CL-002

iOS における TCP 輻輳制御の性能評価：アルゴリズムとパラメータの推定

Performance Evaluation of TCP Congestion Control on iOS : Estimation of Algorithm and Parameters

穂積　勇樹 † 内海　哲史 ‡ サリムザビル §
Yuki Hozumi Satoshi Utsumi Salahuddin Muhammad Salim Zabir

1. まえがき

TCP 輻輳制御には多くのバージョンが存在している．それぞれのバージョンは，その利用環境に合うようなアルゴリズムとなっている．例えば，TCP Hybla は物理的距離が長く，リンクエラー率が高い衛星ネットワークで高性能に機能するアルゴリズムとなっている [1]．

近年，iPhone が流行し，国内でのスマートフォンシェア率で iPhone の割合が過半数を超えた [2]．iPhone の OS である iOS の TCP 輻輳制御は一般にはどのようなアルゴリズムを利用しているのか知られておらず，文献 [3] でも調査されている．しかし，文献 [3] で調査されている iOS バージョンはバージョン 6.0 以降であり，古い可能性がある．

本稿では，最近の iOS（バージョン 9.3.2）の TCP 輻輳制御について性能評価し，どのような TCP 輻輳制御アルゴリズムを採用しているのかを調査する．

2 章では，iPhone5 の TCP 輻輳制御における輻輳ウィンドウサイズの変化に関する関連研究 [3] についてまとめる．3 章では，Linux に実装されている TCP 輻輳制御について述べる．4 章では，iOS の TCP 輻輳制御の性能評価について述べる．5 章では，iOS の TCP 輻輳制御の輻輳ウィンドウサイズの変化について述べる．6 章では，iOS の TCP 輻輳制御で用いられているパラメータの導出について述べる．最後に 7 章で結論を述べる．

2. 関連研究：iPhone5 の TCP 輻輳制御アルゴリズム

Kato ら [3] は，輻輳ウィンドウサイズとその差分の関連性より TCP 輻輳制御アルゴリズムを推測する方法について述べている．この論文では，$cwnd$（輻輳ウィンドウ）の増加フェーズにおいて，個々の RTT（Round Trip Time）における $cwnd$ の値の変化に焦点を当てている．RTT 毎の $cwnd$ 値のシーケンスを $cwnd_i$ として推定する．また，連続する $cwnd$ 値の差分のシーケンスを $\Delta cwnd_i$ として式 (1) に示す．

$$\Delta cwnd_i = cwnd_{i+1} - cwnd_i \tag{1}$$

AIMD，TCP Vegas，TCP Veno，HS TCP，CUBIC TCP，TCP Illinois について上記の方法で各 TCP 輻輳制御を調査している．縦軸に $\Delta cwnd$（$segments$），横軸に $cwnd$（$segments$）とする図を作成し，$cwnd_i$ と $\Delta cwnd_i$ の関連性を調査している．iPhone5 がどの TCP 輻輳制御アルゴリズムを使用しているのかを各 TCP 輻輳制御の $cwnd_i$ と $\Delta cwnd_i$ の関連性から推定している．実験より，iPhone5 の TCP 輻輳制御は AIMD と似た結果になった．これより，文献 [3] では，iPhone5 の TCP 輻輳制御は AIMD を使用していると結論付けている．

3.Linux に実装されている TCP 輻輳制御：TCP CUBIC

TCP CUBIC は，低速ネットワークを想定して設計された従来型の TCP Reno にかわり，近年の高速ネットワークに対応した輻輳制御である．TCP BIC と呼ばれる TCP 輻輳制御の改良型であり，性能を改善している．TCP CUBIC は現在，Linux カーネルにおけるデフォルトの輻輳制御となっている．

TCP CUBIC の輻輳ウィンドウサイズ W_{CUBIC} の変化は式 (2) によって示される．

$$W_{CUBIC} = c(t-K)^3 + W_{max} \tag{2}$$

ただし，

$$K = \sqrt[3]{b * (W_{max} - W')} \tag{3}$$

TCP CUBIC の時刻 t におけるウィンドウサイズを W_{CUBIC}，縮小直前のウィンドウサイズを W_{max}，ウィンドウサイズの縮小後，次に W_{max} に達するまでの時間を K，縮小直後のウィンドウサイズを W'，ウィンドウの変化に関わる係数を c，b とする [4, 5]．TCP CUBIC の輻輳ウィンドウの変化における三次関数部分を図 1 に示す．

