Splitting and Merging of Packet Traffic: Measurement and Modelling

Nicolas Hohn[1] Darryl Veitch[1] Tao Ye[2]

[1]CUBIN, Department of Electrical & Electronic Engineering
University of Melbourne, Vic 3010 Australia

[2]Sprint Advanced Technology Laboratories, Burlingame CA 94010, USA

Workshop on
Mathematical Modeling and Analysis of Computer Networks
ENS Paris, June 21-22, 2007

CUBIN

The Problem

- Concerned with the point process of packet arrivals
- Models often poorly validated (or worse)
- Models typically for *links* only
- Need Split & Merge properties to move to node, then network
- Poisson has it, but too restrictive (no burstiness, LRD)

Prior Work

- Developed Semi-Experiments to find statistical structure
- Resulted in Cluster model for packet arrivals
- In model: flows are Poisson, each bringing a cluster of packets
- Flows are i.i.d. and non-interacting
  (this is how it really is in the core, no TCP dynamics!)

This Paper

Aim I

- Validate model over more data, wider range of rates, utilisation
- Focus on main underpinning, not details

Aim II

- Look at Splitting & Merging properties of Model
- Look at Splitting & Merging properties of Full-Router data
- Evaluate model extensibility from Link -> Node

This Paper

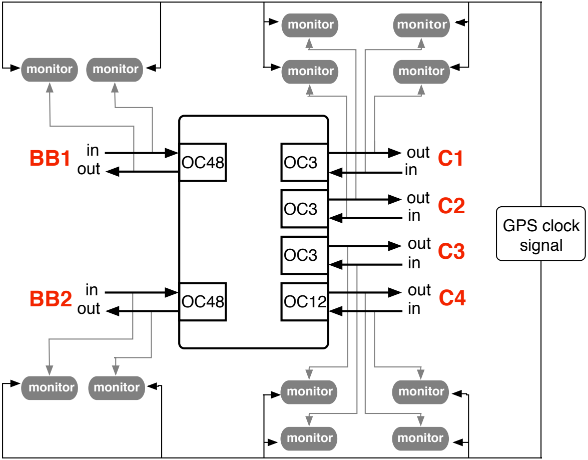
Aim I

- Validate model over more data, wider range of rates, utilisation
- Focus on main underpinning, not details

Aim II

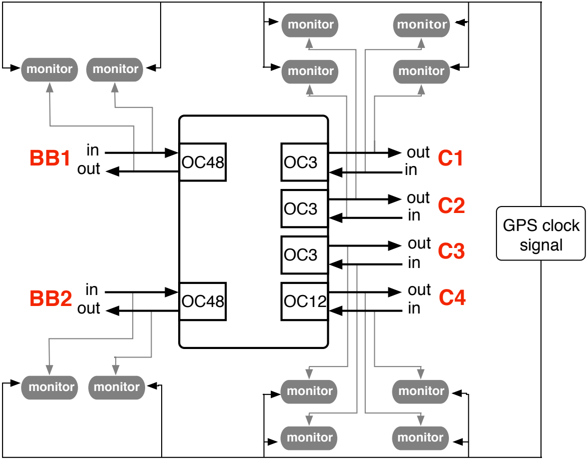
- Look at Splitting & Merging properties of Model
- Look at Splitting & Merging properties of Full-Router data
- Evaluate model extensibility from Link -> Node

THE FULL-ROUTER EXPERIMENT



- Sprint network
- 13h Trace
- 2 backbone links
- 4 customer links
- 7.3 billion packets
- 99.99% monitored

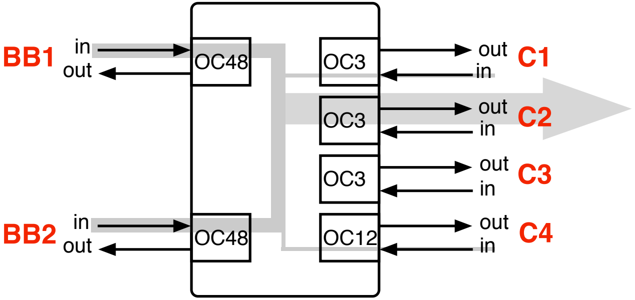
The Full-Router Experiment



- Sprint network
- 13h Trace
- 2 backbone links
- 4 customer links
- 7.3 billion packets
- 99.99% monitored

Packet matching

Identify across all traces the records corresponding to the same packet appearing at different interfaces at different times.



