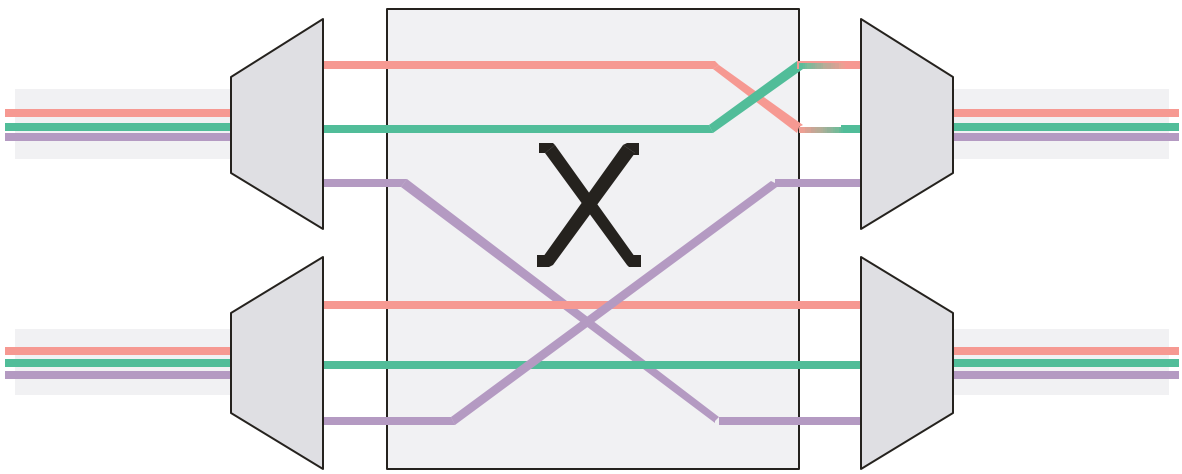
1. Optics

Optical
cross-connect



made of tiny controllable mirrors,
or with gas bubbles which can reflect light
(optionally with wavelength-converters).

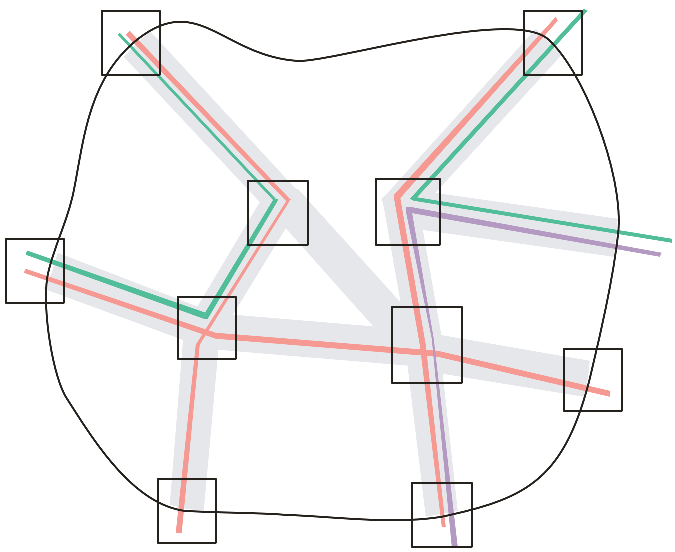
Takes milliseconds to reconfigure.

No buffers.

Inflexible —
but high capacity.

QUESTION:
What is the best way to manage an all-optical network?

