Self-Similarity in Ethernet [Leland94a] and the Web [Crovella97a]

(got to slide 42 on March 23)

CSci551: Computer Networks
SP2006 Thursday Section

John Heidemann

Key Ideas

- [Leland94a]
  – self-similar network traffic
    - similar at different timescales
    - definition: has infinite variance
    - consequenes: variance decays less than exponentially; …
  – looks at Ethernet
  – described math needed to check for self-sim
  – guesses at maybe why tfc is self-sim
- [Crovella97a]
  – web traffic and TCP is self-similar and has heavy tailed on-times
    - heavy tailed: distribution has infinite variance
  – shows *why* internet tfc is self-sim

Traffic Modeling

- Paxson showed microscopic traffic effects
  – ex: reordering, "little stuff", …
- What about macroscopic traffic behavior?
  – what do traffic aggregates look like?
  – conventional wisdom:
    - traffic is generated by Poisson sources
      – or at least, that's a good approximation
    - traffic will "smooth out" at large timescales

Traffic Analysis: A Bit of History

- Phone network is homogeneous and static
  – Call arrivals at trunk groups independent
  – Interarrival times exponentially distributed
  – Call durations are exponentially distributed
- why? *people* are on both ends
- so voice traffic is
  – relatively predictable
  – very amenable to mathematical analysis
    - queueing theory (EE549)
    - with care, some of these techniques can be applied to networking as well
    - examples from class: Markov modeling in Shakih paper
  – Q: will this change with more fax & data?

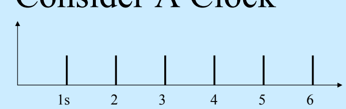
Basic Random Processes

- How to model the process arrival
  – ex. users to a computer room, or packets to a router, or
- Define an *arrival probability*
  – $a(t)$:= P[object arrives at time t] := pdf (probabilty density function)
  – $A(t)$:= P[object arrives at $t_0 < t$] := cdf (cumulative distribution function)
  – arrival rate := λ
    - $\lambda^{-1} = E[a(t)]$

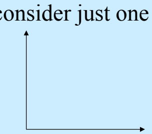
$$A(t) = \int_0^t a(t)dt$$

Consider A Clock

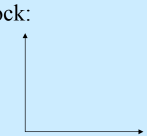
clock ticking once/second



now consider just one tick of the clock:



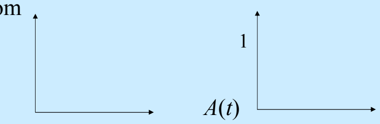
$a(t)$
P[ticks at time t]



$A(t)$
P[ticks at or before time t]

Consider A Probabilistic Clock

• assume the clock ticks uniformly randomly with ticks in the range of 0 to 2s
• note that the *rate* the clock ticks (λ) is the same as a regular clock, but exactly when the clock ticks is random

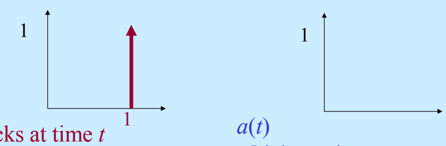


a(*t*)
P[ticks at time *t*]

A(*t*)
P[ticks at or before time *t*]

Consider A Clock and Time

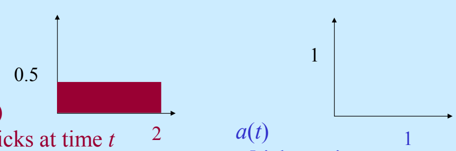
but for a clock that ticks regularly, a(t) *changes* with time (I.e., depending on how long ago the clock ticked)



a(*t*)
P[ticks at time *t* | *clock just ticked*]

a(*t*)
P[ticks at time *t* | *clock ticked ½s ago*]

A Probabilistic Clock and Time

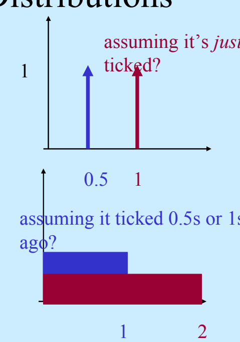


a(*t*)
P[ticks at time *t* | *clock just ticked*]

a(*t*)
P[ticks at time *t* | *clock ticked 1s ago*]

