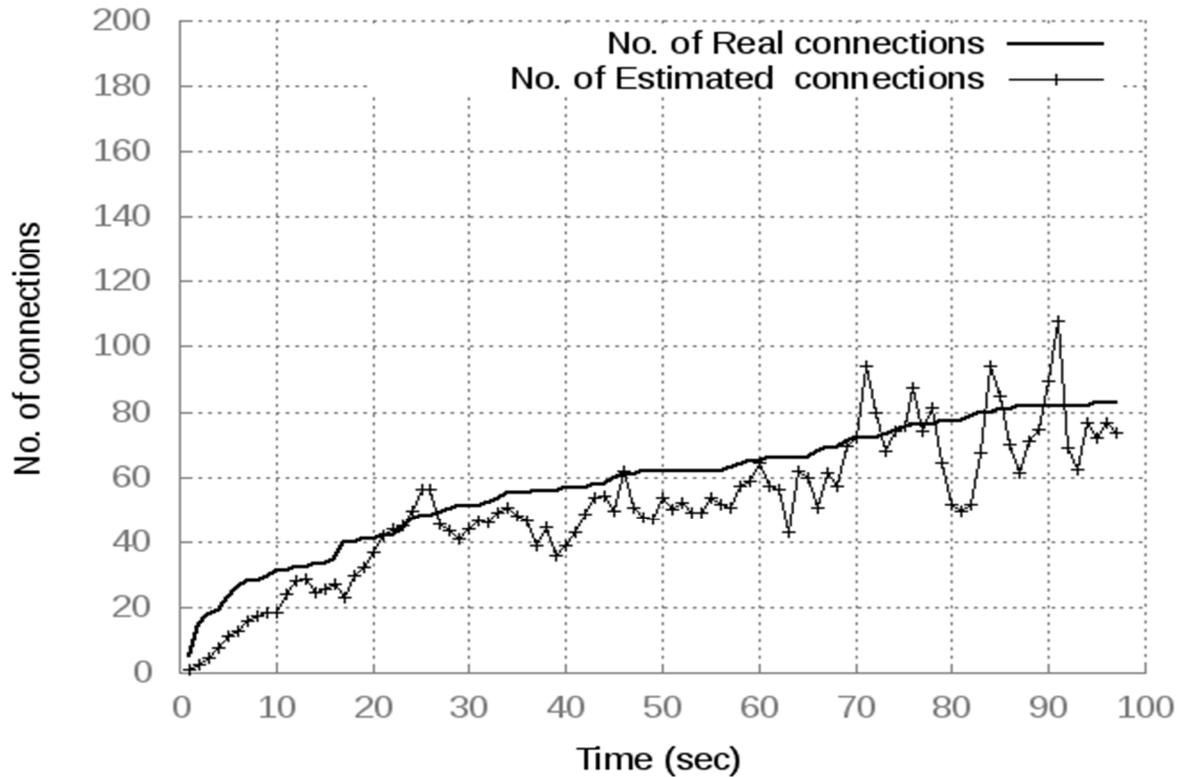
من طول قائمة الاموات الاحياء يتم حساب المعامل  $\alpha$  حيث ان  $M$  (طول قائمة الاحياء الاموات) يؤثر تأثير عكسي على المعامل  $\alpha$  حيث ان زيادته تعني نقص  $\alpha$  ونقصه يعني زيادة المعامل  $\alpha$ , المعامل  $\alpha$  يؤثر تأثير مباشر على تردد المطابقه  $f(t)$  من المعادله حيث ان دقة تقدير عدد الاتصالات من المعادله (4-4) يعتمد على المعامل  $\alpha$ , نستنتج من ذلك ان تقدير عدد الاتصالات بصوره جيده يعتمد مباشره على طول قائمة الاموات الاحياء *Zombie list*, وبالتالي لابد من جعل هذه القائمه طويله كفايه لتقدير عدد الاتصالات بصوره جيده.

تم اجراء تجربه على المحاكى ns-2 لمعرفة تأثير طول قائمة الاموات الاحياء *Zombie list* على عملية تقدير عدد الاتصالات. تم اجراء التجربه على شبكتين متطابقتين في كل المعاملات ولكن جعل طول قائمة الاموات الاحياء في التجربه الاولى مساوي لطول صف الموجه وفي التجربه الثانيه تم جعل طول قائمة الاموات الاحياء مساوي لاربعة اضعاف طول صف الموجه, في التجربه الاولى لم تستطع الخوارزمية تقدير عدد الاتصالات بصوره جيده وذلك موضح في الشكل رقم (2-4) اما في التجربه

الثانيه وبعد زيادة طول قائمة الاموات الاحياء كان تقدير عدد الاتصالات اكثر دقة ومتناسقا مع عدد الاتصالات الحقيقي المار عبر الموجه وذلك موضح في الشكل رقم (3-4).



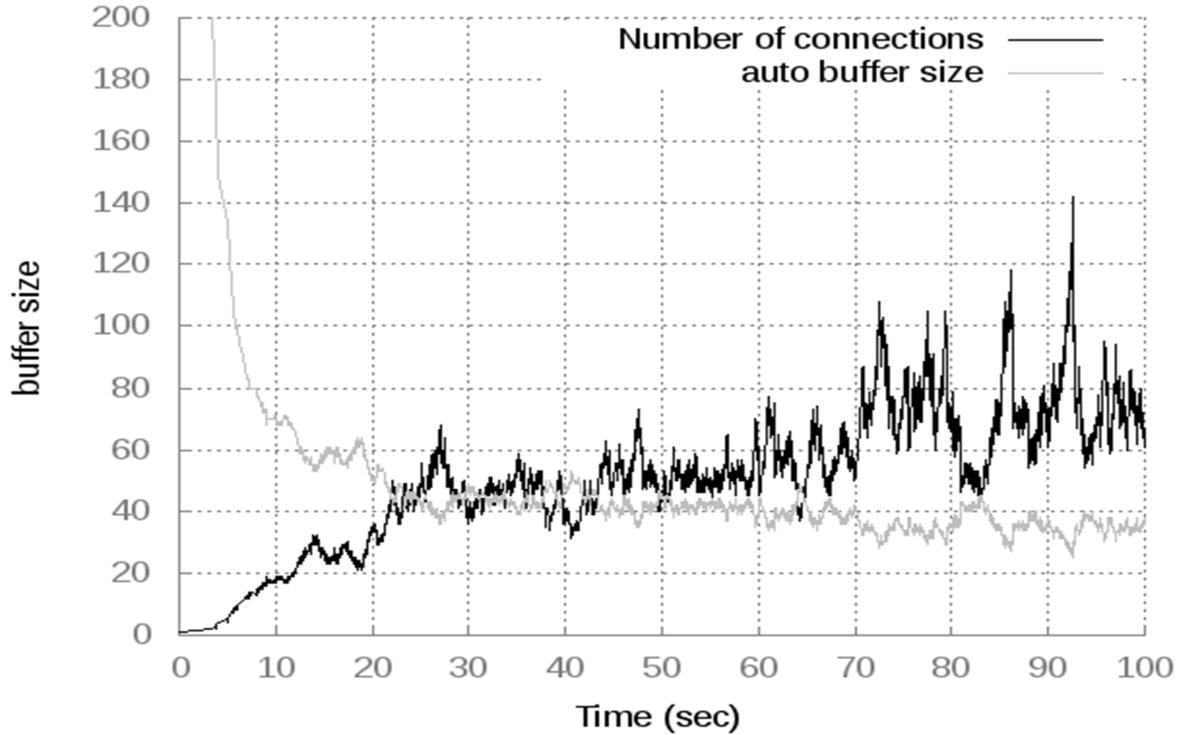
الشكل 2-4 عدم تناسق عدد الاتصالات المقدر مع العدد الحقيقي في auto عندما يكون M صغيرا



الشكل 3-4 تتاسق عدد الاتصالات المقدر مع العدد الحقيقي في auto عندما زيادة M

في الشكل رقم (4-4) المحور الافقي يمثل الزمن والمحور الرأسي يمثل طول الصف المخصص من بداية التجربه للموجه وهو 200 حزمه, تم اجراء التجربه عن طريق زيادة عدد الاتصالات تدريجيا ابتداءا من اتصال وحيد وحتى 100 اتصال كما هو موضح في الشكل, عندما كان هناك اتصال وحيد كان طول الصف 200 حزمه وعند الزمن 10 ثانيه زاد عدد الاتصالات مما ادى الى ان يقل طول الصف الى 40 حزمه, وعند الزمن 30 ثانيه وبعد زيادة عدد الاتصالات مره اخرى قل طول الصف الى 20 حزمه فقط ثم استقر طول الصف عند قيمه تتراوح حول القيمه 10-20 حزمه, الشكل رقم (4-4) يوضح امكانية تقدير طول الصف المناسب في خوارزمية التقدير الذاتي لطول الصف auto من عدد الاتصالات، حيث انه وفي البداية عندما يكون عدد الاتصالات قليلاً كان

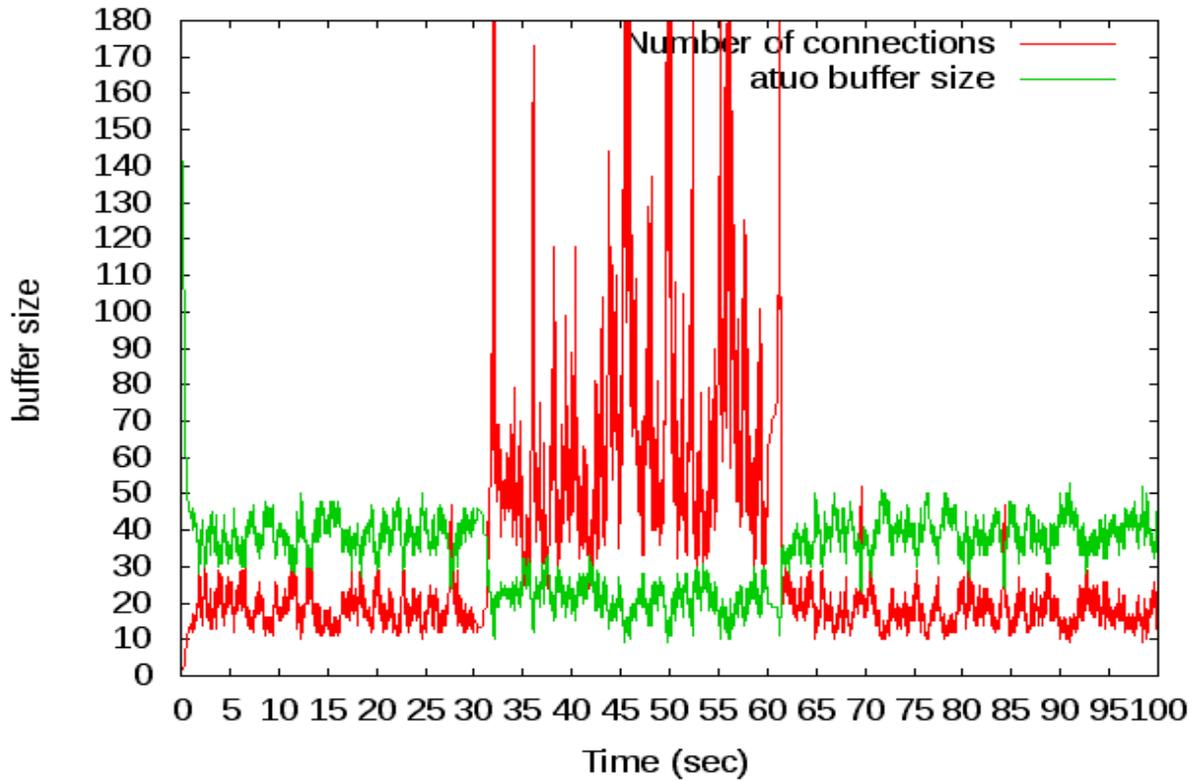
طول الصف كبيرا وذلك حسب القيمة الاساسيه لطول الصف والتي تم تخصيصها للموجه وعند زيادة عدد الاتصالات قل طول الصف.



الشكل 4-4 امكانية تقدير طول الصف المناسب من عدد الاتصالات المارة

في الشكل رقم (4-5) تم تغيير عدد الاتصالات من 20 اتصال الى 70 اتصال ومرة اخرى الى 20 لملاحظة اثر تغيير عدد الاتصالات على طول الصف, وجاءت النتائج مطابقه للتوقعات حيث انه عندما كان عدد الاتصالات 20 اتصال كان الصف طويلا وذلك في اول 30 ثانيه من زمن التجربه وعندما تم تغيير عدد الاتصالات الى 70 اتصال قامت خوارزمية التقدير الذاتي بتعديل طول الصف ذاتيا ليكون متناسبا مع عدد الاتصالات حيث يلاحظ من الشكل رقم (4-5) وعند الزمن 31 زاد عدد الاتصالات وبالتالي قللت الخوارزمية طول الصف بصورة ذاتيه وعند الزمن 61

ثانيه تم ارجاع عدد الاتصالات مره اخرى الى 20 اتصال حيث قامت خوارزمية التقدير الذاتي بسرعه بتعديل طول الصف حيث قامت في هذه المره بزيادة طول الصف ليكون متناسبا مع عدد الاتصالات القليل (20 اتصال) وذلك حسب خوارزمية الصفوف القصيره والتي تحسب من المعادله رقم (1-4).



الشكل 4-5 تحكم auto في طول الصف

تقدير طول الصف امر حيوي جداً ولا بد منه وذلك لضمان عمل الشبكة بصورة جيدة، في الخوارزمية الجديدة للتقدير الذاتي لطول الصف تم الاعتماد على المعادلة (1-4) والتي تقول ان هذا الطول

للصف هو الطول المناسب والذي يضمن تأخير قليل جداً وفي نفس الوقت لا يؤدي الى حدوث ضياع من عرض النطاق عنق الزجاجه والتي نحرض ان تكون مشغولة دائماً.

#### 2.2.4 تقدير طول الصف المناسب من غير فقد عرض النطاق

يعتبر العمل الاساسي لخوارزمية التقدير الذاتي لطول الصف auto هو تقدير طول الصف المناسب للموجه بحيث يحافظ على عرض النطاق عنق الزجاجه بصوره تامه, لمعرفة ان خوارزمية التقدير الذاتي لطول الصف auto تقوم بتقدير طول الصف المناسب للعمل عليه تم قياس عرض النطاق عنق الزجاجه للتأكد من ان الخوارزمية لا تتسبب في اهدار سعة هذه القناة ويقاس ايضا التأخير وهو العامل الهام والذي صممت الخوارزمية من اجله, حيث ان هذه الخوارزمية صممت بالاساس للتخلص من التأخير الزائد في صفوف الموجهات او بكلمات اخرى التخلص من تضخم الصفوف Bufferbloat في هذه الفقره تم اجراء ثلاثة تجارب, في التجربه الاولى جعلت سعة قنوات الدخل مساوي لعرض النطاق الخرج وفي التجربه الثانيه والثالثه تم مضاعفة قنوات الدخل مره واحده ومرتان على التوالي في كل تجربه من التجريبتين, الغرض من هذه التجارب هو التأكد من ان الخوارزمية تحافظ على مقدار تأخير قليل بالاضافه الى المحافظة على عرض النطاق عنق الزجاجه.

#### 1.2.2.4 تقدير طول الصف بواسطة خوارزمية التقدير الذاتي لطول الصف auto

طول الصف المقدر بواسطة خوارزمية التقدير الذاتي لطول الصف auto هو الطول المحسوب بواسطة المعادله (4-1) حيث ان B هو الطول المناسب والذي يحافظ على عرض النطاق عنق الزجاجة حسب الدرسة [21], في الفقرة السابقه تم تحليل قدرة خوارزمية التقدير الذاتي لطول الصف auto على تقدير عدد الاتصالات الحيه n الماره عبر الموجه حيث ان للخوارزميه قدره العاليه على تقدير عدد الاتصالات بصوره دقيقه وجيده, في هذه الفقرة تم تحليل قدرة الخوارزمية على تقدير طول الصف المناسب B والذي يحافظ على عرض النطاق عنق الزجاجة, اجريت عدة تجارب عملية على المحاكى ns-2 لمعرفة قدرة خوارزمية التقدير الذاتي لطول الصف auto على تقدير طول الصف المناسب والذي يحافظ على عرض النطاق عنق الزجاجة.

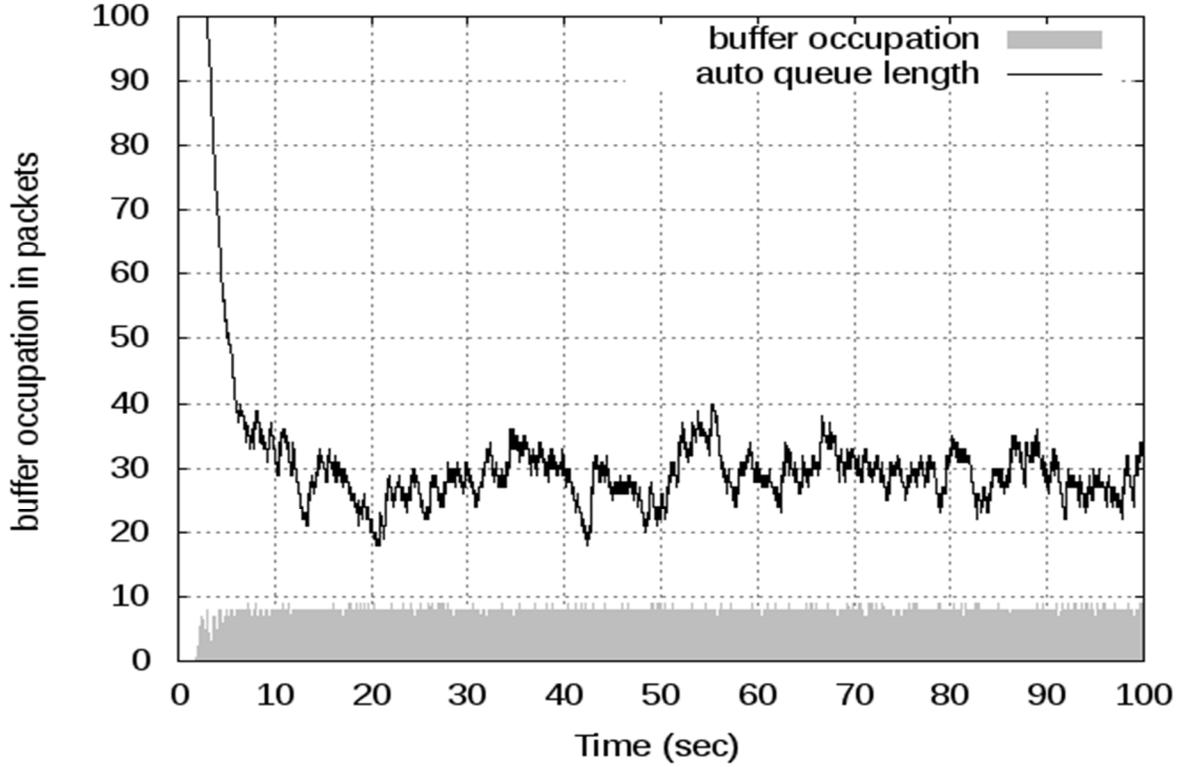
#### 1.1.2.2.4 التجريه الاولى

في التجريه الاولى والتي تهدف لمعرفة قدرة خوارزمية التقدير الذاتي لطول الصف auto على الاستفاده من عرض النطاق عنق الزجاجة, وذلك بعد تقدير طول الصف المناسب جعل عدد الاتصالات في التجريه 100 اتصال بروتوكول التحكم بالنقل TCP ذات خطوط اتصال access link حتى الموجه auto بسعة 100kbps وبالتالي فإن الدخل الكلي الى موجه auto يساوي مجموع عدد الاتصالات 100 اتصال مضروباً في سعة قناة الوصول access link والذي ينتج حمل كلي على الموجه يساوي 10Mbps, اولاً من الشكل رقم (4-6) الخط المتعرج العلوي يمثل

طول الصف B المقدر بواسطة خوارزمية التقدير الذاتي لطول الصف auto من المعادله (1-4),

اعلاه ومن التجريه فأن طول الصف B :

$$B = \left( \frac{2 * 180 \text{ms} * 10 \text{Mbps}}{\sqrt{100}} \right) / 8 * 1500 = 30 \text{ packets}$$

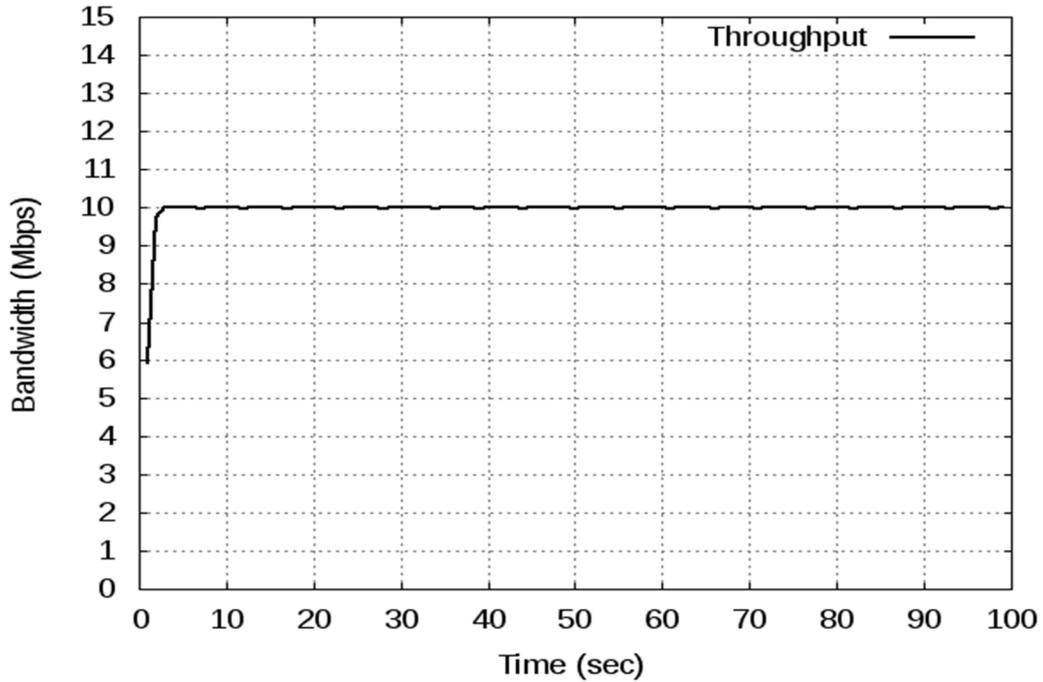


الشكل 4-6 طول الصف العامل المقدر بواسطة auto عند 100kbps

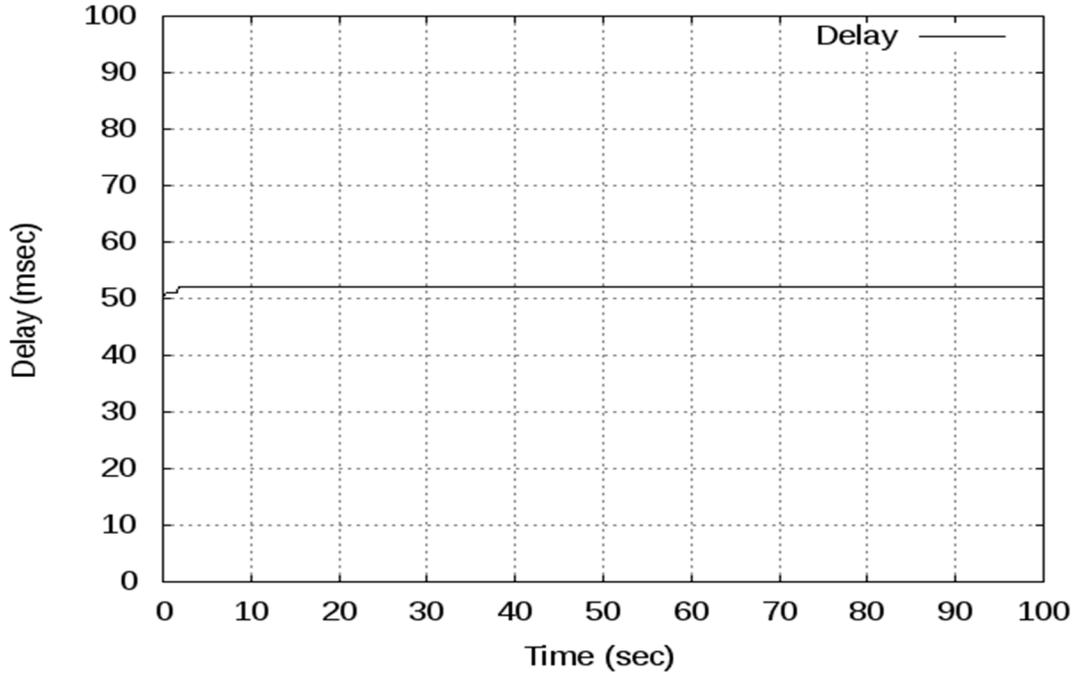
اي ان طول الصف المقدر يساوي 30 حزمه ولكن يظهر الشكل بخط متعرج لان عدد الاتصالات يتغير مع تغير الزمن وذلك لان عدد الاتصالات يقدر لحظيا من مجموع عدد الحزم التي تصل الى صف الموجه auto, المساحه المملوءه في الاسفل تمثل الجزء المملوء من صف الموجه البالغ طوله 30 حزمه, يلاحظ ان الجزء الممتلئ من الصف يتراوح ما بين 10 و 9 حزم وذلك بسبب ان مجموع

ساعات القنوات التي ترسل عليها الاتصالات البالغ عددها 100 اتصال تساوي 10Mbps وبالتالي فإن دخل مجموع هذه الخطوط يساوي خرج خط الموجه auto وهو 10Mbps وهي أيضا عرض النطاق عنق الزجاجة، يلاحظ من هذا الشكل ان تقدير الاتصالات لا يعتمد على مشغولية صف الموجه وانما يعتمد على عدد الاتصالات وبالتالي فان طول صف الموجه يحوم حول 30 حزمه، اما الطول المشغول فعليا من هذا الصف فهو يتراوح ما بين 10 حزم و9 حزم.

الشكل رقم (4-7) يوضح ان الخوارزمية تحافظ بصوره تامه على عرض النطاق عنق الزجاجة اما من الشكل رقم(4-8) فيتضح ان التأخير ذو قيمه ثابتة عند 50ms اي ليس هناك تأخير يذكر، لان التأخير اصلا هو 50ms حتى الموجه التالي، اي ان قيمة التأخير تزيد قليلا عن 50ms من الشكل رقم (4-8) يتضح ذلك بارتفاع الخط الافقي عن القيمه 50ms قليلا.



