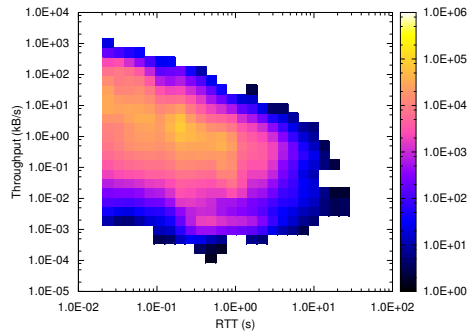
Studying TCP in the wild



Richard G. Clegg (richard@richardclegg.org),João Taveira Araújo, Raul Landa, Eleni Mykoniati, David Griffin, Miguel Rio,
University College London, Department of Electronic Engineering

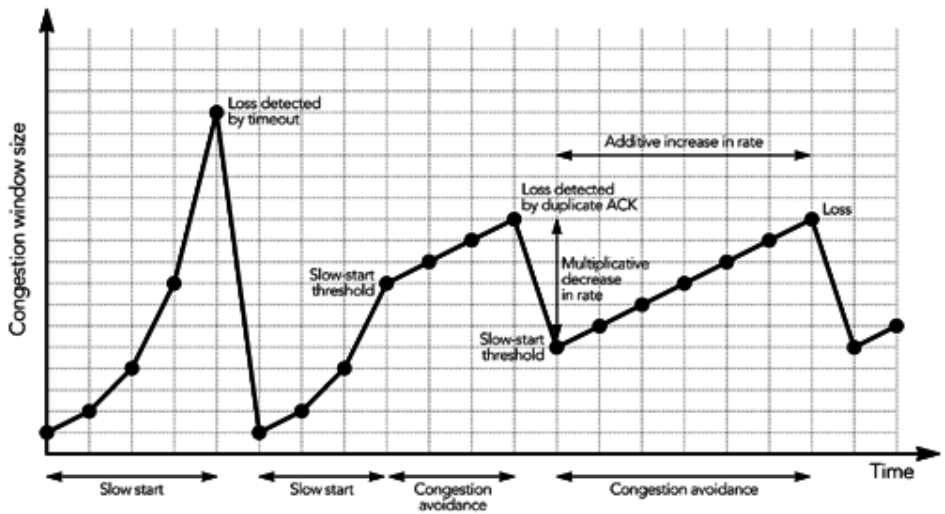
Talk to KCL 2014

(Prepared using LaTeX and beamer.)

Studying TCP in the wild

Concept starting point

What we all know about TCP (but maybe suspect is wrong). TCP "fills a pipe". TCP reacts to loss (or delay) to maximise bandwidth – typically using AIMD.



Studying TCP in the wild

Mathematical starting point

Padhye et al – bandwidth (throughput) of TCP flow at equilibrium:

$$T = \frac{1}{D}\sqrt{\frac{3}{2bp}} + o(1/\sqrt{p}),$$

where D is RTT (delay), p is the probability of packet loss and b is a fixed TCP parameter.

- Result (simplified version presented) is from mathematical model with many assumptions.
- Subsequent work generalises and improves – basic inverse dependence on RTT and $\sqrt{p}$ remain fundamental.
- Beautiful mathematically, how is it statistically?
- Is TCP doing what we think it does and if not, why not?

Two parts to this work

Part One

On the relationship between fundamental measurements in TCP flows Richard G. Clegg, Joao Taveira Araujo, Raul Landa, Eleni Mykoniati, David Griffin, Miguel Rio, ICC 2013.
Looks at several anonymous data sets, attempts to fit equations using standard statistical techniques.

Part Two

A longitudinal analysis of Internet rate limitations Joao Taveira Araujo, Raul Landa, Richard G. Clegg, George Pavlou, Kensuke Fukuda, INFOCOM 2014
Looks at one long non anonymous data set, attempts to understand nature of rate limitation affecting traces.

Commonality – TCP flow reconstruction, data analysis.

On the relationship between fundamental measurements in TCP flows

Part One

On the relationship between fundamental measurements in TCP flows Richard G. Clegg, Joao Taveira Araujo, Raul Landa, Eleni Mykoniati, David Griffin, Miguel Rio, ICC 2013.
Looks at several anonymous data sets, attempts to fit equations using standard statistical techniques.

- Take several well known anonymous data sets.
- Reconstruct TCP flows.
- Create statistical models which fit throughput to loss, RTT, flow length etc etc.

Data and analysis approach

- Basic approach – use lots of freely available packet traces.
- Test both diverse data sets and similar data sets.
- Reconstruct TCP flows – calculate RTT, loss etc. Fit formulae relating these quantities.
- Data used CAIDA (US based data) MAWI (Japanese based data):
  - CAIDA OC48 Traces (2002) — 3 hours of data: 1.4 billion packets originally 876GB of data.
  - CAIDA OC192 (2011A) — 26 minutes of data: 1.3 billion packets originally 662GB of data.
  - CAIDA OC192 (2011B) — 14 minutes of data: 0.927 billion packets, 582 GB of data.
  - CAIDA OC192 (2012) — 29 minutes of data 1.6 billion packets and 1,120 GB of data.
  - MAWI (2006–2012) — 15 minute samples once per month, 1.36 billion packets and 982 GB of data.