Two hour utilisations vary from 2% to 51%.

Wavelet Analysis

Use the Discrete Wavelet Transform (DWT), to transform stationary X to a set of detail process, one per scale j: $\left\{ d(j,k),\ k = 1, 2, \ldots n_j \right\}$

Statistical benefits:

- Detail processes stationary, quasi-decorrelated, $\mathbb{E}[d(j,\cdot)] = 0$
- No LRD in the wavelet domain!
- So classical statistics, $1/n$ convergence.
- Look at variance of coefficients: $\mathbb{E}[d(j,\cdot)^2]$

Unbiased estimate of variance: $\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} |d_X(j,k)|^2$

View in (log) wavelet spectrum plot:

$$\log_2(\mu_j) \text{ vs } j$$

Wavelet Analysis

Use the Discrete Wavelet Transform (DWT), to transform stationary X to a set of detail process, one per scale j: $\left\{ d(j,k),\ k = 1, 2, \ldots n_j \right\}$

Statistical benefits:

- Detail processes stationary, quasi-decorrelated, $\mathbb{E}[d(j,\cdot)] = 0$
- No LRD in the wavelet domain!
- So classical statistics, $1/n$ convergence.
- Look at variance of coefficients: $\mathbb{E}[d(j,\cdot)^2]$

Unbiased estimate of variance: $\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} |d_X(j,k)|^2$

View in (log) wavelet spectrum plot:

$$\log_2(\mu_j) \text{ vs } j$$

Wavelet Analysis

Use the Discrete Wavelet Transform (DWT), to transform stationary X to a set of detail process, one per scale j: $\left\{ d(j,k),\ k = 1, 2, \ldots n_j \right\}$

Statistical benefits:

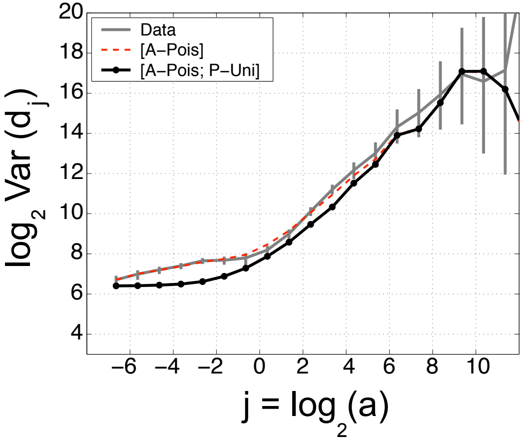
- Detail processes stationary, quasi-decorrelated, $\mathbb{E}[d(j,\cdot)] = 0$
- No LRD in the wavelet domain!
- So classical statistics, $1/n$ convergence.
- Look at variance of coefficients: $\mathbb{E}[d(j,\cdot)^2]$

Unbiased estimate of variance: $\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} |d_X(j,k)|^2$

View in (log) wavelet spectrum plot:

$$\log_2(\mu_j) \text{ vs } j$$

Wavelet Spectra



LRD (or other scale invariance) $\implies$ straight line
Poisson process $\implies$ spectrum is flat

The Semi-Experimental Method

Procedure

- Start with raw data
- Extract flow information
- Perform manipulation on data → modified traffic
- Compare with original using carefully chosen metrics
  – wavelet spectrum compact yet shows behaviour on all scales

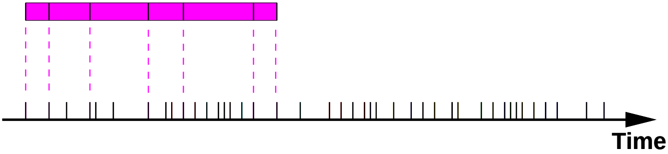
Benefits

- Understand impact of a particular feature on overall statistics
- Avoid need to model all aspects simultaneously
- Reveals physically meaningful structure

Terminology

IP flow: set of packets with same 5-tuple (plus timeout)

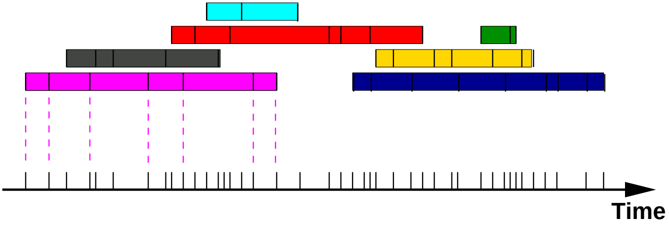
IP protocol	Source Address	Destination Address	Source Port	Destination Port