الشكل 4-7 الحيز المستفاد منه من عرض النطاق موضع الدراسه والبالغ 10Mb/s عند 100kbps



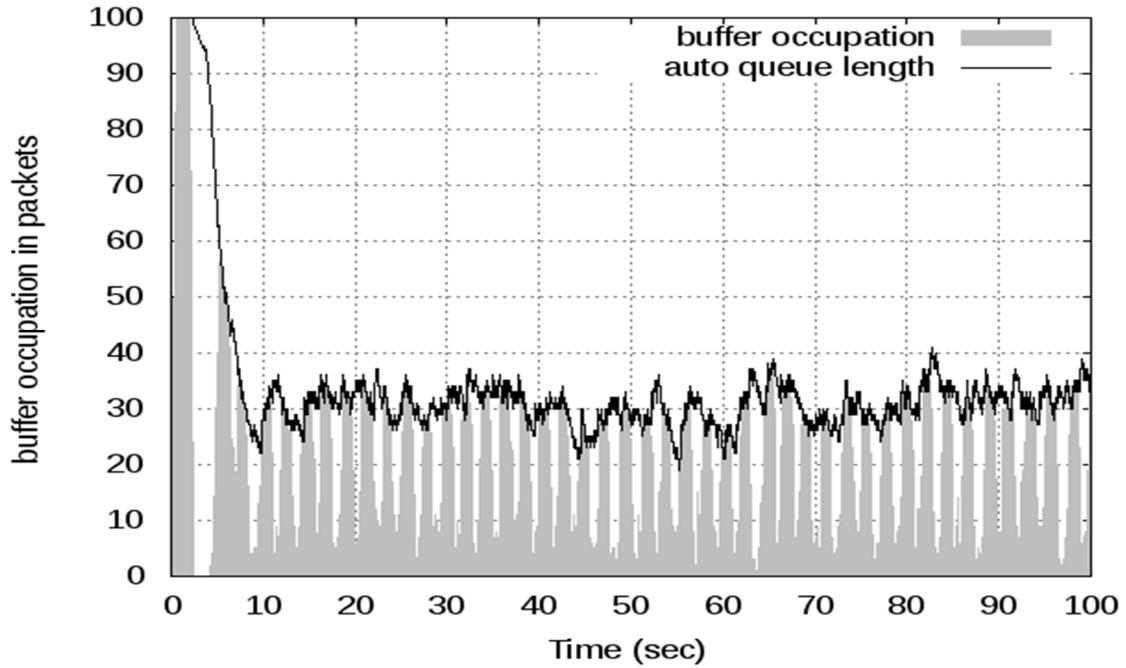
الشكل 4-8 مقدار التأخير الذي تتعرض له الحزم في خوارزمية auto عند 100kbps

#### 2.1.2.2.4 التجربة الثانية

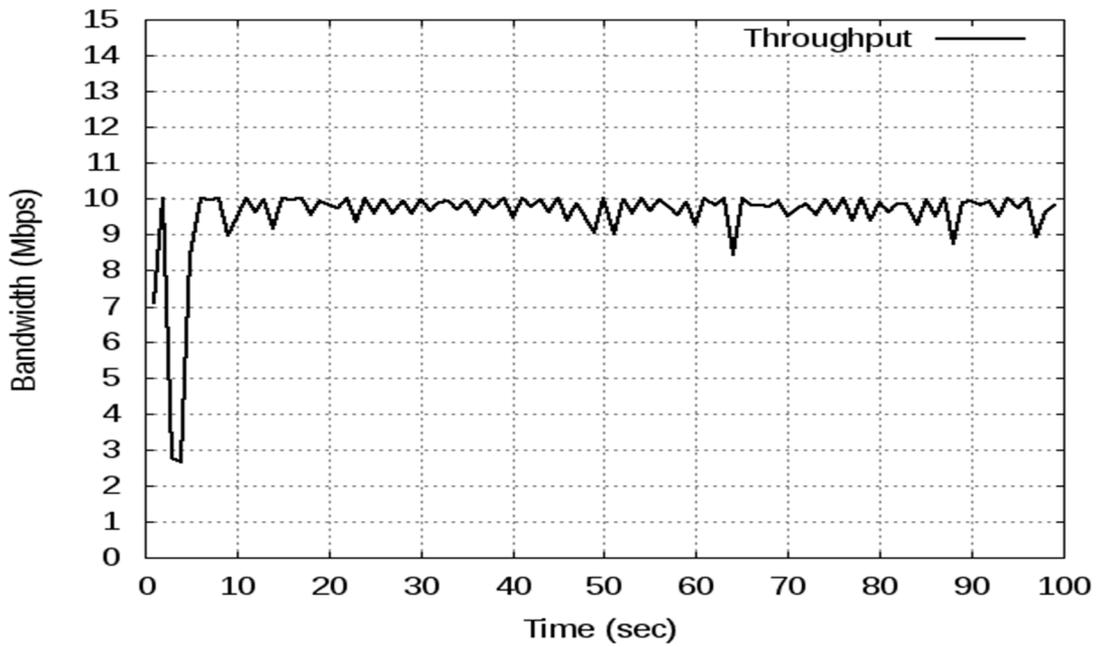
في التجربة الثانية والتي تهدف ايضا لمعرفة طول الصف المناسب والذي يمكن خوارزمية التقدير الذاتي لطول الصف auto من الاستفادة من عرض النطاق عنق الزجاجة, عدد الاتصالات 100 اتصال بروتوكول التحكم بالنقل TCP ذات خطوط اتصال access link حتى الموجه auto بسعة 200kbps اي اكبر بالضعف من الاتصالات في التجربة السابقه, وبالتالي فان الدخل الكلي الى موجه auto يساوي مجموع عدد الاتصالات مضروبا في سعة الوصول access link والتي تنتج 20Mbps, يلاحظ من الشكل رقم (4-9) ان الخط الاسود التخزين والذي يمثل الطول الكلي لصف الموجه auto المقدر بواسطة الخوارزمية, والحيز المملوء بواسطة اللون الرمادي يمثل الحيز المشغول من صف الموجه بواسطة مجموع عدد الاتصالات التي تقوم بالارسال الان, يلاحظ ان

الحيز المملوء بالرمادي لا يستطيع ان يتجاوز الخط الاسود التخين وذلك لان هذا الخط هو الحد الاعلى المسموح باشغاله حاليا من صف الموجه (عمليا يحتوي الموجه على طول اكبر من ذلك وهو بالضبط 200 حزمه) عندما يصل الخط الرمادي الى الخط الاسود فان ذلك يعني ان هناك حزم تصل الى صف الموجه ولا يستطيع الموجه تخزينها حسب خوارزمية التقدير الذاتي لطول الصف auto ولان ذلك يعني زيادة التأخير وبالتالي فان هذه الحزم يتم اسقاطها فورا, هناك بعض الفراغات بين الخطوط الرمادية المرتفعه والتي تمثل حيز من صف الموجه فارغ من الحزم, هذه الارتفاعات تعبر بصوره واضحه عن طريقة بروتوكول التحكم بالنقل TCP في الارسال وهي طريقة الارسال الانفجاري.

الشكل رقم(4-10) يوضح مقدار الاستفاده من عرض النطاق عنق الزجاجة والتي تتراوح ما بين 10Mb/s و 9.5Mb/s تقل الاستفاده بصوره ملحوظه في هذه التجريه عن التجريه السابقه وذلك بسبب ان هناك عدد كبير من الحزم تم اسقاطها والتي سوف تؤدي الى تقليل معدل الارسال حيث ان اي اتصال بروتوكول التحكم بالنقل TCP يتعرض لعملية اسقاط حزمه يقوم بانقاص معدل ارساله في المره القادمه وبالتالي فان معدل الاستفاده من عرض النطاق عنق الزجاجة يقل بمقدار ضئيل عما سبق.

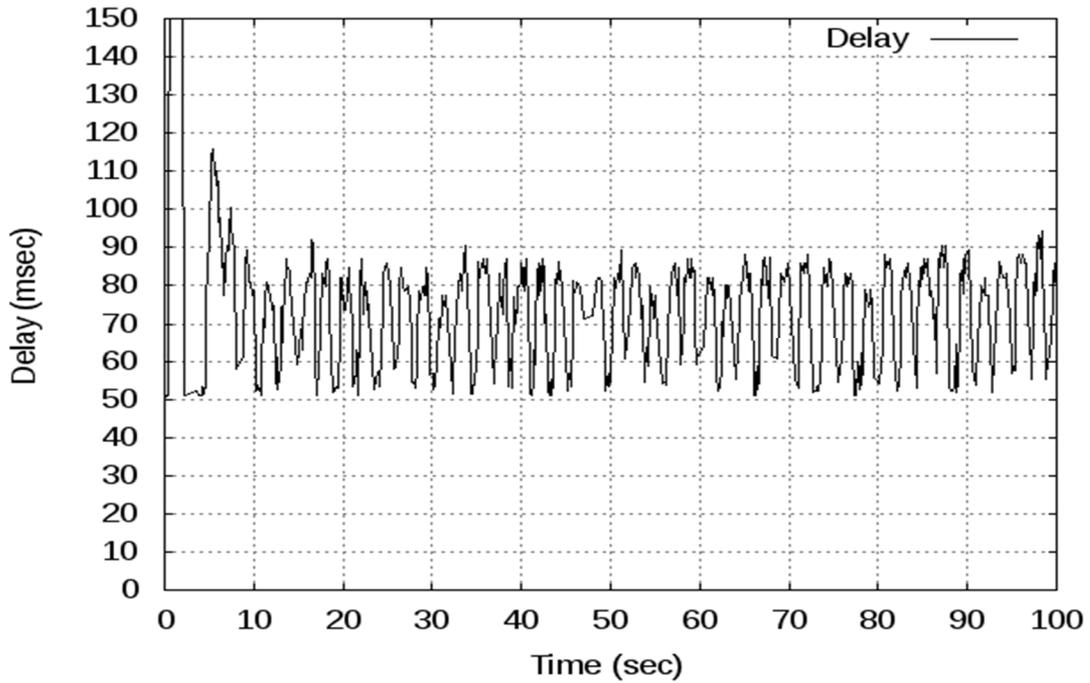


الشكل 4-9 طول الصف العامل المقدر بواسطة auto عند 200kbps



الشكل 4-10 الحيز المستفاد منه من عرض النطاق 10Mb/s عند 200kbps

الشكل رقم(4-11) يوضح مقدار التأخير الذي تتعرض له الحزم والذي يتراوح ما بين 50ms و 90ms اي ان التأخير الكلي ما بين 1ms وحتى 40ms حيث ان تأخير عنق الزجاجة 50ms, اقصى يسار الشكل رقم (4-11) يوضح ان هناك قيمة كبيره للتأخير والتي تصل الى 200ms تنتج هذه القيمة الكبيره للتأخير وذلك بسبب ان خوارزمية التقدير الذاتي لطول الصف auto في البدايه تحتاج الى زمن قليل لتقدير طول الصف المناسب وبالتالي فهي تعمل بالصف المعطى من قبل الموجه وهو 200 حزمه ولكن بعد زمن قليل جدا تتمكن من تقدير عدد الاتصالات الحيه وبالتالي تقوم بانقاص طول الصف العامل الفعلي وبالتالي يقل التأخير الذي تتعرض له الحزم نتيجة لوقوفها في الصف بانتظار الارسال, اي ان صف الموجه يسمح بتأخير اقصى للرسائل على الصف يصل هذه التأخير حتى قيمه 40ms ولكن لن يسمح بتجاوز هذه القيمه من التأخير مطلقا.

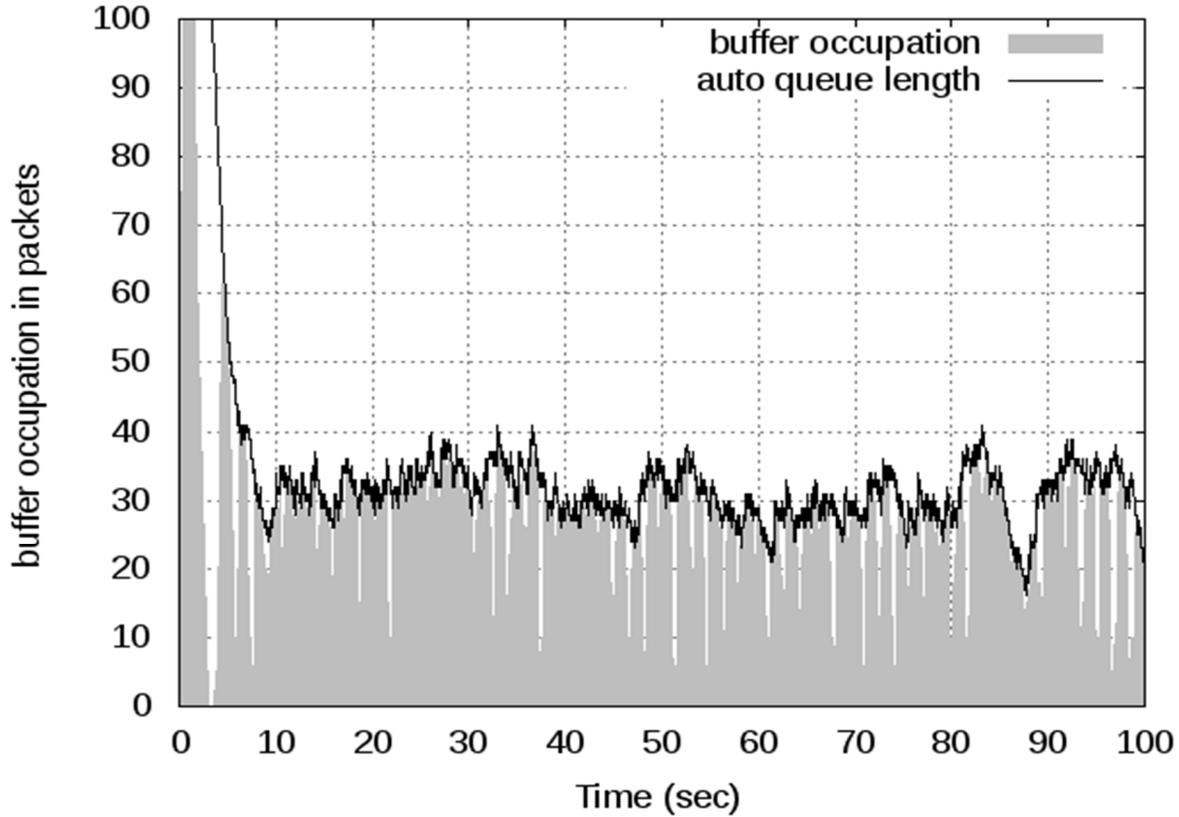


الشكل 4-11 مقدار التأخير الذي تتعرض له الحزم في خوارزمية auto عند 200kbps

#### 3.1.2.2.4 التجربة الثالثة

في التجربة الثالثة والتي ايضا تهدف لمعرفة طول الصف المناسب والذي يحافظ على قدرة خوارزمية التقدير الذاتي لطول الصف auto على الاستفادة من عرض النطاق عنق الزجاجة, عدد الاتصالات 100 اتصال بروتوكول التحكم بالنقل TCP ذات خطوط اتصال access link حتى الموجه auto بسعة 300kbps وبالتالي فان الدخل الكلي الى الموجه auto يساوي مجموع عدد الاتصالات والذي يساوي 100 اتصال مضروبا في عرض النطاق الوصول access link والتي تنتج القيمة 30Mbps من الشكل رقم (4-12) الخط العلوي باللون الاسود يمثل طول الصف B المقدر بواسطة خوارزمية التقدير الذاتي لطول الصف auto من المعادله (4-1) والجزء السفلي باللون الرمادي يمثل الحيز الممتلئ من صف الموجه auto, هناك طبعا بعض الفراغات في الجزء الممتلئ باللون الرمادي والتي تدل على ان الصف لم يمتلئ بصورة كامله عندما يرتفع الخط الرمادي حتى يلامس الخط الاسود فان ذلك يعني ان عدد الحزم التي تدخل الى الصف اكبر من طول الصف الفعلي وبالتالي فإن الحزم في هذه الحاله يتم اسقاطها, يتعرج الخط باللون الاسود بالرغم من ان عدد الاتصالات 100 اتصال بصورة ثابتة ولكن الخوارزمية تقوم بتقدير عدد الاتصالات من الحزم الواصله فعندما يتم اسقاط بعض الحزم من الصف فان ذلك يؤدي الى ان تقوم الاتصالات بتقليل ارسالها في المره القادمه حسب توجيهات بروتوكول التحكم بالنقل TCP وبالتالي فان عدد الحزم التي تصل في المره التاليه تؤدي الى تقليل الصف نسبة لتقليل عدد الحزم الواصله والتي منها يتم تقدير عدد الاتصالات وبالتالي تقليل طول الصف, تظهر هذه النقطه جليه في المنطقه التي تسبق

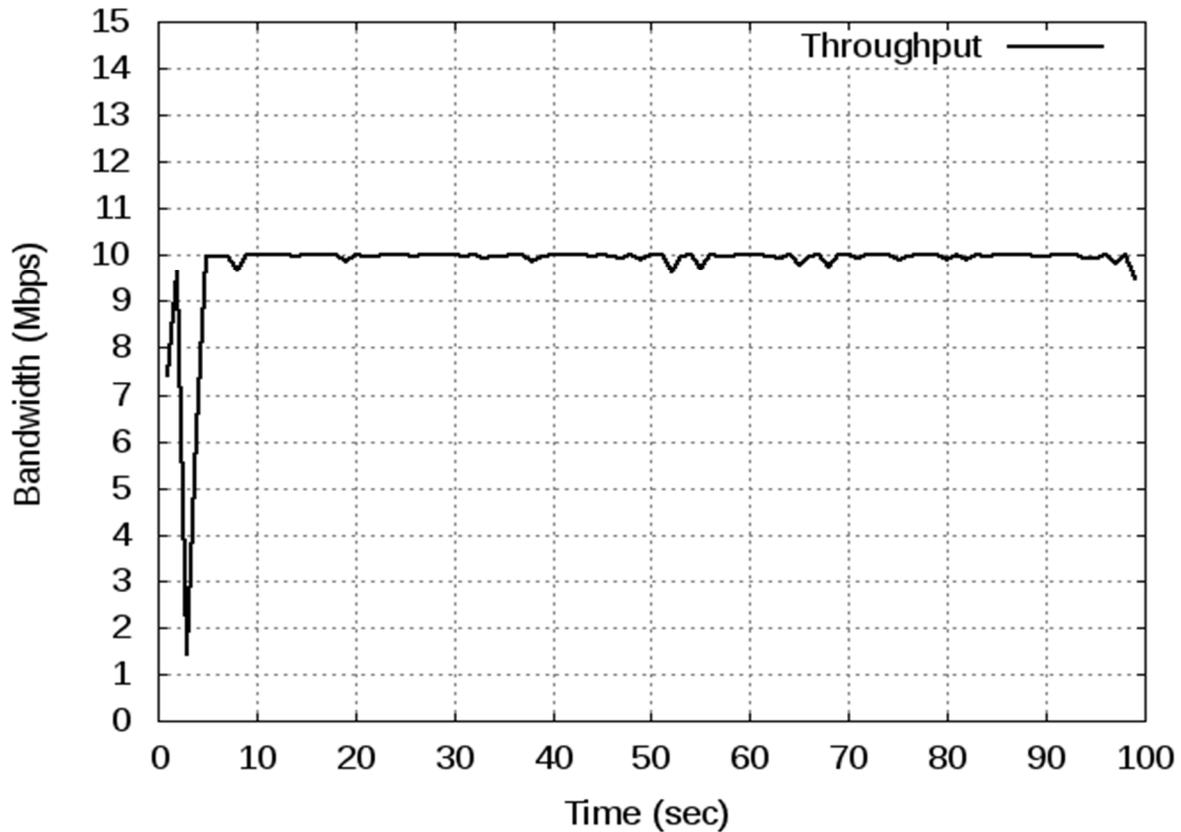
الزمن من 90s من الشكل رقم(4-13) يتضح ان هناك انهيار في طول الصف المقدر وذلك بسبب الاسقاط المكثف في المنطقه التي تسبق منطقه انهيار طول الصف.



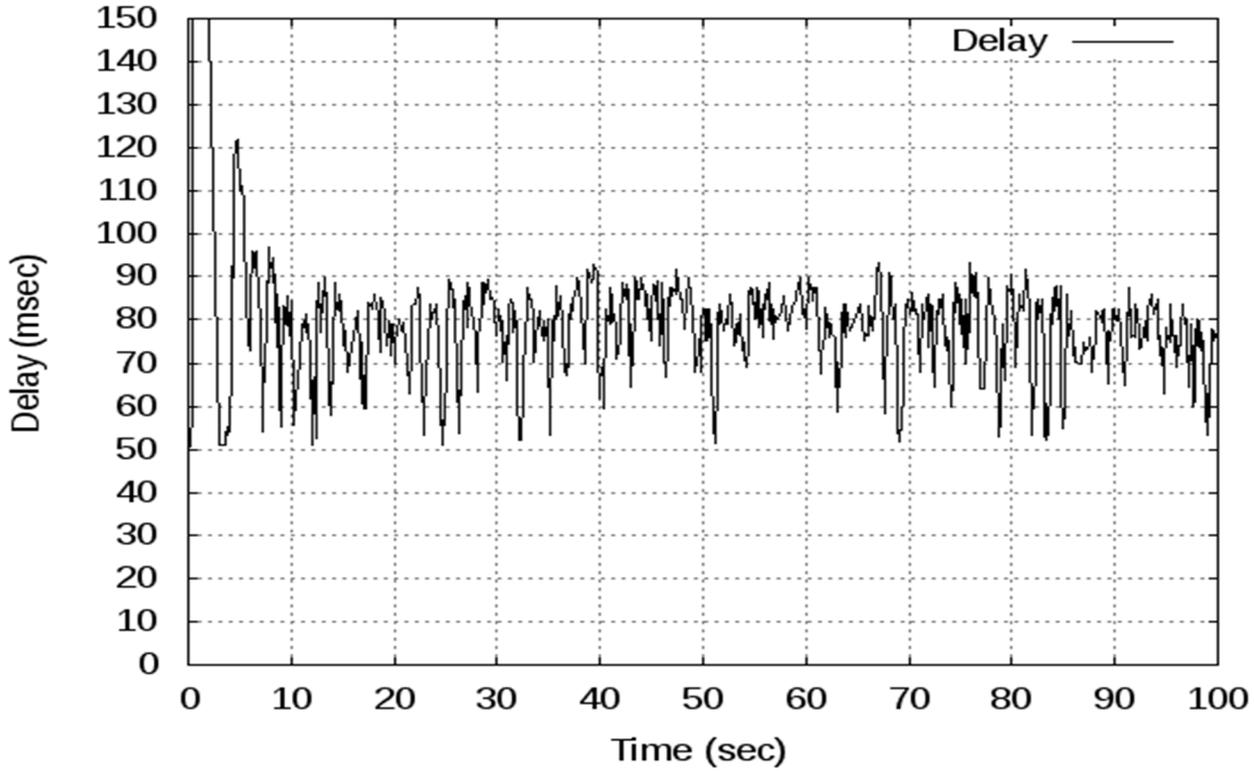
الشكل 4-12 طول الصف العامل المقدر بواسطة auto عند 300kbps

الشكل رقم(4-13) يوضح انه يتم المحافظة على عرض نطاق عنق الزجاجة بصورة عامه فيما عدا ان المنطقه الزمنيه الابتدائيه والتي تكون فيها الخوارزمية لم تقدر بعد طول الصف المناسب يحدث انهيار تام لعرض النطاق وذلك ايضا بسبب الجزء الاول من خوارزمية التحكم بالازدحام في بروتوكول التحكم بالنقل TCP وهي خوارزمية البدايه البطيئه Slow Start والتي يحدث فيها الكثير

من الاسقاط قبل ان تستقر اتصالات بروتوكول التحكم بالنقل TCP بعد ذلك في حيز ارسال مناسب مع سعة القنوات المتاحة, اما في الشكل رقم (4-14) والذي يوضح التأخير الذي تتعرض له الحزم فيلاحظ ان التأخير تناقصت قيمته قليلا ما بين 50ms و 80ms والذي يبلغ 30ms, اي ان التأخير الذي تتعرض له الحزم داخل صف خوارزمية التقدير الذاتي لطول الصف auto يتراوح ما بين 1ms وحتى 30ms ولكن بعد هذه التأخير لن يسمح لاي حزمه بالمكوث في صف الموجه وسيتم اسقاطها فورا.



الشكل 4-13 الحيز المستفاد منه من عرض النطاق 10Mb/s عند 300kbps



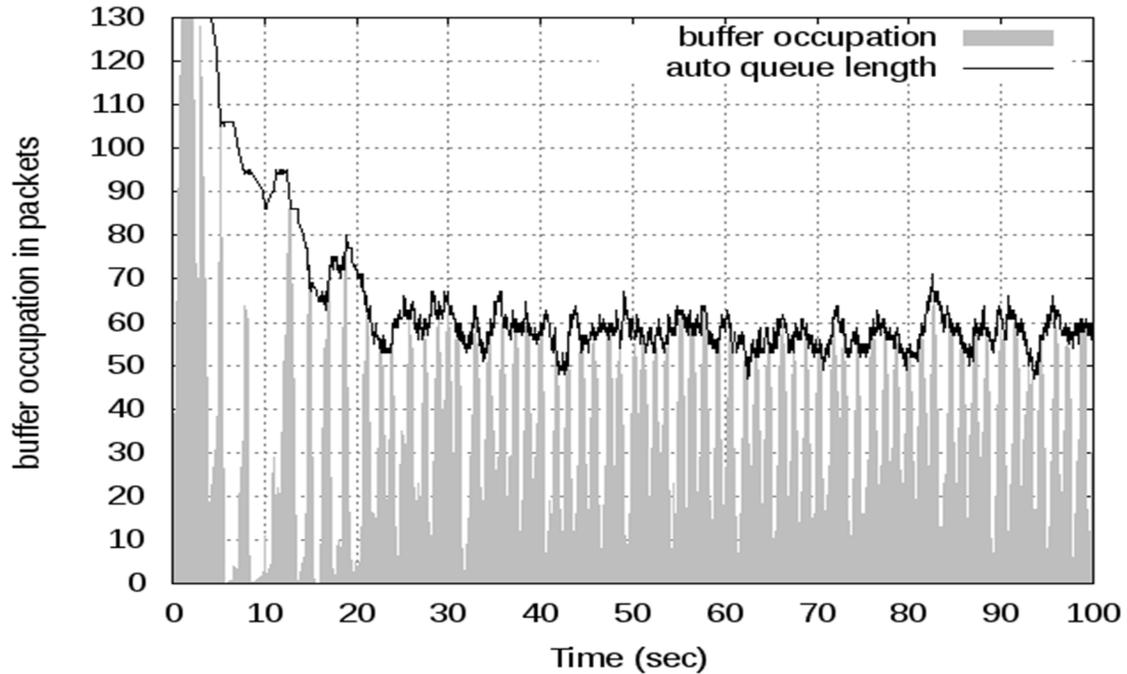
الشكل 4-14 مقدار التأخير الذي تتعرض له الحزم في خوارزمية auto عند 300kbps

### 3.4 تأثير طول الصف على عدد الاتصالات

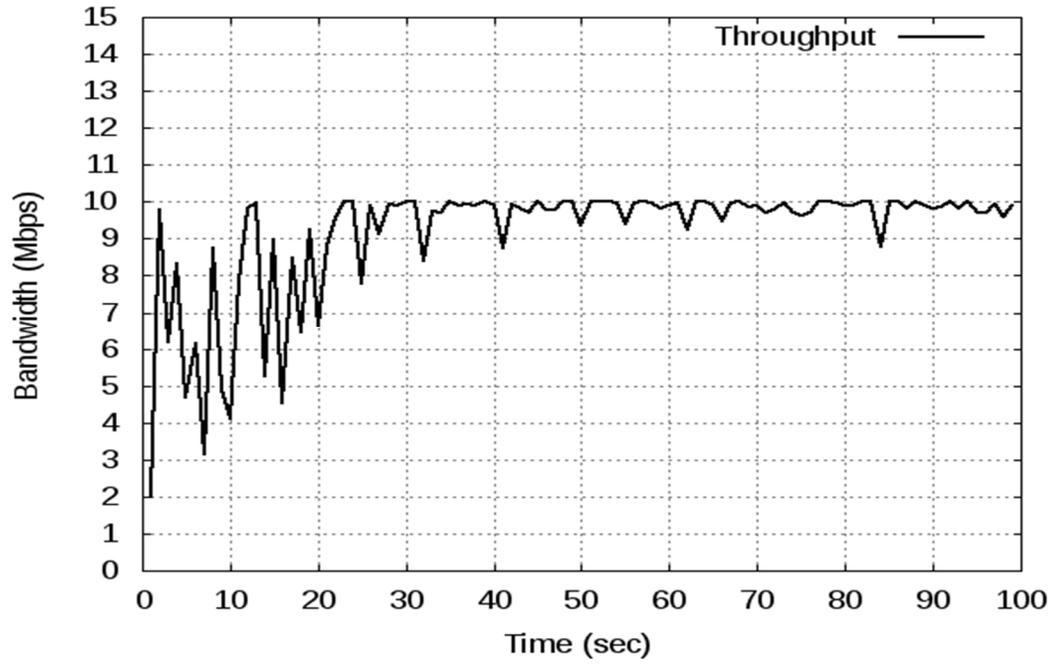
يمثل الصف الطويل على الموجه تأخير كبير، و يمثل الصف القصير تأخير قليل ولكن بالاضافه الى ذلك هناك مشكله اخرى هي تدني معدل الاستفاده من عرض النطاق. عندما يكون عدد الاتصالات قليل لابد ان يكون الصف طويل وذلك للاستفاده القصوى من عرض النطاق [4] وعندما يكون عدد الاتصالات كبير ليس من الضروري ان يكون الصف طويل، في هذه التجربه سنوضح قدرة خوارزمية التقدير الذاتي لطول الصف auto على تقدير طول الصف المناسب مع عدد الاتصالات المتغير، تم تقسيم التجربه الى ثلاثة اقسام مع تثبيت المعاملات التاليه لكل الاقسام،

طول الصف يساوي 150 حزمه حسب [4] وعرض النطاق 10Mbps والتأخير الكلي (RTT) يساوي 180ms وتأخير قناة عنق الزجاجة 50ms .