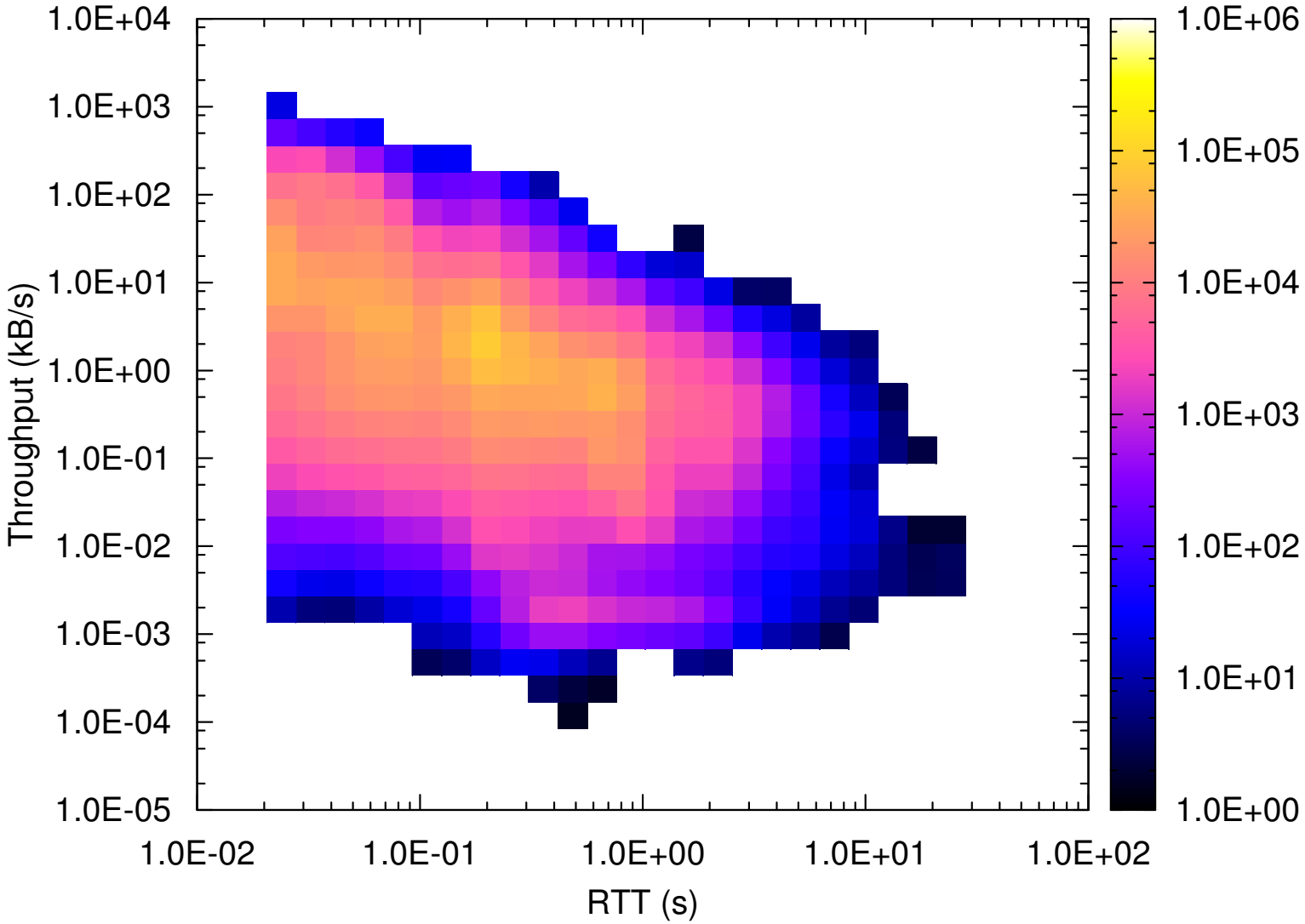
Fundamental relationships within TCP flows

- What is the best relationship which ties network parameters to TCP throughput.
- Let the data speak for itself – find the model which best explains TCP throughput.
- Step 1: graphically investigate the relationships in the data sets.
- Step 2: statistically fit equations which relate the parameters: throughput, loss, RTT, flow length.
- Consider subsets of data to ask questions about equilibrium and transient behaviour.

Data processing/filtering

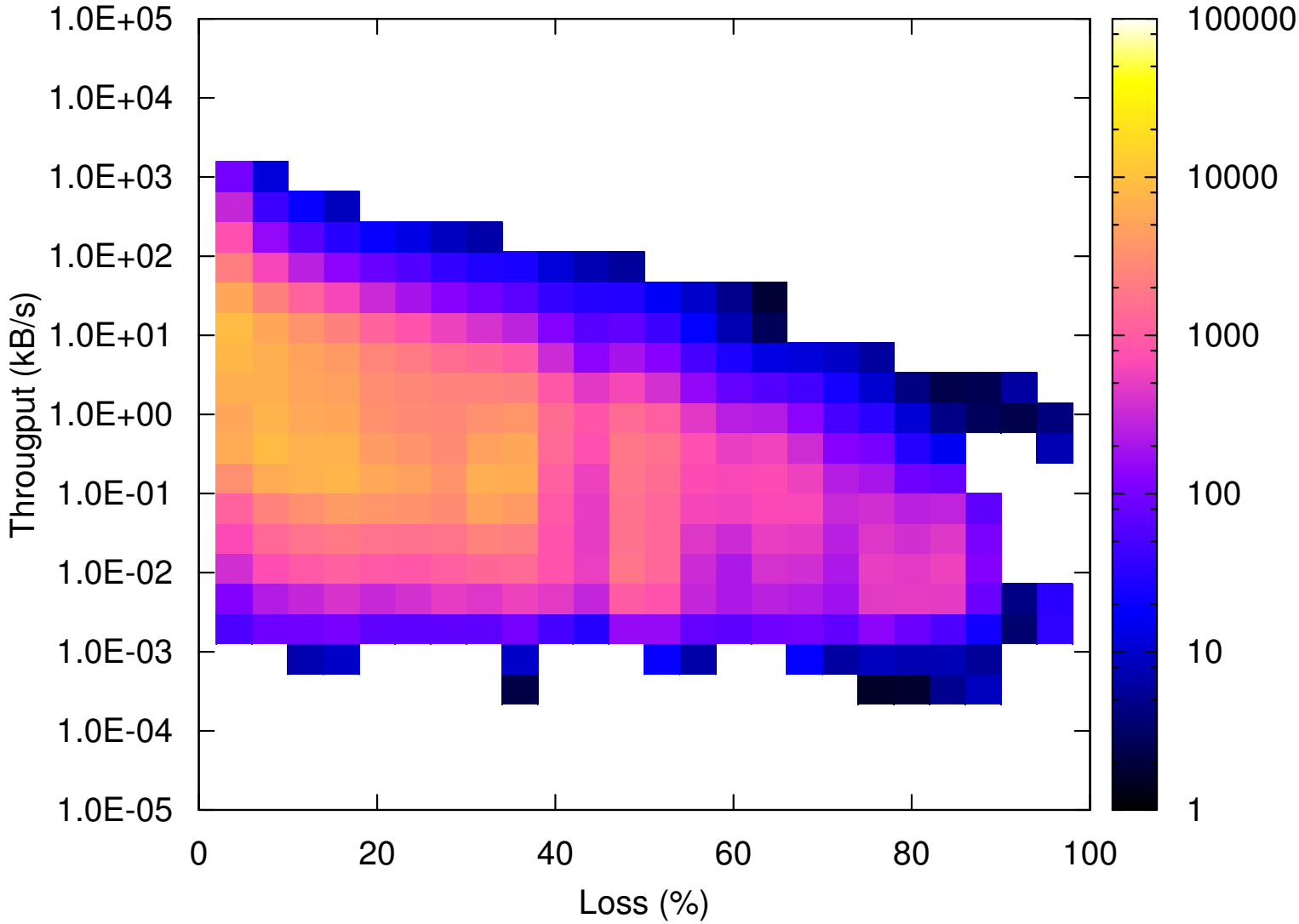
- To get accurate RTT estimates only two-way data is considered.
- RTT can be inferred from SYN/SYNACK/ACK handshake.
- RTT can also be inferred from data transfer when data in both directions (both are used here).
- OC48 and MAWI both directions seen majority of time. OC192 less so.
- Truncation effects mitigated by removing flows do not seem to end within lifetime of capture file.
- Starting point is to visualise correlations in data.
- Most interesting visualisation comes from 3d histograms.

