

通信経路中のアクティブキューマネジメントの検出方法

A Detection Method of Active Queue Management in Communication Paths

80815228 岩谷純至 (Junji Iwatani) Supervisor: 矢向高弘 (Takahiro Yakoh)

1. 序論

近年のハードウェアの進歩により、通信経路の高性能化が進み、グリッドコンピューティングに代表されるような IP 網上で大容量データ転送を必要とするアプリケーションが出現してきた。このような変化に対してソフトウェア面における対応も必要である。現在普及している伝送制御プロトコルである TCP Reno には、パケット損失による輻輳検知、経路によっては十分なスループットが得られないという問題がある。

現在までに TCP の改善研究は数多くなされており、待ち行列遅延やスループット等の情報を用いて輻輳を事前に検知し、回避するようなアルゴリズム (CATs: Congestion Avoidance Techniques) が提案されている。CATs は通信経路中の待ち行列管理方式が Drop Tail であることを前提として設計されているものが多いが、現在のネットワークには作弄的な待ち行列管理方式 (AQM: Active Queue Management) が存在するため、CATs が通信を行う経路中に AQM が存在する場合、不具合が発生することが予想される。

本稿では CATs 及び AQM の代表例として FAST TCP と RED に焦点を当てた。まず、FAST TCP が RED を介して通信した際の問題点を示した。次に、3 節で、通信経路中に RED があるということだけがあったとき、制御方法に単純な修正を加えることで通信性能が改善される例を示した。このことから、FAST TCP が通信を行う際には通信経路中に AQM が存在するかを判断する必要があるということが示された。最後に、FAST TCP での通信内で、経路中に RED が存在するか否かを判断する方法について検討した。

2. 関連技術

・FAST TCP

FAST TCP は待ち行列遅延をフィードバック要素として、フィードバック制御理論に基づいた輻輳制御を行う。FAST TCP の輻輳制御ウィンドウサイズの制御式を(1)式に示す。

$$\frac{dcwnd(t)}{dt} = \alpha \gamma \left(1 - \frac{q(t)x}{\alpha} \right) \quad (1)$$

ここで、 q はルータにおける待ち行列保持数、 γ は平滑化ゲイン係数、 x は待ち行列遅延、 x はスループットを示す。FAST TCP はウィンドウサイズを平衡点に収束させるような制御を行うが、パケット損失時には TCP Reno と同様にウィンドウサイズを半減させる。そのため RED によるパケット棄却により、TCP Reno のようなウィンドウサイズの振動を引き起こす可能性があると考えられる。

・RED

RED はバッファ内の平均待ち行列長 avg を監視し、 avg に応じたパケット棄却率 P_a で待ち行列の中からランダムにパケットを選択し棄却する。 avg と P_a の関係を図 1 に示す。ここで、 max_{th} 、 min_{th} 及び max_p は RED の固有定数パラメータである。 $min_{th} < avg < max_{th}$ のとき P_a は以下の(2)(3)式で決定される。

$$P_b = \frac{\max_p (avg - \min_{th})}{\max_{th} - \min_{th}} \quad (2)$$

$$P_a = \frac{P_b}{1 - count \times P_b} \quad (3)$$

ここで $count$ は前回パケットを棄却してからのパケット数を表す。

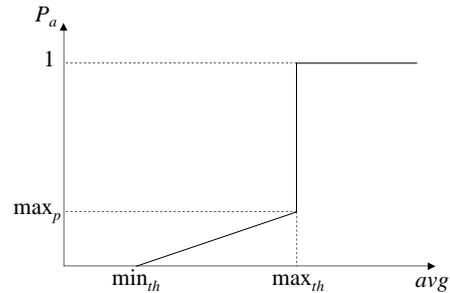


図 1 avg と P_a の関係

3. シミュレーション及び考察

本研究における実験はネットワークシミュレータ ns-2 上で行った。まず、送受信ノード間に RED ルータがあるだけの単純なトポロジで RED 介した FAST TCP の性能を検証した結果を図 2 に示す。結果から、 α の値が大きすぎると TCP Reno のようにウィンドウサイズの振動が発生することがわかった。次に、パケット損失発生時にウィンドウサイズを半減させないよう修正を施した FAST TCP を用いて同様の実験を行った結果を図 3 に示す。図 2 と比較して、少々のパケット損失を無視することでウィンドウサイズが安定した通信を行うことができていくことがわかる。つまり、RED を介した通信では通常とは異なる制御を行ったほうが安定した通信ができる例が示されたと言える。このことから、通信経路中に AQM があるということを検知できるだけでも大きな意味があると言える。

