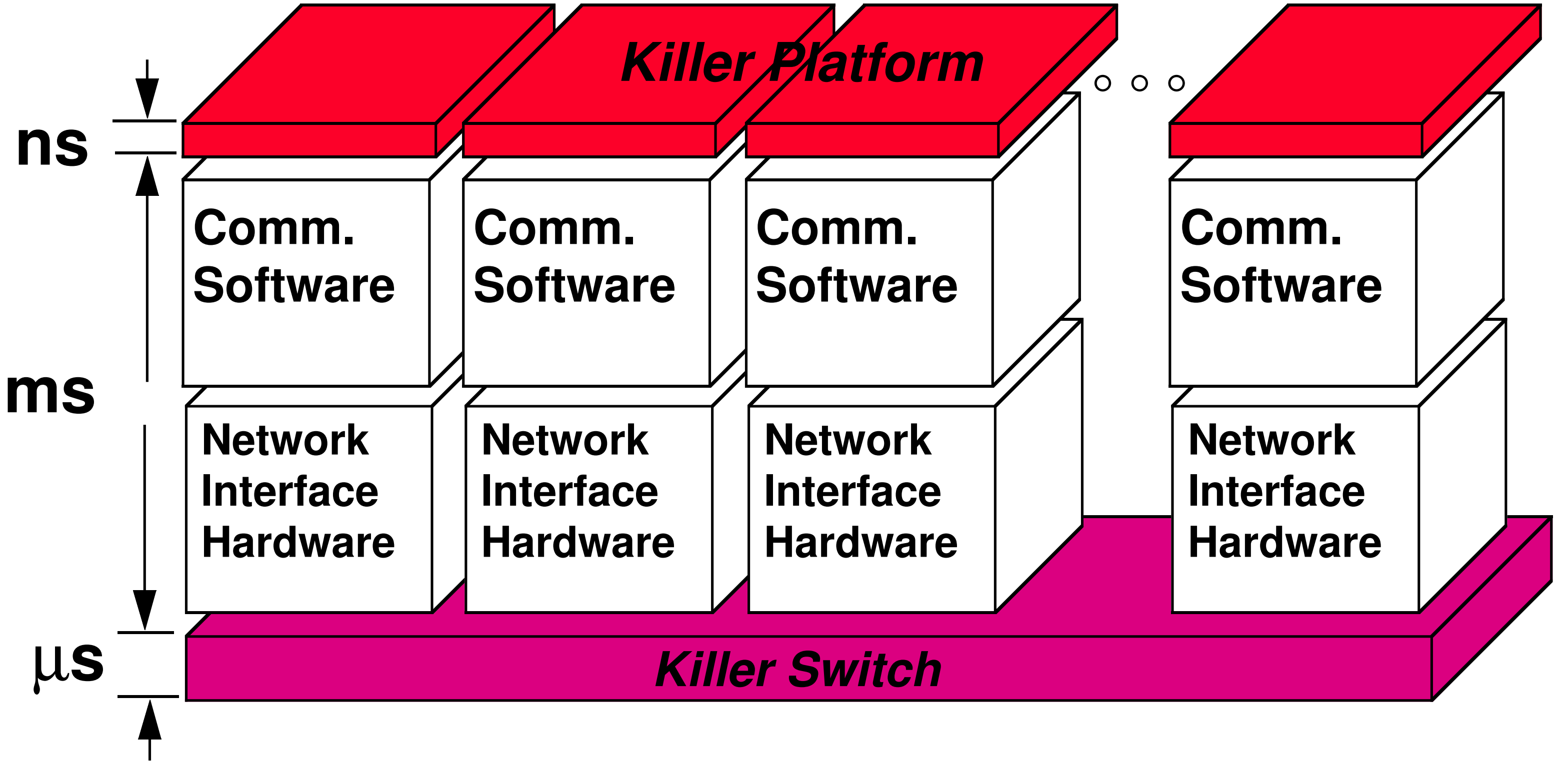
Effects of Communication Latency, Overhead, and Bandwidth in a Cluster Architecture

David E. Culler
Rich Martin, Amin Vahdat
Computer Science Division
University of California, Berkeley