Visualising correlations throughput/RTT



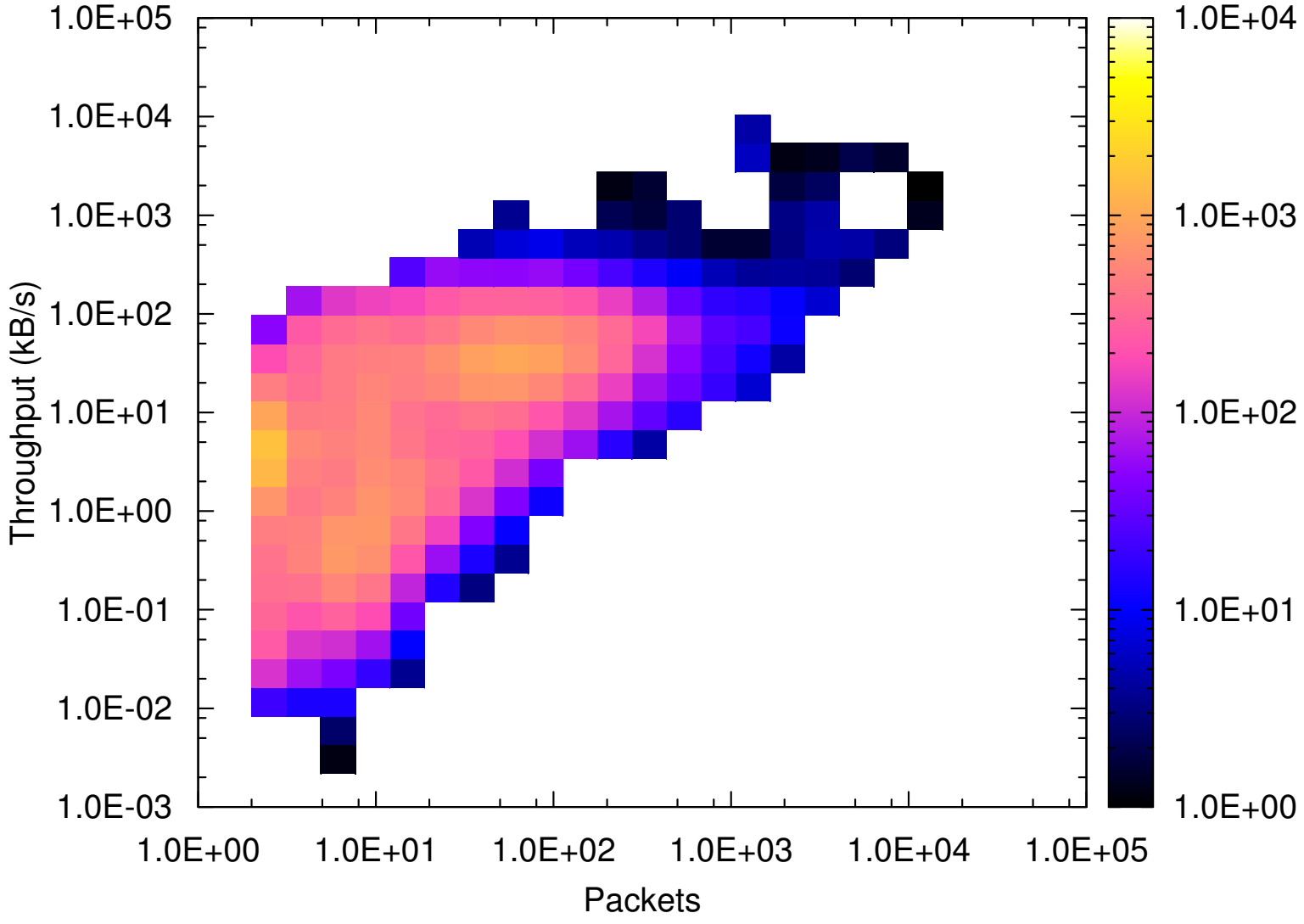
OC48 — relationship between throughput and RTT

Visualising correlations throughput/loss



MAWI — relationship between throughput and loss

Visualising correlations – throughput/packets



OC192 2012 — relationship between throughput and number of packets in flow

Fitting a Linear Model

- Variable Y is observed variable to be explained in terms of variables X_1, X_2 etc.
- Assume a linear relationship $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \varepsilon$ where ε $N(0, \mu)$.
- Want to find β parameters to minimise the error term.
- Fit log of data and use exponential transform to get $T = \beta_0 D^{\beta_1} p^{\beta_2} \varepsilon'$ where ε' is mean 1, lognormal).
- With $\beta_1 = -1$ and $\beta_2 = -0.5$ this is $T = \beta_0 / D\sqrt{p}$ (and error term).
- Goodness of fit judged by R^2 value where $R^2 = 1$ is perfect and $R^2 = 0$ is no fit at all (amount of variance "explained" by model).
- Taking logarithms a problem for loss as sometimes $p = 0$ – use instead $\log p + p_m$ where p_m is a fitted offset parameter.

CAIDA OC192 2012 data

Model for T	R^2	Note
$15.7D^{-0.94}(p+p_m)^{-0.563}P^{0.456}$	0.641	$p_m = 0.105$
$77.2D^{-0.975}P^{0.455}$	0.635	
$316/(D\sqrt{p+p_m})$	0.0227	$p_m = 0.105$

- Excellent fit to data.
- Loss p slightly improves model but not much.
- Best model is approx $T = k\sqrt{P}/D$ where k is constant.

CAIDA OC48 data

Model for T	R^2	Note
$102D^{-0.929}(p+p_m)^{0.391}P^{0.339}$	0.362	$p_m = 0.105$
$29.7D^{-0.89}P^{0.354}$	0.35	
$193/(D\sqrt{p+p_m})$	0.207	$p_m = 0.105$

- Weaker fit to data but not bad for a simple model.
- Again p (loss) has little explanatory power.

CAIDA OC192 2011A data

Model for T	R^2	Note
$0.712 D^{-0.665}(p+p_m)^{-0.661} P^{0.429}$	0.454	$p_m = 0.105$
$4.62 D^{-0.698} P^{0.41}$	0.448	
$251/(D\sqrt{p+p_m})$	0.109	$p_m = 0.105$

- Reasonable fit to data.
- Again loss p not much help.
- Best model approx $T = kP^{0.4}/D^{0}.7$.