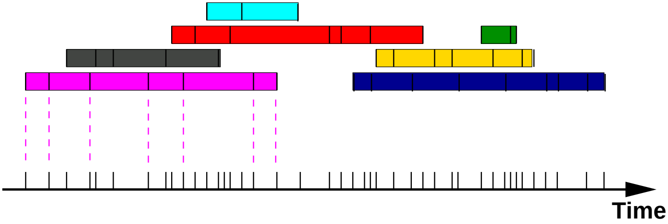
Terminology

IP flow: set of packets with same 5-tuple (plus timeout)

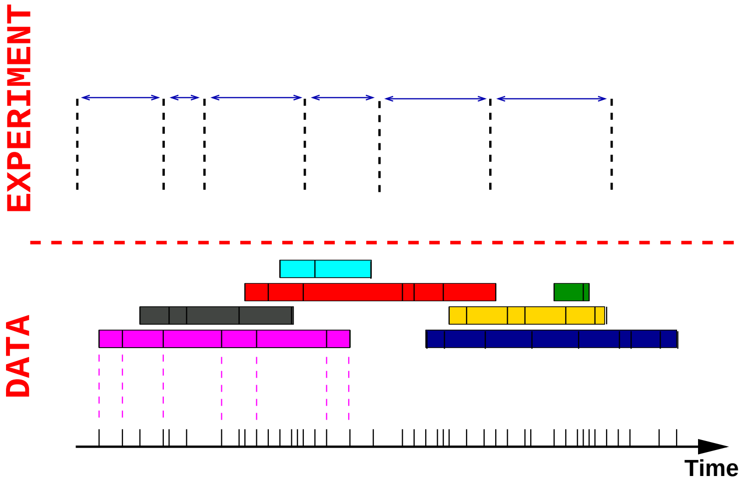
IP protocol	Source Address	Destination Address	Source Port	Destination Port



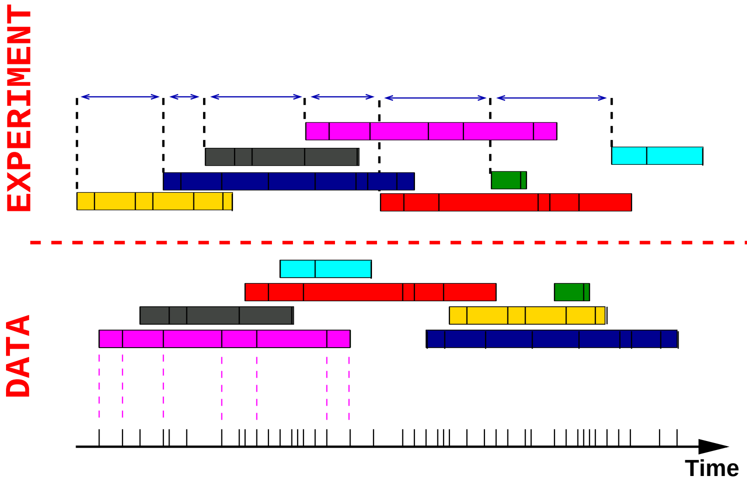
ORIGINAL DATA



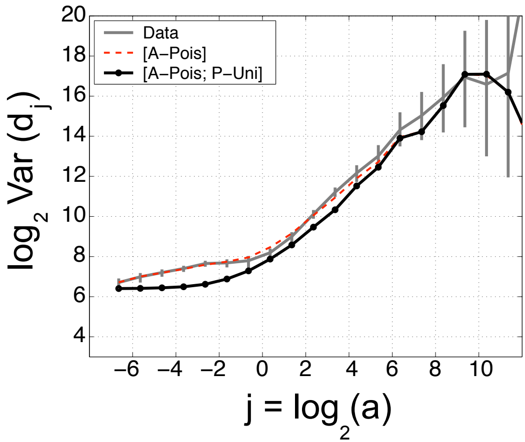
Permuted Poisson Arrivals [A-Pois]



Permuted Poisson Arrivals [A-Pois]