2. Lightpath assignment



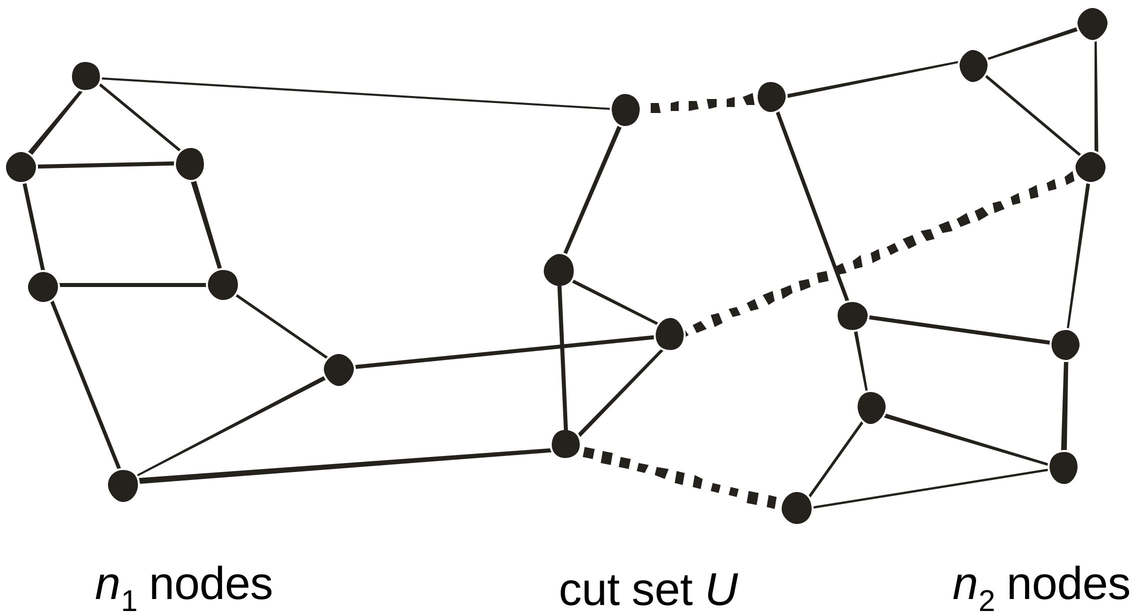
Given a graph *G=(V,E)*, |*V*|=*n*,
and a set of colours *C*:

A *lightpath* between nodes *s* and *t* is a path from *s* to *t*, with associated colour $c \in C$.

A *lightpath assignment* is a collection of lightpaths between all pairs of nodes, such that any two lightpaths of the same colour are edge-disjoint.

QUESTION:
Given a graph *G*, what is the smallest number of colours ?=|*C*| for which there exists a lightpath assigment?

BOTTLENECK CUTS



This graph requires

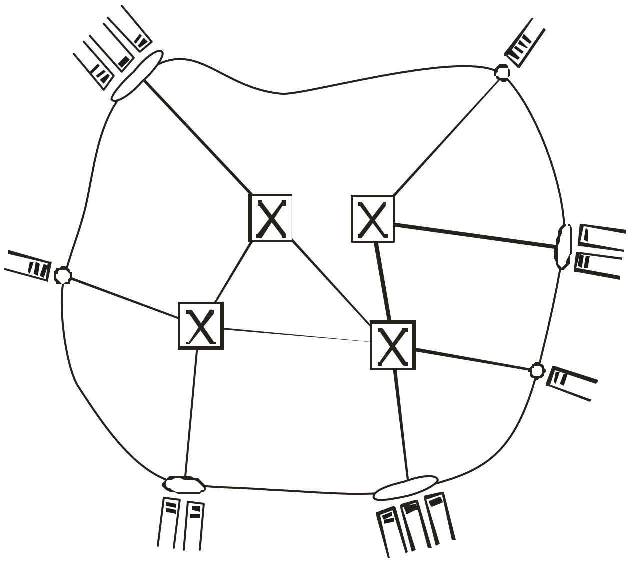
$$? = n_1\, n_2\, |U|^{-1} = 33$$

colours.

Furthermore, one can find a wavelength assignment which uses 33 colours.

For ‘real’ graphs, this often happens:
there is often a bottleneck cut which is tight.

3. Optical Packet Networks



Packets arrive from outside, and are stored in electronic buffers.

From time to time, tunable lasers beam the data Into the optical core network.

QUESTIONS: Packets and circuits? Buffers and multiplexing?