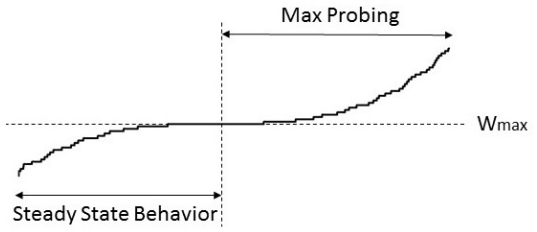

図 1: TCP CUBIC の輻輳ウィンドウの変化（三次関数部分）

TCP CUBIC の輻輳ウィンドウサイズの変化の様子を図 2 に示す．

†福島大学大学院共生システム理工学研究科，Fukushima University，Graduate School of Symbiotic Systems Science and Technology
‡福島大学理工学群共生システム理工学類，Fukushima University，Cluster of Science and Technology，Faculty of Symbiotic Systems Science
§国立高等専門学校機構鶴岡工業高等専門学校創造工学科情報コース，National Institute of Technology ，Tsuruoka College，Department of Creative Engineering，Information Course

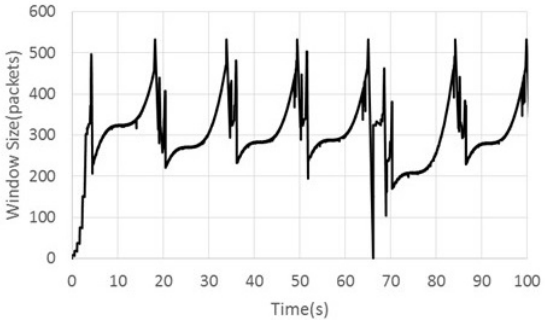

図 2: TCP CUBIC の輻輳ウィンドウの変化

Linux では上記に挙げた TCP CUBIC 以外に TCP NewReno，TCP Hybla，TCP BIC，HS-TCP，H-TCP，Scalable TCP，TCP Vegas，TCP Westwood，TCP Veno，TCP-LP，YeAH-TCP，TCP Illinois などが実装されている．

4. 性能評価：エラー率に対するスループットの測定

ここでは，iPhone5s（iOS バージョン 9.3.2）と Linux マシンを使用し，Dummynet[6] を利用したエミュレーション実験を行い，iOS における TCP 輻輳制御のスループットを測定する．次に Linux マシン 2 台を使用し，Dummynet を利用したエミュレーション実験を行い，Linux（CUBIC）のスループットを測定する．RTT=25ms，RTT=100ms，RTT=550ms でそれぞれ実験を行う．2 つの実験ではリンクエラー率を変更し，それぞれ 10 回測定し，スループットの平均値を取る．RTT 毎にスループットの比較を行う．本実験で用いるパラメータを表 1 に示す．本実験で用いるトポロジーを図 3 に示す．

表 1: スループットの測定で用いるパラメータ

RTT	25 ms	100 ms	550 ms
Link Capacity	64 Mbps	44 Mbps	8 Mbps
Packet Size	1000 Bytes	1000 Bytes	1000 Bytes
Max Window Size	200 packets	550 packets	550 packets
Buffer Size	1000 packets	1000 packets	1000 packets
Emulation Time	100 sec	100 sec	100 sec

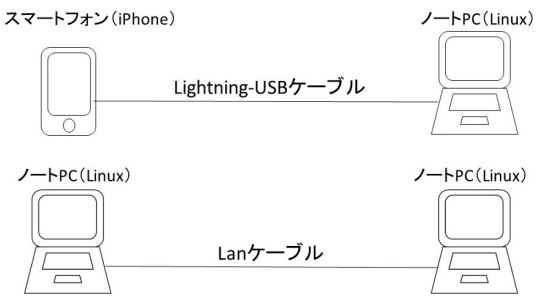

図 3: スループットの測定で用いるトポロジー

RTT=25ms での iOS における TCP 輻輳制御と Linux（CUBIC）の性能の比較を図 4，RTT=100ms での比較を図 5，RTT=550ms での比較を図 6 に示す．