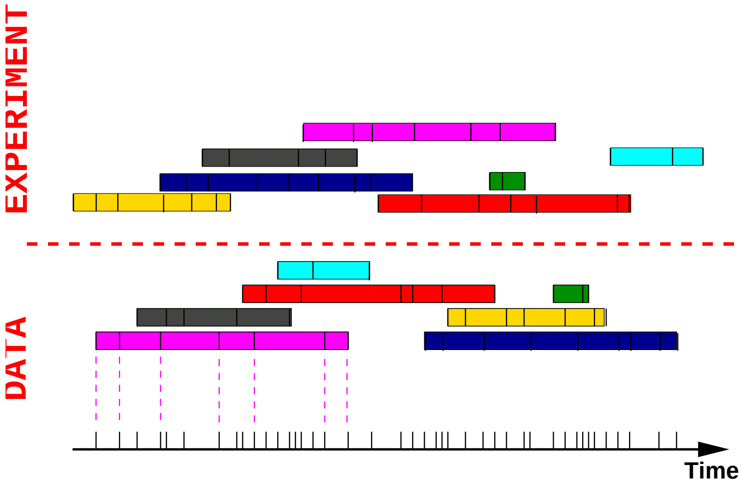


[A-Pois]: Negligible Impact!

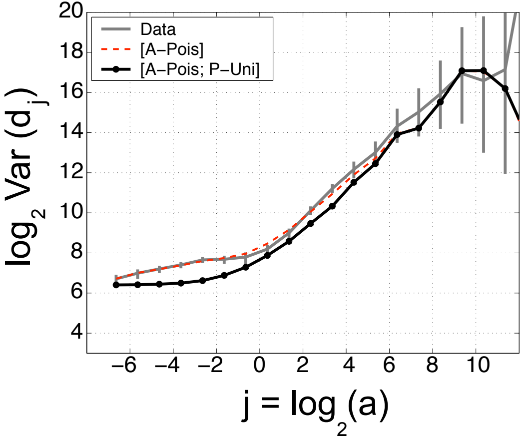


Dependencies between flows, and flow arrival details, can be ignored

Uniform In-Flow Arrivals [A-Pois; P-Uni]



[P-Uni]: Small Impact, at Small Scales



In-flow structure not critical - can replace with ‘uniform burstiness’

(but won’t here)

Poisson (Barlett-Lewis) Cluster Processes

BLPP Definition

- A Poisson process of seeds (flows), initiating independent clusters of points (packets):

$$X(t) = \sum_i \mathcal{G}_i(t - t_{\mathrm{F}}(i))$$

- Cluster: a finite renewal process with P points and inter-arrival distribution A:

$$\mathcal{G}_i(t) = \sum_{j=1}^{P(i)} \delta\left(t - \sum_{l=1}^{j-1} A_i(l) \right)$$

Parameters

- Flow arrivals: constant intensity λ
- Flow structure:
  - Packet arrivals: A, $\frac{1}{\mathbb{E}A} = \lambda_{\mathrm{A}} < \infty$
  - Flow volume: P, $\mathbb{E}P = \mu_{\mathrm{P}} < \infty$

CUBIN

Poisson (Barlett-Lewis) Cluster Processes

BLPP Definition

- A Poisson process of seeds (flows), initiating independent clusters of points (packets):

$$X(t) = \sum_i \mathcal{G}_i(t - t_{\mathrm{F}}(i))$$

- Cluster: a finite renewal process with P points and inter-arrival distribution A:

$$\mathcal{G}_i(t) = \sum_{j=1}^{P(i)} \delta\left(t - \sum_{l=1}^{j-1} A_i(l) \right)$$

Parameters

- Flow arrivals: constant intensity λ
- Flow structure:
  - Packet arrivals: A, $\frac{1}{\mathbb{E}A} = \lambda_{\mathrm{A}} < \infty$
  - Flow volume: P, $\mathbb{E}P = \mu_{\mathrm{P}} < \infty$

From Single to Multi-Class BLPP

Single Class:

- Flows are i.i.d.
- Flow arrivals Poisson: λ
- In-flow packet inter-arrivals: A, $\mathbb{E}[A] = \mu$
- Number of packets per flow: P

FROM SINGLE TO MULTI-CLASS BLPP

SINGLE CLASS:

- Flows are i.i.d.
- Flow arrivals Poisson: λ
- In-flow packet inter-arrivals: A, $\mathbb{E}[A] = \mu$
- Number of packets per flow: P

Now the talk begins...

