The capacity of a burst-switched network

Damon Wischik
Statistical Laboratory, Cambridge
http://www.wischik.com/damon

James Martin
—CCSR, Cambridge
Frank Kelly, Richard Gibbens
—Statistical Laboratory, Cambridge
Polina Bayvel, Michael Dueser, Alec Myers
—Electrical Engineering, UCL

Burst-switching

Suppose there are not enough wavelengths to set up a full static wavelength-routed network.

Q. How to achieve full connectivity?
—Let the ingress nodes collect packets and assemble them into *bursts*.
—Send data in bursts, instead of as a continuous flow; interleave bursts from different flows on the same wavelength.

Q. How to dynamically assign network resources? How to control burst assembly and transmission?
—Bayvel: the ingress node collects several packets then asks 'Please allocate me a lightpath so I can send a burst'.
—Qiao: the ingress node collects several packets, warns the network 'I shall shortly send a burst', then sends it without waiting for a reply.

Burst-slotting

If traffic is predictable, why not
—tell the network 'I shall have bursts to send every T seconds; please ensure I can send them'
—adjust T according to traffic
—signal a new choice of T every T^* seconds

Adaptive burst assembly

$X(s,t]$ = work arriving in $(s,t]$
B = ingress-node buffer size

Choose T to satisfy QoS constraints:

Delay constraint

$$T \leq T_{\mathsf{max}}$$

Loss constraint

$$\mathbb{P}(X(t,t+T] \geq B) \leq e^{-\gamma}$$

Adaptive burst assembly

We have designed an algorithm based on

$$-\log \mathbb{P}(X(0,t] \geq b) \approx \frac{(b-\mu t)^2}{2V_t}$$

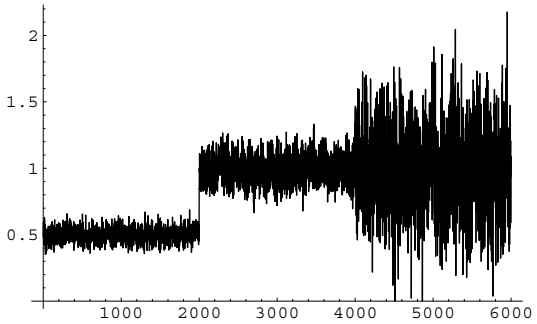
where μ=mean rate and $V_t = \mathrm{Var}\, X(0,t]$.

The algorithm measures how often the buffer contents reach a threshold b, and infers the probability of overflowing the buffer B.

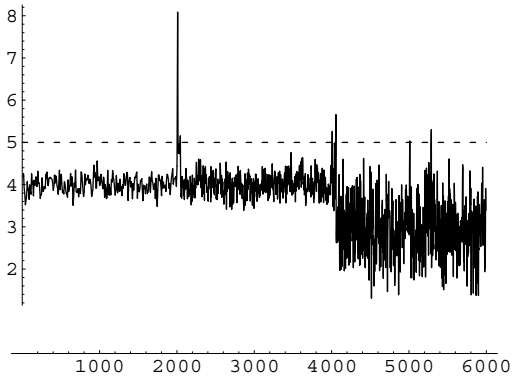
It measures μ and the overflow probability, over a given timescale T^*.

Adaptive burst assembly

Traffic (arrival rate):



Burst sizes:



—When the mean arrival rate doubles, T halves, so that the average burst size stays the same.

—When the input becomes more variable, the mean burst size decreases, to retain quality of service.