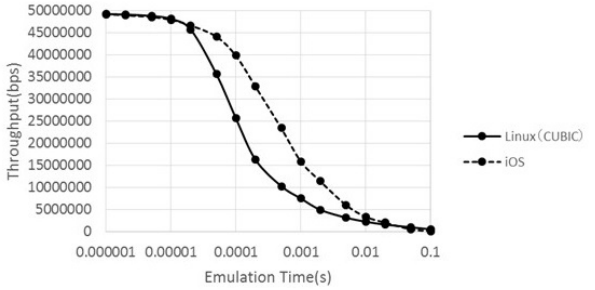

図 4: RTT=25ms の時の iOS と Linux（CUBIC）の比較

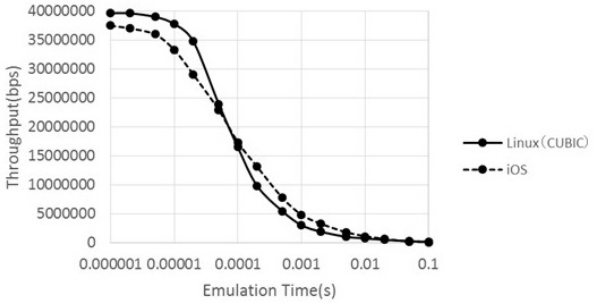

図 5: RTT=100ms の時の iOS と Linux（CUBIC）の比較

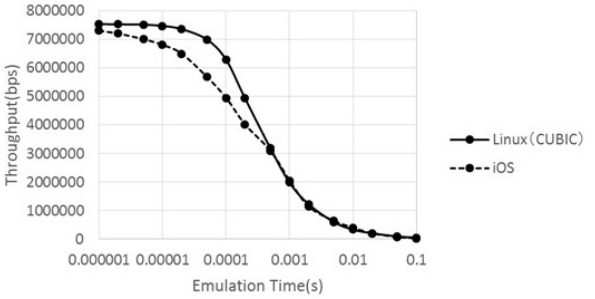

図 6: RTT=550ms の時の iOS と Linux（CUBIC）の比較

性能評価の結果，RTT=25ms の時では，iOS における TCP 輻輳制御が Linux（CUBIC）のスループットを上回っている．RTT=100ms の時では，リンクエラー率 0.0001 を境に Linux（CUBIC）と iOS における TCP 輻輳制御のスループットの大小関係が変化した．RTT=550ms の時では，Linux（CUBIC）が iOS における TCP 輻輳制御のスループットを上回った．RTT とリンクエラー率の変化に応じて，Linux（CUBIC）と iOS における TCP 輻輳制御の性能の大小が異なる結果となった．

5.iOS における TCP 輻輳制御のウィンドウサイズの変化の測定

iPhone5s（iOS バージョン 9.3.2）と Linux マシンを使用し，Dummynet を利用したエミュレーション実験を行い，iOS における TCP 輻輳制御についてウィンドウサイズを測定する．本実験は RTT を変更し，ウィンドウサイズを測定する．本実験でのパラメータを表 2 に示す．本実験のトポロジーを図 7 に示す．

表 2: ウィンドウサイズの測定で用いるパラメータ

Error Rate	0
Packet Size	1000 Bytes
Max Window Size	1000 packets
Buffer Size	1000 packets
Emulation Time	100 sec

図 7: ウィンドウサイズの測定で用いるトポロジー

本実験により得られた iOS における TCP 輻輳制御のウィンドウサイズの変化の様子を図 8 に示す．図 8 は RTT=550ms で測定した時のグラフである．縦軸をウィンドウサイズ（packets），横軸を測定時間（sec）で表す．