From Single to Multi-Class BLPP

Single Class:

- Flows are i.i.d.
- Flow arrivals Poisson: λ
- In-flow packet inter-arrivals: A, $\mathbb{E}[A] = \mu$
- Number of packets per flow: P

Extend to Multi-Class, Definition:

- Flows arrivals Poisson
- Flows randomly allocated to N classes, indexed by c.
- Flow in class c with probability q_c, $\sum_c q_c = 1$
- Within class c: is BLPP with λ_c, (A_c, μ_c), P_c

From Single to Multi-Class BLPP

Single Class:

- Flows are i.i.d.
- Flow arrivals Poisson: λ
- In-flow packet inter-arrivals: A, $\mathbb{E}[A] = \mu$
- Number of packets per flow: P

Multi-Class:

- Flows are i.i.d.
- Flow arrivals: $\lambda = \sum_i \lambda_i$
- In-flow packet inter-arrivals: A doubly stochastic, μ random, $\mathbb{E}[\mu] = \sum_c q_c \mu_c$
- Number of packets per flow: P doubly stochastic

Poisson Splitting and Merging

Theorem (Merging of Poisson Streams)

The superposition of N independent Poisson processes with intensities λ_i is a Poisson process with intensity $\lambda = \sum_i \lambda_i$.

Theorem (Splitting of Poisson Streams)

If each point of a Poisson process with intensity λ is sorted independently into N subsets with probabilities p_i, $i = 1, 2, \cdots N$, then the subsets are mutually independent Poisson processes with intensities $\lambda_i = p_i \lambda$.

Poisson Splitting and Merging

Theorem (Merging of Poisson Streams)

The superposition of N independent Poisson processes with intensities λ_i is a Poisson process with intensity $\lambda = \sum_i \lambda_i$.

Theorem (Splitting of Poisson Streams)

If each point of a Poisson process with intensity λ is sorted independently into N subsets with probabilities p_i, $i = 1, 2, \cdots N$, then the subsets are mutually independent Poisson processes with intensities $\lambda_i = p_i \lambda$.

Single-Class BLPP Splitting and Merging

Splitting is flow based: packets in a flow stay together

Theorem (Merging)

The superposition of N independent BLPP processes $i = 1, 2, \cdots N$ with flow intensities λ_i and the same parameters A and P is a BLPP process with flow intensity $\lambda = \sum_i \lambda_i$ and parameters A and P.

Theorem (Splitting)

If a BLPP process with flow intensity λ and parameters A and P is randomly split into N groups with probabilities $\{p_i\}$, then the new processes are mutually independent BLPP processes with flow intensities $\lambda_i = p_i\lambda$ and parameters A and P.

Single-Class BLPP Splitting and Merging

Splitting is flow based: packets in a flow stay together

Theorem (Merging)

The superposition of N independent BLPP processes $i = 1, 2, \cdots N$ with flow intensities λ_i and the same parameters A and P is a BLPP process with flow intensity $\lambda = \sum_i \lambda_i$ and parameters A and P.

Theorem (Splitting)

If a BLPP process with flow intensity λ and parameters A and P is randomly split into N groups with probabilities $\{p_i\}$, then the new processes are mutually independent BLPP processes with flow intensities $\lambda_i = p_i\lambda$ and parameters A and P.

Multi-Class BLPP Splitting and Merging

Theorem (Merging)

The superposition of N independent BLPP processes with flow intensities λ_i and parameters $\{A_c\}$ and $\{P_c\}$ with class mixes $\{q_{c,i}\}$ is a BLPP process with flow intensity $\lambda = \sum_i \lambda_i$ and parameters $\{A_c\}$ and $\{P_c\}$ with class mix probabilities $q_c = \sum_i \lambda_i q_{c,i} / \lambda$.

Theorem (Splitting)

If a multi-class BLPP process with flow intensity λ, parameters $\{A_c\}$ and $\{P_c\}$ and class mix given by $\{q_c\}$ is randomly split into N groups with probabilities $\{p_i\}$, then the new processes are mutually independent BLPP processes with intensities $\lambda_i = p_i \lambda$ and parameters $\{A_c\}$ and $\{P_c\}$, each with the original class mix $\{q_c\}$.