The Fast Communication Problem

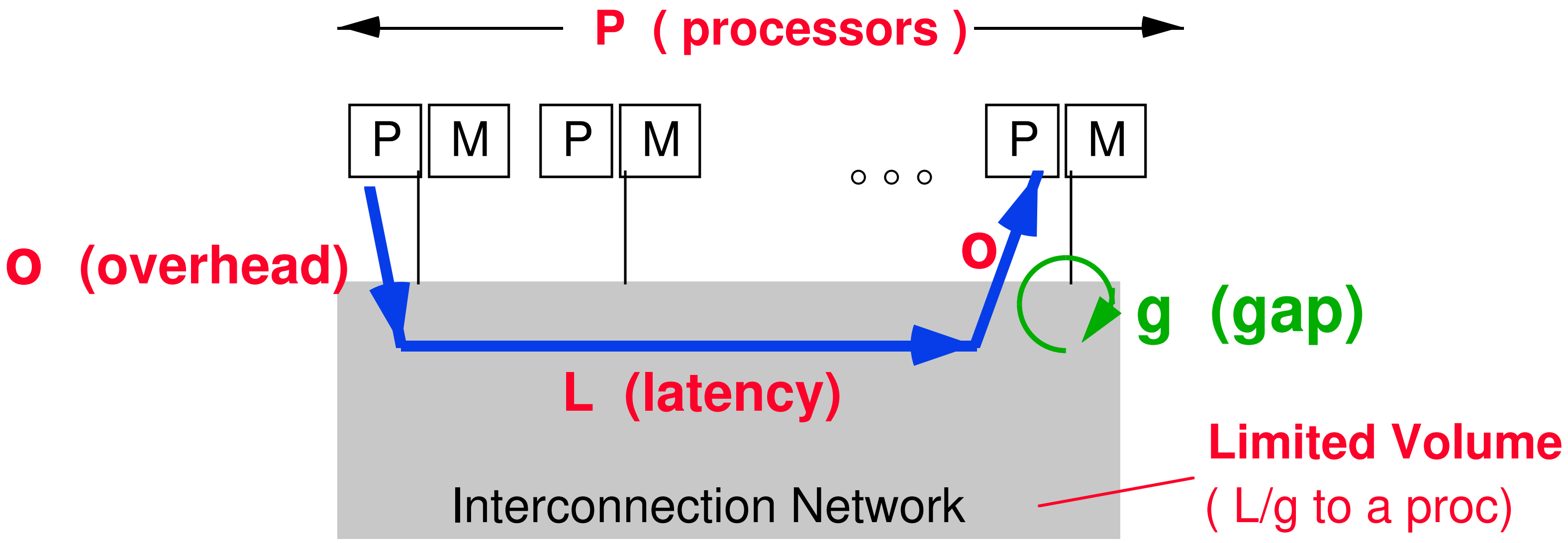


- **NOW Goal: 10 μs end-to-end in general purpose, high-performance clsuters**

Question of the Study

- How much do our efforts really improve application performance?
- Have we gone far enough?
- To what aspects of communication performance are applications most sensitive?
- Can we predict it?