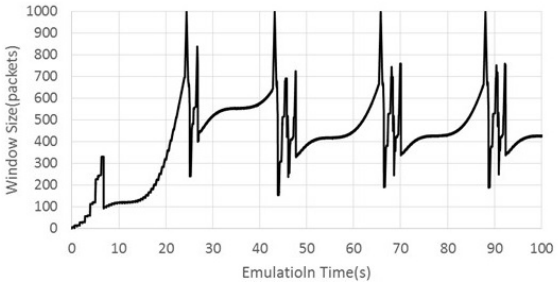

図 8: iOS における TCP 輻輳制御のウィンドウ変化

図 8 の結果より，iOS における TCP 輻輳制御は Linux における TCP CUBIC と同様に三次関数的にウィンドウサイズを増加させ，前回パケット損失が発生した直前でウィンドウサイズの増加を緩めている．

6.iOS における TCP 輻輳制御のパラメータの導出

図 8 の iOS における TCP 輻輳制御のウィンドウの変化の形と，図 2 の TCP CUBIC のウィンドウの変化の形が類似している．図 9 に，図 8 の iOS における TCP 輻輳制御のウィンドウの変化の三次関数部分を示す．

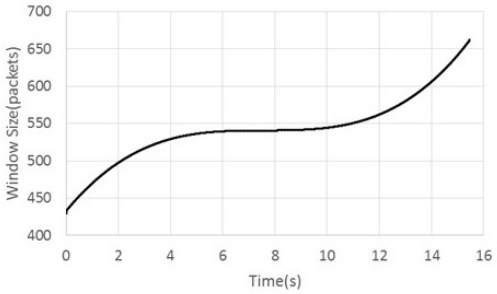

図 9: iOS における TCP 輻輳制御のウィンドウ変化の三次関数部分

iOS における TCP 輻輳制御で Linux と同様の TCP CUBIC が用いられていると仮定する．図 8 の iOS における TCP 輻輳制御の三次関数部分のウィンドウの変化に対し，Linux（CUBIC）のウィンドウの変化の式を当てはめたものを比較する．Linux の TCP CUBIC のウィンドウの変化は式 (2)，式 (3) によって示される．iOS における TCP 輻輳制御と Linux（CUBIC）の三次関数部分から読み取った時刻 t における W_{CUBIC}，K，W_{max}，$W^{'}$ の値を表 3，表 4 に示す．

ウィンドウの変化の増加係数であるパラメータ c を求めるため，式 (2) の変形を式 (4) に示す．

$$c = \frac{W_{CUBIC} - W_{max}}{(t-K)^3} \tag{4}$$

W_{CUBIC} は測定時刻におけるウィンドウサイズ，W_{max} はウィンドウサイズの傾きが 0 になるときのウィンドウサイズの値，t は測定時刻，K はウィンドウサイズを縮小してから，ウィンドウサイズの傾きが 0 になるまでの時間である．表 3，表 4 のそれぞれの値を式 (4) に代入し，c を求める．c を求めた後，式 (2) に c，W_{max}，t，K の値を代入し，ウィンドウサイズを計算する．図 9 の測定値と計算値の比較を図 10 に示す．

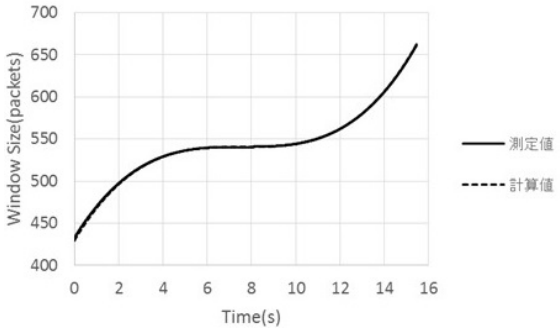

図 10: 測定値と計算値の比較

図 10 より，測定値と計算値がほぼ一致していることから，iOS における TCP 輻輳制御が TCP CUBIC を利用していると推定できる．図 2 の Linux（CUBIC）