Splitting & Merging can be concatenated! Node → Network

Multi-Class BLPP Splitting and Merging

Theorem (Merging)

The superposition of N independent BLPP processes with flow intensities λ_i and parameters $\{A_c\}$ and $\{P_c\}$ with class mixes $\{q_{c,i}\}$ is a BLPP process with flow intensity $\lambda = \sum_i \lambda_i$ and parameters $\{A_c\}$ and $\{P_c\}$ with class mix probabilities $q_c = \sum_i \lambda_i q_{c,i} / \lambda$.

Theorem (Splitting)

If a multi-class BLPP process with flow intensity λ, parameters $\{A_c\}$ and $\{P_c\}$ and class mix given by $\{q_c\}$ is randomly split into N groups with probabilities $\{p_i\}$, then the new processes are mutually independent BLPP processes with intensities $\lambda_i = p_i \lambda$ and parameters $\{A_c\}$ and $\{P_c\}$, each with the original class mix $\{q_c\}$.

Splitting & Merging can be concatenated! Node → Network

VALIDATION OF THE CLUSTER MODEL

Original validation over lightly load links

HERE WE:

- Test over more traces
- Wider range of utilisations and link capacities
- Focus on key Semi-Experiments, not parameter fitting
- Test if experiment outcomes as for earlier work
- Examine both aggregate streams, and substreams

Validation of the Cluster Model

Original validation over lightly load links

Here we:

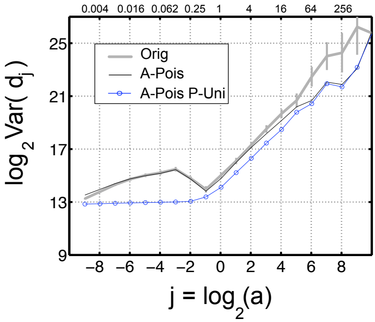
- Test over more traces
- Wider range of utilisations and link capacities
- Focus on key Semi-Experiments, not parameter fitting
- Test if experiment outcomes as for earlier work
- Examine both aggregate streams, and substreams

Input & Output Linecard Traces

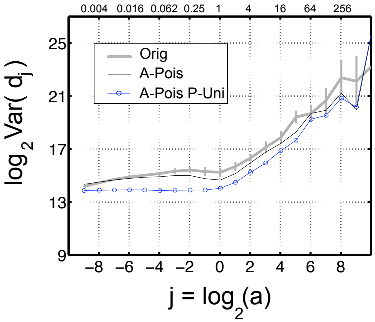
Trace	# Packets	# Flows	Bandwidth (Mbps)	ρ
C1-in	21363721	1845783	16.2	10%
C1-out	17384671	2643529	3.2	2%
C2-in	216140434	27320806	71.7	46%
C2-out	108637851	7857864	79.7	51%
C3-in	0	0	0.0	0%
C3-out	52998594	3945802	57.6	37%
C4-in	49801794	3655830	39.5	6%
C4-out	67797464	6848361	20.4	3%
BB1-in	119808388	9502484	81.2	3%
BB1-out	120286864	15742387	53.6	2%
BB2-in	126566855	11761474	78.9	3%
BB2-out	166385423	16874143	73.7	3%

VALIDATION

C3-out

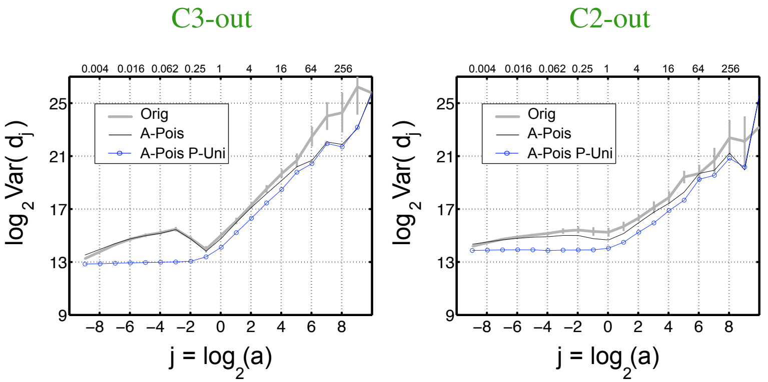


C2-out



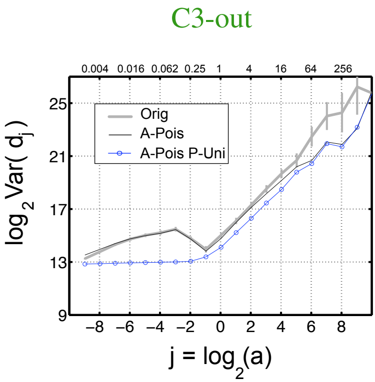
- C3-out: excellent despite $\rho = 0.37$, many worse at much lower ρ
- C2-out: very good despite highest $\rho = 0.51$
- Utilisation does not determine experiment outcome!
- Even worst cases tell same basic story