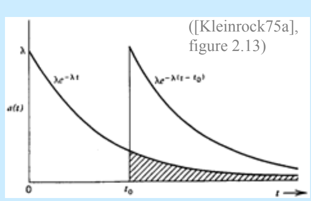
Memoryless Distributions

- in both of these examples, the distribution *changes* based on prior state
  – it has *memory*
- Is there an *a*(*t*) that doesn't change as a function of when the clock last ticked?
  – i.e., is *memoryless*



Poisson Processes

- Poisson traffic
  – $a(t) = \lambda e^{-\lambda t}$ probability next arrival is at *t*
  – λ is arrival rate (steady over time)
  – is *memoryless*
    - a(t) is the same, even after t_0

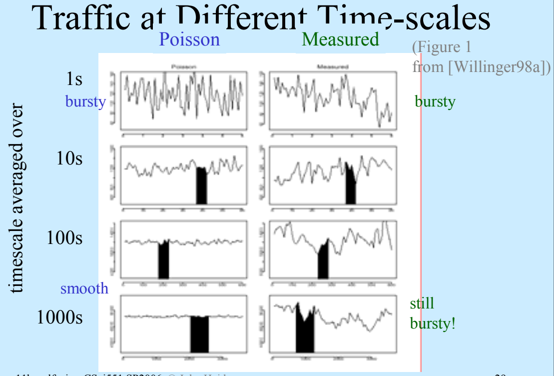


([Kleinrock75a], figure 2.13)

Advantages of Poission Modeling

- analytically tractable
  – can solve hard but relevant problems
- applicable to telephone traffic
  – (may require somewhat more complexity, but basic details are here)
- smoothes out when you combine many independent users
  – allows easier planning
- applicable to many computer problems
  – where did we see this before? xxx

Traffic at Different Time-scales



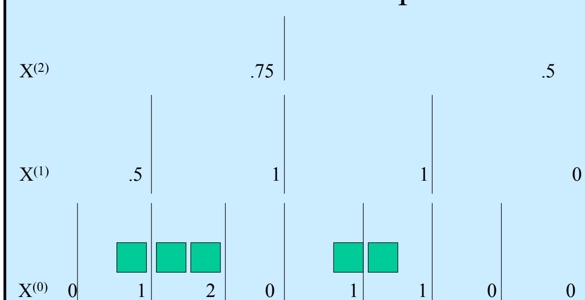
Measured Data

- shows burstiness at many different scales
  - No natural burst length
  - … unlike Poisson
- but what does this mean?
  - hard to model with Poisson
    - but will need many parameters
    - not just arrival rate λ
    - models with just simple λ may not match real net traffic
  - would prefer better model

Defining Time-scale

- Given a random variable X
  - say, pkts/s or bytes/s (or pkts/1ms)
  - let X_t be measurement at time *t*, spaced δ apart
- Define *time-scale m* recursively:
  - $X_t^{(m+1)} := X_t^{(m)} X_{t+\delta}^{(m)}$
  - basically, add things up into larger intervals
- Poisson gets *smoother* at larger *m*, but Internet traffic stays *bursty*
- Question: how can we model bursty traffic?

Time Scale Examples

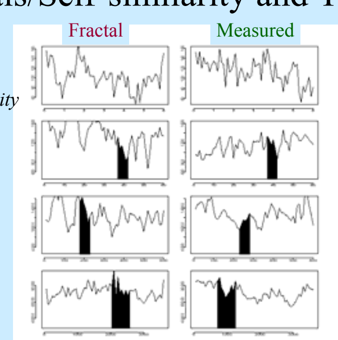


A Hint: Fractals

- Def: A class of objects with surprising scaling properties
- Example: Length of coastline depends on level of detail
- No "natural" length for these objects

Fractals/Self-similarity and Traffic

self-similarity i.e., looks like itself (bursty) at different time-scales



fractal tfc looks like measured tfc

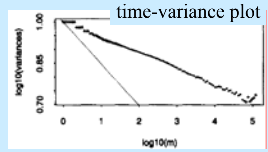
but… how can we *characterize* burstiness

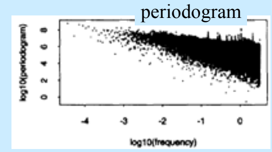
(and *why* does it happen?)