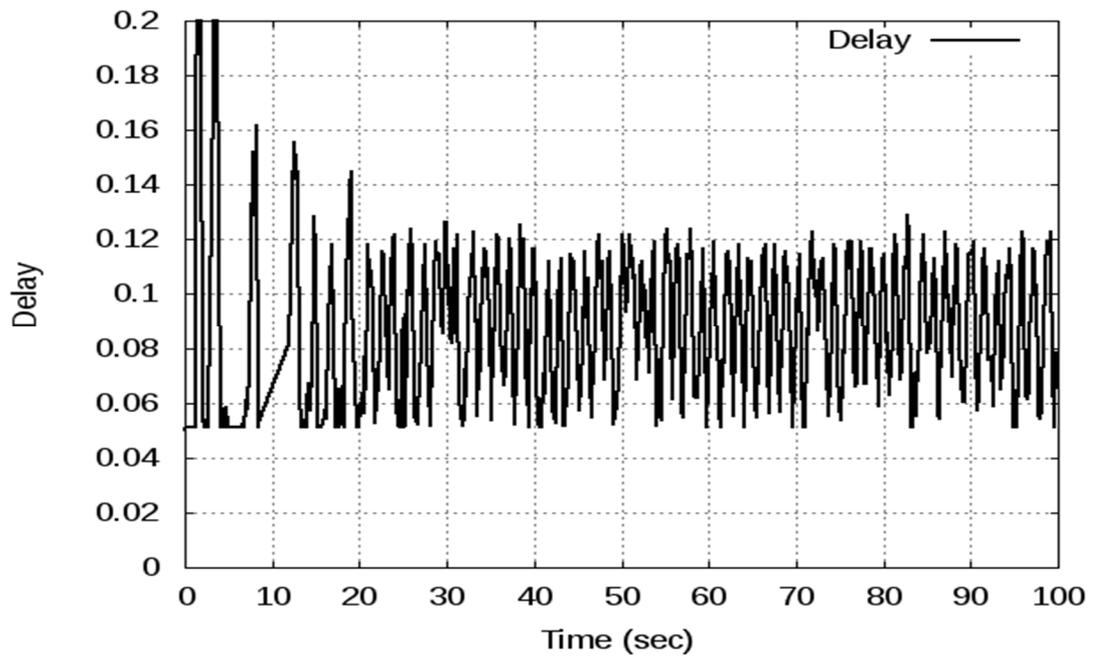
القسم الاول: تم وضع عدد الاتصالات في هذا القسم 30 اتصال, الشكل رقم (4-15) يوضح ان طول الصف المقدر بواسطة خوارزمية التقدير الذاتي لطول الصف auto يتراوح ما بين 50 و 70 حزمه وهو طول صف كبير مقارنة مع عدد الاتصالات القليل ولكن هذا الطول هو الطول المناسب للمحافظة على عرض النطاق عنق الزجاجة والموضحة في الشكل رقم (4-16) حيث ان معدل الاستفادة من عرض النطاق احسن مايكون, اما الشكل رقم (4-17) فيوضح التأخير الذي تتعرض له الحزم داخل الصف.



الشكل 4-15 الصف المقدر بواسطة auto والذي يتراوح ما بين 50 و 70 حزمه ليتناسب مع عدد 30 اتصال

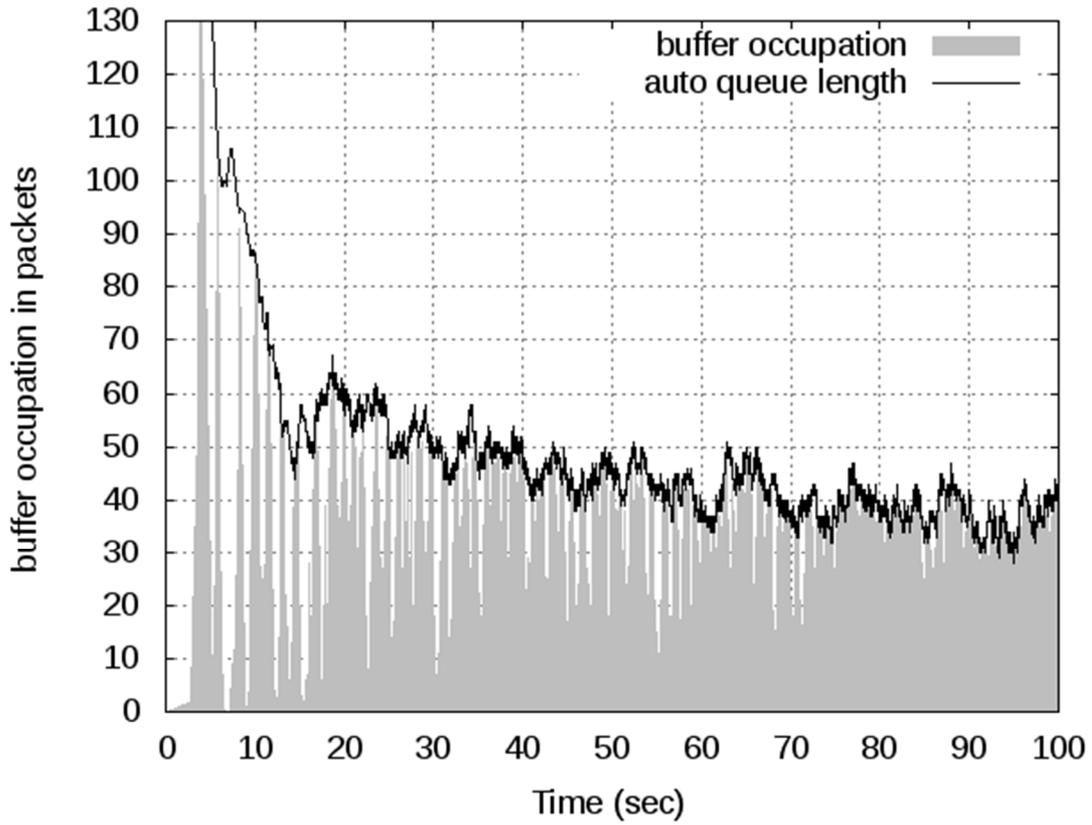


الشكل 4-16 خوارزمية auto تحدد الصف المناسب الذي يضمن الاستفادة من عرض النطاق عنق الزجاجة عند 30 إتصال

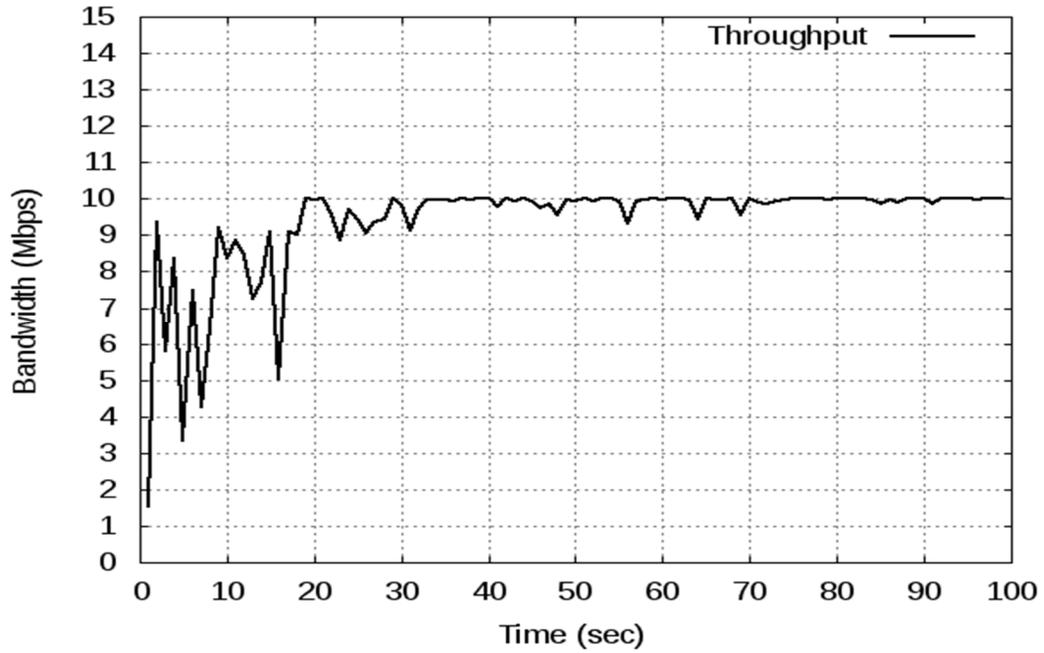


الشكل 4-17 مقدار التأخير الذي تتعرض له الحزم في auto عند 30 إتصال

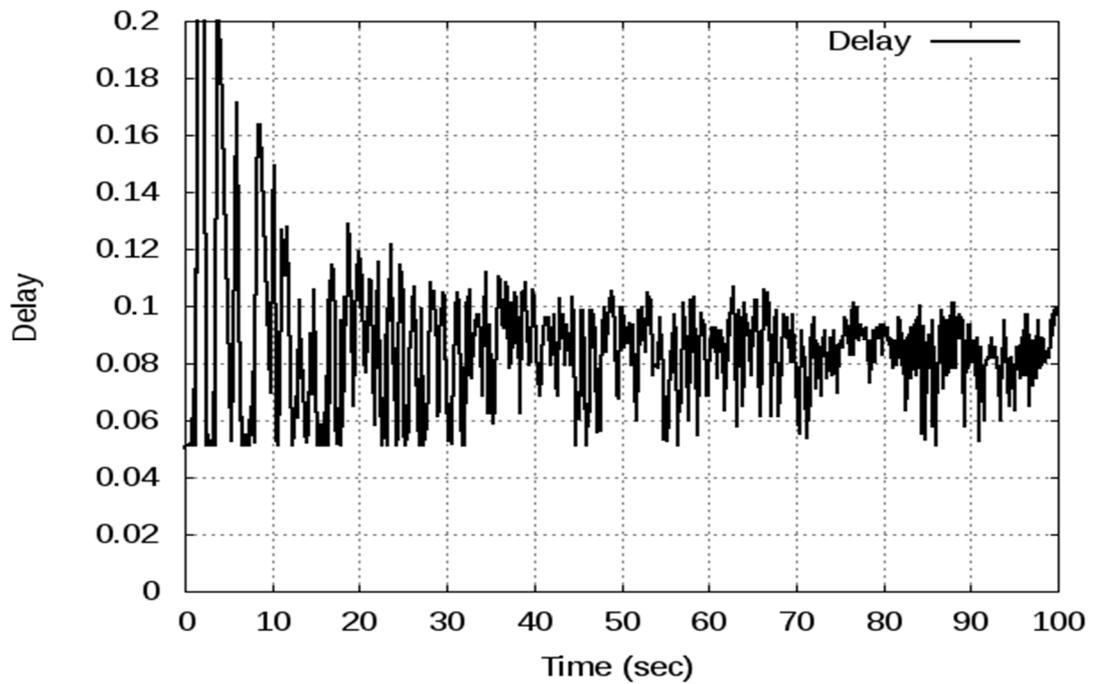
**القسم الثاني:** تم وضع عدد الاتصالات في هذا القسم 70 اتصال, الشكل رقم (4-18) يوضح ان طول الصف المقدر بواسطة خوارزمية التقدير الذاتي لطول الصف auto يتراوح ما بين 30 و 50 وهو طول مناسب مع عدد الاتصالات التي تدخل الى الموجه في تلك اللحظة, يؤدي هذا الطول الى المحافظة على الاستفادة من عرض النطاق والموضح في الشكل رقم (4-19) بالاضافه الى جعل التأخير والموضح في الشكل رقم (4-20) اقل مايمكن.



الشكل 4-18 الصف المقدر بواسطة auto والذي يتراوح ما بين 30 و 50 حزمه ليتناسب مع عدد 70 اتصال

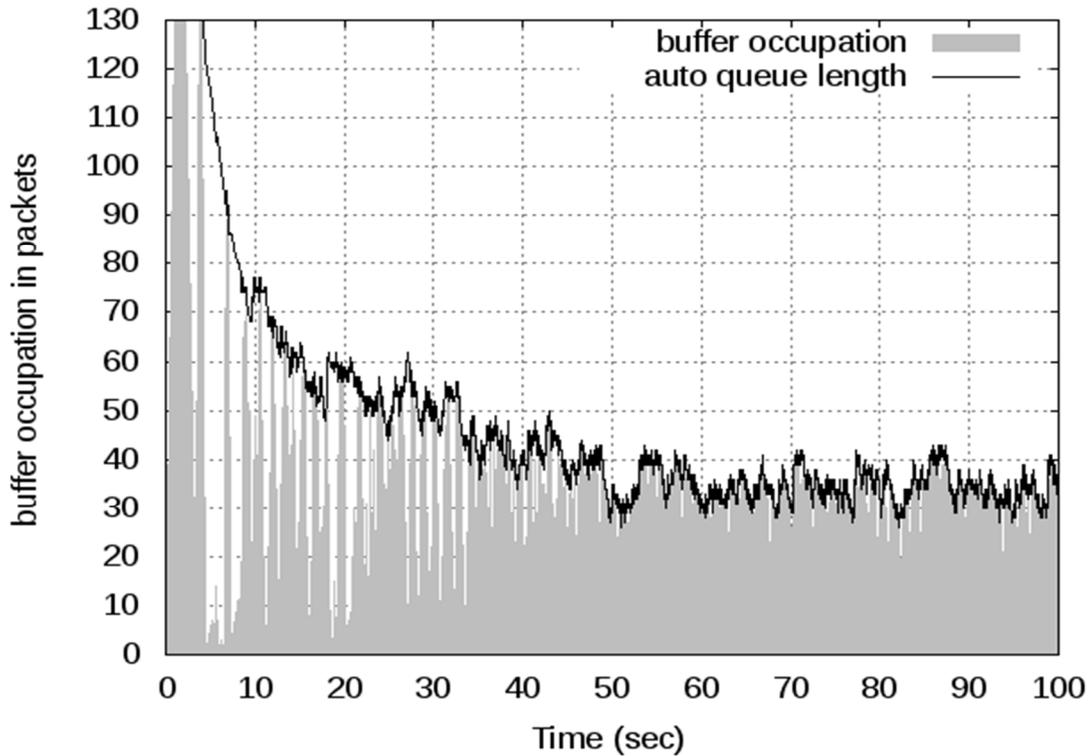


الشكل 4-19 خوارزمية auto تحدد الصف المناسب الذي يضمن الاستفادة من عرض النطاق عند 70 اتصال

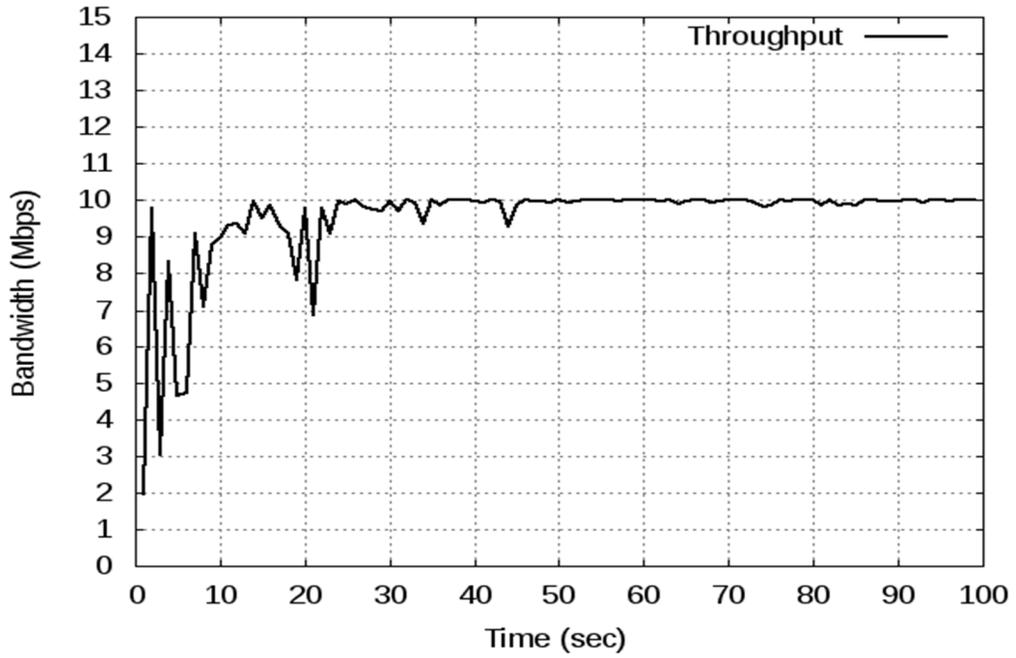


الشكل 4-20 مقدار التأخير الذي تتعرض له الحزم في auto عند 70 إتصال

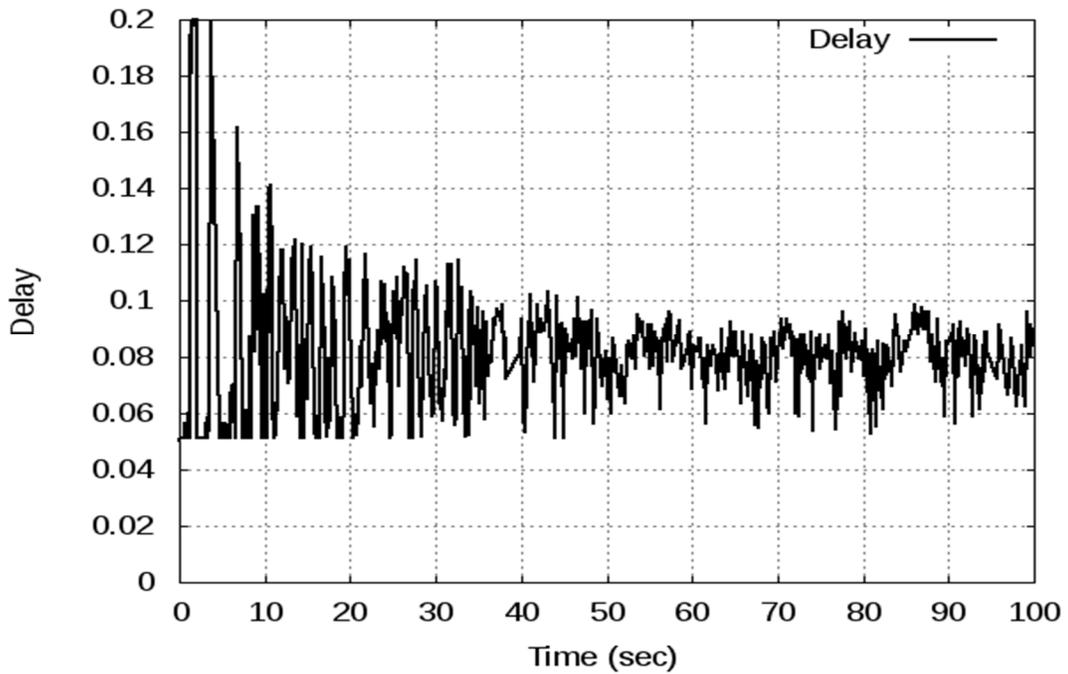
**القسم الثالث:** تم وضع عدد الاتصالات في هذا القسم الاخير 100 اتصال, الشكل رقم (4-21) يوضح طول الصف المقدر بواسطة خوارزمية التقدير الذاتي لطول الصف auto, حيث ان هذا هو الطول المناسب للمحافظة على الاستفادة من عرض النطاق عنق الزجاجة والموضحه في الشكل رقم (4-22) بالاضافه الى جعل قيمة التأخير اقل مايمكن والموضح في الشكل رقم (4-23). من الملاحظ من هذه التجارب الثلاثة انه عندما يتم زيادة عدد الاتصالات يقل طول الصف, حيث ان عدد اتصالات قليل يستلزم صف طويل وذلك للمحافظة على عرض النطاق وعدد اتصالات كبير يستلزم صف قصير حسب [21] وذلك للمحافظة على الاستفادة من عرض النطاق وجعل قيمة التأخير اقل مايمكن.



الشكل 4-21 الصف المقدر بواسطة auto والذي يتراوح ما بين 30 و 40 حزمه ليتناسب مع عدد 100 اتصال



الشكل 4-19 خوارزمية auto تحدد الصف المناسب الذي يضمن الاستفادة من عرض النطاق عند 100 إتصال

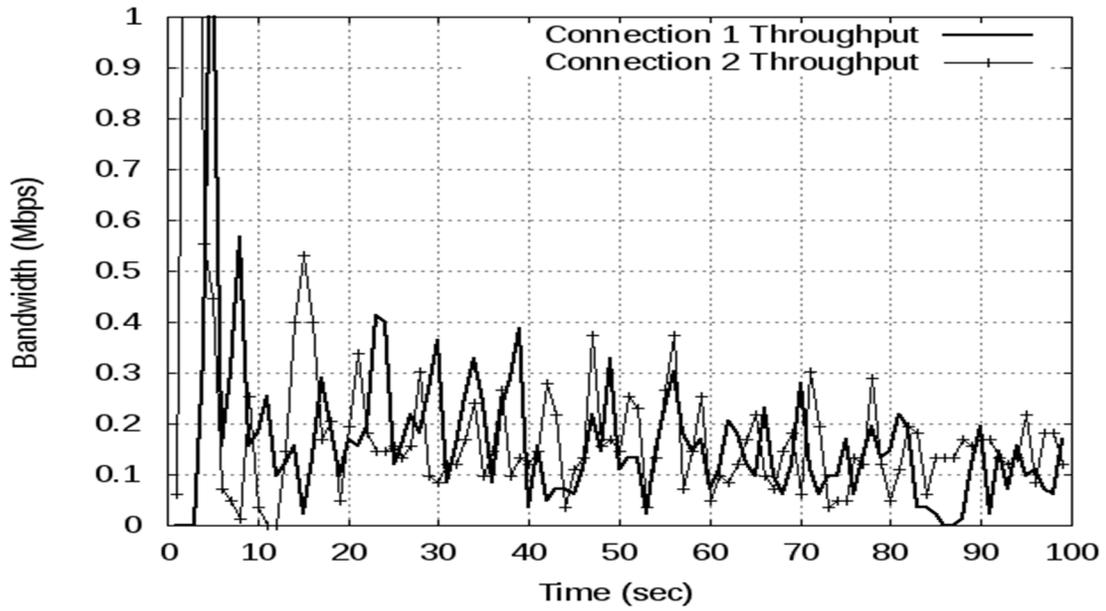


الشكل 4-23 مقدار التأخير الذي تتعرض له الحزم في auto عند 100 إتصال

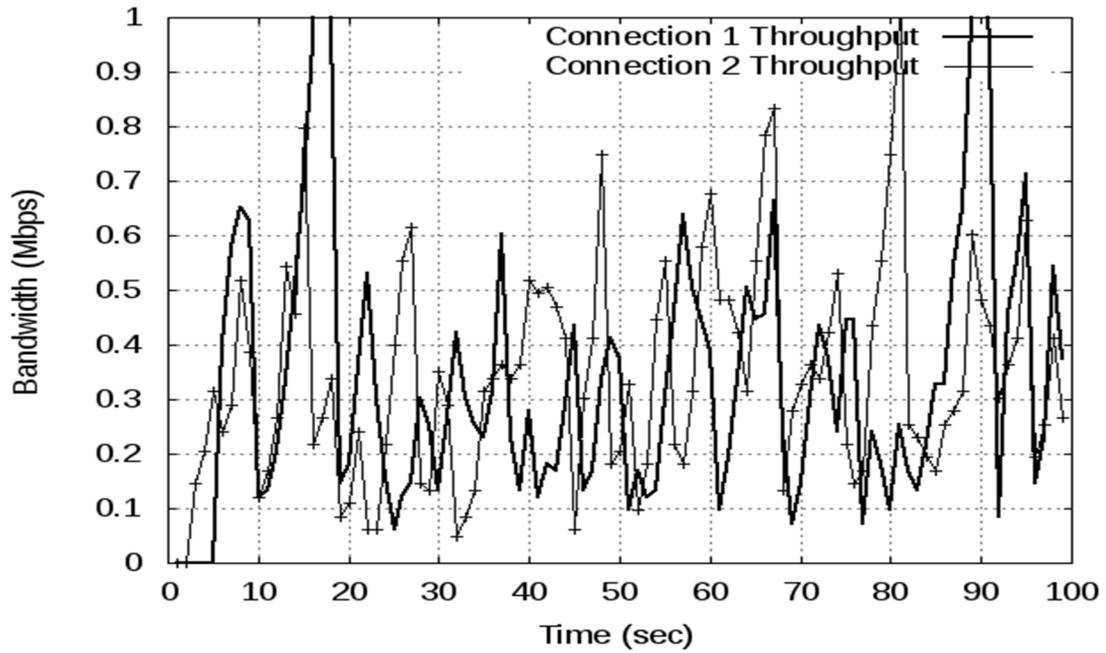
#### 4.4 تأثير العدالة بين الاتصالات في خوارزمية التقدير الذاتي لطول الصف auto

##### بعدد الاتصالات

في الفقرة (1.5.3) تم التحدث عن تأثير عدد الاتصالات على العدالة في خوارزمية التحكم بتأخير الصف CodeI في هذه الفقرة سوف يتم التحدث ايضا عن تأثير عدد الاتصالات على العدالة ولكن هذه المرة على خوارزمية التقدير الذاتي لطول الصف auto, تعتمد خوارزمية التقدير الذاتي لطول الصف auto لحساب طول الصف المناسب للعمل عليه على تقدير عدد الاتصالات الماره وبالتالي ليس هناك اي تأثير لعدد الاتصالات على العدالة, في التجربه الاولى الموضحة في الشكل رقم (4-24) حيث كان عدد الاتصالات 200 اتصال وطول الصف 150 حزمه بالرغم من ان عدد الاتصالات كبير مقارنة مع طول الصف الا ان الخوارزمية استطاعت ان تضمن لكل اتصال حيز معقول من عرض النطاق وايضا هناك نوع من العدالة التقريبيه بين هذه الاتصالات, تم اعاده التجربه بعد انقاص عدد الاتصالات الى 30 اتصال فقط وتثبيت كل المعاملات الاخرى, يتضح من الشكل رقم (4-25) ان خوارزمية التقدير الذاتي لطول الصف auto استطاعت ايضا ان تقدر طول الصف المناسب لعدد الاتصالات وهو في هذه الحاله 30 اتصال, واستطاع كل اتصال ان ينال حيز معقول من عرض النطاق المشترك بين هذه الاتصالات, استطاعت خوارزمية التقدير الذاتي لطول الصف auto ان تقوم بهذا العمل لانها لا تركز فقط على جعل زمن التأخير اقل مايمكن بل تقوم بحساب طول الصف المناسب لعدد الاتصالات وبالتالي فانها تحسب الطول المناسب لعدد الاتصالات الداخلة الى صف الموجه.



الشكل 4-24 الحيز الذي يناله اتصاليين في خوارزمية auto من بين 200 اتصال



الشكل 4-25 يوضح الحيز الذي يناله اتصاليين في خوارزمية auto من بين 30 اتصال

## 5.4 خلاصة

في هذا الباب تناولنا طريقة عمل خوارزمية التقدير الذاتي لطول الصف auto حيث بينا انها خوارزمية خفيفة لا تعتمد علي التعامل مع كل اتصال على حده وإنما تعتمد على تقدير عدد الاتصالات الماره بصوره احتماليه ومنه يتم تقدير طول الصف وذلك لضمان التأخير المناسب, لا ترمي هذه الطريقه بثقل تنفيذ كبير على المعالج, تم بعد ذلك اجراء عدة تجارب على هذه الخوارزمية لدراسة التأخير, معدل الاستفادة من عرض النطاق والعدالة بين الاتصالات الماره, الباب القادم سيتناول نتائج المقارنه بين خوارزميات التحكم بالازدحام وهي خوارزمية الاسقاط من اخر الصف, خوارزمية الاسقاط المبكر العشوائي RED, خوارزمية التحكم بتأخير الصف Code1 وخوارزمية التقدير الذاتي لطول الصف auto.