CAIDA OC192 2011B data

Model for T	R^2	Note
$21.5D^{-0.924}(p+p_m)^{-0.581}P^{0.419}$	0.616	$p_m = 0.105$
$156D^{-0.981}P^{0.386}$	0.611	
$562/(D\sqrt{(p+p_m)})$	0.19	$p_m = 0.105$

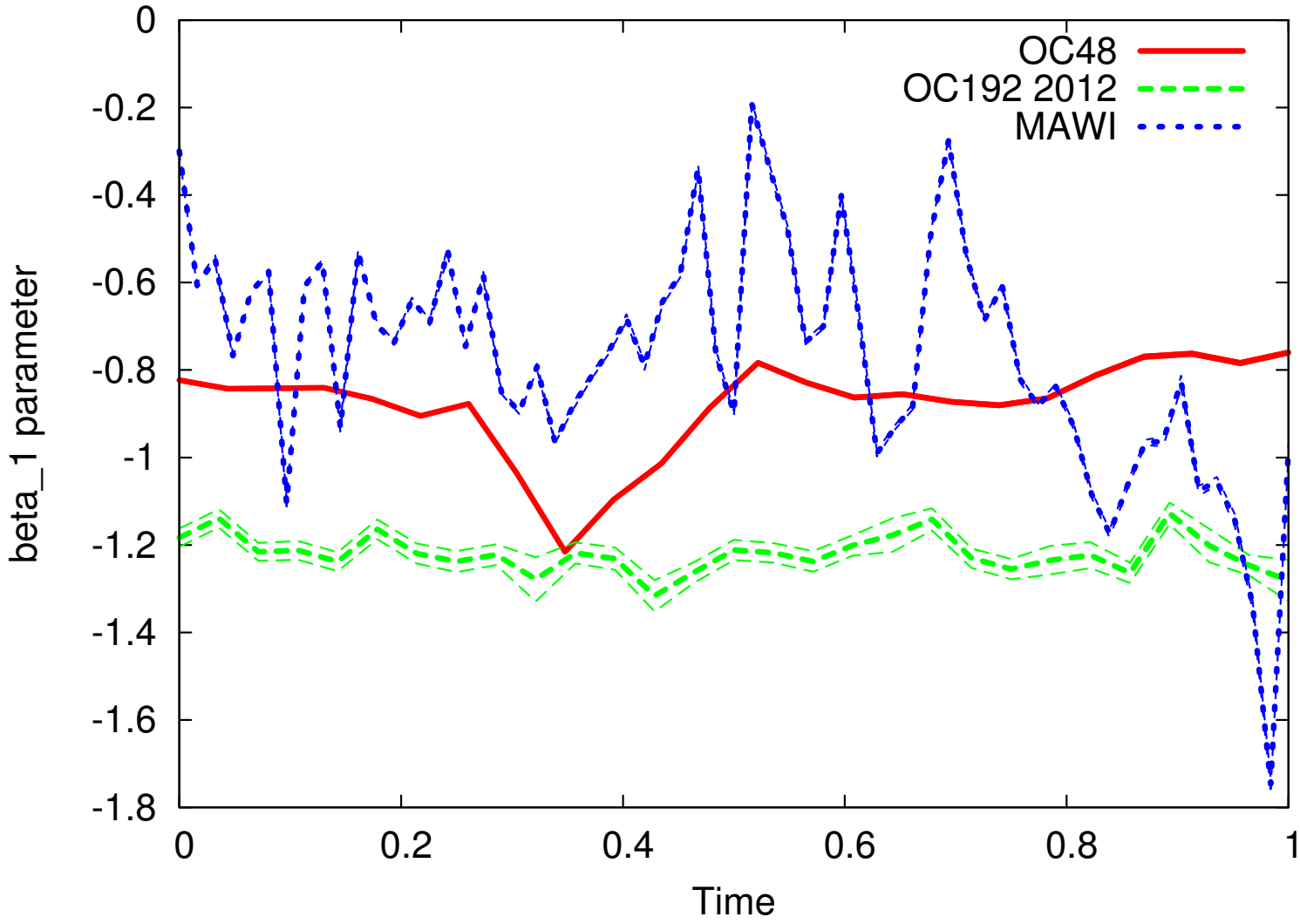
- Much better fit than 2011A.
- Best model approx $T = kP^{0.4}/D$.

MAWI data

Model for T	R^2	Note
$0.15D^{-0.664}(p+p_m)^{-0.416}P^{0.635}$	0.282	$p_m = 0.0132$
$0.648D^{-0.583}P^{0.576}$	0.332	$P > 1000$
$111/(D\sqrt{p+p_m})$	0.0904	$p_m = 0.105$

- Fairly weak fit to data.
- Perhaps because data over long time period.
- Best fit is only for long flows (more than 1000 packets).

Parameter dynamism



Evolution of β_1 parameter in model $T = \beta_0 D^{\beta_1}$ across normalised time

A longitudinal analysis of Internet rate limitations

Part Two

A longitudinal analysis of Internet rate limitations Joao Taveira Araujo, Raul Landa, Richard G. Clegg, George Pavlou, Kensuke Fukuda, INFOCOM 2014

Looks at one long non anonymous data set, attempts to understand nature of rate limitation affecting traces.

- Reconstruct flows from a non anonymous data set (into/out of Japan – MAWI).
- Look in detail at nature of each flow.
- Reconstruct "flights" and hence estimate window size.
- Fit an "explanation" for the throughput of each flow.

Japanese data set WIDE/MAWI

- WIDE backbone network (Japanese version of JANET).
- 5.7 billion flows
- 5 years, 15 minutes a day
- 30 terabytes of TCP traffic
- Study here is on inbound

Year	Days	TCP data flows ($\times 10^6$)	Traffic (TB) In	Traffic (TB) Out	Count ($\times 10^3$) AS	Count ($\times 10^3$) Prefixes
2006	91	20.52	0.43	0.45	10.90	56.86
2007	350	102.56	2.11	2.49	17.21	113.79
2008	358	112.26	2.43	2.10	24.74	156.54
2009	364	113.97	2.48	2.53	19.71	143.87
2010	365	113.70	2.58	3.43	20.38	148.03
2011	358	114.74	3.44	5.14	19.99	140.56
Total	1886	5777.55	13.50	16.14	34.12	341.22

Approach

- Reconstruct raw data into TCP flows as before, estimating loss, throughput etc.
- Estimate transient RTT and break flows into flights.
- From flights estimate and plot window size as a function of time.
- Look for causative mechanisms for delay as seen in the data.
- Create simple ways to automatically classify flows by the major mechanism which causes delay.

Itentified reasons for delay

Application Paced

The application does not want you to get the data – think you tube.

Host limited

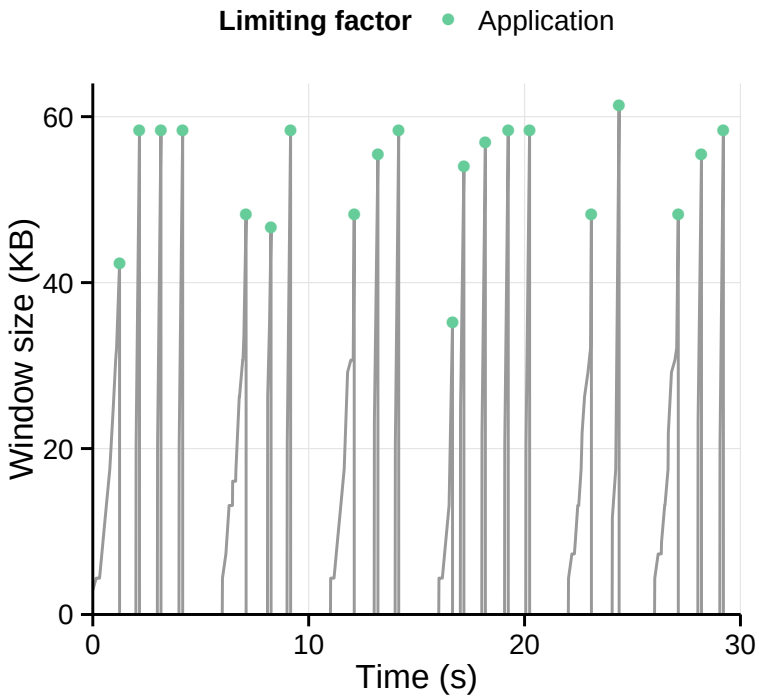
The connection reaches a maximum window size limited by the host OS at one end – think must upgrade my OS.

Receiver shaped

Receiver or middlebox deliberately manipulates advertised window size – think traffic shaping.

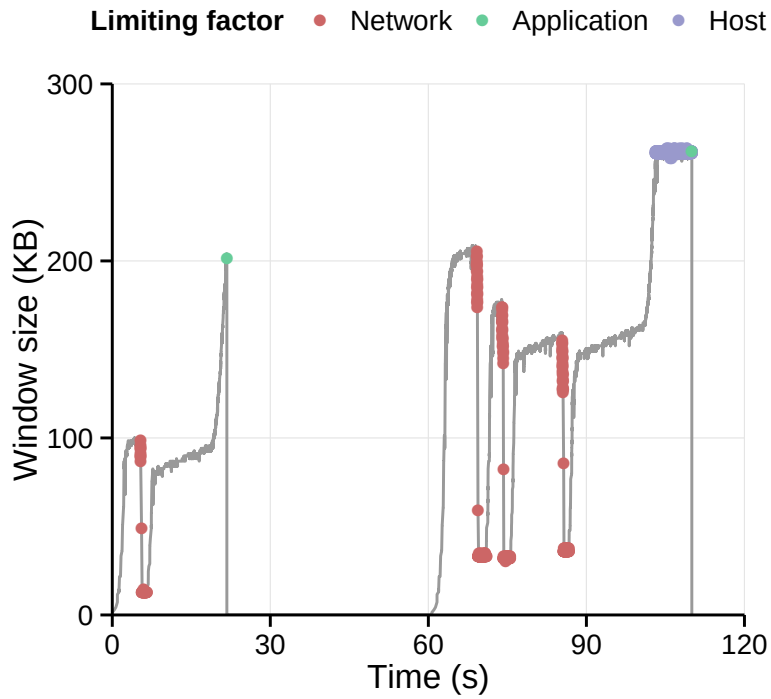
Application paced flows – app does not want traffic faster

- Characteristic – ordinary TCP window behaviour then drop out.
- Application limited flight – terminated by packet less than MSS.
- Application paced flow – period between application limited flights $<$10 secs.
- Std dev of the intermediate pauses $<$1 sec.
- Rule out user controlled limitations (e.g. web browsing).

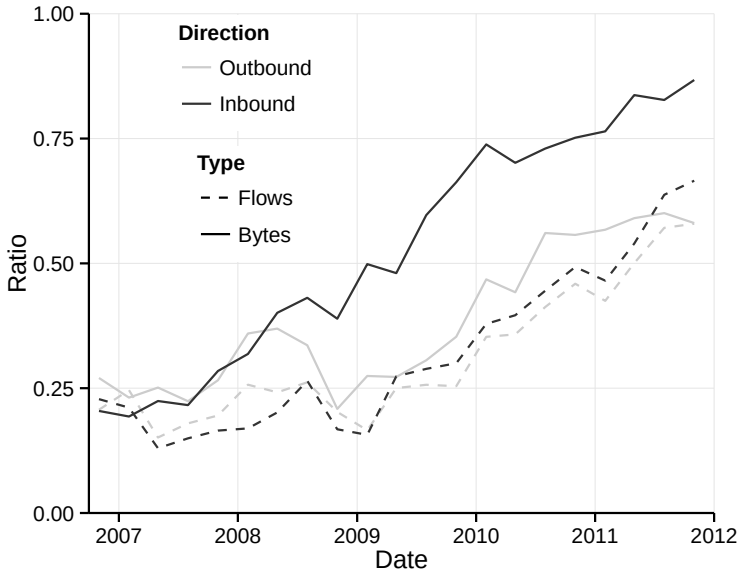


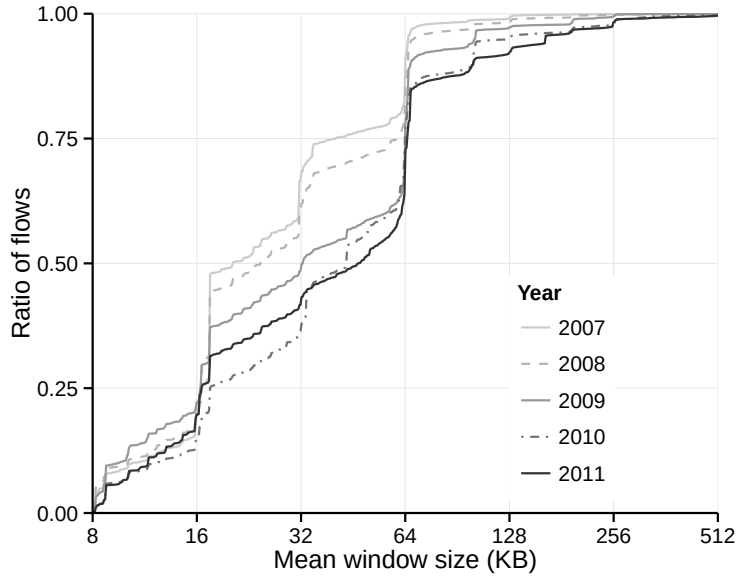
Host limited flows – host hits maximum window size

- Characteristic – window size reaches a hard limit and sticks.
- Max window size same for six RTT.
- Average window size within 10% of max for flow lifetime.
- Later rules out occasions where host limit is not main limit (e.g. right).