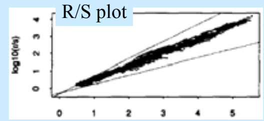
Characteristics of Self-similarity

- variance decays slowly as *m* increases
  - self-sim: Var(X(*m*)) ~ $a\, m^{-\beta}$, $0 < \beta < 1$
  - Poisson: Var(X(*m*)) ~ $a\, m^{-1}$
  - i.e., self-sim *remains bursty* (high variance)
- autocorrelation [r(*k*)] decays hyperbolically rather than exponentially
  - self-sim: $r^{(m)}(k) \sim k^{-\beta}$, where r(*k*) is autocorrelation spaced by *k*
  - Poisson: $r^{(m)}(k) \sim 0$ as $m \to \infty$
  - i.e., self-sim has *long-range dependence*
- spectral density is power-law near origin

Observing Self-Similarity





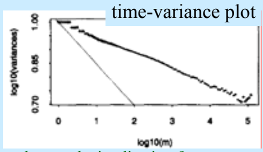


(Figure 5, [Leland94a])

general approach:
• evaluate data over multiple timescales
• measure trend, infer *H*, the *Hurst Parameter*
• somewhat subjective, so validate in multiple ways

Time-Variance Plot

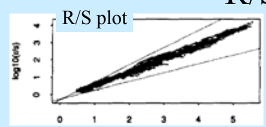
(Figure 5b, [Leland94a])



what are the implications?
• graphs are bursty, even over long averaging intervals
• probably can't design for 100% utilization
  • because bursts force either overprovisioning or high delay
  • or maybe motivates integrated services

- variance decays slowly as *m* increases
  - self-sim: Var(X(*m*)) ~ $a\, m^{-\beta}$, $0 < \beta < 1$
  - i.e., self-sim *remains bursty* (high variance)
- time-variance plot captures decay of variance
  - time (in *m*, scale) vs. variance

R/S Plot

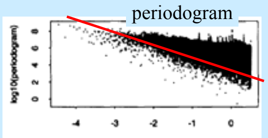


(Figure 5a, [Leland94a])

what are the implications?
• graphs are bursty and bursts sometimes persist for a long time
  • ex: new OS distribution, vs regular tfc variance
• sometimes you'll be over utilized for long periods
  • or maybe even think about protocols and apps that can reduce some big bursts?

- autocorrelation [r(*k*)] decays hyperbolically rather than exponentially
  - self-sim: $r^{(m)}(k) \sim k^{-\beta}$, where r(*k*) is autocorrelation spaced by *k*
  - i.e., self-sim has *long-range dependence*
- R/S Plot captures autocorrelation via *H* (the *Hurst parameter*)
  - H is the slope on the R/S plot
  - $H = 1 - \beta/2$

Spectral Density



- periodogram / Whittle estimator measures spectral density
- can get statistical bounds (confidence intervals) on H
- largely superceeded by wavelet analysis today
  - will see examples next time

(Figure 5, [Leland94a])

got to here 23-Mar-06

Self-Sim: The Big Picture

observations?
- look at network stuff
- file sizes
- different ways
  - dynamic measures: *how often requested, number of users requesting, etc.*
  - static: *sizes of pages in net*

causes?
- superposition of heavy-tailed processes

tfc is self-similar

implications?
- variance decays slowly
- long-range dependence
- spectral density is power-law near origin

observations?
- time variance plot
- R/S plot
- periodogram

modeling implications?
- Pareto heavy tails

modeling implications?
- Hurst parameter

network implications?
- network design,
  - utiliziation
  - long-bursts, etc

Modeling Self-Similiarty

- Can describe with *Hurst parameter, H*
  - $0.5 < H < 1$
  - $H = 1 - (\beta/2)$
- Can generate from (artificial) models
  - fractional Gaussian noise
    - parameters: mean μ, variance σ^2, autocorrelation: $r(k) = (| k+1 |^{2H} - | k |^{2H} + | k-1|^{2H}) / 2$
  - fractional autoregressive integrated moving-average process