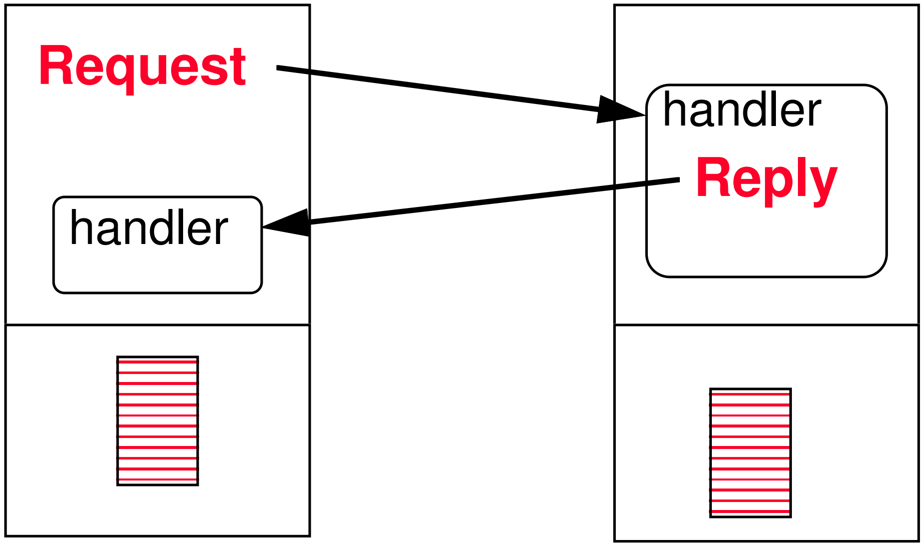
Model Framework: LogP



- Latency in sending a (small) message between modules
- overhead felt by the processor on sending or receiving msg
- gap between successive sends or receives (1/rate)
- Processors

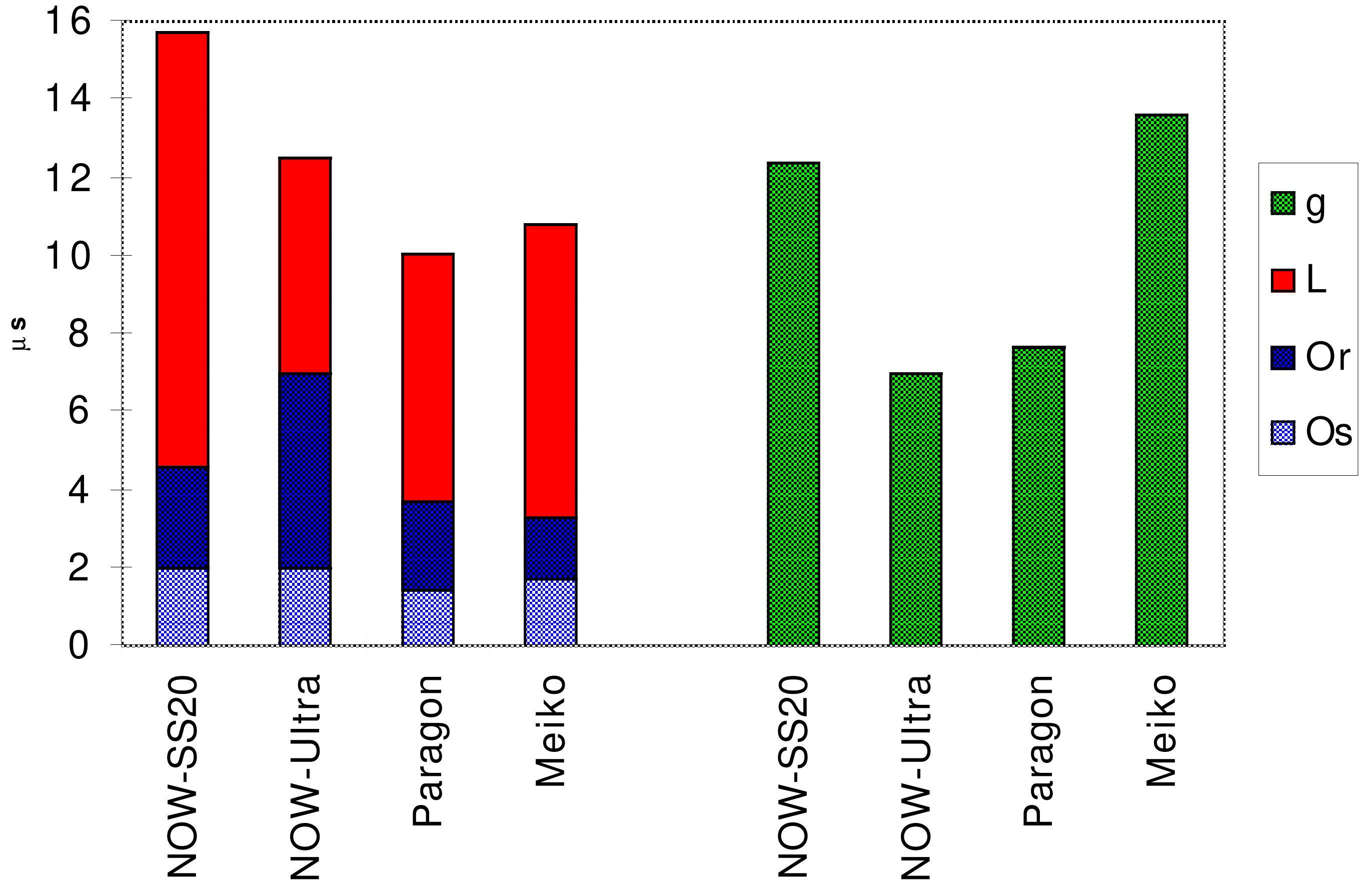
Generic Active Messages (GAM)