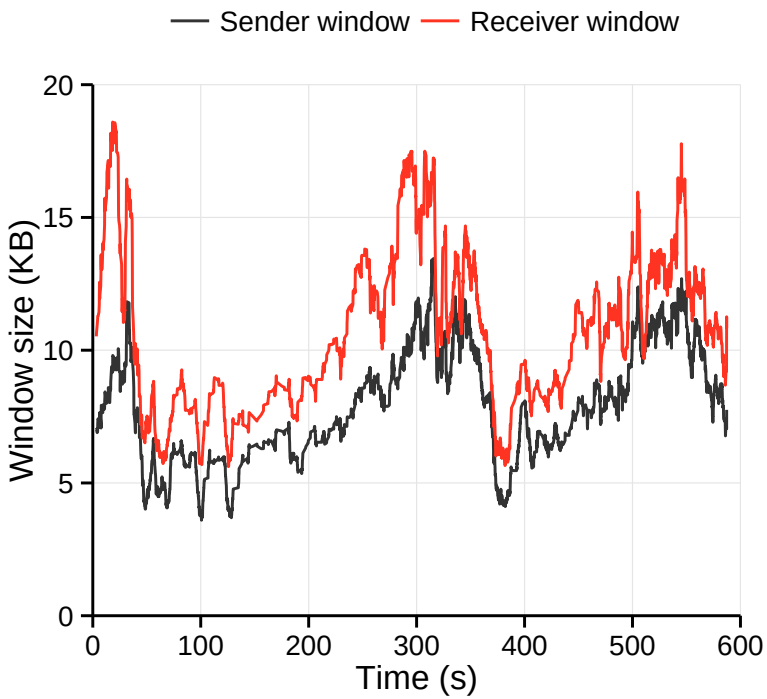
Window sizes in detail





Receiver shaped flows – middlebox/receiver fiddles with window advert

- Characteristic – advertised window manipulated to reduce traffic (e.g. by middlebox).
- In absence of loss advertised window and maximum flight size correlate.
- Statistically significant cross correlation on with $p < 0.05$ over 10 RTT around flight.
- More than half of flights are flagged such. Ignore flights with out-of-order/loss.



Summary by mechanism – flows under 10MB

Year	Limitation (%)				Loss (%)
	App	Host	Receiver	Total	
2007	14.65	5.45	0.10	20.20	1.90
2008	14.99	5.37	0.09	20.44	2.27
2009	15.66	3.83	0.55	20.03	2.39
2010	10.55	4.18	0.36	15.09	2.15
2011	11.13	2.53	0.05	13.71	1.19

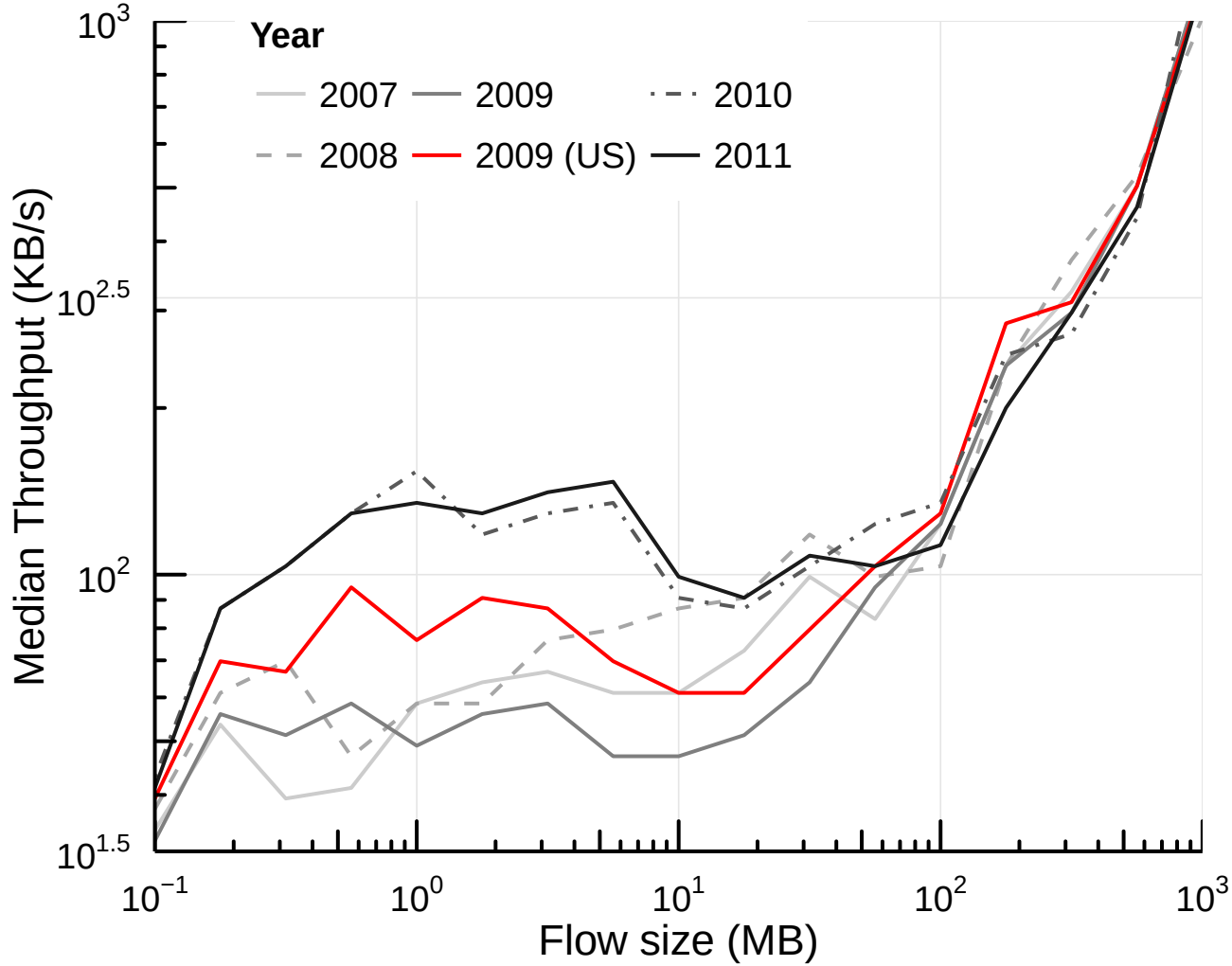
Summary by mechanism – flows over 10MB

	Limitation (%)				
Year	App	Host	Receiver	Total	Loss (%)
2007	61.62	23.07	0.71	85.40	0.96
2008	61.49	21.94	0.92	84.35	0.88
2009	57.86	17.70	3.28	78.85	0.98
2010	43.97	24.45	4.03	72.45	0.71
2011	52.95	15.55	0.71	69.21	0.62