## الباب الخامس

### 1.5 مقارنة الاداء

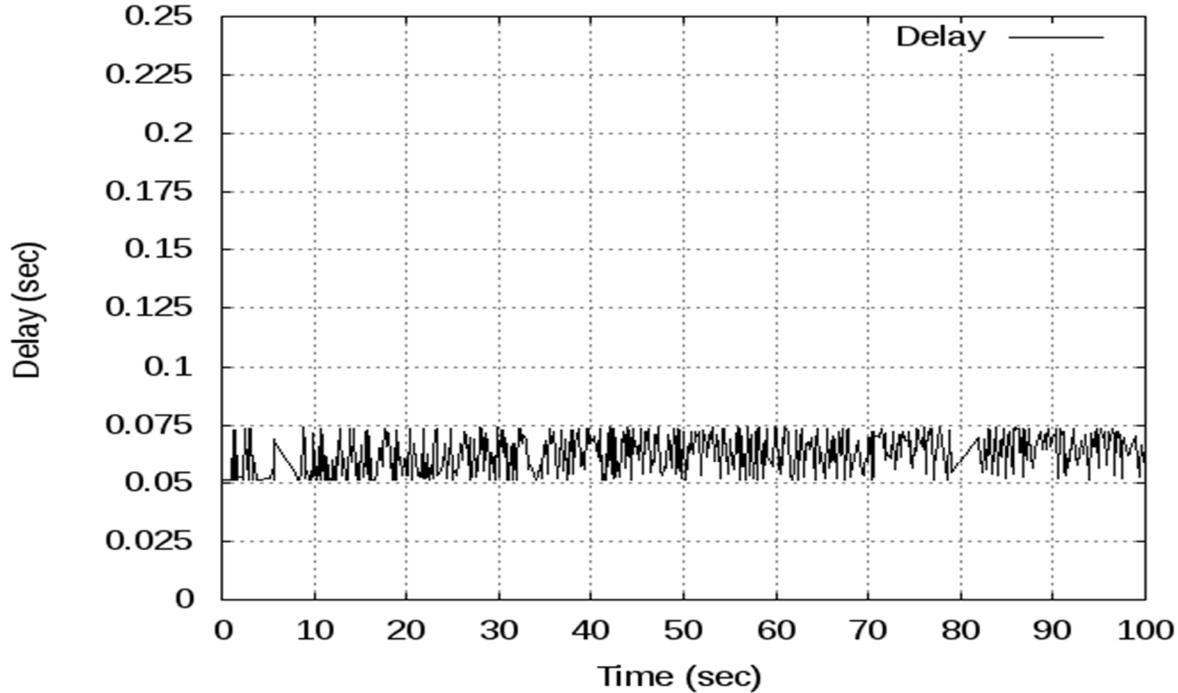
في هذا الباب سوف يتم مقارنة اداء الخوارزميات التي تعمل في مجال التحكم بالازدحام وبالتالي التحكم بالتأخير على الصف, بالاضافة الى الخوارزميات التي تعمل في مجال التحكم بالتأخير فقط وهما خوارزمية التحكم بتأخير الصف Codel وخوارزمية التقدير الذاتي لطول الصف auto, قمنا باجراء تجربتين في الاولى ثبتنا طول الصف عند 20 حزمه لمعرفة التأخير الذي تتعرض له الحزم عند هذا الطول وفي التجربه الثانيه جعلنا طول الصف 100 حزمه وذلك لمعرفة من اين ينتج التأخير الكبير في الشبكة الموضحة في الشكل رقم (4-1).

### 2.5 الصف القصير

لاثبات ان العامل الحاسم في التأخير في صفوف الموجهات هو طول الصف, حتى لا يكون كبيراً فيزيد التأخير او يكون صغيراً فيؤدي الى عدم الاستفاده من عرض النطاق عنق الزجاجة, وذلك عن طريق إجراء تجربة على المحاكى ns-2 لأربع خوارزميات تعمل على المحاكى وهي خوارزمية الاسقاط من آخر الصف Drop-Tail, خوارزمية الاسقاط المبكر العشوائي RED [6], خوارزمية التحكم بتأخير الصف Codel و خوارزمية التقدير الذاتي لطول الصف تم القيام بهذه التجربة لمعرفة تأثير كل خوارزمية من الخوارزميات الأربعة اعلاه على التأخير, ووجد انه عندما يكون طول الصف قصيراً تتشابه كل الخوارزميات تقريباً في التأخير.

## 1.2.5 خوارزمية الاسقاط من آخر الصف Drop-Tail

**التأخير:** تقوم خوارزمية الاسقاط من آخر الصف Drop-Tail بإسقاط الحزم من آخر الصف وهي الحزم التي لم تجد لها مكاناً في الصف، وعندما يكون الصف قصيراً تقوم الخوارزمية إلزامياً بإسقاط كل الحزم الواردة والتي لم تجد لها مكاناً وبالتالي عدد الحزم التي تستطيع هذه الخوارزمية تخزينها في الصف سيكون قليلاً وذلك لأن طول الصف قصير، وبما أن الاسقاط في هذه الخوارزمية مربوط بطول الصف وبما ان طول الصف قصير بالتالي فان التأخير يكون قليلاً ومتناسب مع طول الصف، كما هو موضح في الشكل رقم (5-1) فإن التأخير في منطقة عنق الزجاجة والذي لا يتعدى 10ms قد يصل الى 30ms وذلك عند استخدام خوارزمية الاسقاط من آخر الصف Drop-Tail بالرغم من ان طول الصف لا يتعدى 20 حزمه فقط.

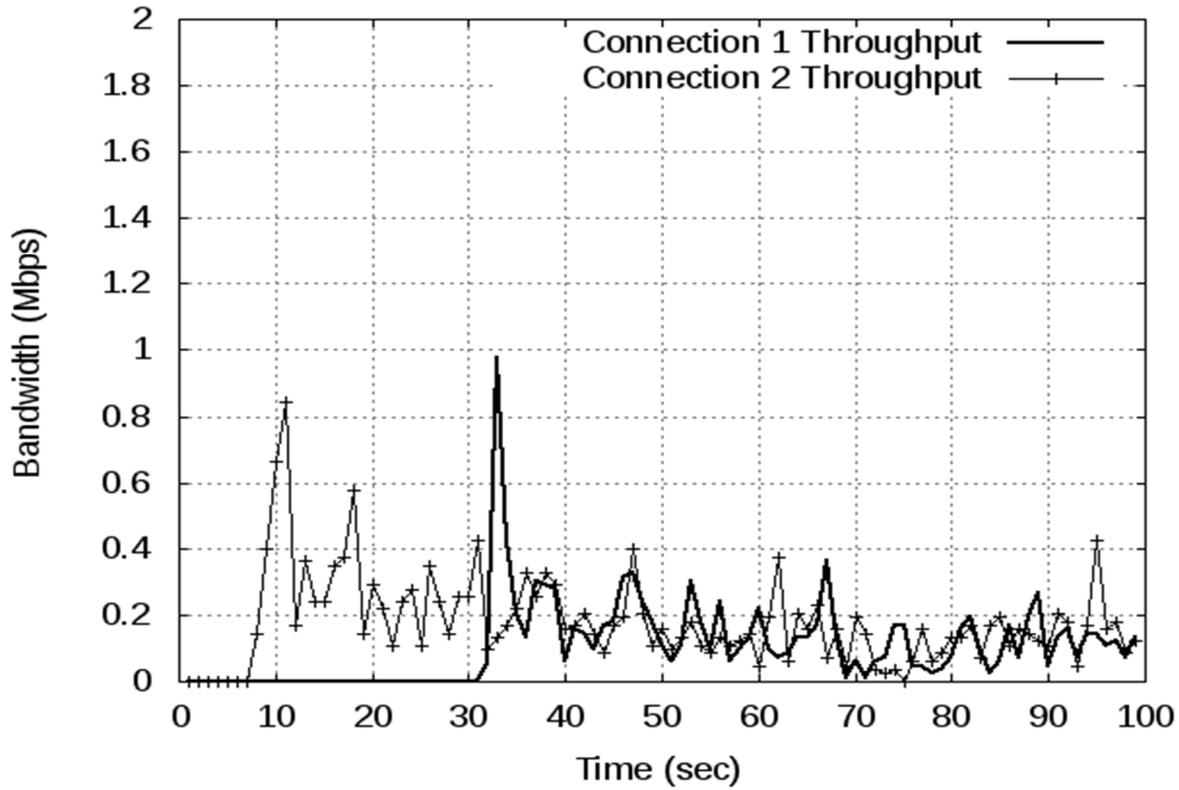


الشكل 5-1 التأخير في منطقة عنق الزجاجة في Drop-Tail عند 20 حزمه

**العدالة:** لدينا في هذه التجربة مائة اتصال تتشارك في قناة عنق الزجاجة وبالتالي فان ضمان اقل حد من العدالة مطلوب بين هذا العدد الكبير من الاتصالات, الشكل رقم (5-2) يوضح الحيز الذي تحصل عليه كل اتصال من اتصاليين تم اختيارهما بصورة عشوائية من مجموعة الاتصالات المائة, من الشكل يتضح ان الاتصال ذو الخط بعلامة الجمع استطاع ان يتحصل على حيز اكبر لانه استطاع ان يبدأ الارسال بصورة مبكره اما الاتصال الاخر فلم يستطع ان ينشئ الاتصال لانه تعرض لحالات اسقاط متكرر عند محاولته انشاء الاتصال لذلك يظهر في الشكل انه لم يبدأ الارسال الا عند الثانيه 30 حيث يظهر جليا عند هذه اللحظة انشاء الاتصال وهو في مرحلة البدايه البطيئه والدليل على ذلك الارتفاع الشديد لهذا الاتصال ثم الانخفاض والاستقرار عند منطقه محده, من العيوب الواضحه مما سبق ان خوارزمية الاسقاط من آخر الصف Drop-Tail لا تستطيع ان تضمن اقل حد للعداله لكنها تتفوق على الخوارزميات الاخرى في مجال المحافظه على عرض النطاق.

## 2.2.5 خوارزمية الاسقاط المبكر العشوائي RED

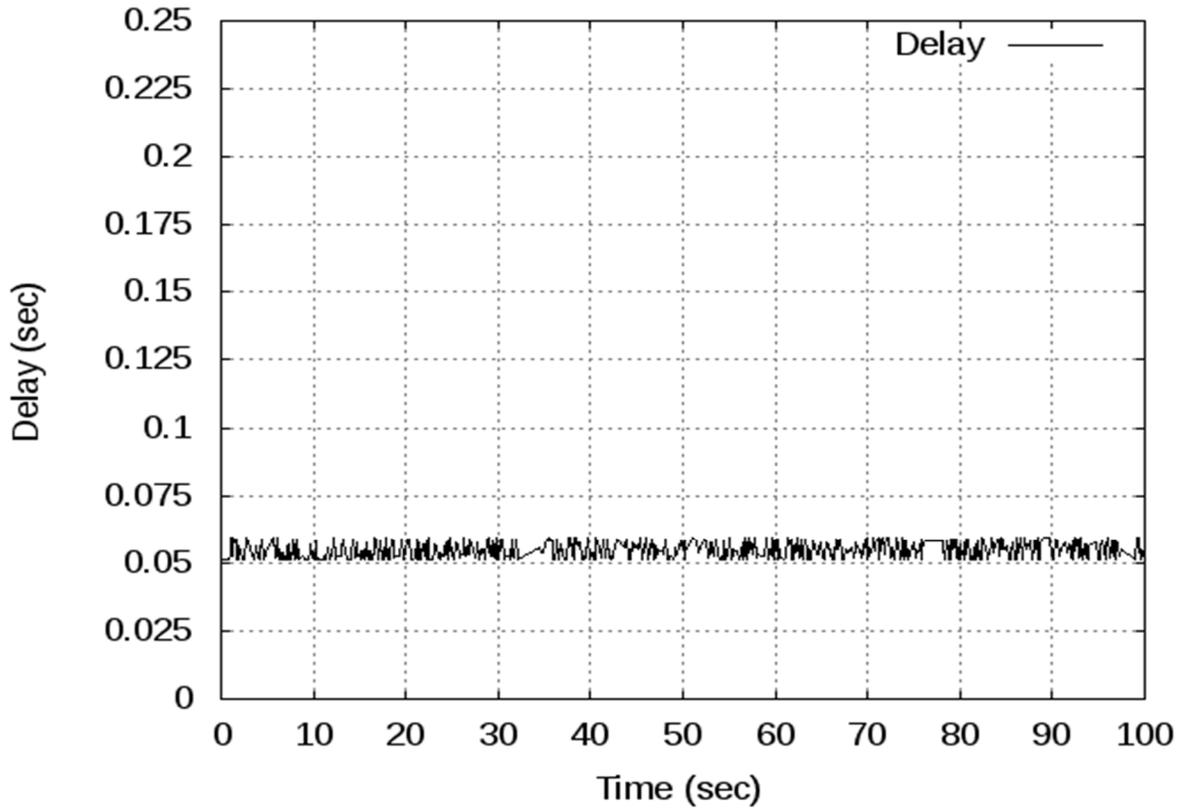
**التأخير:** أما خوارزمية الاسقاط المبكر العشوائي RED فهي ايضاً خوارزمية تعتمد على ضبط طول الصف يدوياً واحياناً في بعض الخوارزميات المطوره عن خوارزمية الاسقاط المبكر العشوائي RED يتم ضبطه يدوياً [46, 48, 50]، حيث ان هذه الخوارزمية لديها معاملان مهمان وهما  $min_{th}$  وهي العتبه الدنيا و  $max_{th}$  وهي العتبه العليا، تقوم خوارزمية الاسقاط المبكر العشوائي RED بتقسيم طول الصف الى ثلاثة مقاطع, المقطع الاول يبتدئ من واحد وينتهي عند  $min_{th}$  في هذا المقطع لا تقوم خوارزمية الاسقاط المبكر العشوائي RED باسقاط اي حزمه بل تسمح لكل الحزم بالدخول



الشكل 5-2 العدالة بين الاتصالات التي تمر عبر موجه Drop-Tail عند 20 حزمه

والمقطع الثاني يبدأ من  $min_{th}$  وينتهي عند  $max_{th}$  في هذا المقطع تقوم الخوارزمية باسقاط الحزمة الواردة باحتمال يتناسب مع عدد الحزم التي تأتي من هذا الاتصال المحدد، والمقطع الثالث يبدأ من  $max_{th}$  وينتهي عند نهاية حدود الصف وتقوم الخوارزمية باسقاط اي حزمه ترد في هذا المقطع. مما سبق يتبين ان خوارزمية الاسقاط المبكر العشوائي RED ايضاً تعتمد على حدود ثابتة يتم وضعها للصف فاذا وضعت حدود كبيرة للمعاملات اعلاه ولم يكن هناك الا عدد قليل من الاتصالات باعتبار ان قناة عنق الزجاجة ذات سعة اقل فان الصف يكون كبيراً نسبة لكبير حجم معاملاته  $(max_{th}, min_{th})$ . وتراكم الحزم في النهاية يؤدي الى زيادة التأخير على الصف وبالتالي فإن التأخير على صف خوارزمية الاسقاط المبكر العشوائي RED مشابه للتأخير على صف خوارزمية

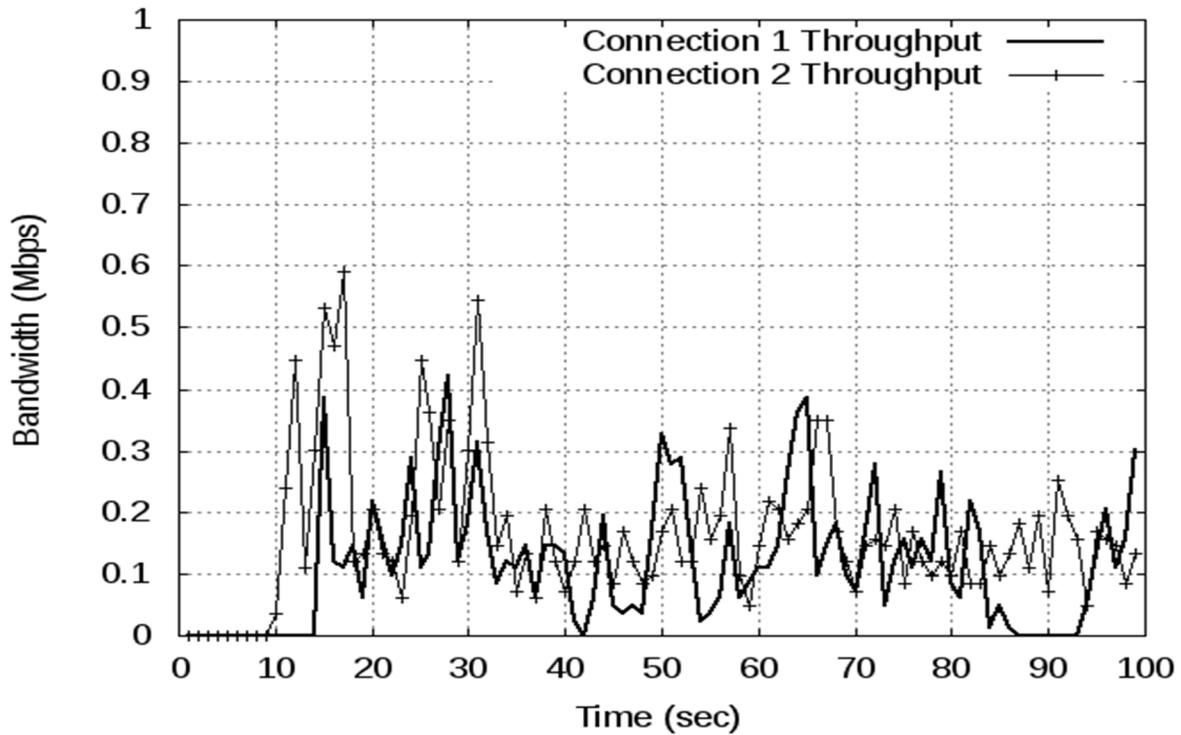
الاسقاط من آخر الصف Drop-Tail وذلك عندما يكون طول الصف في خوارزمية الاسقاط من آخر الصف Drop-tail مساوي لطول المعامل  $\max_{th}$  في الصف المعتمد على خوارزمية الاسقاط المبكر العشوائي RED, في النهاية فان خوارزمية الاسقاط المبكر العشوائي RED تنقصها ميزة تقدير طول الصف بصورة آلية وذلك لتقليل التأخير، كما يتضح من الشكل رقم (3-5) يتراوح التأخير ما بين 10ms - 22ms, حيث تتعرض الكثير من الحزم لتأخير يصل حتى 22ms داخل الصف, اي تنتظر الحزم دورها في الارسال لمدة 12ms داخل الصف حتي يتم ارسالها.



الشكل 3-5 التأخير الذي تتعرض له الحزم داخل صف خوارزمية RED عند 20 حزمه

**العدالة:** في هذه الفقرة تم استكشاف قدرة خوارزمية الاسقاط المبكر العشوائي RED على ضمان الحد الأدنى من العدالة للاتصالات التي تمر عبر الموجه, الشكل رقم (4-5) يوضح الحيز الذي

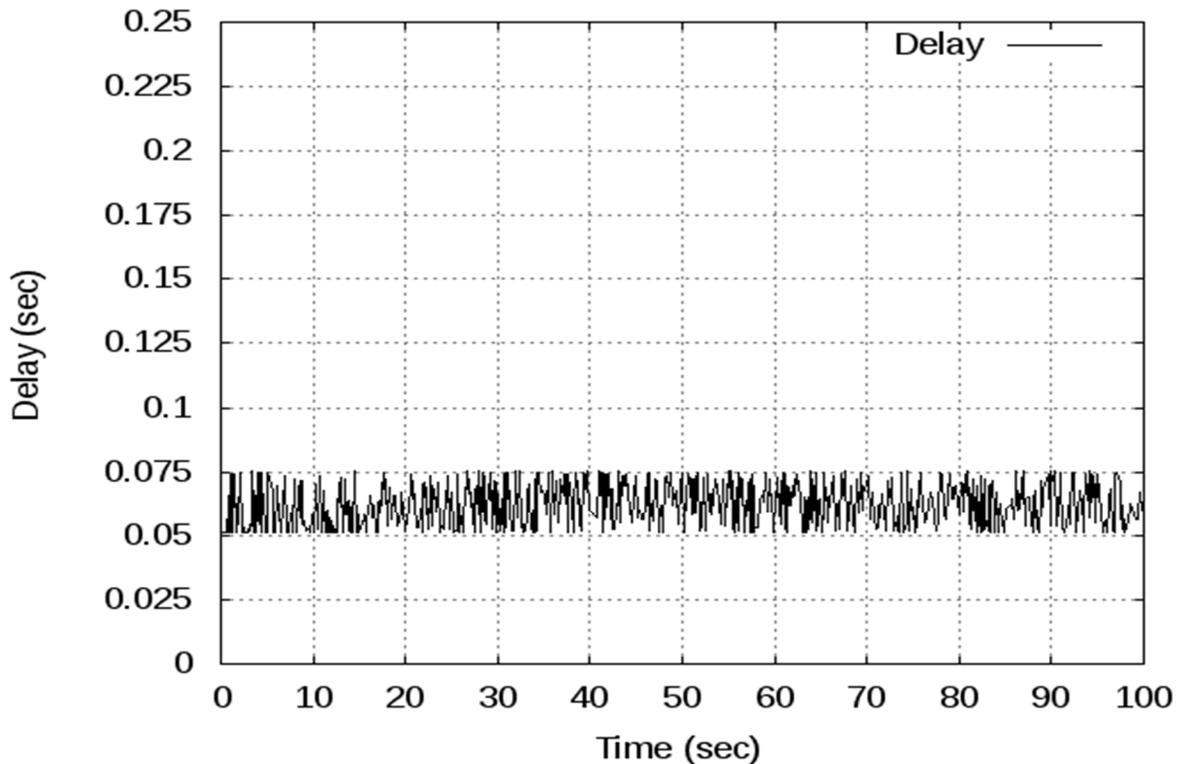
يتحصل عليه اتصاليين من مجموعة الاتصالات التي تتشارك في القناة حيث انه يتساوى مقدار ما يتحصل عليه كل اتصال من عرض النطاق بصوره تقريبيه من الشكل, استطاعت خوارزمية الاسقاط المبكر العشوائي RED ان تضمن العدالة بصوره جيده ولكنها لم تستطع ان تضمن العدالة من بداية التجربه حيث بدء الاتصال الاول بعد 10 ثواني والاتصال الثاني بدء بعد 15 ثانيه بالرغم من ان كل الاتصالات بدأت في زمن واحد, تفسير ذلك يرجع الى ان هذين الاتصاليين لم يستطيعا ان يقيما الاتصال وذلك بسبب الاسقاط المتكرر لرسائل اقامة الاتصال بسبب التزاحم الشديد الناتج عن قصر الصف وذلك بالرغم من ان من اهداف خوارزمية الاسقاط المبكر العشوائي RED العدالة.



الشكل 4-5 العدالة بين الاتصالات التي تمر عبر موجه RED عند 20 حزمه

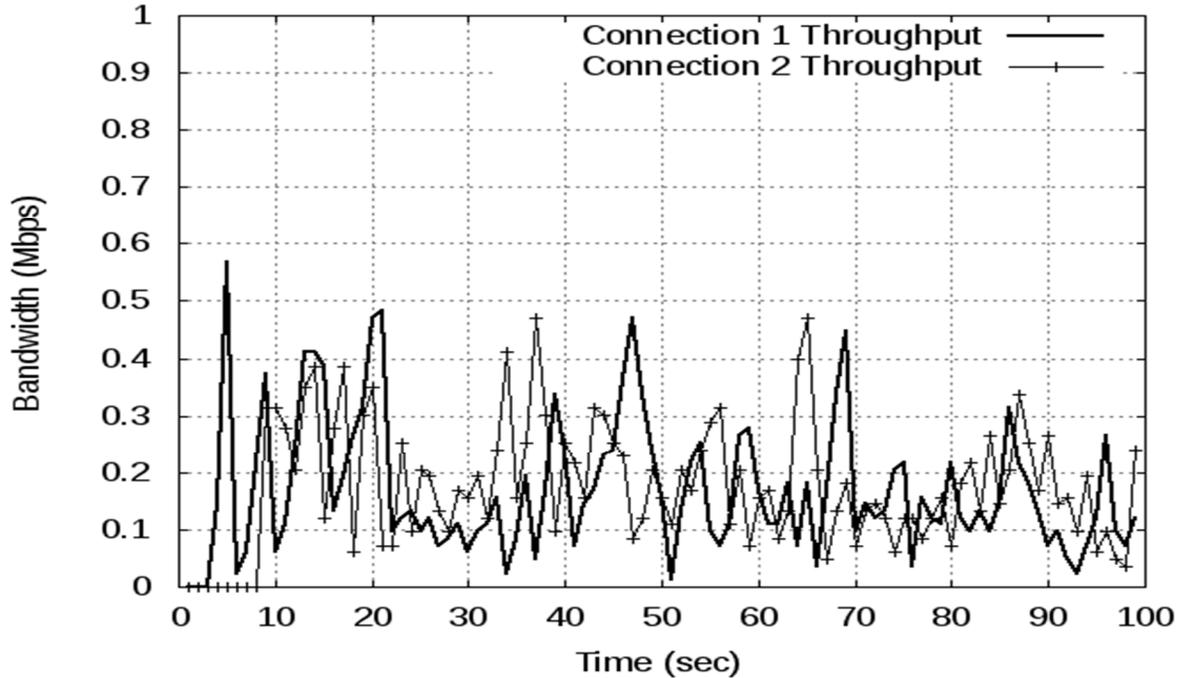
### 3.2.5 خوارزمية التحكم بتأخير الصف Codel

التأخير: اما خوارزمية التحكم بتأخير الصف Codel فعند تزويدها بصف قصير تقوم بضمان تأخير قليل لكل الحزم، تقوم خوارزمية التحكم بتأخير الصف Codel بالاسقاط العشوائي للرسائل وذلك بغض النظر عن حجم الصف المتوفر حيث تحسب زمن دخول اي حزمه وزمن خروجها وتحسب منه زمن الزيارة وتستنتج من هذا الزمن اذا كان هناك ازدحام ام لا فإن كان هناك ازدحام او بوادر ازدحام تقوم الخوارزمية بالاسقاط العشوائي من الصف وبالتالي تضمن أن حجم الصف يكون بصورة دائمه قصيراً , يوضح الشكل رقم (5-5) التأخير الذي تتعرض له الحزم في صف خوارزمية التحكم بتأخير الصف Codel.