表 3: Linux（CUBIC）の t，W_{CUBIC}，K，W_{max}，$W^{'}$ の値

	1サイクル目					2サイクル目					3サイクル目				
	t	W_{CUBIC}	K	W_{max}	$W^{'}$	t	W_{CUBIC}	K	W_{max}	$W^{'}$	t	W_{CUBIC}	K	W_{max}	$W^{'}$
25ms	12.70	328.01	4.88	148.84	104.28	11.48	327.06	4.44	195.29	161.16	10.52	285.35	5.76	243.64	169.69
50ms	14.36	634.21	5.70	378.73	312.84	14.36	634.21	5.72	377.78	311.89	14.36	634.21	5.70	377.78	311.89
100ms	14.07	599.14	5.60	365.45	301.46	14.12	600.08	5.58	357.87	294.83	14.14	601.03	5.58	358.82	294.83
150ms	14.49	582.07	4.88	246.95	202.87	13.97	578.28	5.46	345.55	285.35	14.47	582.07	4.88	246.01	201.92
200ms	13.99	589.66	5.47	356.92	294.83	14.10	589.66	5.52	353.13	291.04	14.06	589.66	5.47	353.13	290.09
250ms	14.17	603.88	5.54	365.45	300.52	14.20	605.77	5.50	361.66	297.67	14.15	604.82	5.49	362.61	299.57
300ms	14.67	599.14	4.95	255.49	209.51	14.79	603.88	4.39	177.75	146.94	14.64	599.14	4.96	258.33	211.40
350ms	14.08	586.81	5.46	349.34	287.24	14.04	584.92	5.49	353.13	290.09	14.01	585.86	5.51	358.82	295.78
400ms	14.15	602.93	5.60	369.25	304.31	14.19	606.72	5.60	370.19	305.26	14.22	605.77	5.59	365.45	301.46
450ms	13.79	568.80	5.42	343.65	284.40	13.85	568.80	5.40	337.96	279.66	13.96	568.80	5.47	337.96	278.71
500ms	14.26	615.25	5.70	377.78	311.89	14.39	617.15	5.67	365.45	300.52	14.32	614.30	5.69	373.04	307.15
550ms	14.14	601.03	5.79	373.99	308.10	14.37	601.98	5.79	356.92	293.88	14.25	601.03	5.67	356.92	294.83

表 4: iOS における TCP 輻輳制御の t，W_{CUBIC}，K，W_{max}，$W^{'}$ の値

	1サイクル目					2サイクル目					3サイクル目				
	t	W_{CUBIC}	K	W_{max}	$W^{'}$	t	W_{CUBIC}	K	W_{max}	$W^{'}$	t	W_{CUBIC}	K	W_{max}	$W^{'}$
25ms	11.90	239.84	4.37	119.92	95.75	11.97	238.90	4.51	125.61	100.49	11.91	241.74	4.41	124.66	99.54
50ms	11.43	476.84	7.15	457.41	364.98	15.55	442.72	5.53	197.66	158.32	15.88	459.78	5.34	178.70	143.15
100ms	15.40	657.91	7.49	534.20	429.44	15.45	662.65	7.60	540.83	429.44	15.67	662.65	7.57	530.41	423.76
150ms	15.55	650.33	7.69	532.30	425.65	15.56	651.28	7.42	519.98	421.86	15.58	650.33	7.62	526.61	420.91
200ms	14.95	629.47	7.52	525.66	419.96	15.27	630.42	7.42	508.60	406.69	16.28	638.95	6.74	409.06	327.06
250ms	16.91	599.14	7.13	399.58	319.48	15.49	592.50	7.76	499.12	399.11	15.74	594.40	7.66	483.01	386.78
300ms	16.52	585.86	6.89	389.15	311.89	16.76	586.81	6.79	368.30	293.88	15.82	580.18	7.34	449.83	360.24
350ms	16.75	621.89	7.10	422.33	338.44	15.33	612.41	7.78	518.08	414.28	15.88	614.30	7.62	493.43	395.32
400ms	14.40	591.55	7.05	483.95	387.73	15.68	601.98	6.40	374.93	300.52	14.39	591.55	7.04	483.01	384.89
450ms	16.57	612.41	7.19	425.18	339.38	15.41	601.98	7.58	494.38	394.37	15.69	603.88	7.58	485.85	388.68
500ms	16.36	605.77	7.14	424.23	337.49	16.60	608.62	7.04	407.17	325.16	16.54	606.72	7.00	408.11	327.06
550ms	13.98	639.90	6.85	556.00	487.27	16.70	659.81	6.65	420.44	350.76	16.38	658.86	6.49	428.97	364.98