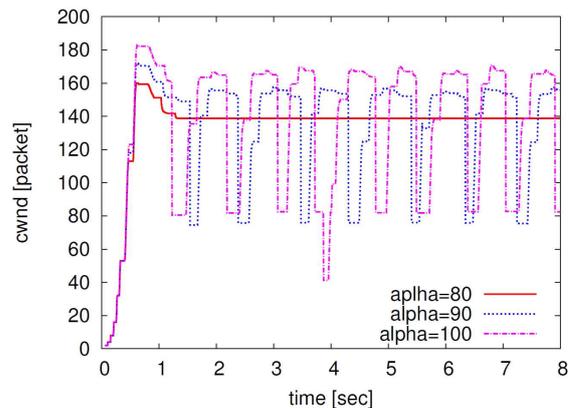


図 2 RED を介した FAST TCP 通信の性能

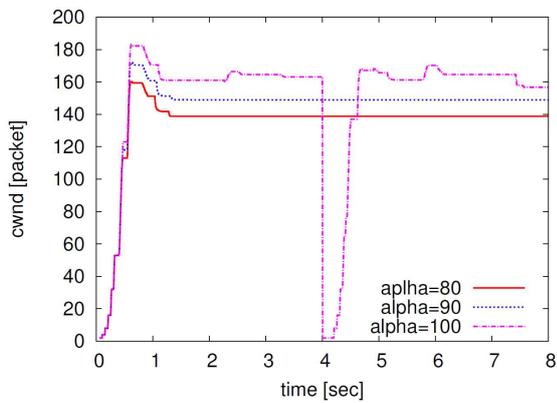


図3 修正後のFAST TCP 通信の性能

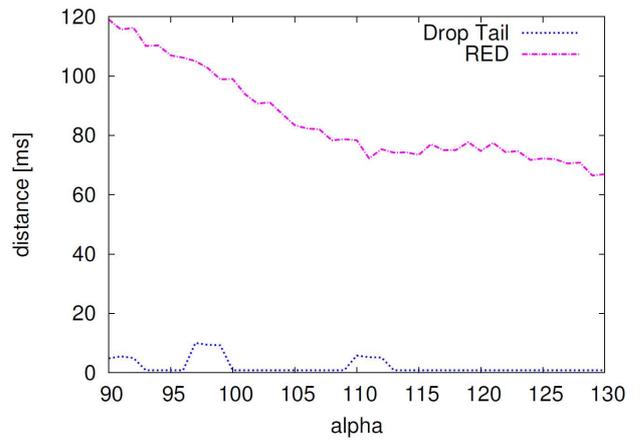


図6 に対するパケット損失間隔の変動

次に、通信経路中のREDを検知する方法を検討するために、特にFAST TCPの α の値を変化させたとき、Drop Tailのみの通信経路とREDを含む経路でどのような応答の違いができるかを調べた。代表例として $\alpha=110$ のときの結果を図4, 5に示す。図4, 5中のインパルス線は1つのパケット損失を表し、高さに意味は無い。また、図6に α の値を90から130まで変化させたときの連続するパケット損失の時間間隔 distance の変動の様子を示す。distance の値が小さいほどパケット損失のバースト性が高いと言える。結果から、REDを含む経路ではパケット損失のバースト性がDrop Tailのみの時より低いことが明らかになった。

4. 結論

まず、REDを介したとき、FAST TCP通信にウィンドウサイズの振動という問題が発生することを確認した。次に、少量のパケット損失を軽度の輻輳と見なしウィンドウサイズを半減させないことで、REDを介するとき通常のFAST TCPより安定した通信が可能になることを確認した。最後に、通信経路中のREDを検知するためにパケット損失のバースト性の情報が有益であることを確認した。

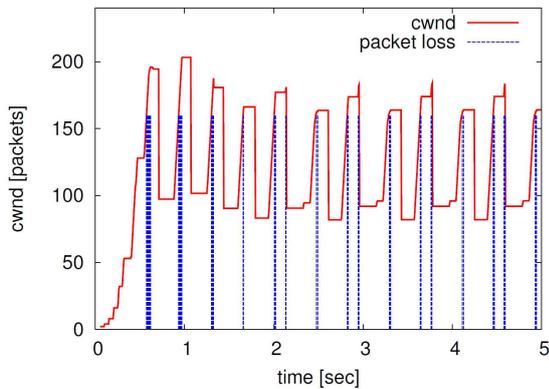


図4 Drop Tail を介したFAST TCP 通信 ($\alpha=110$)

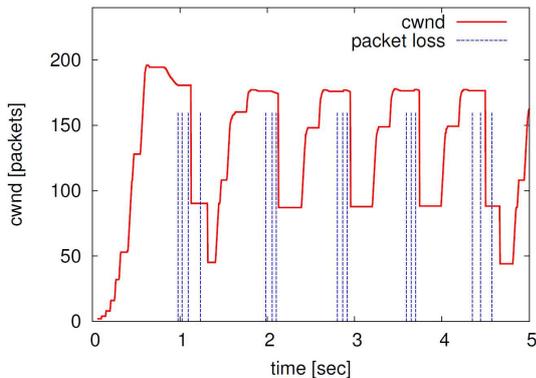


図5 RED を介したFAST TCP 通信 ($\alpha=110$)