الشكل 5-5 التأخير الذي تتعرض له الحزم داخل صف خوارزمية Codel عند 20 حزمه

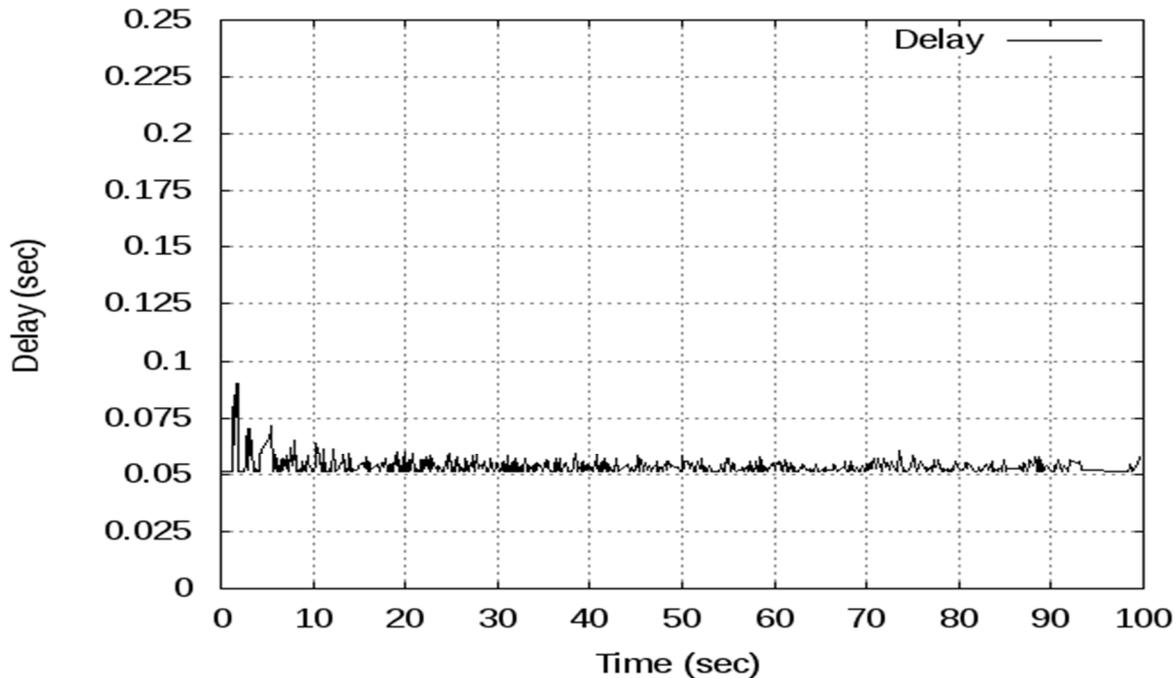
**العدالة :** من الشكل رقم (5-6) يلاحظ ان اداء خوارزمية التحكم بتأخير الصف Codel فيما يتعلق بالعدالة بين الاتصالات انها ذات اداء جيد ولكن لديها نفس المشكله السابقه وهي عدم قدرة الخوارزمية التحكم بتأخير الصف Codel على توفير الحيز الكافي لاي اتصال ليبدأ عمله بطريقه مستقره, حيث لم يبدأ الاتصال الاول الارسال الا عند الثانيه الثالثه اما الاتصال الثاني فلم يبدأ الا عند الثانيه السابعه وذلك كما اوردنا بسبب التزامم الشديد من قبل الاتصالات والذي يسبب اسقاط حزم انشاء الاتصال, ينتج هذا الامر لان الاتصالات الاولى التي تتجح في انشاء الاتصال تدخل في مرحله الارسال الكثيف بواسطة خوارزمية البدايه البطيئه Slow Start يؤدي هذا الارسال الكثيف الى امتلاء الصف بواسطة الحزم المرسله من هذه الاتصالات المحظوظه وعندما ترد حزم من الاتصالات الاخرى تجد ان الصف ممتلئ مما يؤدي الى سقوطها.



الشكل 5-6 العدالة بين الاتصالات التي تمر عبر موجه Codel عند 20 حزمه

## 4.2.5 خوارزمية التقدير الذاتي لطول الصف auto

التأخير: اما الخوارزمية الاخيره فهي خوارزمية التقدير الذاتي لطول الصف auto فعند تزويدها بصف قصير تقوم بضمان تأخير قليل لكل الحزم, تقوم خوارزمية التقدير الذاتي لطول الصف auto اولاً بتقدير طول الصف المناسب لذلك يزيد التأخير بصورة كبيره (تمتد فترة الزيادة الكبيره في الزمن ما بين الثانيه 0 والثانيه 20) ولكن لفته قصيره ثم يبدأ في التناقص مع زيادة عدد الاتصالات الماره, حيث ان التأخير المحسوب في الثواني الاولى من زمن التجربه هو تأخير لاتصال واحد ثم يبدأ هذا التأخير في التناقص مع دخول الاتصال الثاني ويستمر في التناقص حتى يستقر التأخير عند حد معين ثابت يتناسب مع عدد الاتصالات الماره وطول الصف, يتضح من الشكل رقم (5-7) ان خوارزمية التقدير الذاتي تحافظ على تأخير اقل بكثير من التأخير المفروض على الحزم في خوارزمية التحكم بتأخير الصف Code1.

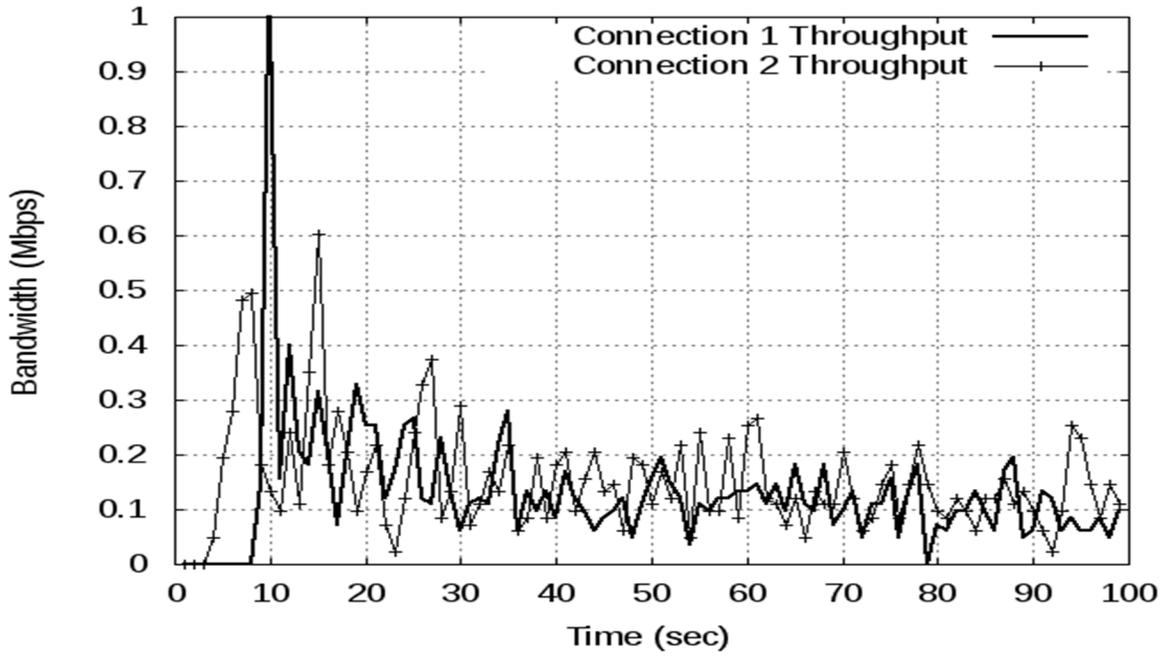


الشكل 7-5 التأخير الذي تتعرض له الحزم داخل صف خوارزمية auto عند 20 حزمه

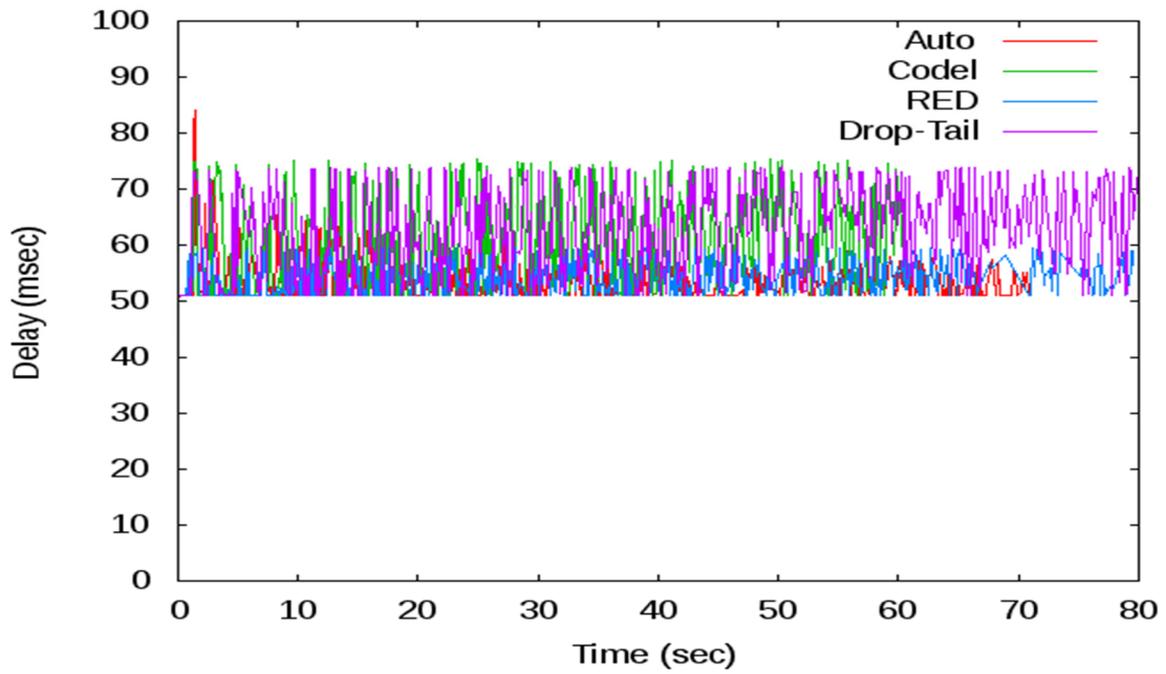
**العدالة:** استطاعت خوارزمية التقدير الذاتي لطول الصف auto ان تضمن قدر جيد من العدالة بين الاتصالات الماره وذلك لان الخوارزمية تقدر طول الصف المناسب للعمل عليه, بالرغم من ان طول الصف اقصر من اللازم الا ان الخوارزمية استطاعت ان تعمل بصورة جيدة لضمان العدالة, ولكن الخوارزمية تواجه نفس المشاكل السابقه وهي عدم قدرة الخوارزمية على ضمان الحيز الكافي من الصف لكل اتصال حيث لم تستطع ان تبدأ الاتصالات الموضحة في الشكل رقم (5-8) الا بعد فترة زمنية ليست بالقصيره, خوارزمية الاسقاط المبكر العشوائي RED وخوارزمية التحكم بتأخير الصف CodeI بالاضافه الى خوارزمية التقدير الذاتي لطول الصف auto تقوم بالاسقاط العشوائي الذي ساعد في ضمان العدالة ولكن خوارزمية التقدير الذاتي لطول الصف auto تساهم زيادة على ذلك بالاسقاط المتناسب مع طول الصف المضاف للموجه.

### 5.2.5 مقارنة التأخير لكل الخوارزميات عند طول الصف 20

الشكل رقم (5-9) يوضح التأخير الذي تفرضه كل خوارزمية من الخوارزميات الاربعه تحت الدراسه وهي خوارزمية الاسقاط من آخر الصف Drop-Tail وخوارزمية الاسقاط المبكر العشوائي RED وخوارزمية التحكم بتأخير الصف CodeI وخوارزمية التقدير الذاتي لطول الصف auto, من الواضح ان خوارزمية التقدير الذاتي لطول الصف auto هي الخوارزمية الاقل تأخيرا للرسائل الماره ويظهر باللون الاحمر اما خوارزمية التحكم بتأخير الصف CodeI باللون الازرق فتأتي في المرتبه الثانيه, في المرتبه الثالثه تأتي خوارزمية الاسقاط المبكر العشوائي RED باللون الاخضر اما الخوارزمية الاكثر تأخيرا للرسائل فهي خوارزمية الاسقاط من آخر الصف.



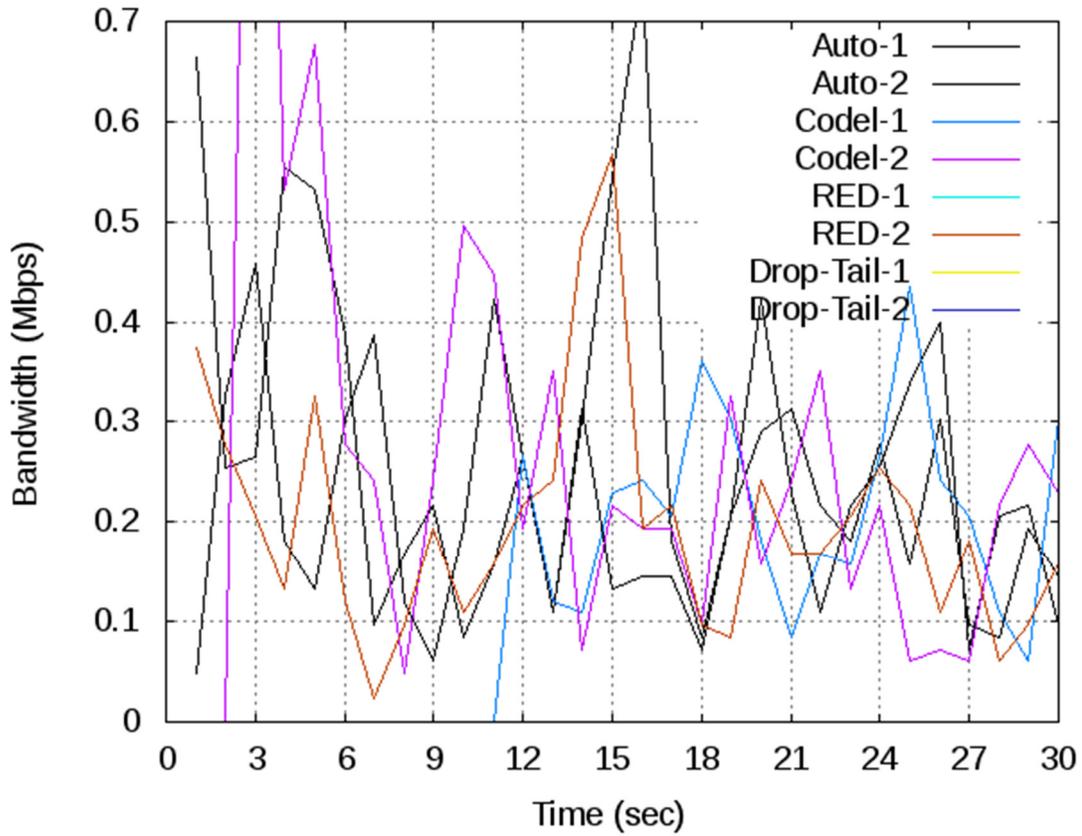
الشكل 5-8 العدالة بين الاتصالات التي تمر عبر موجة auto عند 20 حزمه



الشكل 5-9 التأخير الذي تفرضه كل الخوارزميات تحت الدراسة عند 20 حزمه

## 6.2.5 مقارنة العدالة لكل الخوارزميات عند طول الصف 20

عندما يكون طول الصف 20 حزمه فإن اي إتصال يتحصل على مقدار من عرض النطاق متناسب مع الاتصالات الاخرى التي تتشارك معه في القناة وذلك في اي نوع من انواع الخوارزميات الاربعة المستخدمه, تفسير ذلك يرجع الى انه وعندما يكون عدد الاتصالات كبيرا تكون قناة الخرج مشغوله جدا وبما ان طول الصف قصيرا (20 حزمه) فإنه لا يمكن تخزين عدد كبير من الحزم في هذا الصف القصير وبالتالي فلا يمكن لاي إتصال تخزين عدد كبير من الحزم ليستطيع فيما بعد اي إتصال الحصول على حيز اكبر من عرض النطاق, تم تجميع نتائج مخرجات مقارنة كل إتصاليين من كل خوارزمية من الخوارزميات الاربعة تحت الدراسه في الشكل (5-10), وضحنا في كل من الشكل رقم (5-2) والشكل رقم (5-6) ان ما يتحصل عليه كل إتصال يساوي تقريبا ما يتحصل عليه الإتصال الآخر ولكن يظهر في الشكل رقم (5-4) حيث تعمل خوارزمية الاسقاط المبكر العشوائي RED مستوى جيد من العدالة حيث صممت هذه الخوارزمية بصوره اساسيه من اجل توفير العدالة بين الاتصالات, اما في الشكل رقم (5-8) حيث تعمل خوارزمية التقدير الذاتي لطول الصف auto فإن هناك مستوى عدالة مناسب مع ماتبزله الخوارزمية من جهد مع تركز ان الغرض الاساسي من خوارزمية التقدير الذاتي لطول الصف auto ليس العدالة ولكن تقليل التأخير ولكن الخوارزمية تحافظ على مستوى عدالة يقارب ما توفره خوارزمية الاسقاط المبكر العشوائي RED.



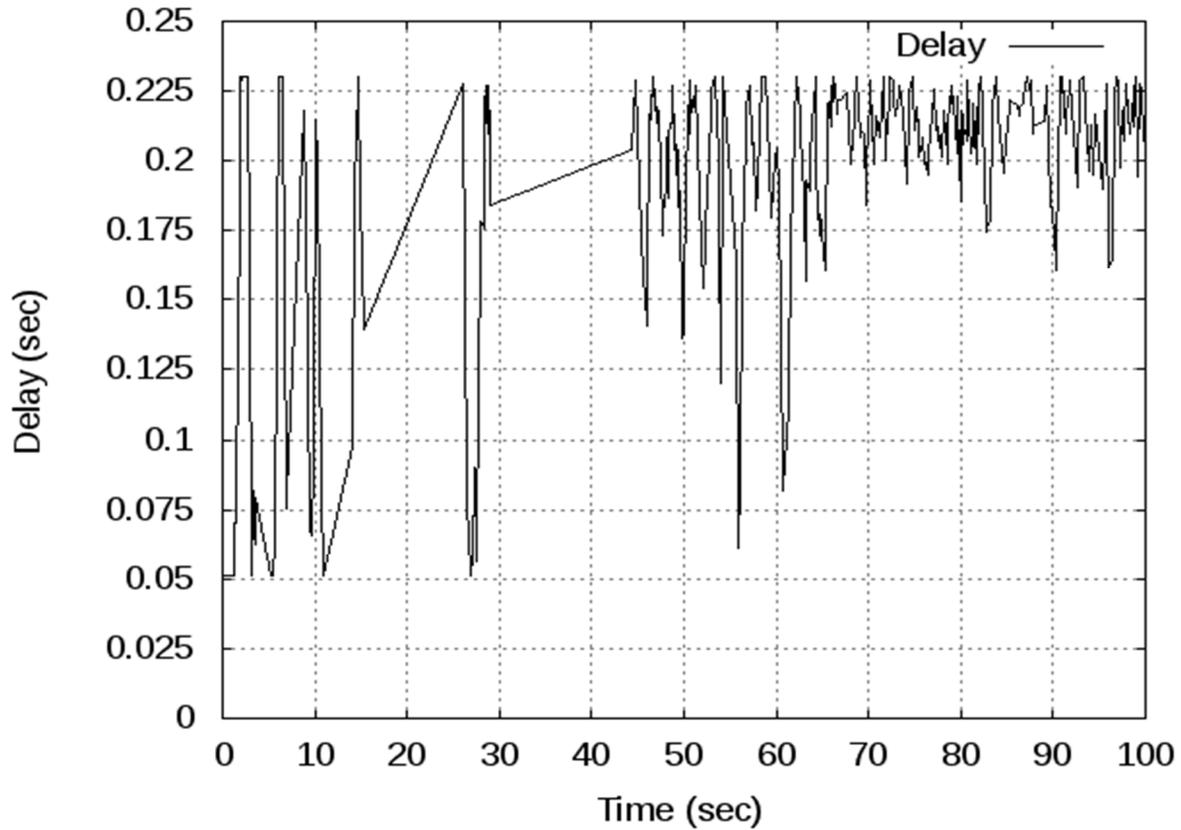
الشكل 5-10 الحيز من عرض النطاق لإتصالين عند 20 حزمه

### 3.5 الصف الطويل

عند إعادة إجراء نفس التجربة السابقة للخوارزميات الأربعة : خوارزمية الإسقاط من آخر الصف وخوارزمية الإسقاط المبكر العشوائي وخوارزمية التحكم بالتأخير وخوارزمية التقدير الذاتي لطول الصف ولكن بعد زيادة حجم الصف من 20 الى 100 حزمه مع تثبيت كل المعاملات الأخرى كان أداء الخوارزميات كالتالي:

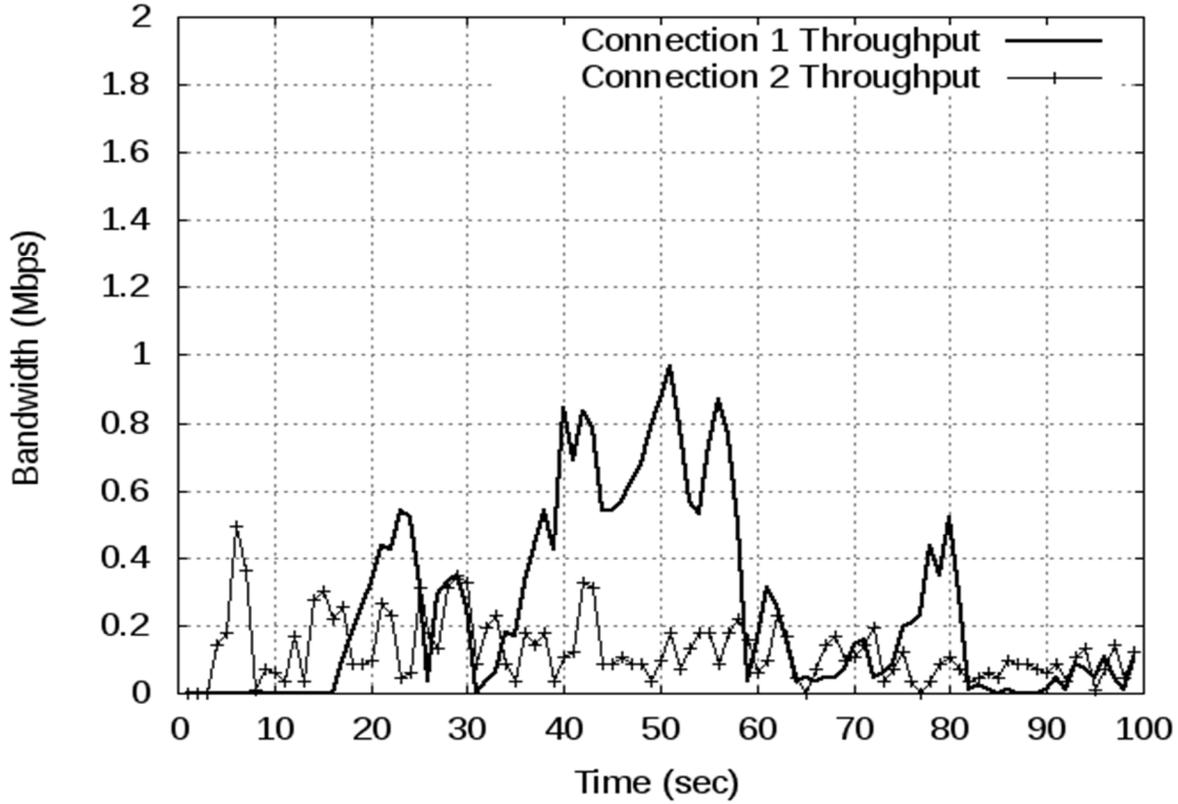
### 1.3.5 خوارزمية الإسقاط من آخر الصف Drop-Tail

التأخير: خوارزمية الإسقاط من آخر الصف Drop-Tail تتميز هذه الخوارزمية بأنها لن تسقط اي حزمه من الصف الا اذا إمتلأ الصف عن آخره وبالتالي وبما ان قناة عنق الزجاجة التي تتحكم بها الخوارزمية مزدحمة دائماً فإن صفاً سوف يتكون ويزداد طوله تدريجياً مع عدم قدرة القناة على تمرير كل الحزم وفي النهاية سوف يمتلئ الصف ويبدأ في اسقاط الحزم التي ترد اليه، الشكل رقم (5-11) يوضح التأخير المحسوب هنا وهو التأخير الذي تمكثه الحزمة داخل صف خوارزمية الإسقاط من آخر الصف Drop-Tail، يلاحظ ان التأخير يزيد عن التأخير المحسوب في التجربة السابقة عندما كان طول الصف 20 حزمه الشكل رقم (5-2) وذلك لأن الحزم في هذه التجربة تمكث فترة أطول في الصف وذلك لأن الصف نفسه طويل، في خوارزمية الإسقاط من آخر الصف يزداد التأخير الذي تفرضه على الحزم مع زيادة طول الصف ولا تملك هذه الخوارزمية اي وسيلة للتحكم بطول الصف المناسب للعمل والذي يوفر معدل تأخير قليل، وفي نفس الوقت تقليل طول الصف بصورة يدويه غير مدروسة قد يؤدي الى ان لا يستفاد من عرض النطاق عنق الزجاجة حيث يستفاد من الحزم الموجودة في الصف في قناة عنق الزجاجة عندما تتراجع الاتصالات عن الارسال.



الشكل 5-11 التأخير الذي تتعرض له الحزم داخل صف Drop-Tail عند 100 حزمه

**العدالة:** تفشل خوارزمية الاسقاط من آخر الصف Drop-Tail تماما عندما يتعلق الامر بضمان العدالة بين الاتصالات الماره بالموجه, في الشكل رقم (5-12) يتضح ان كل اتصال يتحصل على قدر مختلف عن القدر الذي يتحصل عليه الاتصال الاخر, في هذه التجريه والتي تم فيها زيادة طول الصف الى 100 حزمه استطاع الاتصال بالخط التخين ان يقوم بتخزين عدد كبير من الحزم اكثر من الاتصال الاخر ولذلك يتحصل هذا الاتصال على حيز اكبر من عرض النطاق عنق الزجاجه.