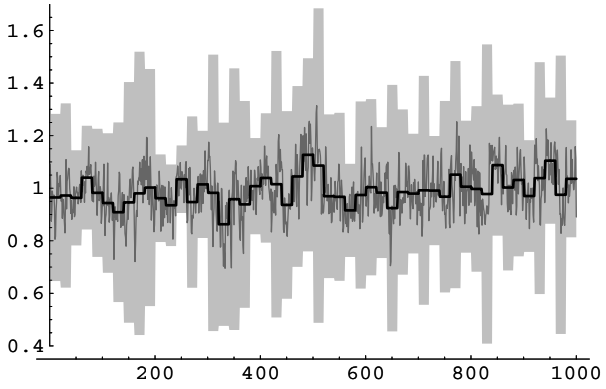
The Algorithm

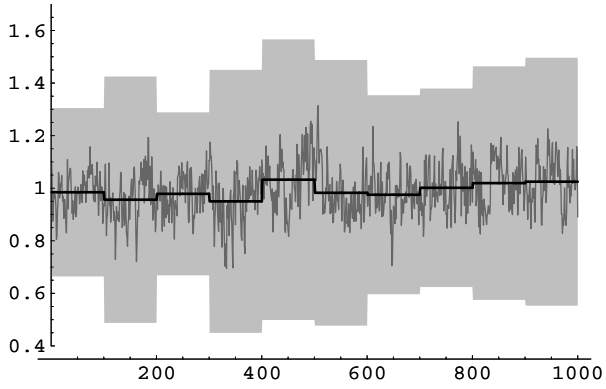
```
updateBurstFreq[B_,γtarget_,ω_][w_,t_,{μold_,γold_}] :=
  Module [ {μ,b,γ,γγ,δt,tnew},
  μ = ω^(1/t) w/t + (1 − ω^(1/t)) μold;
  If [ μt ≥ B,
    {Max[Floor[B/μ − 0.1],1],{μ, γtarget}},
  γ = If [ μold t ≥ B, 0, (μold/μ) ( (B−μ t)/(B−μold t) )^2 γold];
  b = μ t + 0.1 (B-μt);
  γγ = ( (b−μt)/(B−μt) )^2 γ;
  γγ = - Log[ω^(1/t) Indicator[w≥b] + (1−ω^(1/t)) Exp[−γγ]];
  γ = ( (B−μt)/(b−μt) )^2 γγ;
  δt = t ( (γ−γtarget)/γ ) ( (B−μt)/(B+μt) );
  tnew = Max[Round[t+0.2δt],1];
  {tnew,{μ, γ}}
  ] ]

(* B=buffer size;
   γtarget=target loss.prob;
   ω=weight for moving average;
   μold=current est. of mean rate;
   γold=current est. of loss.prob;
   w=size of most recent burst;
   t=current burst-timescale *)
```

Timescales

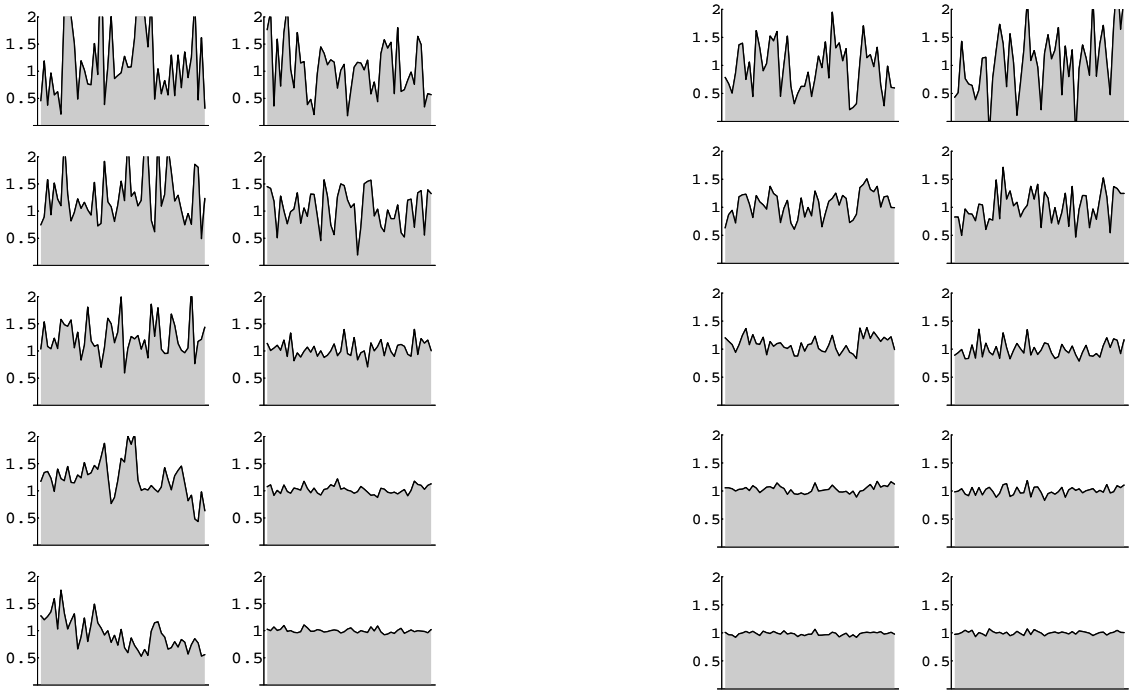
—Let T^* be the timescale of adaptation.
—Measure μ and V_T over timescale T^*.

—The larger T^*, the less often the network needs to be reconfigured.
—The smaller T^*, the smaller the estimate of V_T, and the more efficiently the network can be run.





Self-similarity



Left. Internet traffic (left) is bursty at all timescales, while a traditional Gaussian traffic model (right), matched to have the same mean and variance, is not. The top plots have the finest time-scale; each subsequent plot shows a four-fold longer time-scale.

Right. Internet traffic (left) becomes smoother as it is aggregated, as does a traditional Gaussian traffic model (right), matched to have the same mean and variance. The top plots show traffic from a single flow; each subsequent plot shows a four-fold increase in the level of aggregation.

Source: Bellcore ethernet traffic trace

Multiplexing

Consider a single link:
—L sources
—source i sends a burst every T_i seconds
—time to transmit a burst is d seconds
—Λ wavelengths available

Need

$$\Lambda \geq \sum_{i=1}^{L} d\, T_i^{-1}$$

Erlang model

Contrast to Erlang model, in which bursts:
—arrive in L Poisson streams of rate ν.
—take time $\sim$ exp(mean d/μ) to transmit.