Implications for Networking

- Queueing delays can be much higher than if all sources were Poisson
  - ⇒ Need large buffers to avoid dropping packets
  - ⇒ or accept that losses will always be possible
- Cannot determine future buffer requirements based on recent past
- Shouldn't assume Poisson traffic models represent all traffic
- But, we haven't yet looked at *why:*
  - what aspect of networks causes self-similarity

But *why* is the traffic Self-similar?

- Observation: Superposition of *heavy tailed on-off processes* are self-similar
  - and many, many things have heavy tails
    - web pages, files sizes, CPU job duration,…
  - and network traffic is sort of on-off
    - get web page, think, repeat
    - but not completely (ex. TCP dynamics)
- Suggested in [Leland94a], verified in [Crovella97a]
- Still a matter of some debate if it's the only cause

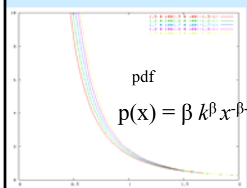
Causes for Self-Similarity

- Ethernet traffic is self-similar
- But where is this coming from?
  - *i.e.,* what physical processes contribute to self-similarity?
- Keys:
  - look for ON/OFF processes with heavy tailed durations
  - what is most Internet traffic? web or maybe p2p, over TCP

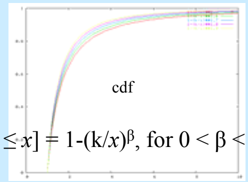
Pareto Distributions: a simple heavy-tailed distribution

- Pareto:
  - pdf: $p(x) = \beta k^{\beta} x^{-\beta-1}$
  - cdf: $P[X \le x] = x^{-\beta}$, for $0 < \beta < 2$
- when $\beta < 2$, the variance of the distribution is infinite, when $\beta < 1$, the mean is infinite
- for network traffic, $1.2 < \beta < 1.8$

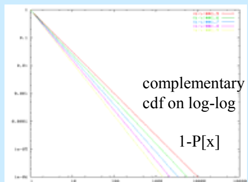
Pareto Distributions



$p(x) = \beta\, k^{\beta} x^{-\beta-1}$



P[X ≤ x] = 1-(k/x)$^{\beta}$, for $0 < \beta < 2$



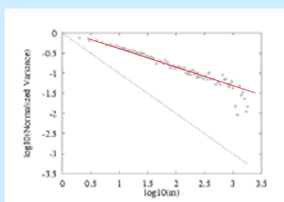
Measuring Heavy Tails

- Plot the complementary cumulative distribution function
  - P'(x) := 1 - P(x)
  - on log-log plot
- If "enough" of the tail is linear => heavy tail
  - "enough" means 3 orders of magnitude or more
  - estimate β from slope of log-log plot

$$\frac{\partial \log P(x)}{\partial \log x} = -\alpha, \text{ for } x > \theta$$

Web Traffic Self-Similar

- time-variance plot shows that the Hurst parameter is 0.76
  - web traffic at a campus site, at high enough loads, is self-similar
- confirms results of Leland et al. study
  - (Leland et al. looked at aggregate of all traffic)



([Crovella97a] figure 1)

Analyzing Web Traffic

- update browser to log request start and end times, and file sizes transferred
- then analyze the four busiest hour periods
  - more long-range dependence in the busy periods
- show that the Hurst parameter is significantly different from 0.5
  - use variance-time plots, R/S plots, periodogram

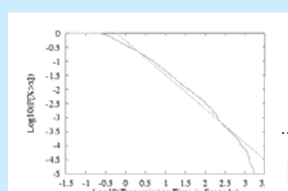
Superimposing Heavy Tails

- consider an ON/OFF process
  - with a heavy tailed distribution (parameter β_1) distribution of ON times, heavy tailed distribution of OFF times (parameter β_2), or both
- superimpose many such processes
  - Count the number of ON processes at any given time
  - this process is self-similar with Hurst parameter $H = [(3 - \min(\beta_1, \beta_2)) / 2]$
- so, is an aspect of Web tfc is heavy-tailed?