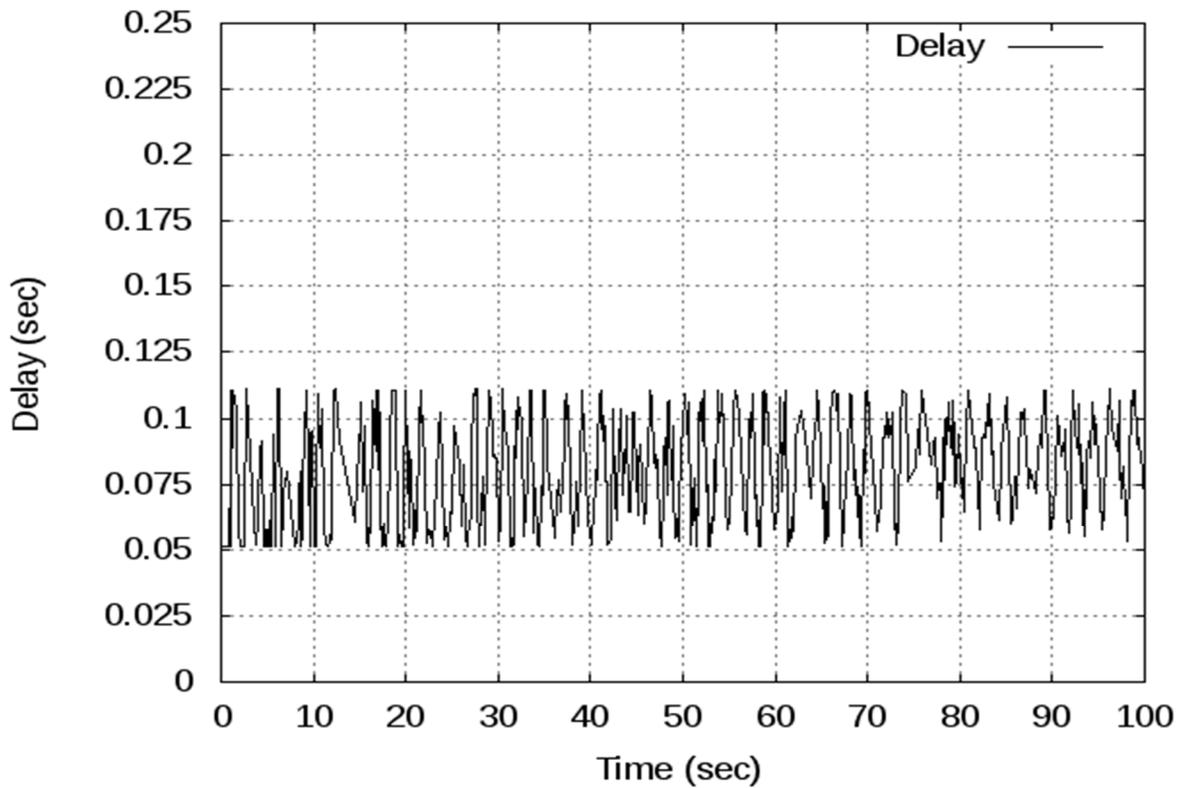
الشكل 5-12 العدالة بين الاتصالات التي تمر عبر موجه Drop-Tail عند 100 حزمه

### 2.3.5 خوارزمية الاسقاط العشوائي المبكر RED

التأخير: اما خوارزمية الاسقاط المبكر العشوائي RED فهي لا تختلف كثيراً عن خوارزمية الاسقاط من آخر الصف سوى أن لها معامل يتحكم في عدم الاسقاط ومعامل للتحكم في الاسقاط العشوائي او الاسقاط الالزامي وهو العتبه العليا  $max_{th}$  فعند زيادة طول الصف في هذه التجربة الى 100 حزمه وزيادة حدود المعاملات ( $max_{th}$  ,  $min_{th}$ ) لوحظ ان قيمة التأخير الذي تفرضه هذه الخوارزمية على الحزم المارة يزداد مع زيادة طول الصف والمعاملات. بصورة أوضح يتناسب التأخير مع المعاملات  $max_{th}$  و  $min_{th}$  والتي تتحكم في الاسقاط والتي يتم ضبطها يدوياً للاستفادة من عرض النطاق عنق الزجاجة, يتم ضبط هذه المعاملات الى قيم كبيره حسب توصيات المصنعين

لتقليل اسقاط الحزم, هذه الخوارزمية تم تطويرها أصلاً للتخلص من مشكلة التزامن وتوفير بعض العدالة ولم تصمم لتقليل التأخير, يتضح من الشكل رقم (5-13) التأخير الذي تتعرض له الحزم في صف خوارزمية الاسقاط المبكر العشوائي RED, حيث ان التأخير اقل بكثير من التأخير على خوارزمية الاسقاط من آخر الصف وذلك لأن هذه الخوارزمية تعتمد الاسقاط المكثف من الاتصالات

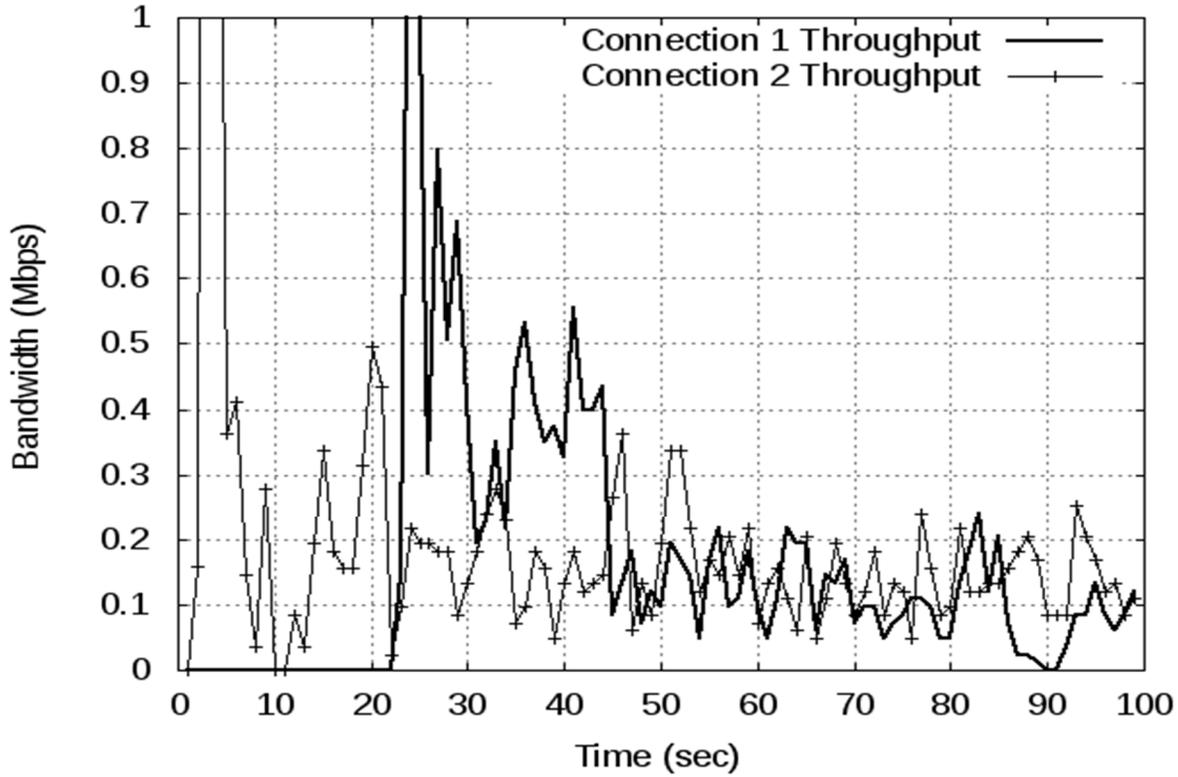
الماره



الشكل 5-13 التأخير الذي تتعرض له الحزم داخل صف خوارزمية RED عند 100 حزمه

**العدالة:** عند زيادة طول الصف الى 100 حزمه يتعرض موضوع العدالة الى الكثير من الازدحام وذلك بسبب ان كل اتصال يستطيع ان يخزن عدد مختلف من الحزم في ذلك الصف الطويل, من الشكل

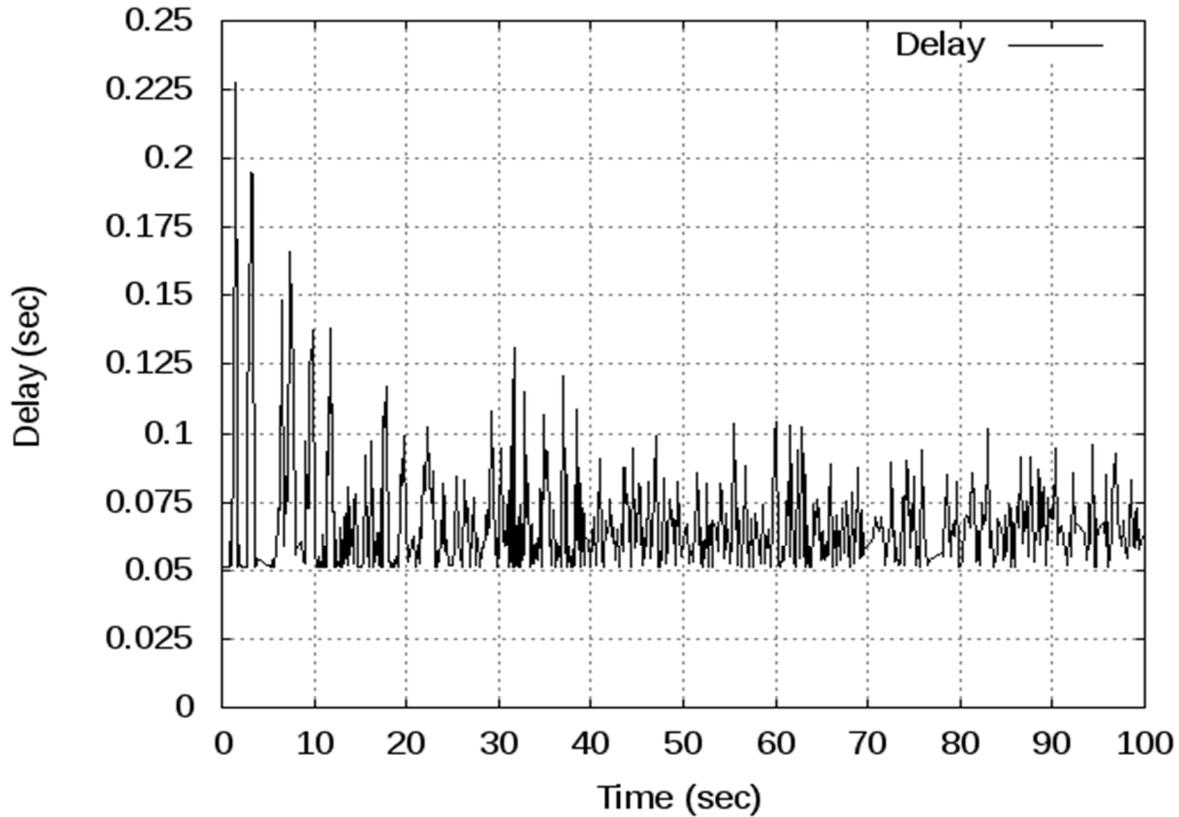
رقم (5-14) تخزين عدد متفاوت من الحزم في الصف ينعكس سلبيًا على مقدار الحيز الذي يناله كل اتصال من الصف وبالتالي على مقدار التأخير الذي تتعرض له الحزم.



الشكل 5-14 العدالة بين الاتصالات التي تمر عبر موجه RED عند 100 حزمه

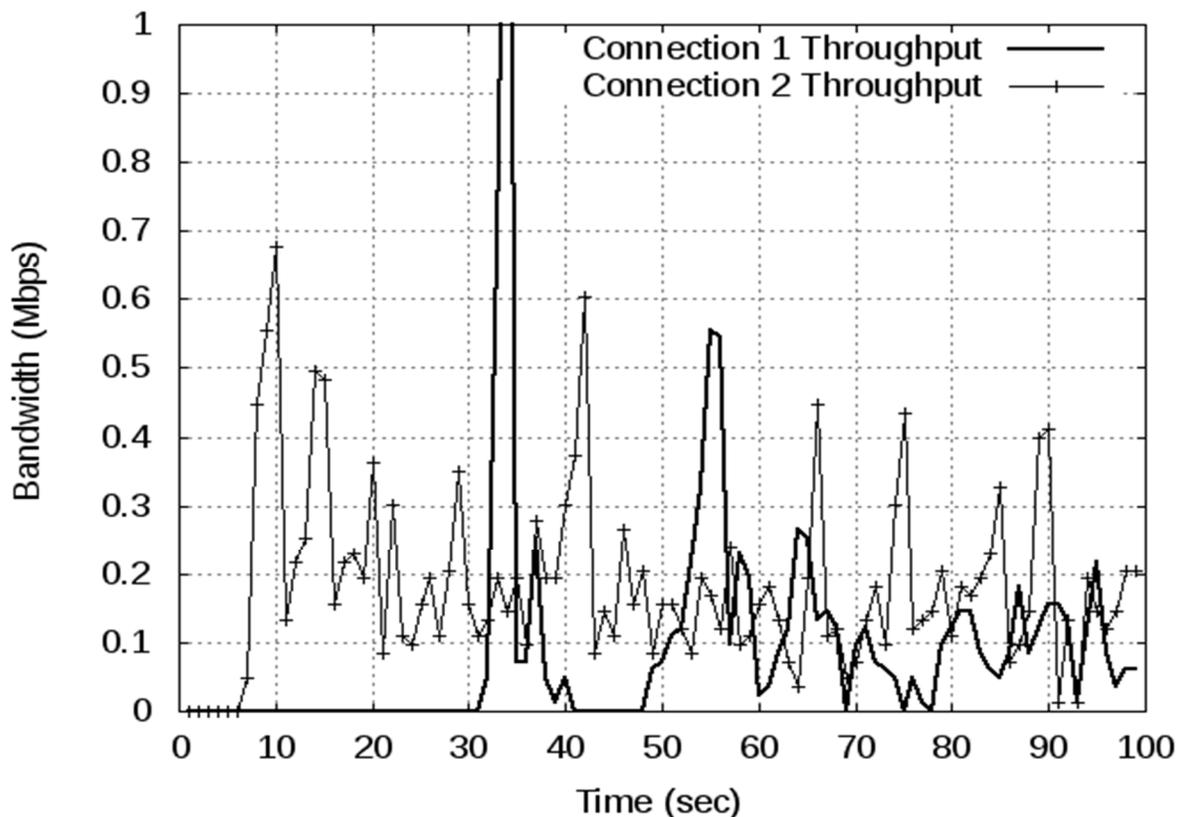
### 3.3.5 خوارزمية التحكم بتأخير الصف Codel

**التأخير:** أما في خوارزمية التحكم بتأخير الصف Codel وعند زيادة طول الصف الى 100 حزمه لوحظ ان التأخير يظل كما هو وذلك بالمقارنة مع التجربة السابقة والتي كان فيها طول الصف 20 حزمه يتضح ذلك من الشكل رقم (5-15) حيث يكون التأخير ثابت عند كلا الطولين في التجريبتين وبالتالي فإن خوارزمية التحكم بتأخير الصف Codel تتحكم في طول الصف وذلك بعزل طول الصف عن التأثير في التأخير.



الشكل 5-15 التأخير الذي تتعرض له الحزم داخل صف خوارزمية Code1 عند 100 حزمه

**العدالة:** تتفوق خوارزمية التحكم بتأخير الصف Code1 على خوارزمية الاسقاط المبكر العشوائي RED في موضوع العدالة بين الاتصالات الماره في القناة من الشكل رقم (5-16) يتضح ان هناك تحسن طفيف يؤدي الى تناسب الى درجه مقبوله من العدالة بين الاتصاليين الموضحين في الشكل، ولكن هناك تأخر ملحوظ للاتصاليين في انشاء الاتصال حيث يبدأ الاتصال الاول عند الثانيه السابعه اما الاتصال الثاني فيبدأ عند الثانيه الواحده والثلاثون وذلك طبعا بسبب الاسقاط الكثيف الذي يسقط حزم انشاء الاتصال بالنسبه للاتصاليين تحت الدراسه.

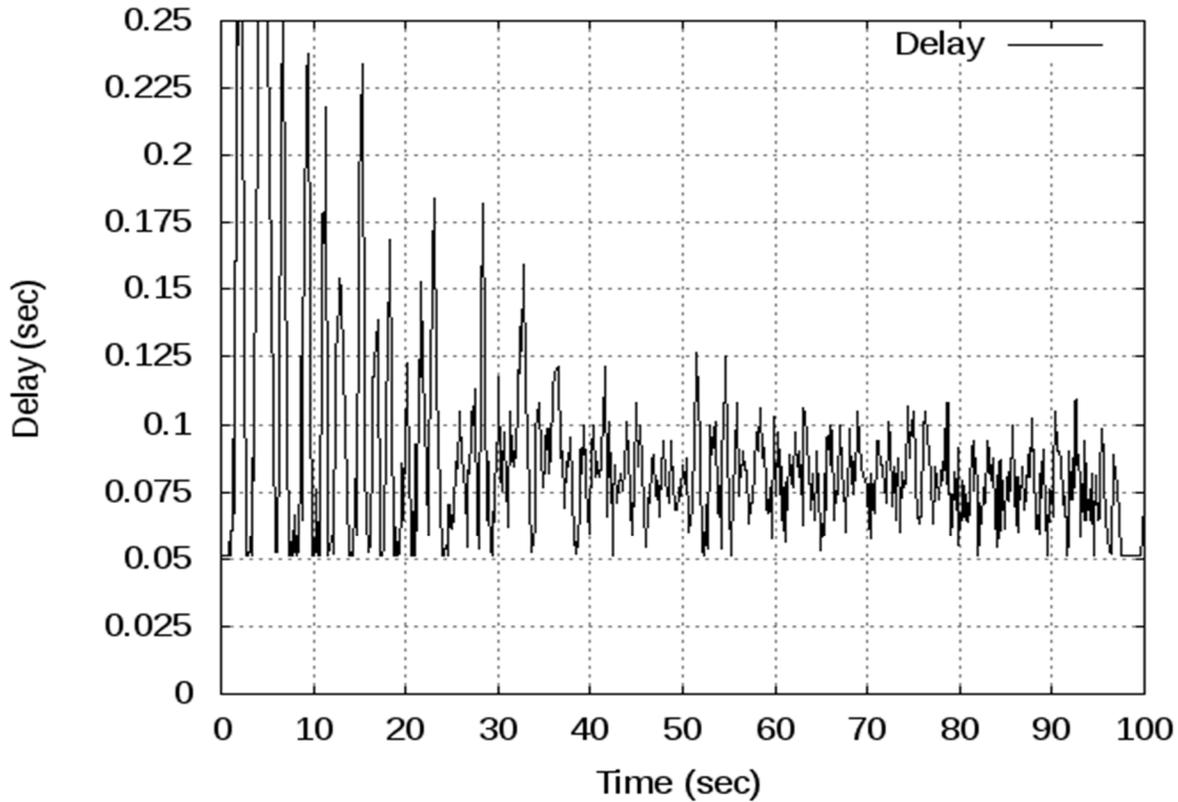


الشكل 5-16 العدالة بين الاتصالات التي تمر عبر موجه Codel عند 100 حزمه

### 4.3.5 خوارزمية التقدير الذاتي لطول الصف auto

**التأخير:** في خوارزمية التقدير الذاتي لطول الصف auto وعند زيادة طول الصف الى 100 حزمه فإن التأخير يظل ثابتا كما لو ان طول الصف لم يتغير وذلك عند مقارنة الشكل رقم (5-7) مع الشكل رقم (5-17) حيث ان التأخير في الشكل رقم (5-7) نتج عن طول صف يساوي 20 حزمه والتأخير في الشكل رقم (5-17) نتج عن طول صف يساوي 100 حزمه, حيث يمكن تجاهل الثواني الخمسه الاولى من نتائج المحاكاة والتي يكون فيها التأخير كبيرا, ينتج هذا التأخير الكبير لأن هذه الخوارزمية تقوم بتقدير طول الصف استنادا على عدد الاتصالات الماره, وبالتالي فإنه وعندما يكون عدد الاتصالات قليلا وطول الصف كبير فان التأخير يكون كبيرا ولم يتم تقديره بصورة

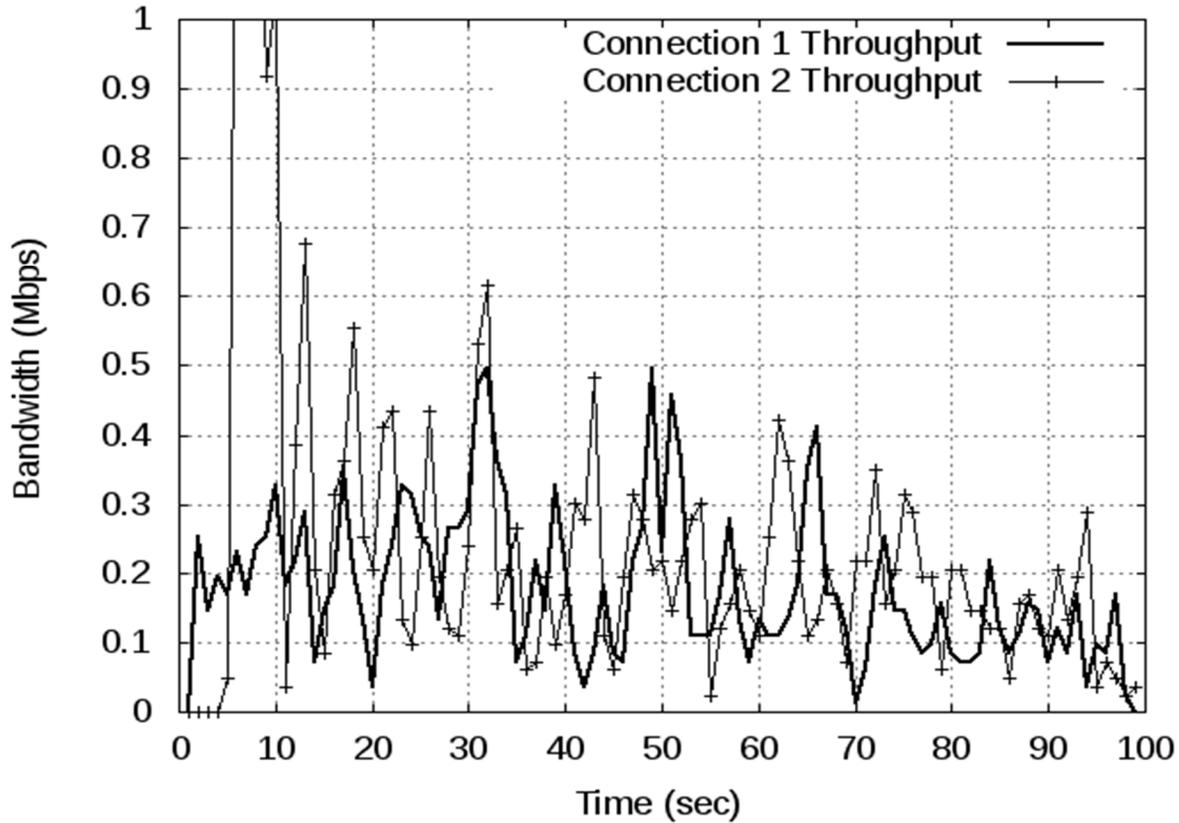
جيده بعد, يكون التأخير كبير كما هو الحال في بداية هذه التجربه حيث تبدأ المحاكاة باتصال واحد ويزداد عدد الاتصالات بعد ذلك بصورة متتاليه, يتضح من الشكل رقم (5-15) ان خوارزمية التقدير تحافظ على تأخير مشابه للتأخير المفروض على الحزم في خوارزمية التحكم بتأخير الصف .Codel



الشكل 5-17 التأخير الذي تتعرض له الحزم داخل صف خوارزمية auto عند 100 حزمه

**العدالة:** تتفوق خوارزمية التقدير الذاتي لطول الصف auto على خوارزمية التحكم بتأخير الصف Codel في عدة امور اولها ان الاتصالين يبدآن الارسال في ازمان متقاربه وذلك من بداية التجربه, قدرة الاتصالين على الارسال ترجع الى قدرتهما على انشاء الاتصال وهو الامر المفقود في الخوارزميات السابقه والتي لم تستطع ان تضمن حيز كافي لكل اتصال لينشئ الاتصال الخاص به

ويستمر في الارسال فيما بعد, الامر الثاني تضمن الخوارزمية قدر كبير من العدالة بين الاتصالات وذلك واضح من الشكل رقم (5-18) حيث يتساوى الاتصاليين في القدر الذي يناله كل اتصال من عرض النطاق وذلك طبعا بصوره تقريبيه من الشكل.

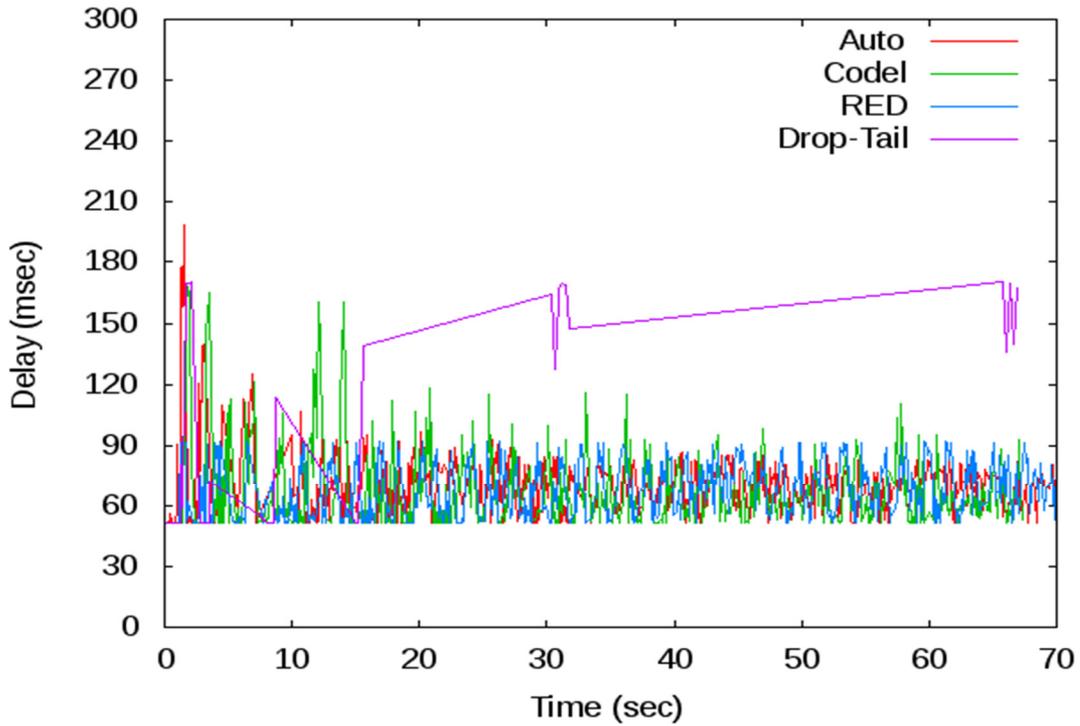


الشكل 5-18 العدالة بين الاتصالات التي تمر عبر موجه auto عند 100 حزمه

### 5.3.5 مقارنة التأخير لكل الخوارزميات عند طول الصف 100

الشكل رقم (5-19) يوضح التأخير الذي تفرضه كل خوارزمية من الخوارزميات الاربعه تحت الدراسه وهي خوارزمية الاسقاط من آخر الصف Drop-Tail وخوارزمية الاسقاط المبكر العشوائي RED وخوارزمية التحكم بتأخير الصف CodeI وخوارزمية التقدير الذاتي لطول الصف auto, من

الواضح ان خوارزمية التقدير الذاتي لطول الصف auto هي الخوارزمية الاقل تأخيرا للرسائل الماره ويظهر باللون الاحمر اما خوارزمية التحكم بتأخير الصف CodeI باللون الازرق فتأتي في المرتبه الثانيه, في المرتبه الثالثه تأتي خوارزمية الاسقاط المبكر العشوائي RED باللون الاخضر اما الخوارزمية الاكثر تأخيرا لحزم فهي خوارزمية الاسقاط من آخر الصف.

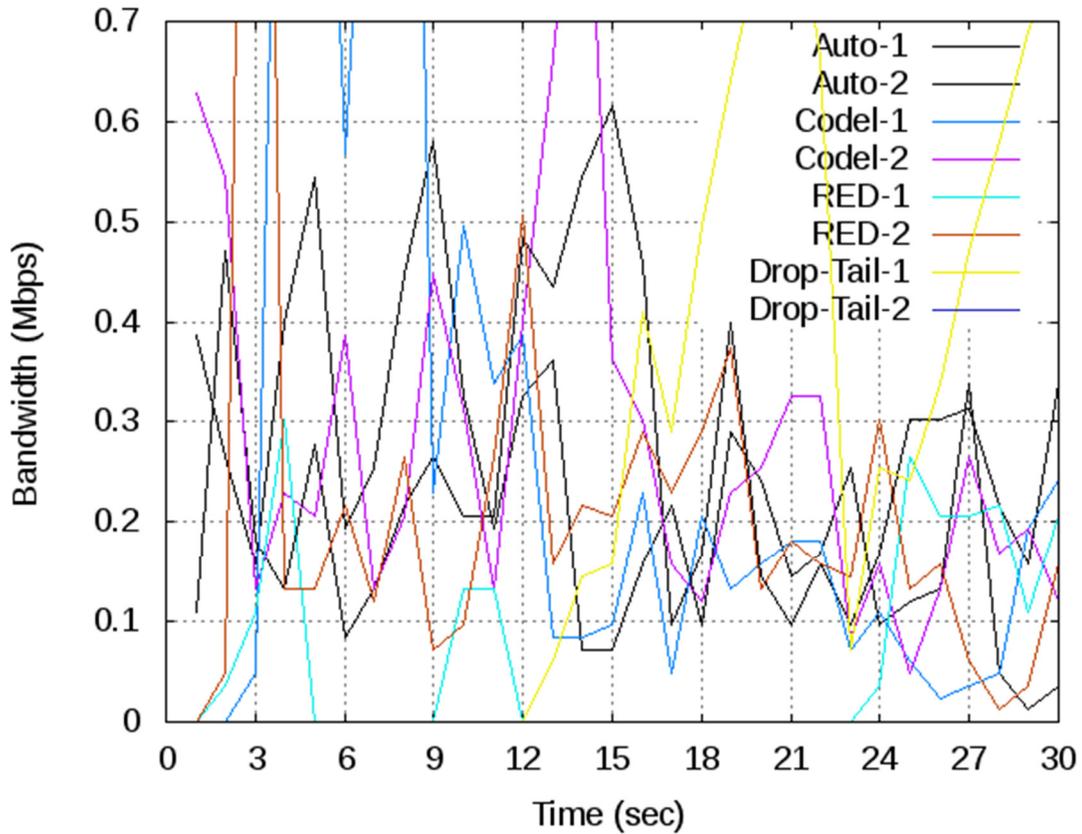


الشكل 5-19 التأخير الذي تفرضه كل الخوارزميات تحت الدراسه عند 100 حزمه

### 6.3.5 مقارنة العدالة لكل الخوارزميات عند طول الصف 100

عندما يكون طول الصف 100 حزمه فإنه في هذه الحاله يظهر جليا ان اي إتصال يكون ذو معدل إرسال اكبر من الاخرين يتحصل على حيز اكبر من عرض النطاق الارسال, تفسير ذلك يعود الى ان الصف الطويل يسمح لاتصال ما بأن يحتفظ بعدد حزم اكبر فيه وعند إرسال هذه الحزم عندما

يأتي دورها في الخرج يتحصل هذا الاتصال على الحيز الاكبر من عرض النطاق, تم تجميع نتائج مقارنة كل إتصاليين من الاشكال رقم (5-12) والشكل رقم (5-14) والشكل رقم (5-16) والشكل رقم (5-18) في شكل واحد وهو الشكل رقم (5-20), من الاشكال المذكوره سابقا يتضح مستوى العدالة بين كل اتصاليين في كل خوارزمية على حده, تجميع هذه الاشكال في شكل واحد ليس بغرض معرفة مستوى العدالة فقط ولكن بغرض معرفة مستوى العدالة بالنسبه لعرض النطاق الارسال حيث يمكن ان يكون ما يتحصل عليه الاتصاليين تحت الدراسه متناسب مع بعضه او انها متعادلين ولكن قد تكون الخوارزمية تضيع حيز عرض النطاق بغرض المحافظة على العدالة, تحتوي هذه التجربه على 100 إتصال وعرض النطاق الخرج تحت التنافس بسعة 10Mbps عند قسمة 10Mbps على عدد الإتصالات يتحصل كل إتصال على حيز قدره 100Kbps وذلك طبعا عند التقسيم بصورة عادله, الخوارزمية الاكثر عدلا بين الاتصالات هي التي يحوم الحيز الذي يتحصل عليه الاتصال حول قيمه 100Kbps اما الخوارزمية الغير عادله فهي التي تعطي إتصال قيمة اكبر بكثير من 100Kbps وإتصال آخر قيمه اقل بكثير من 100kbps, اما الخوارزمية الاسوأ فهي التي تعطي كل او معظم الاتصالات قيمه متعادله ولكن اقل بكثير من 100kbps حيث ان الغرض الاساسي يكون في هذه الحالة هو المحافظة على العدالة بين الاتصالات.