1. STATIC
   Set up permanent end-to-end lightpaths.
2. PACKET
   Send packets in, and hope they don’t collide.
3. BURST

a. Accumulate a *burst* of packets;
   signal to request a lightpath;
   once the request is acknowledged, send the burst;
   tear down the lightpath.

b. Accumulate a burst of packets;
   at time *t*, send a burst-coming notification;
   at time *t*+*e*, send the burst.

4. TIME-DIVISION-SLOTTED
   Signal to the network to say:
   ‘I will send a burst of size *B* every *T* seconds’

4. Multiplexing

EXAMPLE 1:
Consider a link with ? available wavelengths.
Let there be *L* independent arrival processes,
each a Poisson flow of packets of rate *?*,
where the size of each packet is Exp(μ).

How large can the total arrival rate be?

PACKET:
Let packets be immediately assigned a free wavelength, if one is available, and otherwise dropped.
Suppose the system is provisioned so that the probability of a drop is less than $e^{-?}$.

SLOT:
For each arrival process:
accumulate packets in a buffer of size *B*,
and every *T* seconds send out a burst of packets.
Suppose the system is provisioned so that the probability of buffer overflow is less than $e^{-?}$.

CONCLUDE
PACKET shows slightly more multiplexing gain;
SLOT shows much more buffering gain.

4. Multiplexing

EXAMPLE 2:
Consider a link with ? available wavelengths.
Let there be L_2 independent arrival processes,
each a Poisson flow of packets of rate $L_1?$,
where the size of each packet is 1.

How many wavelengths do we need?

BURST:
Accumulate packets in a buffer of size $L_1 B$,
and whenever it fills, send out a burst of size $L_1 B$.
Suppose the system is provisioned so that the
probability that bursts collide is less than $e^{-?}$.

SLOT:
For each arrival process:
accumulate packets in a buffer of size $L_1 B$,
and every T seconds send out a burst of packets.
Suppose the system is provisioned so that the
probability of buffer overflow is less than $e^{-?}$.

CONCLUDE
BURST benefits from L_1 multiplexing,
SLOT benefits from L_2 multiplexing;
SLOT shows much more buffering gain.

4. Multiplexing

CONCLUSION

Networks typically deal with variable input traffic using

- buffers
- multiplexing.

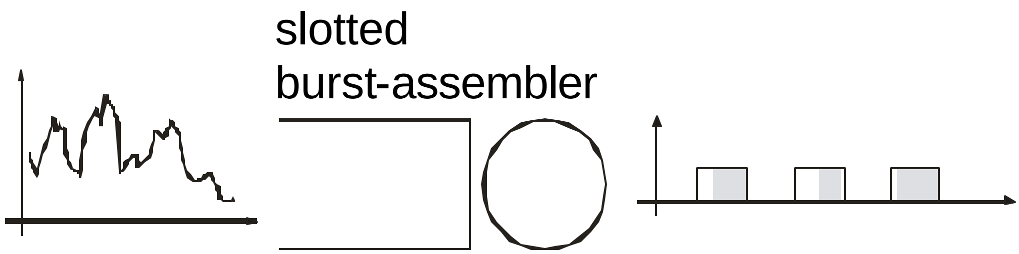
An optical network has

- a flexible (electronic) boundary, where buffers and multiplexing can be exploited
- an inflexible (optical) core, with no buffering and limited multiplexing ability.

This suggests we design the core to work like a circuit-switched (possibly TDM) network.

(Still, packet networks are far easier to manage!)

5. Edge-buffers



Suppose the input process is stationary.
Let $A(t)$ be the amount of work arriving in $(0,t)$.
Let $\mu = \mathrm{E}\, A(t)/t$ and $V(t) = \mathrm{Var}\, A(t)$.

Suppose the burst-assembler has a buffer of size B,
and emits a burst every T seconds.

Seek to adaptively set T,
to ensure that the probability of buffer overflow is small.

Using $\log P(\text{overflow}) \sim -(B-\mu T)^2 / 2V(t)$,
a simple algorithm gives: