

TCP 再送信タイマ管理の変更による低量 DoS 攻撃被害の緩和効果

細井 琢朗† 松浦 幹太†

† 東京大学生産技術研究所
153-8505 東京都目黒区駒場 4-6-1
{hosoi, kanta}@iis.u-tokyo.jp

あらまし インターネットにおける通信の輻輳制御として、TCP の再送信タイムアウトの機構は大きく役立っている。しかし既存研究で指摘されているように、この機構は低平均通信量での DoS 攻撃を可能にする。この攻撃の被害緩和策として、我々はこれまでに、TCP 再送信タイマの長さを連続するタイムアウト毎で丁度二倍にしないという、再送信タイマの管理方法の変更を提案した。本稿ではこの変更による被害緩和の効果を検証する。

Effectiveness of a Change in TCP Retransmission Timer Management for Low-rate DoS Attack Mitigation

HOSOI Takuro† Kanta Matsuura†

†Institute of Industrial Science, The University of Tokyo
4-6-1, Komaba, Meguro-ku, Tokyo 153-8505, Japan
{hosoi, kanta}@iis.u-tokyo.jp

Abstract The mechanism of TCP retransmission timeout is essential to the Internet congestion control. But existing research pointed out that this mechanism allows DoS attack with low-rate mean traffic. We proposed a change in TCP retransmission timeout management, in which length of TCP retransmission timer is increased not to precisely twice of the prior timer length in successive timeout waiting. We investigate the effectiveness of above change for DoS attack mitigation.

1 はじめに

インターネットはその通信規約において、通信量が回線容量を超えた場合（輻輳）にその通信パケットの一部を廃棄してネットワークの安定動作を確保する仕組みを備えている。この通信廃棄が TCP において行われた場合、廃棄に対応した通信データを再送信することが TCP の通信規約で決められている [3]。再送信は通信の輻輳が疑われる状況で行われるため、再送信パケットが輻輳を強めることのないように、一定の待

ち時間の後に行われる。この輻輳制御には二つの時間尺度がある。まず、正常に送られた通信への応答は RTT (round trip time, ~数十ミリ秒) 程度で送信側に返ってくる。輻輳などの理由により複数の応答が返ってこない場合、再送信タイムアウト (RTO : retransmission timeout, 推奨最小値 = 1sec [2]) だけ待ってからパケットを再送信する。

低量 DoS (Denial-of-Service) 攻撃 [1] は、RTT の時間尺度の間だけ続くバースト通信を一定周期で繰り返し送信することで、この二つ

の時間尺度の違いの分だけ低い平均攻撃通信量で通信妨害を行う DoS 攻撃である。この時間尺度の違いは輻輳制御で重要な役割を担っており、そのままにしておく必要がある。また RTT はネットワーク環境やその利用状況で決まるため、送信側で操作することは難しい。そこで再送信タイマへの工夫で低量 DoS 攻撃対策ができることが望ましい。文献 [1] では RTO の最小値 (minRTO) を一定の幅の中でランダムに選ぶことで、TCP の各通信が DoS 状態に入り難くする対策を試している。この方法では、対策が無い場合に較べて攻撃が最も強く働くパラメータ設定 (バースト通信の周期が RTO の最小値 minRTO と同じ) での平均回線容量は改善する。しかしこの対策では、バースト通信の周期を minRTO からランダム変化の幅の分だけずらしたところで対策前と同程度の攻撃被害が発生するという実験結果になった。そのためこの対策の実行に気付いている攻撃者にとっては最強攻撃パラメータ設定の位置が少しずれるだけとなり、有効な対策とはなっていない。

そこで我々は、TCP 再送信タイマの長さ (RTO) が連続するタイムアウト毎に二倍される [4] 点がこの DoS 攻撃の被害を深刻にしていることに着目した。再送信タイマの増加が整数倍で行われると、一度低量 DoS 攻撃に捕まった通信はそこから抜け出すことが難しく、平均回線容量が低いままに抑えられてしまう。この点を改良するべく、我々は文献 [5] で再送信タイマの長さを連続するタイムアウト毎に $(1 + u)$ 倍 ($0 < u < 1$) するという再送信タイマの管理方法の変更を提案した。この文献内ではその攻撃緩和効果について十分な検証ができていないため、本稿でその効果を調査する。調査は攻撃被害を簡単にモデル化した上で解析的に行い、効果の傾向も明らかにする。

2 TCP 再送信タイムアウトと低量 DoS 攻撃

この節では RFC6298 [4] で規定されている再送信タイマの管理方法と、その性質を利用する低量 DoS 攻撃 [1] について簡単に説明する。

2.1 TCP 再送信タイムアウト

RTT が計測済みの場合、パケットを $t = 0$ に送信する際、RFC6298 [4] によって RTO の最初の値が次の式で設定される。

$$\begin{aligned} \text{RTO} = \max\{ & \text{minRTO}, \\ & (\text{smoothed RTT}) \\ & + \max[(\text{clock granularity}), \\ & 4(\text{RTT variation})]\} \end{aligned} \quad (1)$$

ここで minRTO は RTO の最小値で、RFC6298 [4] により 1 秒にすることが推奨されている。多くの場合、最初の最大値比較の中の第二項は第一項に較べて小さい (RTT の時間尺度が RTO の時間尺度よりずっと小さい)。そこでこれ以降は RTO は最初に minRTO に設定されるものとして議論する。

$$\text{RTO}_1 = \text{minRTO} \quad (2)$$

送信されたパケットが無事に宛先に届き、その応答が RTT 程度で送信側に到着した場合は、そのまま通常の通信が続く。しかし輻輳などによりパケットが宛先に届かない、もしくは応答が到着しない場合、送信側は RTO だけ待ってから、送信に失敗したパケットの内の最初の一つを再送信する。その際 RTO の値を次の式で再設定する (exponential backoff phase)。

$$\text{RTO}_i = 2\text{RTO}_{i-1} \quad (3)$$

ここで添字の i は RTO の設定回数を表しており、当該パケットの送信に $i - 1$ 回連続して失敗し、 i 回目の再送信を行っていることを示す。

この新しい RTO の間にパケットが宛先に届き、その応答が送信側に到着した場合は、RTO の値はそのまま、先程送信に失敗したパケットの内の次の二個を再送信する。このパケット送信と応答が無事にできれば、RTO は (1) 式で再計算される (minRTO に再設定される)。一方、再送信した一個のパケットの送信とその応答が (3) 式の RTO の間にできなかった場合、先程と同様、RTO の値を (3) 式の通りに増やし、送信に失敗したパケット一個を再送信する。

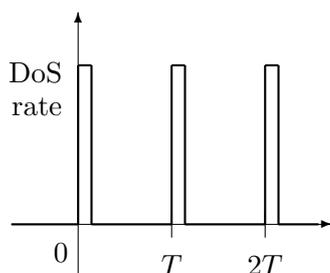


図 1: 矩形波状の DoS 攻撃通信.

以下これを繰り返す. ただし RTO の値には上限値 (60 秒以上) が付けられており [4], それを超える場合はこの上限値が使われる.

2.2 低量 DoS 攻撃

文献 [1] で提案された, 2.1 節の再送信タイムアウトの仕組みを利用した低量 DoS 攻撃の概略を説明する.

攻撃通信には図 1 のような, RTT の時間尺度程度続く短時間のバーストと通信無しが周期 T で繰り返される矩形波状のものを考える. 以下では簡単のため, 短時間のバーストは他の通信ができない程に通信帯域を消費する強度と長さを持つ理想的なもののみを考える.

この攻撃通信が始まると, まず最初のバースト通信により, (ほとんど) 全ての TCP 通信が失敗し, 再送信タイムアウトを待つ. 2.1 節で解説した通り, この際の RTO は全ての通信で一律に minRTO に設定されている. そのため (ほとんど) 全ての通信が同期して再送信タイムアウトを待つことになる.

再送信タイムアウトが DoS 攻撃のバースト通信に重ならなければ, それ以降はほぼ通常の通信が行われた後, 次のバースト通信により再度 (ほとんど) 全ての TCP 通信が失敗し, 再送信タイムアウトを待つ. これを攻撃が止むまで繰り返すことになる.

再送信タイムアウトが DoS 攻撃の次のバースト通信に重なる

$$\text{minRTO} \simeq T \quad (4)$$

と, 再送信も失敗し, (3) 式により 2 倍に増や

された RTO だけ待ってから再度再送信を行うことになる. しかしこの再度の再送信タイムアウトも (eq:minRTO: eq:T) 式の条件からバースト通信に重なる. 以降, RTO は minRTO の整数倍の値を取り続けるため, 再送信タイムアウトは全てバースト通信に重なり, 攻撃の間は通信が (ほとんど) できない.

バーストにより全ての TCP 通信が送信失敗になる場合の正規化平均回線容量は, 文献 [1] により次の式で表せる.

$$\rho = \frac{T - \text{minRTO}}{T} \quad (5)$$

(for $T \geq \text{minRTO}$)

上式は常に 1 より小さく, その 1 との差 (通信回線容量の減少) がサービス妨害の被害に当たる. 攻撃のバースト通信の周期 T と再送信タイムアウトの最小値 minRTO が等しい ((4) 式) と, 回線容量は 0 になり, 攻撃の被害が最も大きい.

3 提案方式の再送信タイム管理

低量 DoS 攻撃が行われると, 2.2 節で解説した通り, 特に DoS 攻撃のバースト通信の周期が再送信タイムアウトに同期している場合 ($T \simeq \text{minRTO}$: (4) 式) に大きなサービス妨害が発生する. これは連続して再送信を続ける場合, (1) 式と (3) 式により RTO が minRTO の整数倍に増やされるため, 毎タイムアウトが必ず DoS 攻撃のバースト通信と重なることが原因である. 文献 [1] では minRTO を一定の幅の範囲でランダムに選ぶことでこの DoS 攻撃への対策を試みたが, TCP 通信の輻輳制御を働かせるためにはこの幅を大きく取れず, 大きな被害低減効果は得られなかった.

そこで我々は, (3) 式による RTO の増加方法を以下のように変更することを文献 [5] で提案した.

$$\text{RTO}_i = (1 + u)\text{RTO}_{i-1} \quad (6)$$

($0 < u < 1$)

すると連続した再送信での RTO は

$$\text{RTO}_i = (1 + u)^{(i-1)} \text{minRTO} \quad (7)$$

となり, u に有理数を選ぶことで minRTO の整数倍にはならない系列が得られる. この系列を使って RTO を増やせば, 低量 DoS 攻撃の一周期内のバーストの回数を増やし続けられない限り, 再送信タイムアウトがバーストと重ならない機会が生まれ, サービス妨害の被害の低減が期待できる.

2.2 節の低量 DoS 攻撃と同じ攻撃を考える. 再送信タイムアウトが攻撃のバースト通信と重ならない場合は, 各通信は一回目のタイムアウトの後に通常の通信に戻り, 連続してタイムアウト待ちになることはない. RTO の最初の値の設定方法は提案方式と元の再送信タイマ管理方法では同じなため, この場合の正規化平均回線容量は提案方式も元の再送信タイマ管理方法と同じ (5) 式で表せる.

$$\rho' = \frac{T - \text{minRTO}}{T} \quad (\text{for } T > \text{minRTO}) \quad (8)$$

攻撃のバースト通信の周期が再送信タイマの最小値に一致している場合, 提案方式では以下のような遷移をする.

- ($t = 0$)
バーストによりパケットの送信に失敗する.
(再送信タイマ $\text{RTO}_1 = \text{minRTO}$)
- ($t = \text{minRTO}$)
再送信タイムアウト. バーストによりパケットの再送信に失敗する.
(再送信タイマ $\text{RTO}_2 = (1 + u)\text{minRTO}$)
- ($t = 2\text{minRTO}$)
再送信タイムアウトを待つ.
- ($t = (2 + u)\text{minRTO}$)
再送信タイムアウト. バーストが重ならないため, パケットの再送信に成功する.
- (以下繰り返し.)

この遷移による回線容量の推移をグラフにすると図 2 の実線のようにになる. この図からわかる通り, 元の再送信タイマ管理方法 (図中の RFC6298 の破線) では回線容量は 0 のままであるが, 提案方式では期間 3minRTO 毎に一回, 通常通信に戻る期間ができる. この正規化平均

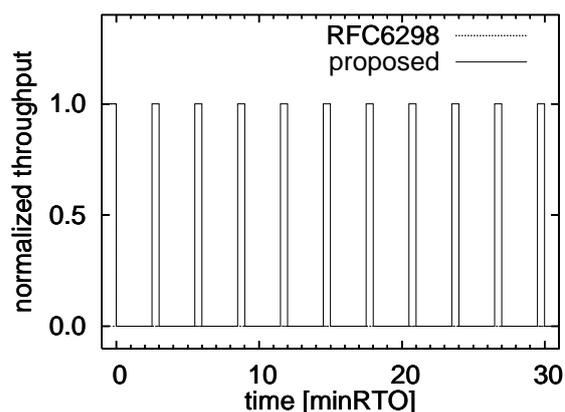


図 2: バーストにより全ての通信が送信失敗になる場合の正規化回線容量の推移.

回線容量は簡単に計算でき, 以下のようにになる.

$$\rho' = \frac{1 - u}{3} \quad (\text{for } T = \text{minRTO}) \quad (9)$$

この分だけ, 提案方式は元的方式に較べて低量 DoS 攻撃の被害緩和効果がある.

4 被害のモデル化

低量 DoS 攻撃のバースト通信は, 幾つかの理由から一部の TCP 通信を止められないことがある. まず, 低量 DoS 攻撃の攻撃者が「低量」を志向すると, 攻撃に用いるバースト通信の強度や期間をなるべく抑えることになる. そのため一部の通信は運良くバースト下でもパケットの送信に成功したり, 大きい RTT のためにバーストを避けたりする可能性が出てくる. また, 再送信タイマ RTO の計時の粒度が大きいと, タイムアウトの認識時刻がパケットの送信失敗時刻から再送信タイマの大きさだけ正確に進んだ分からずれる. そのためタイムアウトが本来重なるはずのバーストから外れることがある. さらに文献 [1] で試みられていた対策 (RTO の最小値 minRTO の設定値にランダムな揺らぎを加える) が使われた場合, 一部の通信はバーストを避けるようになる.

そこでここでは簡単のため, 低量 DoS 攻撃の様子とその被害を次のようにモデル化する.

1. バースト通信が行われた際, バーストが無ければその時刻に通信を行っていたはずの

通信（再送信タイムアウトする通信を含む）のうち、一定の回避率 r の通信だけがパケットの送信に成功し、通信を続行する。

2. 残りの $(1 - r)$ の通信はパケットの送信に失敗し、再送信タイムアウトを待つ。
3. バーストがない時にはすべての通信がパケットの送信に常に成功する。

これは RTT 程度の時間で起こる現象を一纏めにして変数 r に代表させたものになっている。以下ではこのモデルを使って低量 DoS 攻撃の被害の程度を評価する。

5 攻撃被害の評価

この節では元の再送信タイム管理方法と提案方式について、4 節のモデルを用いて低量 DoS 攻撃の被害の程度を評価する。

5.1 攻撃周期が minRTO と異なる場合

低量 DoS 攻撃のバースト通信の周期 T が再送信タイムの最小値 minRTO より大きい場合は、(5) 式、(8) 式と同様の議論から、正規化平均回線容量は元の再送信タイム管理方法と提案方式で同じになり、以下のように表せる。

$$\rho = \rho' = \frac{T - (1 - r)\text{minRTO}}{T} \quad (10)$$

(for $T > \text{minRTO}$)

正規化平均回線容量は (10) 式の分子の第 2 項にある r の分だけ増える (図 3)。

5.2 攻撃周期が minRTO と等しい場合

低量 DoS 攻撃のバースト通信の周期 T が再送信タイムの最小値 minRTO と同じ場合は、連続してパケットの送信に失敗する通信が出る。そのため元の再送信タイム管理方法と提案方式の間では回線容量の減少に違いが現れる。

5.2.1 元の再送信タイム管理方法の場合

元の再送信タイム管理方法では、各再送信タイムアウトがバースト通信と重なる。そのため

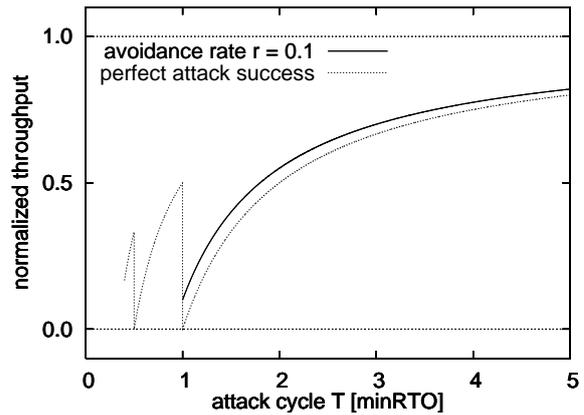


図 3: 元の再送信タイム管理方法での、バーストにより全ての通信が送信失敗になる場合と回避率 r で一部が通信に成功する場合の正規化回線容量。

連続してパケットの送信に失敗する通信が出る。攻撃前から続く通信のうち、どれだけが通信を続けられるかを追うと表 1 のようになる。 $t = 0$ のバースト通信で一部 (r) はパケットの送信に成功する。残りの $(1 - r)$ はパケットの送信に失敗し、再送信タイム $\text{RTO}_1 = \text{minRTO}$ を設定されて一回目のタイムアウト待ちに入る。

$t = \text{minRTO}$ のバースト通信で一部 (r^2) はパケットの送信に成功し続け、再送信タイムアウトした分の一部 ($r(1 - r)$) も再送信に成功する。 $(r(1 - r))$ はパケットの送信に失敗して一回目のタイムアウト待ちに入り、 $((1 - r)^2)$ はパケットの再送信に失敗して再送信タイム $\text{RTO}_2 = 2\text{minRTO}$ を設定されて二回連続のタイムアウト待ちに入る。この二回連続してタイムアウト待ちに入った分は再送信タイムの値が 2 倍に増やされているため、次のバースト通信時は何もせず、二回連続のタイムアウト待ち 2 に移り、その次のバースト通信時にタイムアウトする。この遷移から通信続行の割合の例をグラフに表すと、図 4 の破線のようになる。

時刻 $t = N \cdot \text{minRTO}$ で通信を続行する割合は、下限が r^N 、上限が r になることが表 1 の遷移を追うことで簡単に導ける。通信を続行する回避率 r が 1 よりずっと小さい場合は、時刻 $t = 0$ から連続してタイムアウト待ちをし続けている分だけが order 1 の大きさを担っている。

表 1: 元の再送信タイマ管理方法で攻撃周期が minRTO と等しい場合の通信続行の割合とタイムアウト待ちの割合の遷移.

$t [\text{minRTO}]$	通信成功	タイムアウト待ち			
		1 回目	2 回連続 (待ち 1)	2 回連続 (待ち 2)	3 回連続 (待ち 1) ...
(< 0)	1	0			
0	r	$(1-r)$			
1	$r^2 + r(1-r)$	$r(1-r)$	$(1-r)^2$		
2	$r^3 + 2r^2(1-r)$	$r^2(1-r)$ $+ r(1-r)^2$	$r(1-r)^2$	$(1-r)^2$	
3	$r^4 + 3r^3(1-r)$ $+ r^2(1-r)^2$ $+ r(1-r)^2$	$r^3(1-r)$ $+ 2r^2(1-r)^2$	$r^2(1-r)^2$ $+ r(1-r)^3$	$r(1-r)^2$	$(1-r)^3$
...					

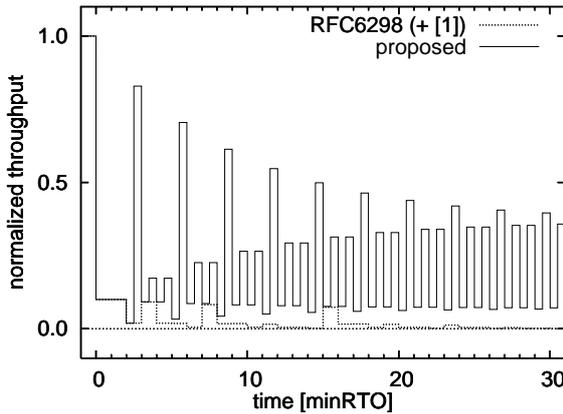


図 4: 元の再送信タイマ管理方法と提案方式での、バースト時も回避率 r で一部が通信に成功する場合の正規化回線容量 ($r = 0.1$) .

時刻 $t = 0$ から k_N 回連続してタイムアウト待ちをした分は時刻 $t = \sum_{k=0}^{k_N-1} 2^k \text{minRTO} = (2^{k_N} - 1) \text{minRTO}$ でタイムアウトを迎え、その r 倍 (: order r) が通信を回復させる. これ以降 2^{k_N}minRTO の期間は、表 1 の遷移からわかるように、大きくても order r^2 だけしか通信を行う分がない. そのため低量 DoS 攻撃が長く行われた場合、正規化平均回線容量は order r^2 かそれより小さい値になる.

$$\rho \lesssim O(r^2) \quad (\text{for } T = \text{minRTO}) \quad (11)$$

図 4 の一点鎖線はこの一例として $t = 60 \text{minRTO}$ での計算値を図にしたものである.

5.2.2 提案方式の場合

提案方式では、一回目の再送信タイムアウトはバースト通信と重なるが、二回連続した再送信タイムアウト待ちはバースト通信と重ならない. そのためパケットの送信に失敗するのは連続 2 回までとなり、再送信タイムアウトの遷移は期間 3minRTO で一周する.

時刻 $t = N \cdot \text{minRTO}$ のバースト通信の後、通信が続行できている割合を A_N , 一回目のタイムアウト待ちに入った割合を B_N , 二回連続のタイムアウト待ちに入った割合を C_N , 二回連続のタイムアウト待ちに入り続けている割合を D_N とすると、その後の遷移は付録 A のようになる. またこの遷移から通信続行の割合の例をグラフに表すと、図 4 の実線のようになる.

このタイムアウト待ちの遷移の様子から、初めにそれぞれの割合が 0 から 1 の間の値を取るとき ($0 \leq (A_N, B_N, C_N, D_N) \leq 1$) , 以下のことが証明できた¹.

- $0 \leq (A_k, B_k, C_k, D_k) \leq 1$,
 $0 \leq (A_{k+u}, B_{k+u}, C_{k+u}, D_{k+u}) \leq 1$
($k \geq N$) .
- 系列 $\{A_{N+1}, A_{N+4}, A_{N+7}, \dots\}$ は単調に増加・減少する.
- 系列 $\{A_{N+1+u}, A_{N+4+u}, A_{N+7+u}, \dots\}$ は単調に増加・減少する.
- 増減は 2. と 3. では逆になる.
- $(A_{N+1} + A_{N+2} + A_{N+3})$ は単調に増加・減

¹紙面の都合により証明は省略する.

少する.

6. $(A_{N+1+u} + A_{N+2+u} + A_{N+3+u})$ は単調に増加・減少する.

7. 増減は 5. と 6. では逆になる.

再送信タイムアウトの遷移が一周する期間 3minRTO で回線容量を平均したものは

$$\rho'_{N+1,3} = \frac{1}{3} [uA_{N+1} + (1-u)A_{N+1+u} + uA_{N+2} + (1-u)A_{N+2+u} + uA_{N+3} + (1-u)A_{N+3+u}] \quad (12)$$

で表され, これの増減は上記の証明から単調であること, またその増減が

$$1 - u > r \Rightarrow 5. \text{ と逆}$$

$$1 - u < r \Rightarrow 5. \text{ と同じ}$$

になることが証明できた. これらの証明と付録 A から, 低量 DoS 攻撃が長く続いた場合, タイムアウト待ちの遷移の各割合がそれぞれ

$$\begin{aligned} A_N &\rightarrow \frac{1 - (1-r)^2}{1 + (1-r) + (1-r)^2} \\ B_N &\rightarrow \frac{(1-r)}{1 + (1-r) + (1-r)^2} \\ C_N &\rightarrow \frac{(1-r)^2}{1 + (1-r) + (1-r)^2} \\ D_N &\rightarrow \frac{(1-r)^2}{1 + (1-r) + (1-r)^2} \end{aligned} \quad (13)$$

に収束することが分かった.

図4は破線 ("at outage") がバースト通信が行われた後の正規化回線量の収束値 ($A_N(N \rightarrow \infty)$), 点線 ("at timeout2") が二回連続のタイムアウト待ちが終わった時刻での正規化回線量の収束値 ($A_N + D_N(N \rightarrow \infty)$) を示している. 図4と(13)式から分かる通り, 低量 DoS 攻撃のバーストにより全ての通信が再送信タイムアウト待ちに入ってしまう ($r = 0$), その攻撃が長く続いた場合でも, 提案方式では正規化平均回線容量が 0 にならない. この極限では, DoS 攻撃のバースト通信から $u \cdot \text{minRTO}$ までの間は回線容量は 0 になり, その後, 次のバースト通信がなされるまでの期間で元の容量の $\frac{1}{3}$ だけ回線容量が残っている. これらを平均することで, 提案方式の正規化平均回線容量の最悪値は

$$\rho' \rightarrow \frac{1}{3}(1-u) \quad (14)$$

(for $T = \text{minRTO}$)

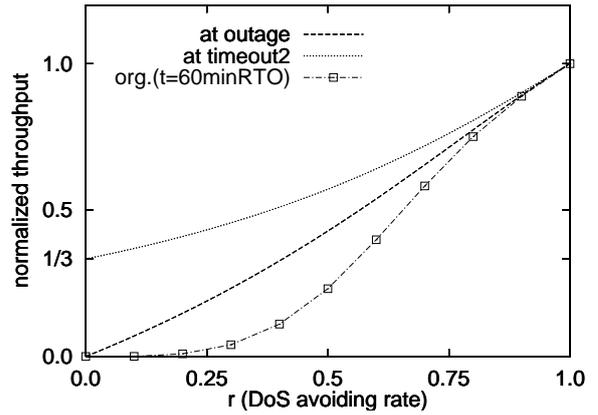


図 5: 提案方式での, バースト時も回避率 r で一部が通信に成功する場合の正規化回線容量の収束値.

に収束することが分かる. これは (9) 式と同じ値である. これに対応する図2の実線が収束しないのは, 初期値で通信続行の割合 $A = 1$ としており, 振動解となっているためである.

6 まとめ

本稿では, 文献 [1] で提案された低量 DoS 攻撃の被害を緩和するような再送信タイムの管理方法の変更 [5] について, その被害緩和効果を調査した. 調査は簡単なモデルの下で解析的に行い, 攻撃が行われている際の通信可能分を正規化平均回線容量として定量的に導出した. その結果, 攻撃のバースト通信の周期が再送信タイムの最小値よりも大きい場合は, 提案方式と元の再送信タイム管理方法で被害は全く同じになること, また攻撃のバースト通信の周期と再送信タイムの最小値が一致する場合は, 提案方式が元の再送信タイム管理方法に較べて被害を有意に緩和することが分かった. 特に後者の場合で元の再送信タイム管理方法での攻撃被害が大きい状況 ($r \rightarrow 0$) でも 0 でない平均回線容量を確保できる点は大きな改善点と言える.

今後は実験により, 攻撃緩和性能の実証と既存のネットワーク環境への適合性の検証を行う予定である.

参考文献

- [1] A. Kuzmanovic and E.W. Knightly, “Low-rate TCP-targeted Denial of Service Attacks: the Shrew versus the Mice and Elephants”, In Proceedings of ACM SIGCOMM’03, pp.75-86 (August 2003)
- [2] M. Allman and V. Paxson, “On Estimating End-to-end Network Path Properties”, In Proceedings of ACM SIGCOMM’99, pp.263-274 (September 1999).
- [3] J. Postel (Ed.), “Transmission Control Protocol”, Internet RFC 793, IETF (September 1981).
- [4] V. Paxson, M. Allman, J. Chu, M. Sargent, “Computing TCP’s Retransmission Timer”, Internet RFC 6298, IETF (June 2011).
- [5] 細井 琢朗, 松浦 幹太, “低量 DoS 攻撃を緩和する TCP 再送信タイム管理の一検討”, 情報処理学会コンピュータセキュリティ研究会 (研究報告コンピュータセキュリティ (CSEC)) , vol.62, no.51, pp.1-5 (July 2013)

A 提案方式において回避率 r で一部が通信に成功する場合の遷移

回避率 r で一部が通信に成功する場合, 提案方式の再送信タイムアウトの遷移は遷移は以下のようになる.

$$(t = N \cdot \text{minRTO})$$

$$A_N = A \quad (15)$$

$$B_N = B \quad (16)$$

$$C_N = C \quad (17)$$

$$D_N = D \quad (18)$$

$$(t = (N + u)\text{minRTO})$$

$$A_{N+u} = A + D \quad (19)$$

$$B_{N+u} = B \quad (20)$$

$$C_{N+u} = C \quad (21)$$

$$D_{N+u} = 0 \quad (22)$$

$$(t = (N + 1)\text{minRTO})$$

$$A_{N+1} = r(A + D + B) \quad (23)$$

$$B_{N+1} = (1 - r)(A + D) \quad (24)$$

$$C_{N+1} = (1 - r)B \quad (25)$$

$$D_{N+1} = C \quad (26)$$

$$(t = (N + 1 + u)\text{minRTO})$$

$$A_{N+1+u} = r(A + D + B) + C \quad (27)$$

$$B_{N+1+u} = (1 - r)(A + D) \quad (28)$$

$$C_{N+1+u} = (1 - r)B \quad (29)$$

$$D_{N+1+u} = 0 \quad (30)$$

$$(t = (N + 2)\text{minRTO})$$

$$A_{N+2} = r(A + D + C) + r^2B \quad (31)$$

$$B_{N+2} = r(1 - r)(A + D + B) + (1 - r)C \quad (32)$$

$$C_{N+2} = (1 - r)^2(A + D) \quad (33)$$

$$D_{N+2} = (1 - r)B \quad (34)$$

$$(t = (N + 2 + u)\text{minRTO})$$

$$A_{N+2+u} = r(A + D + C) + (1 - r + r^2)B \quad (35)$$

$$B_{N+2+u} = r(1 - r)(A + D + B) + (1 - r)C \quad (36)$$

$$C_{N+2+u} = (1 - r)^2(A + D) \quad (37)$$

$$D_{N+2+u} = 0 \quad (38)$$

$$(t = (N + 3)\text{minRTO})$$

$$A_{N+3} = r^2(2 - r)(A + D) + r(B + C) \quad (39)$$

$$B_{N+3} = r(1 - r)(A + D + C) + (1 - r)(1 - r + r^2)B \quad (40)$$

$$C_{N+3} = r(1 - r)^2(A + D + B) + (1 - r)^2C \quad (41)$$

$$D_{N+3} = (1 - r)^2(A + D) \quad (42)$$

$$(t = (N + 3 + u)\text{minRTO})$$

$$A_{N+3+u} = (1 - 2r + 3r^2 - r^3)(A + D) + r(B + C) \quad (43)$$

$$B_{N+3+u} = r(1 - r)(A + D + C) + (1 - r)(1 - r + r^2)B \quad (44)$$

$$C_{N+3+u} = r(1 - r)^2(A + D + B) + (1 - r)^2C \quad (45)$$

$$D_{N+3+u} = 0 \quad (46)$$

$$(t = (N + 4)\text{minRTO})$$

$$A_{N+4} = r(1 - r + 2r^2 - r^3)(A + D + B) + r(2 - r)C \quad (47)$$

$$B_{N+4} = (1 - r)(1 - 2r + 3r^2 - r^3)(A + D) + r(1 - r)(B + C) \quad (48)$$

$$C_{N+4} = r(1 - r)^2(A + D + C) + (1 - r)^2(1 - r + r^2)B \quad (49)$$

$$D_{N+4} = r(1 - r)^2(A + D + B) + (1 - r)^2C \quad (50)$$

(以下繰り返し.)