- **Highly optimized communication layer on a range of HW**
- **Request / Reply small active messages (RPC)**
- **Bulk-Transfer (store & get)**



Round Trip time: 2 x (2o + L)

LogP Summary of Current Machines



Max MB/s: 38 141 47

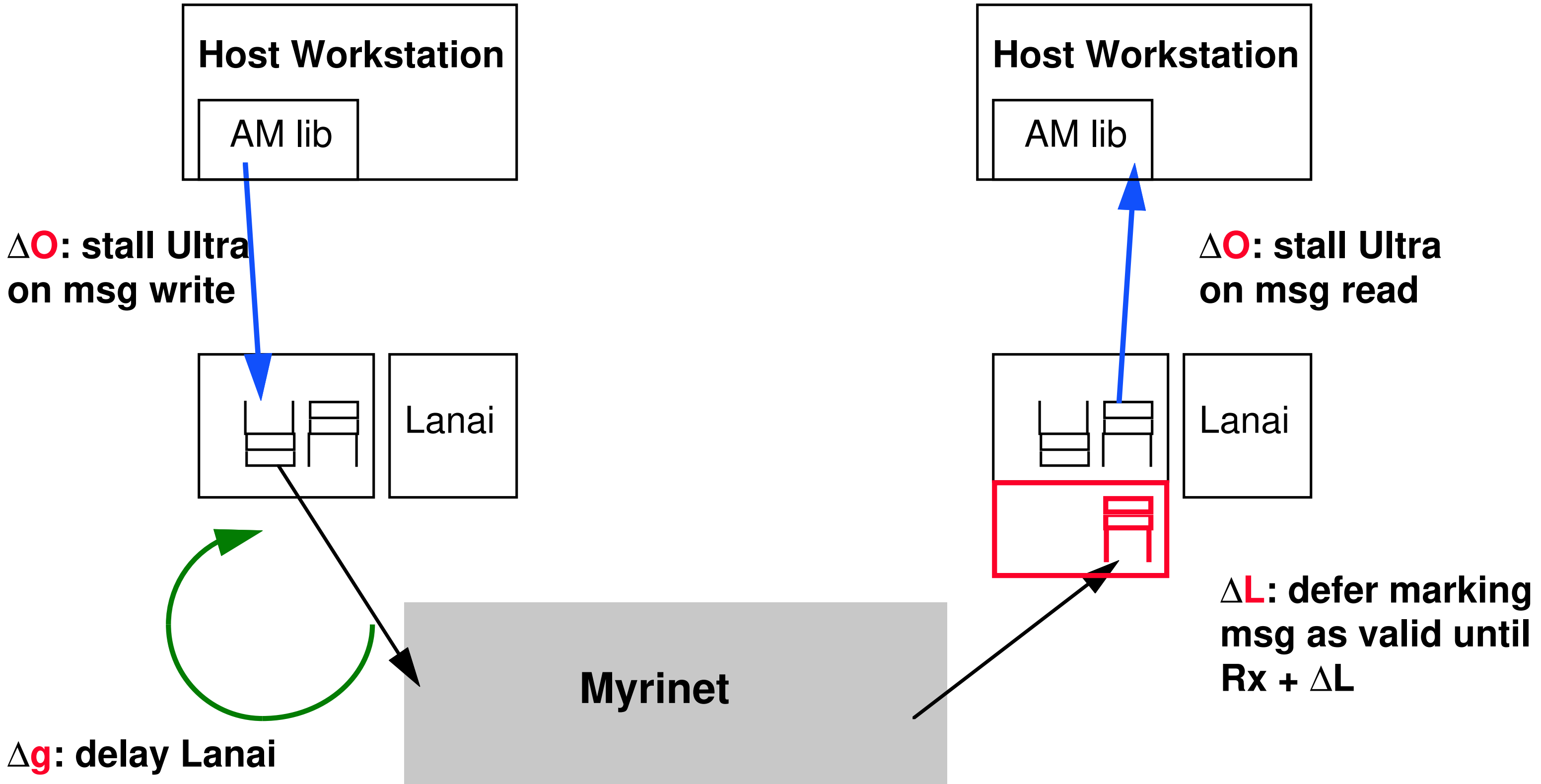
Outline of the Talk

- Framework for Communication Performance
- Empirical Method
- Base Application Characteristics
- Sensitivity to Overhead
- to gap
- to Latency
- to bulk transfer bandwidth
- Predictive models
- Conclusions

Methodology

- **Apparatus:**
  - 35 Ultra 170s (64 MB, .5 MB L2, Solaris 2.5)
  - M2F Lanai + Myricom Net in Fat Tree variant
  - GAM + Split-C
- **Modify the Active Message layer to inflate L, o, g, or G independently**