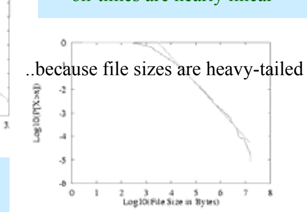
Web ON Times

what does transmit time depend on?

- on-times are nearly linear



On times are heavy-tailed…



..because file sizes are heavy-tailed

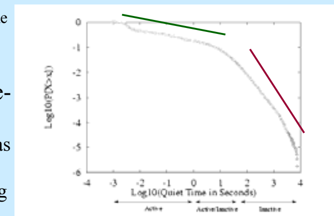
([Crovella97a] figures 3 & 4)

Why are file sizes heavy tailed?

- 1. look at network traffic, TCP connections are HT
- 2. file sizes on web servers are HT
- just the way things work (natural phenomena), but why?
  - not just file type
  - most things should work quickly, therefore have small files
  - and many people have a few big things

Web OFF Times

- Two different regimes
  - Active OFF
    - Display, rendering time
  - Inactive OFF
    - User-think time
- Crovella says inactive-off is heavy tailed
  - others have modeled as Poisson
  - ⇒ be careful interpreting graphs!



Summary

- Aggregate Ethernet traffic is self-similar
  - shown statstically
- Web transfers are heavy tailed
  - ON times are heavy-tailed because file sizes are heavy tailed
  - provides a *physical interpretation*
- General approach:
  - make statistical observation (self-sim)
  - but also find *physical interpretation* (heavy tailed files)

questions

- how does this change protocol evaluation?
  - previously: people just assumed tfc was Possion
    - enabled queueing theory and mathematical modeling of tfc
    - but could give incorrect answers for network tfc since it's not possion
  - now:
    - if answers depend on tfc model, then you should definitely consider a self-similar traffic model
      - vary object sizes
      - think about variable burstiness
    - what about effects on protocols?
  - and what do people do today? lots of things

A More Complex Story

- but traffic *cannot* be *exactly* self-similar
  - traces are finite duration
  - computers/connections are finite
  - can never really see infinite variance
- and, what happens at fine timescales?
- *instead,* traffic is *multi-fractal*
  - some behavior at fine timescales (<few RTT)
  - self-similarity at medium timescales (few RTT to hour/hours)
  - longer-term behavior dominated by outside factors (daily work cycles)
    - but *computer-originated* traffic could remove some of this in the future, maybe

Are There *Other* Causes of Self-Similarity?

- heavy-tailed on-off processes
  - but could be just *a* cause, not necessarily *the only* cause
- what about protocols?
  - chaotic TCP interactions?
  - TCP retransmit behavior (Figueiredo et al, U. Mass)
    - explains self-sim over several orders of magnitude
- does topology matter? (Feldmann et al, SIGCOMM 1999)

Traffic at High Levels of Aggregation

- what about traffic levels at very high levels of aggregation (i.e., Gb/s)
  - some recent work suggests that traffic does smooth out with *enough* users and *enough* traffic
  - intuition: each individual can only give so much burstiness, in a big enough pipe they get lost
  - certainly human generated traffic has limits (but what about computer generated traffic?!?)

Implications

- For network traffic: discussed earlier
- For modeling network traffic:
  - should use appropriate models (not just Poisson)
  - should study at *many* time-scales (not just mean and std. dev)
    - suggests *structural modeling*
      - model user arrivals, web connections, TCP details
    - has been shown to accurately reproduce Internet traffic

Other questions/observations?

- xxx