Summary by AS

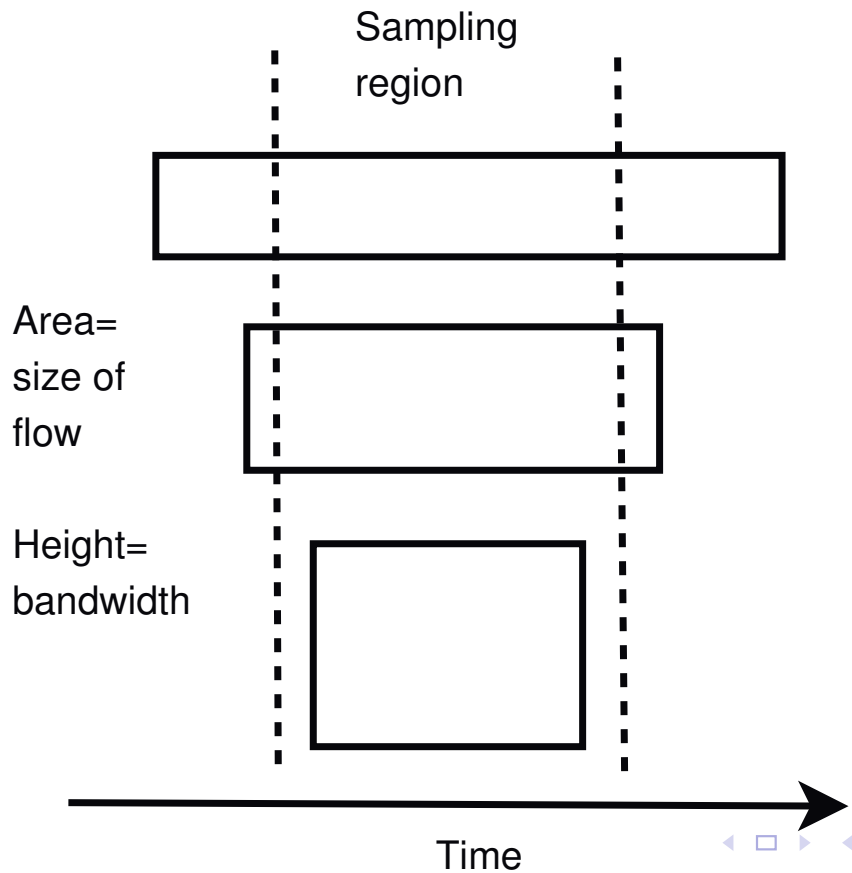
Year	ASN	AS Name	Traffic (%)	Limitation (%) App	Host	Receiver
	3462	HiNet	9.93	60.07	4.82	0.05
	15169	Google	8.78	74.79	12.16	0.02
2009	43515	Google/YT	8.08	83.46	9.83	0.14
	2914	NTT	5.69	39.76	8.37	0.16
	46742	Carpathia	4.27	41.04	48.01	2.03
	2914	NTT	7.39	21.80	4.91	0.00
	31976	Red Hat	7.03	9.62	41.63	0.00
2010	7366	Lemuria	5.88	51.95	15.72	5.85
	43515	Google/YT	5.22	77.76	8.41	0.14
	46742	Carpathia	4.69	33.06	42.71	4.21
	2914	NTT	10.37	50.33	8.19	0.18
	20473	Choopa	8.92	54.03	19.24	0.21
2011	43515	Google/YT	8.69	69.71	7.56	0.16
	35415	Webazilla	6.05	40.02	11.23	0.95
	40824	WZ Comm.	4.83	42.08	17.43	0.05