الشكل 5-20 الحيز من عرض النطاق الذي يتحصل عليه إتصالين من مجموعة الاتصالات عند 100 حزمه

#### 4.5 الاستفادة من عرض النطاق

تضمن خوارزمية التقدير الذاتي للصف معدل استفاضة كبير جداً لعرض النطاق عنق الزجاجة، تم مقارنة خوارزمية التقدير الذاتي مع عدة خوارزميات اخرى هي خوارزمية الاسقاط من آخر الصف Drop-Tail، خوارزمية الاسقاط العشوائي المبكر RED، خوارزمية التحكم بتأخير الصف Codel فى الشبكة المحددة فى الشكل (4-1) والتي تحتوى على قناة عنق الزجاجة ذات السعة 10Mbps، الجدول (5-1) يوضح حجم الاستفاضة من قناة عنق الزجاجة فى كل خوارزمية من الخوارزميات الاربعه اعلاه وذلك عندما يكون طول الصف هو 20 حزمه وهو طول قصير مقارنة مع حجم

البيانات وعدد الاتصالات في السيناريو حيث يوضح الجدول ان الاستفادة الاكبر تكون لخوارزمية الاسقاط من آخر الصف ثم لخوارزمية الاسقاط العشوائي ومن بعدها لخوارزمية التقدير الذاتي وأخيراً خوارزمية التحكم بتأخير الصف Codel. ومع معدل تأخير قليل لكل الخوارزميات وذلك بسبب قصر طول الصف وعند زيادة طول الصف الى 100 حزمه كان ترتيب الاستفادة من عرض النطاق نفس الترتيب السابق في الجدول (5-1) مع زيادة كبيرة جدا في التأخير لخوارزميتي الاسقاط من آخر الصف وخوارزمية الاسقاط العشوائي, اذاً خوارزمية التقدير الذاتي وخوارزمية التحكم بتأخير الصف Codel هما الخوارزميتين الفائزتين بمعدل تأخير اقل يضمن عمل تطبيقات الزمن الحقيقي بالاضافة الى ان خوارزمية التقدير الذاتي تفوق خوارزمية التحكم بتأخير الصف Codel في معدل الاستفادة من عرض النطاق عنق الزجاجة, عند استخدام نفس الخوارزمية ولكن بطول صف مختلف اي 20 حزمه و 100 حزمه في خوارزمية الاسقاط من آخر الصف فان هناك زيادة في الاستفادة من عرض النطاق يرجع ذلك الى ان خوارزمية الاسقاط من آخر الصف Drop-Tail لا تقوم باسقاط اي حزمه إلا اذا امتلأ الصف وبالتالي فان هذه الخوارزمية هي الافضل في المحافظة على عرض النطاق وبالتالي فان زيادة الصف تترجم مباشرة الى زيادة في الاستفادة من عرض النطاق, ايضا يتناسب طول الصف طرديا مع ماتتصل عليه خوارزمية الاسقاط المبكر العشوائي RED من عرض نطاق القناة وذلك بسبب انها ايضا لا تقوم بالاسقاط من الصف إلا اذا وصل طول الصف الى حدود معينه هذه مرتبطه بطول الصف, خوارزمية الاسقاط المبكر العشوائي RED لا تقوم بالاسقاط العشوائي الا اذا تجاوز طول الصف العتبه الدنيا  $min_{th}$  ولا تقوم بالاسقاط الالزامي الا اذا تجاوز طول الصف العتبه العليا  $max_{th}$  وبالتالي زيادة طول الصف تترجم ايضا الى زيادة

في الاستفادة من عرض النطاق، يختلف الامر قليلا فيما يتعلق بخوارزمية التحكم بتأخير الصف Codel فان زيادة طول الصف لا تؤثر على الاستفادة من عرض النطاق يرجع ذلك الى ان خوارزمية التحكم بتأخير الصف Codel تعتمد في اسقاط الحزم من الصف على تأخير تلك الحزم، فاذا تجاوز التأخير حدا معيناً تقوم الخوارزمية بالاسقاط وذلك حتى اذا تبقى اكثر من طول غير مستخدماً، خوارزمية التقدير الذاتي لطول الصف auto لا تعتمد في الاسقاط على طول الصف مثل خوارزمتي الاسقاط من آخر الصف Drop-Tail وخوارزمية الاسقاط المبكر العشوائي RED ولا على التأخير مثل خوارزمية التحكم بتأخير الصف Codel ولكن تقوم بتقدير طول الصف المناسب ليعمل عليه الموجه، اي انها تمزج المحافظة على عرض النطاق مع المحافظة على معدل تأخير مقبول للخدمات في خوارزمية واحدة، وبالتالي فان هناك زيادة ملحوظة في الاستفادة من عرض نطاق القناة مع زيادة طول الصف.

نوع الخوارزمية	الاستفاده من عرض النطاق عند طول الصف 20 حزمه	الاستفاده من عرض النطاق عند طول 100 حزمه
Drop-Tail	93.1%	95.7%
RED	84.7%	95.2%
Codel	92.5%	92.3%
Auto	91.5%	95.8%

الجدول 5-1 الاستفادة من عرض نطاق عنق الزجاجة لكل واحدة من الخوارزميات تحت الدراسة

## 5.5 خلاصة

تناول هذا الباب نتائج مقارنة خوارزميات التحكم بالتأخير، وهي خوارزمية الاسقاط من آخر الصف وخوارزمية الاسقاط المبكر العشوائي RED حيث تم تصنيفهما على انهما تعتمدان لاداء عملهما على طول الصف، خوارزمية التحكم بتأخير الصف Codel وخوارزمية التقدير الذاتي لطول الصف auto المصنفتان على انهما يعتمدان على التأخير في الصف، العوامل التي تم مقارنتها هي قياس التأخير وهو العامل الاساسي الذي يميز بين نوعي الخوارزميات، ثم قياس العدالة بين الاتصالات الماره عبر الموجه، خوارزمية الاسقاط من آخر الصف هي الاسواء في كل نتائج المقارنات اما خوارزمية الاسقاط المبكر العشوائي RED فكانت هي الافضل من حيث توفير العدالة ولكنها ذات قيم تأخير كبيرة لانها تعتمد على صف طويل لتوفير العدالة، اما خوارزمية التقدير الذاتي لطول الصف auto فكانت متفوقه دائما علي خوارزمية التحكم بتأخير الصف Codel من حيث توفير العدالة بين الاتصالات ومن حيث المحافظة على معدلات التأخير المناسبه.

## الباب السادس

### 1.6 الخاتمة والتوصيات

الهدف الاساسي لهذا البحث هو ايجاد خوارزميه تساعد في حل مشكلة الازدحام المسبب لمشكلة تضخم الصفوف Bufferbloat وبناءا على الدراسه المكثفه للخوارزميات العامله في مجال فك الازدحام المسبب لمشكلة تضخم الصفوف بالاضافه للخوارزميات الاقدم والعامله ايضا في مجال فك الازدحام والتحكم به خرجت هذه الدراسه بخوارزميه جديده تعمل بصوره عامه على فك الازدحام وبصوره خاصه تعمل على فك الازدحام المسبب لمشكلة تضخم الصفوف, تتفوق هذه الخوارزميه على الخوارزميات الاخرى بامكانيه تطبيقها على الموجهات المفصليه وذلك لانها تعتمد على ابسط الطرق لاداء عملها, تتمثل الطرق البسيطه هذه في نقطتين, النقطه الاولى هي تقدير عدد الاتصالات الحيه الماره بالموجه حاليا والنقطه الثانيه تتمثل في تقدير طول الصف المناسب للعمل عليه في الموجه والذي يضمن التخلص من تضخم الصف.

**تقدير عدد الاتصالات الحيه الماره بالموجه:** تقوم خوارزميه التقدير الذاتي لطول الصف auto لحساب طول الصف المناسب للعمل عليه والذي يحافظ على قيمه التأخير اقل ما يمكن على تقدير عدد الاتصالات الحيه, طريقه تقدير عدد الاتصالات الحيه طريقه بسيطه وسهله حيث تستخلص هذه الطريقه عدد الاتصالات من الحزم الداخله الى صف الموجه, حيث يتم مقارنة الحزمه الواصله حديثا مع حزمه واحده من مجموعه الحزم التي وصلت قبلها وذلك بصوره عشوائيه, من عمليه المقارنه العشوائيه هذه للرسائل يتم تقدير عدد الاتصالات الحيه, يمكن تحديد عدد الاتصالات الحيه بصوره دقيقه ولكن ذلك يحتاج الى جهد كبير من الموجه لذلك فان عمليه التقدير التي تسلكها

خوارزمية التقدير الذاتي لطول الصف auto لن تشكل اي حمل كبير على معالج الموجه او على حيز التخزين.

حساب طول الصف المناسب للعمل بالموجه: لحساب طول الصف المناسب ليعمل عليه الموجه والذي يحافظ على قيمة التأخير في قيمتها الصغرى وذلك طبعاً من غير التفريط بعرض النطاق عنق الزجاجه, يتم حساب طول الصف بصوره تقريبيه وذلك اعتماداً على عدد الاتصالات الحيه وعرض النطاق عنق الزجاجه بالاضافه الى قيمه متوسطه للتأخير الكلى RTT على القناة, تعتمد خوارزمية التقدير الذاتي لطول الصف auto لفك الازدحام والتخلص منه على تقدير طول الصف العامل بالموجه, الهدف الاساسي لتقدير طول الصف هو منع الحزم من التراكم والانتظار لازمان طويله تؤدي لان تكون هذه الحزم غير ذات فائده عند وصولها الى الهدف, اختارت خوارزمية التقدير الذاتي لطول الصف auto حساب طول الصف بصوره ديناميكيه وذلك عند ورود كل حزمه لعدة اسباب منها:

- الخوارزميات السابقه والتي حاولت حل مشكله الازدحام وبالتالي المساعدة في تقليل التأخير على صف الموجه كانت تركز على موضوع اسقاط الحزم من صف الموجه, كمثال على الخوارزميات التي تعتمد على التحكم بالاسقاط من الصف خوارزمية الاسقاط المبكر العشوائي RED, ظن المصممون لخوارزمية الاسقاط المبكر العشوائي RED ان الطريق للتحكم بالازدحام هو التحكم بالاسقاط عن طريق تقسيم طول الصف الى اقسام محدده ويكون لكل قسم طريقه اسقاط معينه, وبما ان القسم المعين للاسقاط منه هو طول محدد للصف كانت خوارزمية الاسقاط المبكر العشوائي RED لا تقوم بتغيير طريقه الاسقاط الا

إذا تجاوزت الحزم الحد المعين إلى الحد الآخر، وبالتالي فإن خوارزمية الإسقاط المبكر العشوائي RED وعند أطوال محدده للصف لا تقوم بالإسقاط وإن زاد التأخير عن المسموح وادى إلى تراكم الحزم وادى في النهاية إلى مشكلة تضخم الصف Bufferbloat.

- مجموعه كبيره من الخوارزميات كانت تركز على توفير العدالة بين الاتصالات اما بصوره تقليديه عن طريق تقسيم الصف إلى اقسام محدده تعتمد على عدد الاتصالات الماره وبالتالي فإن التأخير في هذه الحاله سيكون كبيرا وخاصة عند زيادة عدد الاتصالات بصوره كبيره، او أيضا توفير العدالة ولكن بصوره غير تقليديه عن طريق الإسقاط العشوائي من الحزم الموجوده بالصف ولكن بصوره غير متوقعه تعتمد هذه الطريقه أيضا على طول الصف فالإسقاط بصوره عادله لابد ان يكون الصف طويلا وذلك ليتم تمثيل كل اتصال باعداد متقاربه من الحزم وذلك لتتمكن طريقه الإسقاط العشوائي من الإسقاط بصوره عادله من مجموعه الحزم.

## 2.6 محددات عمل خوارزمية التقدير الذاتي لطول الصف auto

لا تدعي خوارزمية التقدير الذاتي لطول الصف auto انها تحل كل المشاكل الخاصه بالازدحام المسبب لتضخم الصفوف حيث ان هناك بعض الاشكالات والتي تحتاج إلى بحوث اضافيه لتتمكن الخوارزمية من التغلب عليها من هذه العوائق:

عدم قدرة الخوارزمية على تقدير التأخير الكلي قبل الإسقاط: التأخير الكلي هو التأخير الذي تتحمله الحزمه منذ خروجها من المرسل وحتى وصولها إلى المستقبل، وكما هو معلوم فإن التأخير

الكلي يتفاوت من اتصال الى اتصال اخر وحتى بين الاتصالات ذات المصدر الواحد والاهداف المختلفة, عند ورود هذه الحزم ذات ازمان التأخير المختلفة الى الموجه auto فانه لا يستطيع ان يميز بين الحزم ذات قيمة التأخير الصغير والتي تتحمل المزيد من التأخير قبل ان تصل الى الهدف النهائي والحزم التي لن تتحمل اي تاخير في الموجه وبالتالي فانه عند حدوث الازدحام على الموجه فان الموجه يقوم باسقاط الحزم من كلا الاتصاليين, يعتبر الاسقاط من الاتصال الذي لا يتحمل اي تاخير مفيد للشبكه حيث ان هذه الحزم تصبح غير ذات فائده بعد فتره محدده من الارسال, اما الاسقاط من الاتصال الذي يتحمل قيمة تاخير كبيره يعتبر غير مفيد للشبكه حيث ان هذه الحزم كان يمكن ان تنتظر فتره اطول وذلك اذا لم يسقطها الموجه auto.

**الخوارزمية تقوم بتقدير طول الصف عند كل حزمه تدخل الى الموجه:** يعني ذلك ان الموجه auto يقوم وعند استلام اي حزمه بتقدير طول الصف المناسب للعمل عليه, وهي عملية مكلفه بالنسبه لمعالج الموجه وتستهلك الكثير من زمن تنفيذ المعالج.

### 3.6 التوصيات

تناول هذا البحث موضوع الزيادة الكبيره في عدد مستخدمي شبكة الانترنت والذي لم تقابله الزيادة المناسبه في قدرة الشبكه على خدمة كل هولاء المستخدمين والتخلص من التأخير الناتج عن ذلك. كانت نتيجة هذا البحث هو خوارزمية تحافظ على قيمة التأخير اقل مايمكن والتي نشرت في ورقه بحثيه [62], ولكن هناك ثمن سيدفعه معالج الموجه للمحافظه على هذا التأخير, يعني ذلك ان خوارزمية التقدير الذاتي لطول الصف auto تقوم وعند استلام اي حزمه بتقدير طول الصف

المناسب للعمل عليه, حيث انه يمكن تبسيط هذه العملية وبالتالي تحسين كفاءة الخوارزمية, بما ان طول الصف غير حساس جدا ليتم تعديله عند كل حزمه, فان القيام بعملية التعديل بعد عدد اكبر من عدد واحد حزمه داخله يعتبر ميزه اضافيه خوارزمية التقدير الذاتي لطول الصف auto, لذلك يمكن اجراء بعض التعديلات على خوارزمية التقدير الذاتي لطول الصف auto بحيث يتم تقدير طول الصف بعد دخول عدد  $n$  من الحزم, تساهم هذه العملية في تبسيط الخوارزمية بصورة كبيره حيث ان الخوارزمية تستمد قوتها من كونها بسيطه وسهلة التطبيق.

1. Jain, R., *Congestion control in computer networks: Issues and trends*. IEEE Network, 1990. **4**(3): p. 24-30.
2. Paul, S., J. Pan, and R. Jain, *Architectures for the future networks and the next generation Internet: A survey*. Computer Communications, 2011. **34**(1): p. 2-42.
3. Floyd, S. and V. Jacobson, *Random early detection gateways for congestion avoidance*. Networking, IEEE/ACM Transactions on, 1993. **1**(4): p. 397-413.
4. Villamizar, C. and C. Song, *High performance TCP in ANSNET*. ACM SIGCOMM Computer Communication Review, 1994. **24**(5): p. 45-60.
5. Mathis, M., et al., *The macroscopic behavior of the TCP congestion avoidance algorithm*. ACM SIGCOMM Computer Communication Review, 1997. **27**(3): p. 67-82.
6. Shenker, S., L. Zhang, and D.D. Clark, *Some observations on the dynamics of a congestion control algorithm*. ACM SIGCOMM Computer Communication Review, 1990. **20**(5): p. 30-39.
7. Pahdye, J. and S. Floyd, *On inferring TCP behavior*. ACM SIGCOMM Computer Communication Review, 2001. **31**(4): p. 287-298.
8. Jiang, H. and C. Dovrolis. *Why is the internet traffic bursty in short time scales?* in *ACM SIGMETRICS Performance Evaluation Review*. 2005: ACM.
9. Allman, M. and E. Blanton, *Notes on burst mitigation for transport protocols*. ACM SIGCOMM Computer Communication Review, 2005. **35**(2): p. 53-60.
10. Blanton, E. and M. Allman, *On the impact of bursting on TCP performance*, in *Passive and Active Network Measurement*. 2005, Springer. p. 1-12.
11. Beheshti, N., et al. *Experimental study of router buffer sizing*. in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. 2008: ACM.
12. Wischik, D. and N. McKeown, *Part I: Buffer sizes for core routers*. ACM SIGCOMM Computer Communication Review, 2005. **35**(3): p. 75-78.
13. Raina, G., D. Towsley, and D. Wischik, *Part II: Control theory for buffer sizing*. ACM SIGCOMM Computer Communication Review, 2005. **35**(3): p. 79-82.
14. Enachescu, M., et al., *Part III: Routers with very small buffers*. ACM SIGCOMM Computer Communication Review, 2005. **35**(3): p. 83-90.
15. Gerla, M., H. Chan, and J.B. De Marca. *Fairness in computer networks*. in *Proc. IEEE Int. Conf. Commun*. 1985.
16. Jain, R., D.-M. Chiu, and W.R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. 1984: Eastern Research Laboratory, Digital Equipment Corporation.
17. Floyd, S. and V. Jacobson, *Link-sharing and resource management models for packet networks*. Networking, IEEE/ACM Transactions on, 1995. **3**(4): p. 365-386.
18. Bashir, O.A., et al., *Performance Evaluation of Controlling High Bandwidth Flows by RED-PD*. Performance Evaluation, 2013. **67**(13).
19. Mahajan, R., S. Floyd, and D. Wetherall. *Controlling high-bandwidth flows at the congested router*. in *Network Protocols, 2001. Ninth International Conference on*. 2001: IEEE.
20. Wang, M. and Y. Ganjali, *The effects of fairness in buffer sizing*, in *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*. 2007, Springer. p. 867-878.

21. Appenzeller, G., I. Keslassy, and N. McKeown, *Sizing router buffers*. Vol. 34. 2004: ACM.
22. Postel, J., *User datagram protocol*. Isi, 1980.
23. Postel, J., *Transmission control protocol*. 1981.
24. Stevens, W.R., M. Allman, and V. Paxson, *TCP congestion control*. Consultant, 1999.
25. Jacobson, V. *Congestion avoidance and control*. in *ACM SIGCOMM Computer Communication Review*. 1988: ACM.
26. Zhang, L., S. Shenker, and D.D. Clark. *Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic*. in *ACM SIGCOMM Computer Communication Review*. 1991: ACM.
27. Gettys, J. and K. Nichols, *Bufferbloat: Dark buffers in the internet*. Queue, 2011. **9**(11): p. 40.
28. Chirichella, C. and D. Rossi. *To the Moon and back: are Internet bufferbloat delays really that large?* in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. 2013: IEEE.
29. Groenewegen, D. and H. Kleppe, *Detecting and quantifying bufferbloat in network paths*. 2011, Tech. rep., 2011.[bib]. pdf].
30. Allman, M., *Comments on bufferbloat*. ACM SIGCOMM Computer Communication Review, 2012. **43**(1): p. 30-37.
31. Gijssbers, D.D.A.B. and D.D. Akkoorath, *Performance simulation of buffer bloat in routers*. 2011, March.
32. Nagle, J., *On packet switches with infinite storage*. 1985.
33. Staff, C., *BufferBloat: what's wrong with the internet?'*. Commun. ACM, 2012. **55**(2): p. 40-47.
34. The International Telecommunication Union (ITU) Recommendation G.114 and O.W.T. Time.
35. Nichols, K. and V. Jacobson, *A Modern AQM is just one piece of the solution to bufferbloat*. ACM Queue Networks, 2012. **10**(5).
36. McCanne, S. and S. Floyd, *ns-LBNL network simulator*. 2005.
37. Jacobson, V., *Modified TCP congestion avoidance algorithm*. end2end-interest mailing list, 1990.
38. Pan, R., B. Prabhakar, and K. Psounis. *CHoKE-a stateless active queue management scheme for approximating fair bandwidth allocation*. in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. 2000: IEEE.
39. Fall, K. and S. Floyd, *Simulation-based comparisons of Tahoe, Reno and SACK TCP*. ACM SIGCOMM Computer Communication Review, 1996. **26**(3): p. 5-21.
40. Kelly, T., S. Floyd, and S. Shenker, *Patterns of congestion collapse*. International Computer Science Institute, and University of Cambridge, 2003.
41. Shreedhar, M. and G. Varghese, *Efficient fair queuing using deficit round-robin*. Networking, IEEE/ACM Transactions on, 1996. **4**(3): p. 375-385.
42. Hahne, E.L. and R.G. Gallager, *Round robin scheduling for fair flow control in data communication networks*. 1986, DTIC Document.
43. Stoica, I., S. Shenker, and H. Zhang, *Core-stateless fair queueing: a scalable architecture to approximate fair bandwidth allocations in high-speed networks*. IEEE/ACM Transactions on Networking (TON), 2003. **11**(1): p. 33-46.
44. Lin, D. and R. Morris. *Dynamics of random early detection*. in *ACM SIGCOMM Computer Communication Review*. 1997: ACM.

45. Braden, B., et al., *Recommendations on queue management and congestion avoidance in the Internet*. 1998.
46. Alemu, T. and A. Jean-Marie. *Dynamic configuration of RED parameters [random early detection]*. in *Global Telecommunications Conference, 2004. GLOBECOM'04*. IEEE. 2004: IEEE.
47. May, M., et al. *Reasons not to deploy RED*. in *Quality of Service, 1999. IWQoS'99. 1999 Seventh International Workshop on*. 1999: IEEE.
48. Christiansen, M., et al., *Tuning RED for web traffic*. IEEE/ACM Transactions on Networking (TON), 2001. **9**(3): p. 249-264.
49. Feng, W.-c., et al. *A self-configuring RED gateway*. in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. 1999: IEEE.
50. Floyd, S., R. Gummadi, and S. Shenker, *Adaptive RED: An algorithm for increasing the robustness of RED's active queue management*. Preprint, available at <http://www.icir.org/floyd/papers.html>, 2001.
51. Ott, T.J., T. Lakshman, and L.H. Wong. *Sred: stabilized red*. in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. 1999: IEEE.
52. Tang, A., J. Wang, and S.H. Low, *Understanding CHOKe: throughput and spatial characteristics*. IEEE/ACM Transactions on Networking (TON), 2004. **12**(4): p. 694-707.
53. Eshete, A. and Y. Jiang, *Generalizing the CHOKe flow protection*. Computer Networks, 2013. **57**(1): p. 147-161.
54. Stanojevic, R., R.N. Shorten, and C.M. Kellett, *Adaptive tuning of drop-tail buffers for reducing queueing delays*. Communications Letters, IEEE, 2006. **10**(7): p. 570-572.
55. Enachescu, M., et al. *Routers with Very Small Buffers*. in *INFOCOM*. 2006.
56. Pan, R., et al. *Pie: A lightweight control scheme to address the bufferbloat problem*. in *High Performance Switching and Routing (HPSR), 2013 IEEE 14th International Conference on*. 2013: IEEE.
57. Chirichella, C., et al. *Passive bufferbloat measurement exploiting transport layer information*. in *IEEE GLOBECOM*. 2013.
58. Chirichella, C., et al. *Remotely gauging upstream bufferbloat delays*. in *Passive and Active Measurement*. 2013: Springer.
59. Raghuvanshi, D.M., B. Annappa, and M.P. Tahiliani. *On the Effectiveness of CoDel for Active Queue Management*. in *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on*. 2013: IEEE.
60. Allman, M., *On the generation and use of TCP acknowledgments*. ACM SIGCOMM Computer Communication Review, 1998. **28**(5): p. 4-21.
61. Yi, S., et al., *Technique for estimating the number of active flows in high-speed networks*. ETRI journal, 2008. **30**(2): p. 194-204.
62. Bashir, O.A., Y.A. Mohamed, and M.A. Elshaikh, *Delay Reduction on Routers using Adaptive Queue Length Calculation*. Journal of Engineering and Computer Science, 2015. **16**(2).

## 8 الملاحق

يحتوي هذا الجزء من البحث على الشفرة المصدرية لملفي الرأس والجسم بلغة C++ والتي تحتوي على التعديلات المطلوبة على محاكي ns-2 لاجراء التجربه بواسطة الخوارزمية الجديده خوارزمية التقدير الذاتي لطول الصف auto, ويحتوي ايضا على برنامج المحاكاة المكتوب بلغة TCL.

### 8.1 برنامج المحاكاة بلغة TCL

في الفقرات التاليه سنقوم بعرض وشرح البرنامج الذي تم بواسطته اجراء التجارب المعملية بواسطة المحاكي ns2, كتب هذا البرنامج بلغة البرمجه TCL, حيث ان المحاكي كتب بواسطة لغتي برمجه هما لغة C++ و لغة TCL, نسبة لقوة لغة C++ لذلك كتبت بها البنيه التحتيه للمحاكي اما واجهة المحاكي فكتبت بواسطة لغة TCL, سوف يتم شرح محتويات البرنامج وذلك لفهم التجارب التي تم اجراءها لمقارنة الخوارزميات المختلفه في هذه الدراسه.