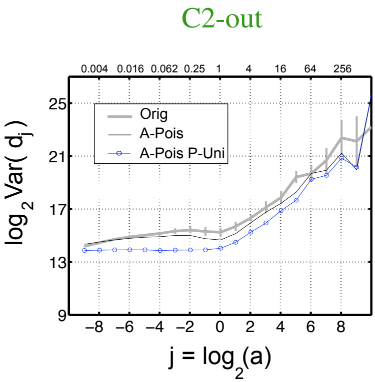
Validation



- C3-out: excellent despite $\rho = 0.37$, many worse at much lower ρ
- C2-out: very good despite highest $\rho = 0.51$
- Utilisation does not determine experiment outcome!
- Even worst cases tell same basic story

Validation





- C3-out: excellent despite $\rho = 0.37$, many worse at much lower ρ
- C2-out: very good despite highest $\rho = 0.51$
- Utilisation does not determine experiment outcome!
- Even worst cases tell same basic story

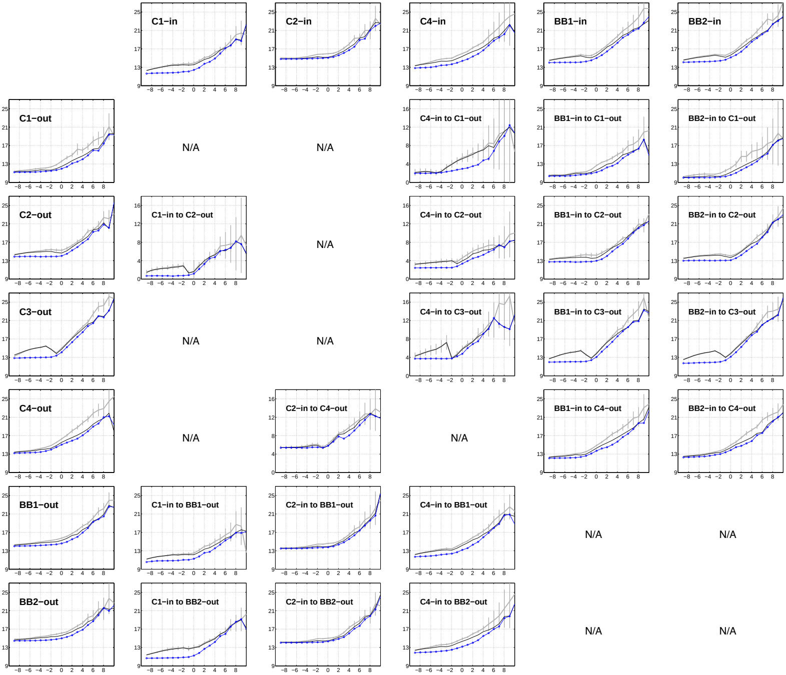
Matrix of Substreams

Substream	# Packets	# Flows	Bandwidth (Mbps)	ρ (of out link)
C1-in to C2-out	12445	1052	0.004	0.003%
C1-in to BB1-out	9664976	853932	7.0	0.28%
C1-in to BB2-out	11669672	988326	9.2	0.37%
C2-in to C4-out	300495	35445	0.05	0.008%
C2-in to BB1-out	87430709	13294988	28.6	1.2%
C2-in to BB2-out	127968526	13814708	43.0	1.7%
C4-in to C1-out	29419	2308	0.003	0.002%
C4-in to C2-out	39039	4768	0.02	0.013%
C4-in to C3-out	98359	7087	0.09	0.058%
C4-in to BB1-out	22955506	1573749	17.9	0.72%
C4-in to BB2-out	26577591	2056484	21.5	0.87%
BB1-in to C1-out	9634095	1414227	1.88	1.2%
BB1-in to C2-out	50210170	3403399	36.6	23.6%
BB1-in to C3-out	28653178	1966237	32.4	20.9%
BB1-in to C4-out	31184234	2705399	11.0	1.8%
BB2-in to C1-out	7709435	1226433	1.27	0.92%
BB2-in to C2-out	58258428	4423855	43.1	27.8%
BB2-in to C3-out	24224258	1969708	25.2	16.3%
BB2-in to C4-out	36207136	4107360	9.3	1.5%