Throughput as a function of flow size

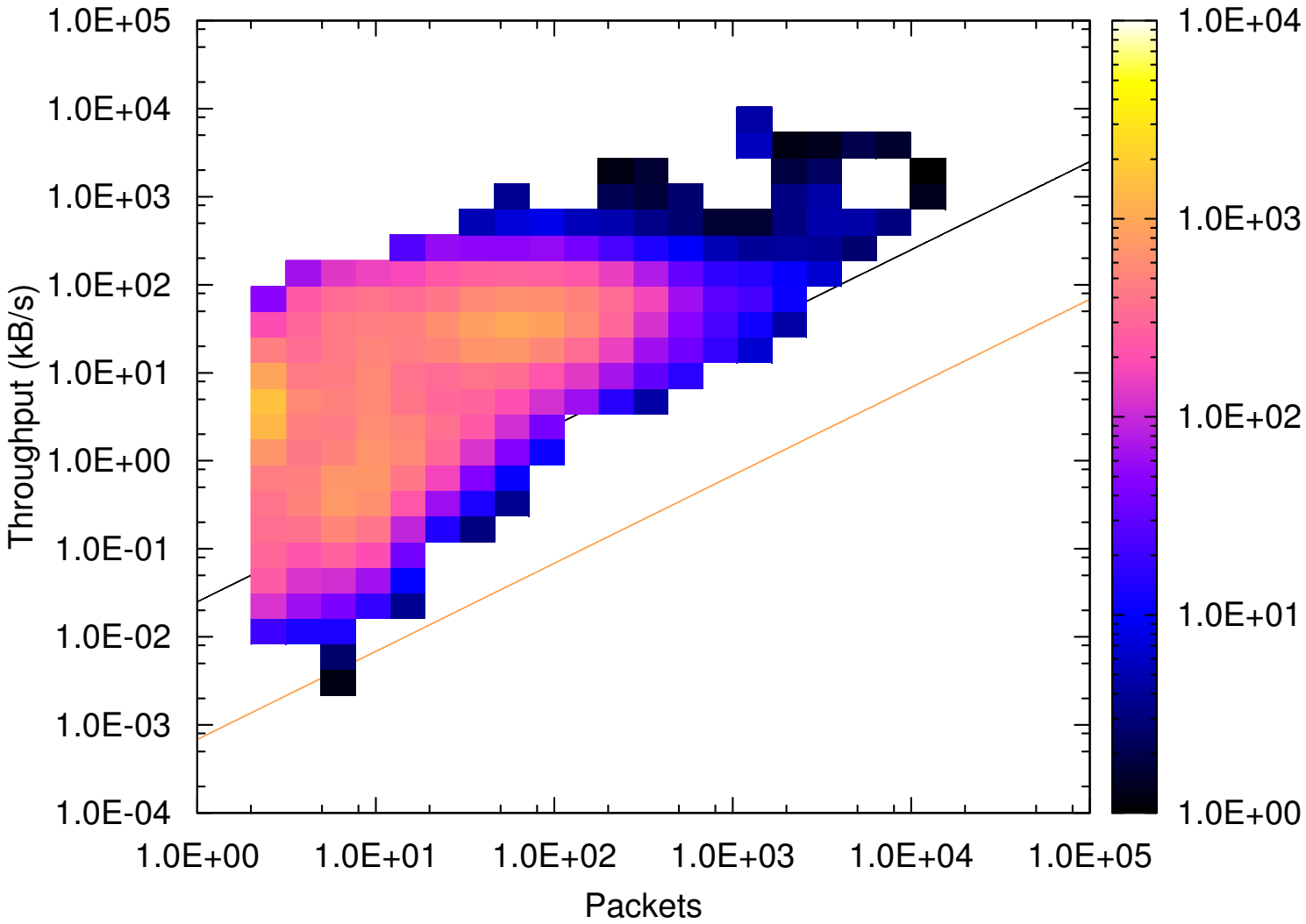


Relationship between flow length and throughput

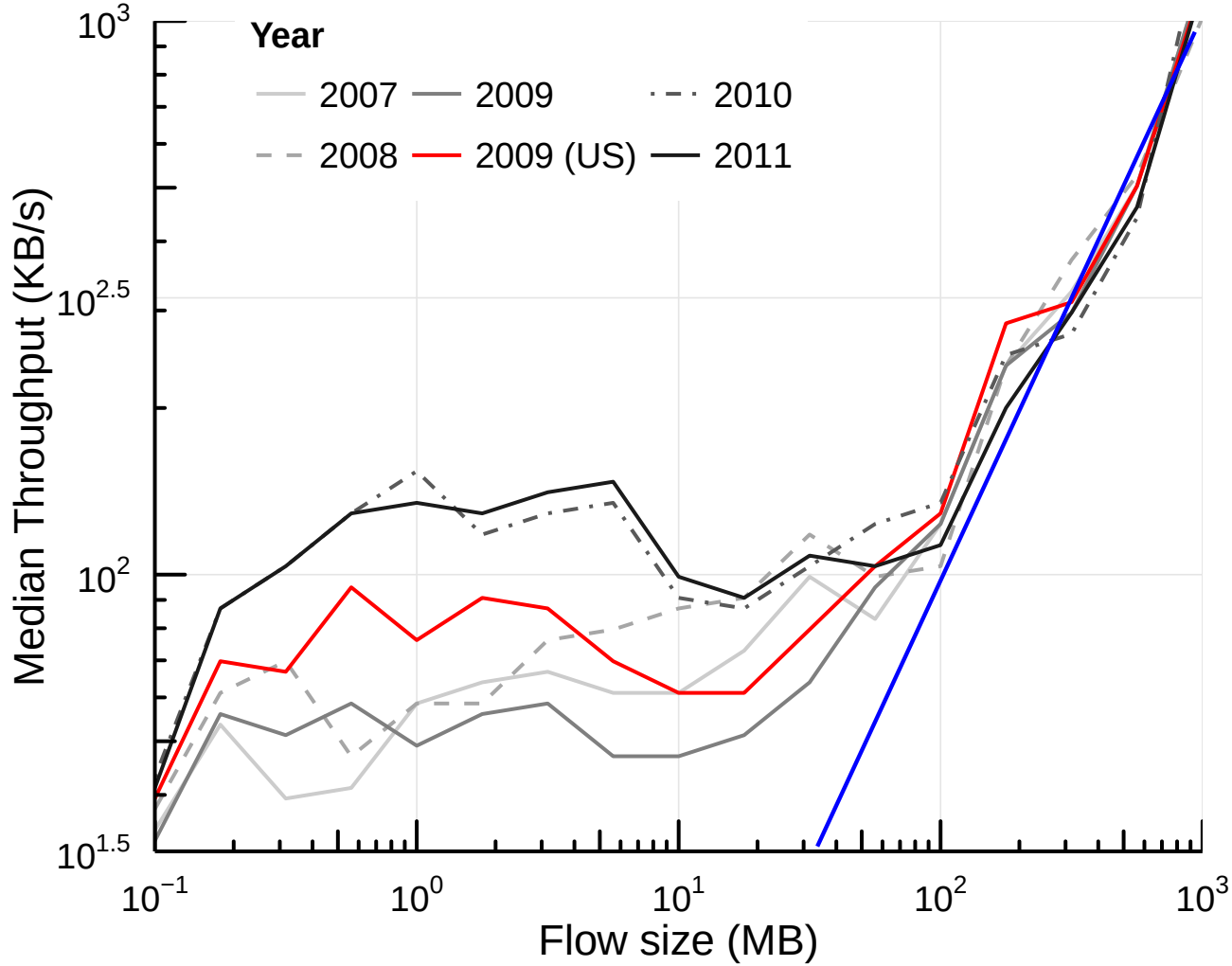
- Flow length is correlated with throughput even outside slow start region
- This has been long observed e.g. [Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the characteristics and origins of internet flow rates 2002].
- Could it in fact be a sampling issue?



Throughput and flow size with excluded region



Throughput as a function of flow size



Questioning TCP wisdom (1)

TCP throughput is loss driven

The correlation between observed loss and throughput found was minimal. Models including loss were usually little if any better than those including delay and packet length. Delay and session length are far more important.

Questioning TCP wisdom (1)

TCP throughput is loss driven

The correlation between observed loss and throughput found was minimal. Models including loss were usually little if any better than those including delay and packet length. Delay and session length are far more important.

TCP congestion control governs network behaviour

As of early 2012 we find that less than 40% of all inbound traffic had TCP congestion control as the primary rate control mechanism.

Questioning TCP wisdom (1)

TCP throughput is loss driven

The correlation between observed loss and throughput found was minimal. Models including loss were usually little if any better than those including delay and packet length. Delay and session length are far more important.

TCP congestion control governs network behaviour

As of early 2012 we find that less than 40% of all inbound traffic had TCP congestion control as the primary rate control mechanism.

TCP throughput is proportional to $1/\sqrt{p}RTT$

A better model was $B = \sqrt{P}/RTT$ where P is the length in packets. Simple models were surprisingly accurate.

Questioning TCP wisdom (2)

TCP throughput is primarily sender driven

In the data considered receiver shaping and host limitations together affect up to 24% of all traffic.

Questioning TCP wisdom (2)

TCP throughput is primarily sender driven

In the data considered receiver shaping and host limitations together affect up to 24% of all traffic.

Longer flows have higher throughput

While this was the overall pattern, in fact, for mid-range flows there was little effect. Short flows were "slow" and long flows "fast". The long flows could be a sampling effect.

Further work

- Make the classification more rigorous (how can we get ground truth?)
- Investigate potential sampling issues.
- Better estimators for RTT within flow and for window sizes.
- Get more data into the problem from more different sources.
- Currently working (with Richard Mortier) on new flow analysis tools (naturally written in ocaml).
- Get more data, process more data, do more rigorous statistics, learn more.

Questions?

?