في السطر الاول من البرنامج تم تحديد طول زمن التجربه بمائة ثانيه

```
set stopTime 100  
set ns [new Simulator]
```

يحدد هذا السطر عدد الاتصالات بمائة اتصال تقوم جميعها بالارسال في نفس الوقت

```
set num_ftp 100
```

يتم تحديد عرض النطاق عنق الزجاجة الى 10Mbps

```
set bottleneck 10Mb
```

تم تحديد طول الملف الذي يقوم اي اتصال من الاتصالات المائه بارساله, تم اختيار هذا الرقم الكبير حتى يكون لاي اتصال كمية كبيره من البيانات ليقوم بارسالها وحتى لايتوقف من الارسال بسبب انتهاء كمية البيانات التي يقوم بارسالها.

```
#for a 10MB ftp
```

```
set filesize 1000000000000000
```

في هذا السطر من البرنامج تم تحديد اسم الملف out.tr ليكون هو الملف الذي تكتب عليه كل مخرجات التجربه, بعد انتهاء التجربه يأخذ كل جزء من دوال تحليل عمل الخوارزميه مدخلاته من هذا الملف.

```
set nf [open out.tr w]
```

```
$ns trace-all $nf
```

تم تحديد طول الحزمه حيث يساوي 1500byte بواسطة الثابت psize

```
# experiment settings
```

```
set psize 1500
```

الثابت accessdly هو التأخير من الجهاز المرسل الى الموجه

```
set accessdly 20
```

الثابت bdelay هو التأخير بين الموجهين 0 و 1

```
set bdelay 50
```

## حاصل ضرب عرض النطاق في التأخير (BDP)

تم حساب عرض النطاق الارسال من الجهاز المرسل الى الجهاز الهدف بواسطة المعادلة ادناه, حيث ضربت عرض النطاق في `realrtt` وهو زمن ارسال الحزمة من الجهاز المرسل وعودة حزمة الاقرار عليها ثم قسم الناتج على طول الحزمة مضروباً في العدد 8 وذلك لتحسب سعة على انها عدد رسائل.

```
set bdp [expr round($bw*$realrtt/(8*$psize))]
```

بداية زمن المحاكاة هو الزمن 0.0

```
set startTime 0.0
```

نافذة الازدحام لكل جهاز مرسل على الشبكة هو  $bdp * 16$  اي عرض النطاق مضروب في الثابت 16 وذلك ليكون حجم النافذة كبير كفايه وذلك حتى لا يؤثر في اداء الجهاز المرسل :

```
Agent/TCP set window_ [expr $bdp*16]
```

حجم الحزمة هو 1460 بايت

```
Agent/TCP set segsize_ [expr $psize-40]
```

```
Agent/TCP set packetSize_ [expr $psize-40]
```

نافذة الازدحام الاولى لخوارزمية البدايه البطيئه

```
Agent/TCP set windowInit_ 4
```

وجوب استخدام خيار الختم الزمني المستخدم للحصول على قيمه دقيقه للزمن rtt

```
Agent/TCP set timestamps_ true
```

ضبط طول الصف

```
#buffersizes
```

```
set buffersize1 [expr $bdp*10]
```

تصميم شبكة المحاكاة - الطوبولوجي

```
proc build_topology { ns which } {
```

```
    global bw bdelay accessdly buffersize buffersize1 filesize node_cnt
```

```
    set node_cnt 2
```

العقدة n0 هي الموجه 0 والعقدة n1 هي الموجه 1

```
    # congested link
```

```
    global n0 n1
```

```
    set n0 [$ns node]
```

```
    set n1 [$ns node]
```

في هذا السطر من البرنامج تم ربط الموجهين الاول والثاني بواسطة قناة اتصال ترسل وتستقبل

البيانات duplex-link, اما المتغير bw فهو يحدد عرض النطاق المستخدم في التجربه والمتغير

`bdelay` فهو يحدد التأخير الكلي من الموجه 0 الى الموجه 1, اما خوارزمية التقدير الذاتي لطول الصف `auto` فهي تحدد ان الخوارزميه المستخدمه في الموجه هي خوارزمية `auto` في هذه التجريه.

```
$ns duplex-link $n0 $n1 $bw ${bdelay}ms auto
```

### ضبط طول الصف

طول الصف في الموجه 0 هو 20 حزمه وطول الصف في الموجه 1 هو ايضا 20 حزمه

```
$ns queue-limit $n0 $n1 20
```

```
$ns queue-limit $n1 $n0 20
```

```
global num_ftps web_rate num_cbrs greedy num_revs
```

```
set linkbw [expr $bw*1]
```

```
#need to fix the angles if use nam
```

انشاء الاتصالات التي تقوم بارسال واستقبال البيانات في هذه الفقره تم انشاء الاتصالات التي تقوم بارسال البيانات `num_ftps` يحدد عدد الاتصالات بمائة اتصال وبالتالي فان الداله `for` تقوم بانشاء مائة اتصال, في الجزئيه الاولى يتم انشاء المخدمات وفي الجزئيه الثانيه يتم انشاء الزبائن

```
for {set k 0} {$k < $num_ftps} {incr k 1} {
```

المخدمات

```
# servers
set j $node_cnt
global n$j
set n$j [$ns node]
```

في هذا السطر من البرنامج تم ربط الموجه الاول والزيون n\$j بواسطة قناة اتصال ترسل وتستقبل البيانات duplex-link, اما المتغير linkbw فهو يحدد عرض النطاق المستخدم في التجريه والمتغير accessdly فهو يحدد التأخير الكلي من الموجه 0 الى الزيون, اما خوارزمية الاسقاط من آخر الصف Drop-Tail فهي تحدد ان الخوارزميه المستخدمه في الموجه هي خوارزمية الاسقاط من آخر الصف Drop-Tail في هذه التجريه.

```
$ns duplex-link [set n$j] $n0 $linkbw ${accessdly}ms DropTail
```

تحديد طول الصف

في هذين السطرين يتم تحديد طول الصف من جهاز الزيون الى الموجه 0 ومن الموجه 0 الى جهاز الزيون, حيث ان buffersize1 يساوي حاصل ضرب  $10 * bdp$

```
$ns queue-limit [set n$j] $n0 $buffersize1
$ns queue-limit $n0 [set n$j] $buffersize1
```

```
incr node_cnt
```

الزيائن

```
# clients
set j $node_cnt
```

```

global n$j
set n$j [$ns node]
set dly [expr ${accessdly} +($k+1)]

```

في هذا السطر من البرنامج تم ربط الموجه الاول والزيون n\$j بواسطة قناة اتصال ترسل وتستقبل البيانات duplex-link, اما المتغير linkbw فهو يحدد عرض النطاق المستخدم في التجربه والمتغير dly فهو يحدد التأخير الكلي من الزيون الى الموجه 0, اما الاسقاط من آخر الصف Drop-Tail فهي تحدد ان الخوارزميه المستخدمه في الموجه هي خوارزمية الاسقاط من آخر الصف Drop-Tail في هذه التجربه.

```

$ns duplex-link $n1 [set n$j] $linkbw ${dly}ms DropTail

```

تحديد طول الصف

في هذين السطرين يتم تحديد طول الصف من جهاز الزيون الى الموجه 0 ومن الموجه 0 الى جهاز الزيون, حيث ان buffersize1 يساوي حاصل ضرب  $bdp * 10$

```

$ns queue-limit $n1 [set n$j] $buffersize1
$ns queue-limit [set n$j] $n1 $buffersize1
incr node_cnt
}

```

بناء تطبيقات المرسلين والمستقبلين

في هذه الفقرة يتم انشاء وربط تطبيقات الزبائن cnd وتطبيقات المخدمات snd بواسطة الداله proc build\_ftpcient, حيث ان startTime هو زمن بدء الارسال و timeToStop هو زمن انتهاء الارسال, Flow\_id هو رقم فريد لكل اتصال

```
# cnd is client node, snd is server node
```

```
proc build_ftpcient {cnd snd startTime timeToStop Flow_id} {
```

```
global ns filesize
```

هذا السطر يوضح انشاء وربط الزبون مع المخدم

```
set ctcp [$ns create-connection TCP/Reno $snd TCPSink $cnd $Flow_id]
```

انشاء تطبيق من نوع FTP

```
set ftp [$ctcp attach-app FTP]
```

```
$ftp set type_ FTP
```

تحديد زمن بدء الارسال

```
$ns at $startTime "$ftp send $filesize"
```

تحديد زمن انتهاء الارسال

```
$ns at $timeToStop "$ftp stop"
```

## 8.ب نموذج خوارزمية الاسقاط المبكر العشوائي المستقره SRED لتقدير عدد

### الاتصالات

تتمركز فكرة خوارزمية الاسقاط المبكر العشوائي المستقره SRED حول إيجاد علاقة بين تردد تتطابق الحزم مع عدد الاتصالات الحية المارة، في هذه الفقرة سوف يتم توضيح النموذج بواسطة سلسلة ماركوف Markov chain لاستيعاب مفاهيم خوارزمية الاسقاط المبكر العشوائي المستقره SRED وتحليل سلوكها في تقدير عدد الاتصالات.

افتراض ان هناك ذاكرة مخبئية صغيرة او ما يسمى بقائمة الاموات Zombie list ذات حجم  $M$  و  $N$  هو عدد الاتصالات المستقلة، لكل إتصال معدل وصول  $\lambda_i$  حزمة في الثانية، نفترض ان الحزم التي تصل لكل إتصال مستقلة، افتراض ان  $X_i$  هو عدد حجرات التخزين للاتصال ذو الرقم الفريد  $i$  و  $X$  هو متجه يحتوي على عدد الحجرات التي يشغلها اي إتصال، بالتالي فإن  $X$  يمكن تمثيله في شكل سلسلة ماركوف Markov chain ذات تحولات عند وصول الحزم في الزمن، وبالتالي فإن عدد الحجرات الذي سوف تشغله من قائمة الاموات يحسب من المعادله (1-8)

$$\sum_{i=1}^N X_i = M \quad (1-8)$$

و

$$\bar{X} \in \{ m = (m_1 \dots m_i \dots m_N)^T \mid 0 \leq \sum m_i \leq M, \sum m_i = M \} = S_{M,N}$$

معدل تحولات سلسلة ماركوف مستقلة عند حالتين الاولى هي الاحلال الاحتمالي (الاحلال هو ان تحل الحزمة التي تصل محل الحزمة الموجوده اصلا) والحاله الثانية هي حالة المقارنة بين الحزم التي تصل والحزمة المختاره عشوائيا. اعتمادا على خوارزمية الاسقاط المبكر العشوائي المستقره SRED فانه لكل حالة قائمة الاموات معطاة  $\bar{m}$  فإن عدد حجرات التخزين لكل إتصال موجود يظل كما هو عند اي من حالتي التطابق او عدم التطابق باحتمال  $1-r$ , ولكن تتغير حالة قائمة الاموات من  $\bar{m}$  الى  $\delta_{ij} \bar{m}$  عند حالة عدم التطابق باحتمال  $r$ .

يكون للحزمة التي تصل للاتصال ID  $i$  حجرة التخزين المختارة بصورة عشوائيه من قائمة الاموات تخص الاتصال ID  $j$  عند حالة قائمة الاموات  $\bar{m}$  يتم تبديلها بواسطة الاتصال ID  $i$  بواسطة إحتمال الاحلال  $r$  فإن الحاله الجديدة للقائمة  $\delta_{ij} \bar{m}$  تعرف بواسطة المعادلة (2-8)

$$\delta_{ij} \bar{m} = (m_1 \dots m_i + 1 \dots m_j, -1 \dots m_N)^T \text{ if } i < j, \quad (2-8)$$

لكل حالات قائمة الاموات

$$\bar{m} \quad \text{IF } i > j, \quad m_i < M \text{ and } 0 < m_j, \quad \text{حيث ان}$$

$i < j$  فإن  $i$  و  $j$  يمكن تبديلهما. عند حالة خاصة من حالات القائمة  $\bar{m}$ , فإن إحتمال المطابقه هو مضروب احتمال وصول حزمه ذات رقم فريد (ID)  $i$  في إحتمال إختيار حزمه بصورة عشوائية لنفس الاتصال ذو الرقم الفريد ID. وبصورة مشابهة فإن إحتمال عدم المطابقه هو مضروب إحتمال وصول حزمة ذات رقم فريد (ID)  $i$  في إحتمال إختيار حزمه بصوره عشوائية لاتصال مختلف عن

الاتصال  $i$ . وبالتالي فإن احتمال التغير لنموذج قائمة الاموات يمكن كتابته لحالات المطابقيه او حالات عدم المطابقيه من غير إحلال بواسطة المعادله (3-8)

$$\begin{aligned}
 P ( \bar{X}(t+1) = \bar{m} | \bar{X}(t) = m ) \\
 &\equiv P_{hit} ( \bar{m}, \bar{m} ) + P_{miss} ( \bar{m}, \bar{m} ) (1-r) \quad (3-8) \\
 &= \sum_{i=1}^N \frac{m_i}{M} \cdot \frac{\lambda_i}{\sum_{k=1}^N \lambda_k} + \sum_{\substack{i,j \\ i \neq j}}^N \frac{m_j}{M} \cdot \frac{\lambda_i}{\sum_{k=1}^N \lambda_k} (1-r)
 \end{aligned}$$

لكل حالة عدم تطابق في القائمة تم احلالها بواسطة الاحتمال  $r$  يمكن حسابها بواسطة المعادلة (4-8)

$$\begin{aligned}
 P ( \bar{X}(t+1) = \delta_{ij} \bar{m} | \bar{X}(t) = \bar{m} ) \\
 &\equiv P ( \bar{m} , \delta_{ij} \bar{m} ) r = \sum_{\substack{i,j \\ i \neq j}}^N \frac{m_j}{M} \cdot \frac{\lambda_i}{\sum_{k=1}^N \lambda_k} r \quad (4-8)
 \end{aligned}$$

في المعادلة (4-8) فإن  $\frac{\lambda_i}{\sum_{k=1}^N \lambda_k}$  يوضح ان احتمال وصول حزمه ذات رقم فريد ID  $i$  والحددين  $(m_i / M)$  و  $(m_j / M)$  يمثلان احتمال إختيار حجرة من القائمة بصورة عشوائية تملك ذات الرقم الفريد او رقم فريد مختلف. الحالة المسقرة لتردد التوافق لحالة معطاة من نموذج القائمة يمكن توضيحه بواسطة النظرية الموضحة في الفقرة التاليه.

## النظرية 1:

باعتبار الافتراض ان وصول الحزم لكل إتصال مستقل, فإن تردد التطابق والذي يحسب بواسطة عملية الدرجة الاولى لنظام قائمة وحيد في خوارزمية الاسقاط المبكر العشوائي المستقره SRED فإن تقارب الحالة المستقره يحسب من المعادلة (5-8)

$$H = \sum_{\bar{m} \in S_{M,N}} \left( \frac{\prod_{n=1}^N \frac{\lambda_n^{m_n}}{m_n!}}{G_{MN}} \cdot \frac{\sum_{i=1}^N \frac{m_j}{M} \cdot \frac{\lambda_i}{\sum_{k=1}^N \lambda_k}} \right) \quad (5-8)$$

إثبات: لحاله معطاة z وباعتبار ان  $m_i < M$  و  $0 < m_j$ , فإن معادلات التوازن لهذا النظام لكل إاحتمالات  $\pi$  سوف يصبح حسب المعادله (6-8):

$$\pi(\bar{m}) P(\bar{m}, \delta_{ij} \bar{m}) = \pi(\delta_{ij} \bar{m}) P(\delta_{ij} \bar{m}, \bar{m}), \quad (6-8)$$

حيث ان

$$\text{Where } P(\bar{m}, \delta_{ij} \bar{m}) = \frac{m_j}{M} \cdot \frac{\lambda_i}{\sum_{k=1}^N \lambda_k} \text{ and } P(\delta_{ij} \bar{m}, \bar{m}) = \frac{m_i + 1}{M} \cdot \frac{\lambda_j}{\sum_{k=1}^N \lambda_k}$$

$\pi(\bar{m})$  هو توزيع الحالة المستقرة للمعامل  $\bar{X}$  وبالتالي فإن المعادلة (6-8) تصبح

$$\pi(\bar{m}) = \pi(\delta_{ij}\bar{m}) \frac{\lambda_j}{m_j} \cdot \frac{m_i + 1}{\lambda_i} \quad (7-8)$$

وبالتالي فإنه يمكن حل معادلة التوازن التفصيلية:

$$\pi(\bar{m}) = \prod_{n=1}^N \frac{\lambda_n^{m_n}}{m_n!} / G_{MN} \quad (8-8)$$

حيث ان الثابت الاعتباري:

$$G_{MN} = \sum_{\bar{m} \in S_{MN}} \prod_{n=1}^N \frac{\lambda_n^{m_n}}{m_n!} \quad (9-8)$$

وبالتالي فإن  $\bar{X}$  عبارة عن معكوس زمني من المعدلة (7-8)، (8-8) و (9-8) نحصل على المعادلة

$$(10-8)$$

$$\pi(\delta_{ij}\bar{m}) = \prod_{n \neq i,j}^N \frac{\lambda_n^{m_n}}{m_n!} \cdot \frac{\lambda_i^{m_i+1}}{(m_i+1)!} \cdot \frac{\lambda_j^{m_j-1}}{(m_j-1)!} / G_{MN}$$

وبالتالي فإن  $f(t)$  يمثل تردد التطابق، و  $H$  يمثل حالة احتمال التطابق المسفرة لقائمة الاموات.

وبالتالي فإن  $H$  تمثل مجموع كل حالات قائمة الاموات من المعادلة (11-8):

$$H = \sum_{\bar{m} \in S_{MN}} \pi(\bar{m}) P_{hit}(\bar{m}, \bar{m}) \quad (11-8)$$

عمليا يمكن تقدير H بواسطة عملية الدرجة الاولى من المعادلة (12-8)

$$f(t) = (1-\alpha)f(t-1) + \alpha \cdot 1 \quad (12-8)$$

لكل  $0 < \alpha < 1$  في هذه المعادلة 1 (تطابق الحزمة عند الزمن t) يمثل عامل مؤثر لحالة التطابق.

من الواضح ايضا ان H عبارة عن نقطة تحديد للدالة f, وهي حسب المعدلة (13-8)

$$\lim_{t \rightarrow \infty} f(t) \equiv H \quad (13-8)$$

نفترض ان إختيار  $\alpha$  يؤدي الى ان يتقارب f(t) بسرعة اكبر من معدل تغير عدد الاتصالات N.

وبالتالي فإن  $\alpha$  يلعب دورا مهما كثابت زمني يحدد السرعة التي بواسطتها يستطيع النموذج ان يصل

الى مرحلة الاستقرار, في النهاية فإنه من المعادلات (3-8), (8-8) و (11-8) نحصل على

المعادلة (5-8) وبالتالي هذا إثبات للنظرية رقم (1)

بملاحظة ان معدل الوصول  $\lambda_i$  هو المجموع للبسط في المعادلة (5-8) وتم استبدال القيمة القصوى

$\lambda_{\max}$  وبالتالي فإن قيمة H اصغر من او تساوي  $\sum_{k=1}^N \lambda_k$ . بصورة مشابهة اذا كان المعدل

و  $\lambda_{\min}$  هو مجموع البسط في المعادلة (5-8) وتم استبداله بواسطة اقل قيمة, فإن H يصبح اكبر

من او يساوي  $\sum_{k=1}^N \lambda_k$ . وبالتالي نحصل على البديهيات التالية:

**البديهية 1:**

$$\lambda_{\min} \sum_{k=1}^N \lambda_k \leq H \leq \lambda_{\max} \sum_{k=1}^N \lambda_k \quad (14-8)$$

## البديهيه 2:

اذا كانت كل معدلات الوصول  $\lambda$  متساوية لكل الاتصالات  $N$  فإن حدود القيمة القصوى والدنيا للمعامل  $H$  في المعادلة (10) متساوية, يقود ذلك الى ان

$$H = 1/N \quad (15-8)$$

وبالتالي فإن هذه المعادلة تخبرنا ان عدد الاتصالات  $N$  يمكن حسابه من الحالة المستقرة لتردد التظابق للاتصالات بواسطة المعادلة (5-8).

## ج.8 الشفرة المصدرية لملفات ns-2

يحتوي هذا الجزء من البحث على شفرة خوارزمية التقدير الذاتي لطول الصف auto والتي تم تنفيذها على المحاكى ns-2, تنقسم الشفرة الى قسمين هما شفرة الترويسة وشفرة جسم البرنامج.

### شفرة الترويسة

```
#ifndef ns_auto_h
#define ns_auto_h

#include <string.h>
#include "queue.h"
#include "config.h"

#define maxEntries 1000 /* maximum entries allowed
in zombie table*/

class LinkDelay;

struct zombie

{
    int flowid;
    int count;
```

```

    double timestamp;

};

class AUTOQueue : public Queue {

    public:

    AUTOQueue();
    ~AUTOQueue();

    protected:

    int command(int argc, const char*const* argv);
    void enqueue(Packet*);
    Packet* deque();
    PacketQueue *q_;
    int drop_front_;
    TracedInt nm_;
    TracedInt sn_;
    TracedInt rt_;
    void trace(TracedVar*);
    Tcl_Channel tchan_;
    int k; /*no. of entries in zombie list with which
to compare incoming packet*/
    double p_write; /*probability of over writing a
zombie entry in case of a miss*/

```

```

double p_auto; /*the p_auto(q) function*/
double p_zap; /*dropping probability*/
double p_max; /*max drop probability*/
int L; /*for calculating f(t)*/
long double alpha; /*weighting constant for
calculating f(t)*/
double P_t; /*Hit frequency, f(t)*/
double P_n; /*No of flows, 1/f(t)*/
double P_t_prev; /*P(t-1)*/
int hit; /*either 0 or 1*/
int aut;
int at;
int rt;
long double zt;
int numFlows; /*estimated no. of active flows*/
int augment; /*ratio of zombie table capacity to
queue capacity*/
int nextEntry; /* next entry in the zombie
table*/
int zombieCap; /*this is the zommbi capacity in a
given simulation scenario and will*/
/*depend on queue capacity and augmentation
ratio*/
bool zombie_full; /*zombie table full or not*/
struct zombie zombie_table[maxEntries];
void zombie_add(Packet* p);
void zombie_pick(Packet* p);

```

```
    int drop_early(Packet* p);  
};
```

```
#endif
```

## شفرة جسم البرنامج

```
#include <math.h>  
#include <stdlib.h>  
#include "auto.h"  
#include "delay.h"  
#include "random.h"  
#include "flags.h"
```

```
static class AUTOClass : public TclClass {  
public:  
    AUTOClass() : TclClass("Queue/AUTO") {}  
    TclObject* create(int, const char*const*) {  
        return (new AUTOQueue);  
    }  
} class_auto;
```

```
AUTOQueue::AUTOQueue() : tchan_(0)  
{
```

```

q_ = new PacketQueue();
nextEntry = 0;
zombie_full = false;
for (int i = 0; i < maxEntries; i++) {
    zombie_table[i].flowid = -1;
    zombie_table[i].count = 0;
    zombie_table[i].timestamp = 0.0;
}

pq_ = q_;
bind("k", &k);
bind("p_write", &p_write);
bind("p_max", &p_max);
bind("epoch", &L);
bind("augment", &augment);
bind("nm_", &nm_);
bind("sn_", &sn_);
//bind("alpha", &alpha);

/*initialise variables*/
p_auto = 0.0;
p_zap = 0.0;
P_t = 1.0; /*Initially assume just one flow*/
P_t_prev = 0.0;
hit = 0;
numFlows = 0;
nm_ = 0;

```

```
sn_ = 0;
at  = 0;
rt  = 0;
}
```

```
AUTOQueue::~~AUTOQueue() {
```

```
    delete q_;
}
```

```
int AUTOQueue::command(int argc, const char*const*
argv) {
```

```
    Tcl& tcl = Tcl::instance();
    if (argc == 2) {
        if (strcmp(argv[1], "reset") == 0) {
            reset();
            return (TCL_OK);
        }
    } else if (argc == 3) {
        // attach a file for variable tracing
        if (strcmp(argv[1], "attach") == 0) {
            int mode;
            const char* id = argv[2];
            tchan_ = Tcl_GetChannel(tcl.interp(),
(char*)id, &mode);
```

```

        if (tchan_ == 0) {
            tcl.resultf("CoDel trace: can't
attach %s for writing", id);
            return (TCL_ERROR);
        }
        return (TCL_OK);
    }

    if (!strcmp(argv[1], "packetqueue-attach")) {
        delete q_;
        if (!(q_ = (PacketQueue*)
TclObject::lookup(argv[2])))
            return (TCL_ERROR);
        else {
            pq_ = q_;
            return (TCL_OK);

        }
    }

    if (strcmp(argv[1], "link") == 0) {
        return (TCL_OK);

    }
}

return Queue::command(argc, argv);
}

```

```

void AUTOQueue::zombie_add(Packet* p)
{
    hdr_ip *iph = hdr_ip::access(p);
    int fid = iph->flowid();
    double time = Scheduler::instance().clock();

    zombie_table[nextEntry].flowid = fid;
    zombie_table[nextEntry].count = 0;
    zombie_table[nextEntry].timestamp = time;

    nextEntry++;
    if (nextEntry >= zombieCap)
    {
        zombie_full = true;
    }
}

```

```

void AUTOQueue::zombie_pick(Packet* p)
{

    hdr_ip* iph = hdr_ip::access(p);
    int fid = iph->flowid();
    double time = Scheduler::instance().clock();
    /*first pick a random zombie entry*/

```



```

        zombie_table[j].timestamp = time;
    }
    /*else do nothing*/
}
}
/*update estimate of hit frequency*/

P_t = (1 - alpha)*P_t + (alpha*hit);

}

void AUTOQueue::enqueue(Packet* p) {

    long double time_spent;
    clock_t begin, end;
    begin = Scheduler::instance().clock();

    /*First set no. of entries that zombie table will
hold before*/
    /*being set to full status*/
    zombieCap = augment*qlim_;
    alpha = 1.0/zombieCap;

    //double time = Scheduler::instance().clock();
    if (!zombie_full)
        zombie_add(p); /*add entry of this packet to
zombie list*/
}

```

```

    zombie_pick(p); /*compare this packet to random
zombie entry*/

    P_n = sqrt(1/P_t);

    int no = 1/P_t;

    double autol = 2*150/P_n ;

    at = autol ;
    nm_ = no;

    int qlimBytes = autol * 1500;
    rt = q_>length() ;
    if (((q_>length() + 1) >= autol) ||
        ((q_>byteLength() + hdr_cmn::access(p)-
>size()) >= qlimBytes)){
        // if the queue would overflow if we added
this packet...
        /* remove from head of queue */
        q_>enqueue(p);
            int qlen = q_>length();
            int j =
(int)(double(qlen)*rand()/(RAND_MAX + 1.0));
            Packet *pp = q_>lookup(j);
            q_>remove(pp);

```

```

        drop(pp);

    } else {
        q_>enqueue(p);
    }

    end = Scheduler::instance().clock();
    time_spent = (long double)(end - begin);
    long double rf = time_spent;
    zt = zt + rf ;

}

Packet* AUTOQueue::deque() {

    return q_>deque();
}

void
AUTOQueue::trace(TracedVar* v)
{
    char wrk[500];
    const char *p;

    if ((p = strstr(v->name(), "nm")) == NULL) &&
        ((p = strstr(v->name(), "sn")) ==
NULL)) {

```

```

        fprintf(stderr, "AUTO:unknown trace var
%s\n",
            v->name());
        return;
    }

    if (tchan_) {
        int n;
        double t = Scheduler::instance().clock()
// XXX: be compatible with nsv1 RED trace
entries
            if (*p == 'n') {
                sprintf(wrk, "n %g %d %d %d",
t, int(*(TracedInt*) v), at, rt );
            } else if(*p == 's') {
                sprintf(wrk, "s %g %d", t,
int(*(TracedInt*) v));

            }
        n = strlen(wrk);
        wrk[n] = '\n';
        wrk[n+1] = 0;
        (void)Tcl_Write(tchan_, wrk, n+1);
    }
    return;
}

```

## 8. قائمة الاوراق المنشوره

تحتوي هذه الفقره على الاوراق المنشوره اثناء هذا البحث

1. Bashir, O.A., et al., *Performance Evaluation of Controlling High Bandwidth Flows by RED-PD*. Performance Evaluation, 2013. **67**(13).
2. Bashir, O.A., Y.A. Mohamed, and M.A. Elshaikh, *Delay Reduction on Routers using Adaptive Queue Length Calculation*. Journal of Engineering and Computer Science, 2015. **16**(2).