If we require

$$\mathbb{P}(\text{burst blocked}) \leq e^{-\gamma}$$

then the maximum allowed arrival rate is

$$L\nu \leq \frac{\Lambda\mu}{d}\,\Lambda^{-\gamma/\Lambda}.$$

Contrast to burst-slotted model. If we require

$$\mathbb{P}(\text{burst-assembly buffer overlows}) \leq e^{-\gamma}$$

then

$$T \leq \frac{1}{\nu}\left(\sqrt{\mu B} - \sqrt{\gamma}\right)^2$$

and maximum allowed arrival rate is

$$L\nu \leq \frac{\Lambda\mu}{d}\left(\sqrt{B} - \sqrt{\gamma/\mu}\right)^2.$$

Notice: the benefits of buffering; the more efficient use of wavelengths.

Multiplexing model

In a simple multiplexing model:
—Λ wavelengths available
—L streams of packets
—each stream is a Poisson process of rate ν
—all packets the same size
—assembled into bursts of size B
—takes time d to transmit a burst.

If we require

$$\log \mathbb{P}(\text{burst blocked}) \leq e^{-\gamma}$$

then

$$\frac{\Lambda}{L} \geq \frac{d\nu}{B} + \sqrt{\frac{2\gamma}{L}\frac{d\nu}{B}\left(1 - \frac{d\nu}{B}\right)}.$$

Contrast to the burst-slotted model:

$$\frac{\Lambda}{L} \geq \frac{d\nu}{B} + \frac{d\nu}{B}\left(\frac{\sqrt{2\gamma}}{1 - \sqrt{2\gamma}}\right).$$

Notice: burst-slotted model makes more efficient use of the buffer; but does not benefit from multiplexing.

Networks

Adaptation. How can networks adapt to changing traffic loads?

Timescales. Over what timescale does this adaptation take place?

Capacity. What is the most traffic a network can carry? Where are the bottlenecks?

Network adaptation

Fast optical switch model. Suppose that an optical switch can reconfigure itself very quickly, and send each burst in a different direction.

Q. Given the demands T_i from all the flows, what switching schedules should all the switch follow?

Network adaptation

Reflector model. Suppose that optical switches reconfigure slowly, but that ingress-nodes can adapt quickly.

—Designate some interior nodes as *reflectors*.

—Set up a static wavelength-routed network, in which some paths are direct and others go via (a choice of) reflectors.

—Let each reflector convert incoming bursts into electronic packets, classify them, convert them back into optical bursts (possibly on a different wavelength).

Q. How to adapt, over two timescales?

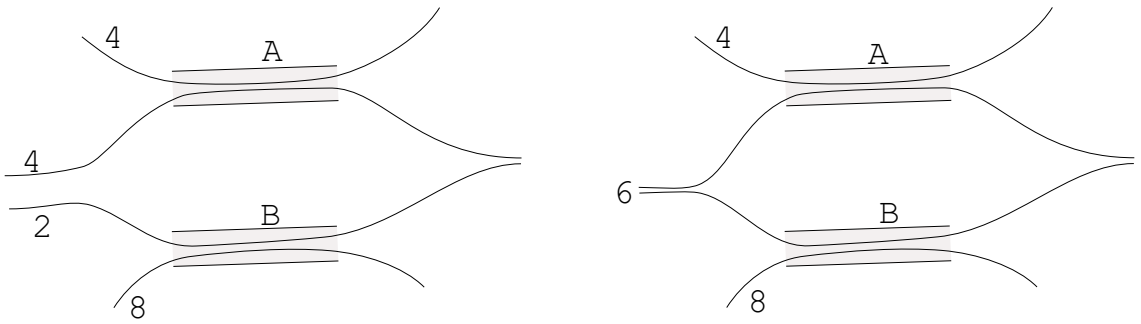
—*Long timescale:* Which nodes should be reflectors? How should the routing be done? Which paths should be direct, and which go via a reflector?

—*Short timescale:* How should an ingress node divide its traffic among reflectors?

Bottleneck cuts

Routing choice leads naturally to the idea of *bottleneck cuts*. If for any set of links C there is total traffic $\rho(C)$ which must go across C, then we need

$$\Lambda|C| \geq \rho(C) \quad \text{for all } C.$$



Left. With fixed routing, there is a single bottleneck link B, which determines
$\Lambda \geq 8 + 2$.

Right. When there is routing choice, we find a bottleneck cut $\{A, B\}$, which determines
$2\Lambda \geq 4 + 6 + 8$.

Network capacity

As demand changes, the bottleneck cut will move around the network.

The network adapts over two timescales (under the reflector model):

Slow timescale. Network should select reflectors and routes to minimize the bottleneck cut constraint.

Fast timescale. Ingress nodes should choose the proportion of flow to send via each reflector, so as to avoid the bottleneck cut where possible.

These decisions should be made using traffic measurements *taken over the respective timescales*.