と図 8 の iOS における TCP 輻輳制御の三次関数部分の b を計算する．パラメータ b は式 (3) を変形した式 (5) より求める．

$$b = \frac{K^3}{(W_{max} - W')} \tag{5}$$

W_{max} はウィンドウサイズの傾きが 0 になるときのウィンドウサイズの値，W' はウィンドウサイズの最小値，K はウィンドウサイズ縮小後，ウィンドウサイズの傾きが 0 になるまでの時間である．表 3，表 4 のそれぞれの値を式 (5) に代入し，パラメータ b を求める．

Linux の TCP CUBIC で用いられる式 (2)，式 (3) よりパラメータ c，パラメータ b は，$t = 0$ において式 (6) のように示せる．

$$\begin{aligned} W^{'} &= c * (0 - \sqrt[3]{b * (W_{max} - W^{'})})^3 + W_{max} \\ W^{'} - W_{max} &= cb * (W^{'} - W_{max}) \\ cb &= 1 \\ b &= \frac{1}{c} \end{aligned} \tag{6}$$

RTT 毎に得られるウィンドウの変化において，三次関数的にウィンドウを変化させている部分を図 9 のように抽出する．Linux（CUBIC），iOS における TCP ウィンドウの抽出した三次関数部分のパラメータ c，パラメータ b，式 (6) より求められる $1/c$ を RTT 毎に 3 サイクル分ずつ計算した結果を表 5，表 6 に示す．

表 5: Linux（CUBIC）のパラメータ c，パラメータ b と $1/c$

	1 サイクル目			2 サイクル目			3 サイクル目		
	c	b	$1/c$	c	b	$1/c$	c	b	$1/c$
25ms	0.374	2.61	2.67	0.379	2.57	2.64	0.387	2.59	2.58
50ms	0.393	2.81	2.54	0.397	2.83	2.52	0.395	2.82	2.53
100ms	0.385	2.75	2.60	0.388	2.75	2.58	0.387	2.72	2.58
150ms	0.377	2.63	2.65	0.378	2.71	2.65	0.381	2.64	2.62
200ms	0.377	2.64	2.65	0.373	2.70	2.68	0.372	2.59	2.69
250ms	0.370	2.61	2.70	0.371	2.60	2.70	0.373	2.63	2.68
300ms	0.375	2.64	2.67	0.379	2.75	2.64	0.376	2.60	2.66
350ms	0.371	2.63	2.70	0.372	2.63	2.69	0.370	2.66	2.70
400ms	0.374	2.70	2.67	0.372	2.70	2.69	0.374	2.73	2.67
450ms	0.383	2.68	2.61	0.382	2.70	2.62	0.378	2.77	2.65
500ms	0.379	2.81	2.64	0.379	2.80	2.64	0.376	2.80	2.66
550ms	0.390	2.95	2.56	0.387	3.08	2.58	0.387	2.94	2.58

表 6: iOS における TCP 輻輳制御のパラメータ c，パラメータ b と $1/c$

	1 サイクル目			2 サイクル目			3 サイクル目		
	c	b	$1/c$	c	b	$1/c$	c	b	$1/c$
25ms	0.281	3.46	3.56	0.273	3.66	3.66	0.278	3.42	3.60
50ms	0.249	3.96	4.02	0.243	4.29	4.12	0.240	4.29	4.17
100ms	0.250	4.01	4.00	0.252	3.94	3.97	0.249	4.07	4.02
150ms	0.242	4.26	4.13	0.244	4.16	4.10	0.246	4.18	4.07
200ms	0.253	4.03	3.95	0.252	4.00	3.97	0.265	3.73	3.77
250ms	0.213	4.53	4.69	0.203	4.68	4.93	0.211	4.66	4.74
300ms	0.220	4.23	4.55	0.220	4.21	4.55	0.214	4.41	4.67
350ms	0.222	4.26	4.50	0.219	4.53	4.57	0.214	4.52	4.67
400ms	0.271	3.64	3.69	0.283	3.52	3.53	0.273	3.55	3.66
450ms	0.227	4.32	4.41	0.224	4.36	4.46	0.222	4.49	4.50
500ms	0.232	4.20	4.31	0.230	4.26	4.35	0.229	4.23	4.37
550ms	0.231	4.67	4.33	0.236	4.22	4.24	0.238	4.27	4.20