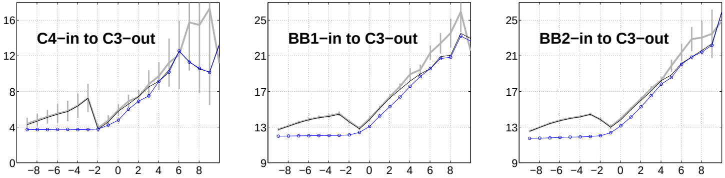
Matrix of Substreams

	C1-in	C2-in	C3-in	C4-in	BB1-in	BB2-in
C1-out				✔	✔	✔
C2-out	✔			✔	✔	✔
C3-out				✔	✔	✔
C4-out		✔			✔	✔
BB1-out	✔	✔		✔		
BB2-out	✔	✔		✔		

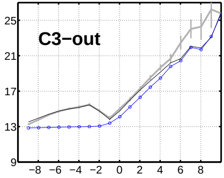
Table: Router ‘matrix’ showing packet substreams through router. Empty boxes mean no traffic.


C1−in
C2−in
C4−in
BB1−in
BB2−in
C1−out
N/A
N/A
C4−in to C1−out
BB1−in to C1−out
BB2−in to C1−out
C2−out
C1−in to C2−out
N/A
C4−in to C2−out
BB1−in to C2−out
BB2−in to C2−out
C3−out
N/A
N/A
C4−in to C3−out
BB1−in to C3−out
BB2−in to C3−out
C4−out
N/A
C2−in to C4−out
N/A
BB1−in to C4−out
BB2−in to C4−out
BB1−out
C1−in to BB1−out
C2−in to BB1−out
C4−in to BB1−out
N/A
N/A
BB2−out
C1−in to BB2−out
C2−in to BB2−out
C4−in to BB2−out
N/A
N/A

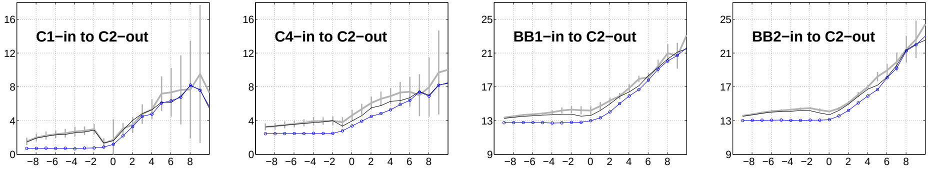
The Substreams of C3-out



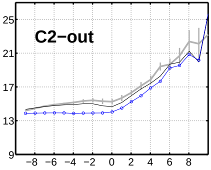
Each agrees with the model, hence C3-out does



THE SUBSTREAMS OF C2-OUT



Those that agree less well have many packets, dominate C2-out



Conclusions

- BLPP can be extended to a multi-class form
- Multi-class BLPP has splitting and merging properties
- The BLPP further validated over wider range of links
- Also validated over substreams
- Suggests multi-class BLPP for modelling router multiplexing
  $\Longrightarrow$ paradigm for (core) network traffic model
- Utilisation irrelevant for model validity
- Validity depends on substream validity, and validity upstream
- Could use Semi-Experiments as bottleneck detector

CONCLUSIONS

- BLPP can be extended to a multi-class form
- Multi-class BLPP has splitting and merging properties
- The BLPP further validated over wider range of links
- Also validated over substreams
- Suggests multi-class BLPP for modelling router multiplexing
  $\Longrightarrow$ paradigm for (core) network traffic model
- Utilisation irrelevant for model validity
- Validity depends on substream validity, and validity upstream
- Could use Semi-Experiments as bottleneck detector

CONCLUSIONS

- BLPP can be extended to a multi-class form
- Multi-class BLPP has splitting and merging properties
- The BLPP further validated over wider range of links
- Also validated over substreams
- Suggests multi-class BLPP for modelling router multiplexing
  $\Longrightarrow$ paradigm for (core) network traffic model
- Utilisation irrelevant for model validity
- Validity depends on substream validity, and validity upstream
- Could use Semi-Experiments as bottleneck detector