表 5 と表 6 を比較すると，Linux（CUBIC）のウィンドウの変化に関するパラメータ c の値はおよそ 0.4，パラメータ b の値はおよそ 2.5〜3.0 になった．iOS における TCP 輻輳制御のパラメータ c の値はおよそ 0.2〜0.3，パラメータ b の値はおよそ 3.5〜4.5 になった．Linux（CUBIC）より iOS における TCP 輻輳制御のパラメータ c の値は小さく，パラメータ b は値が大きくなった．これより，iOS における TCP 輻輳制御は Linux（CUBIC）と比較すると，ウィンドウサイズの増加が緩やかになっていると言える．

パラメータ c，パラメータ b はウィンドウサイズを縮小してから，ウィンドウサイズの傾きが 0 になるまでの時間 K の目算値より計算したが，パラメータ b と $1/c$ の値の差異は Linux（CUBIC）で 20 %以下，iOS における TCP 輻輳制御で 8 %以下であった．

7. まとめ

本研究では iOS における TCP 輻輳制御の性能評価とアルゴリズム，パラメータの推定を行った．

性能評価では，iOS における TCP 輻輳制御と Linux（CUBIC）のスループットを測定すると，RTT=25ms では iOS における TCP 輻輳制御，RTT=550ms では Linux（CUBIC）の方が高いスループットとなった．RTT=100ms では，リンクエラー率 0.0001 を境に Linux（CUBIC）と iOS における TCP 輻輳制御のスループットの大小関係が変化した．

アルゴリズム，パラメータの推定では iOS における TCP 輻輳制御と TCP CUBIC のウィンドウサイズの変化の形が似ており，iOS における TCP 輻輳制御に TCP CUBIC で用いるウィンドウサイズの式を代入し，測定値と計算値を比較するとほとんど一致した．このことから，iOS における TCP 輻輳制御は TCP CUBIC であることが推定できる．ウィンドウサイズの変化に関するパラメータ c，パラメータ b を計算すると，Linux（CUBIC）のパラメータ c の値はおよそ 0.4，パラメータ b の値はおよそ 2.5〜3.0 であるのに対し，iOS における TCP 輻輳制御のパラメータ c の値はおよそ 0.2〜0.3，パラメータ b の値はおよそ 3.5〜4.5 である．iOS における TCP 輻輳制御のパラメータ c，パラメータ b の値にばらつきが見られ，値が定まらなかった．しかし，Linux（CUBIC）のパラメータ c，パラメータ b と比較するとパラメータ c の値は小さく，パラメータ b の値は大きくなっていることがわかった．iOS における TCP 輻輳制御は，Linux（CUBIC）よりウィンドウサイズの増加が緩やかになっていると言える．

参考文献

[1] Carlo Caini and Rosario Firrincieli，”TCP Hybla: a TCP enhancement for heterogeneous networks”，International Journal of Satellite Communications and Networking，vol.22，pp547-566，2004.

[2] Kantar Worldpanel，”Smartphone OS sales market share “ Kantar Worldpanel ComTech”，https://www.kantarworldpanel.com/global/smartphone-os-market-share/，Jone 2017.

[3] Toshihiko Kato，Atsushi Oda，Shun Ayukawa，Celimuge Wu and Satoshi Ohzahata，”Inferring TCP Congestion Control Algorithms by Correlating congestion Window Sizes and their Differences”，Proceedings of The Ninth International Conference on Systems and Networks Communications (ICSNC 2014)，pp42-47，2014.

[4] Sangtae Ha，Injong Rhee，and Lisong Xu，”CUBIC:A New TCP-Friendly High-speed TCP Variant”，ACM SIGOPS Operating System Review，vol.42，no.5，pp64-74，2008.

[5] D.J.Leith，R.N.Shorten，G.McCullagh，Hamilton Institute，Ireland，”Experimental evaluation of Cubic-TCP”，IEEE/ACM Transactions on Networking (ToN)，vol.15，no.5，pp1109-1102，2008.

[6] Marta Carbone，Luigi Rizzo，”Dummynet revisited”，ACM SIGCOMM Computer Communication Review，vol.40，no.5，pp12-20，2010.