- **Execute a diverse suite of applications and observe effect**
- **Evaluate against natural performance models**

Adjusting L, o, and g (and G) *in situ*



ΔO: stall Ultra on msg write

ΔO: stall Ultra on msg read

ΔL: defer marking msg as valid until Rx + ΔL

Δg: delay Lanai after msg injection

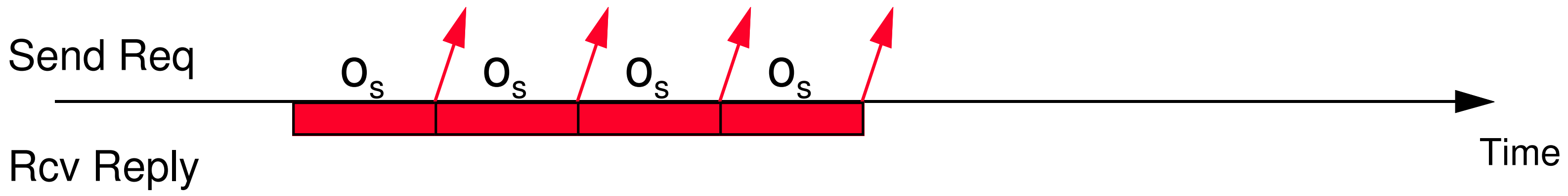
(after fragment for bulk transfers)

Microbenchmarks for Calibrating the Empirical Adjustments

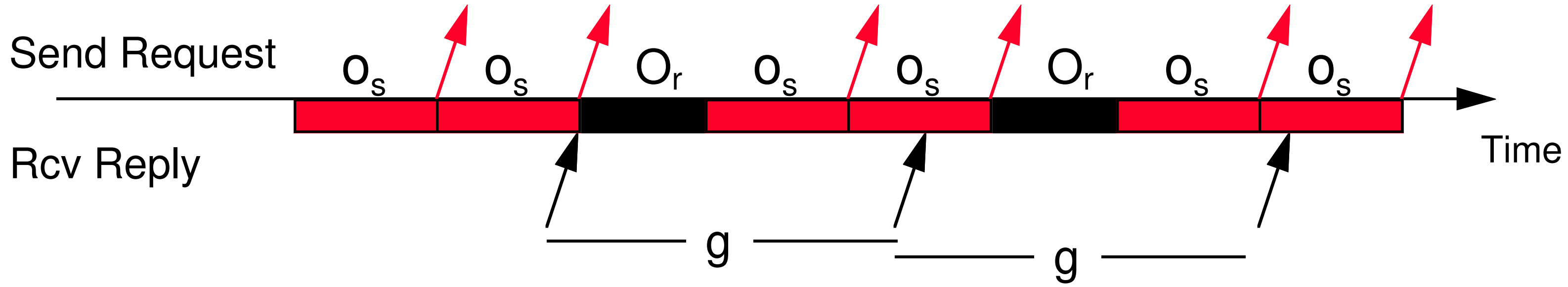
```
For Range of $\Delta$ and $M$

    Repeat $M$ times              Time($M$, $\Delta$)
        Issue request
        Compute for $\Delta$

    ... things complete
```

Time to Issue Sequence

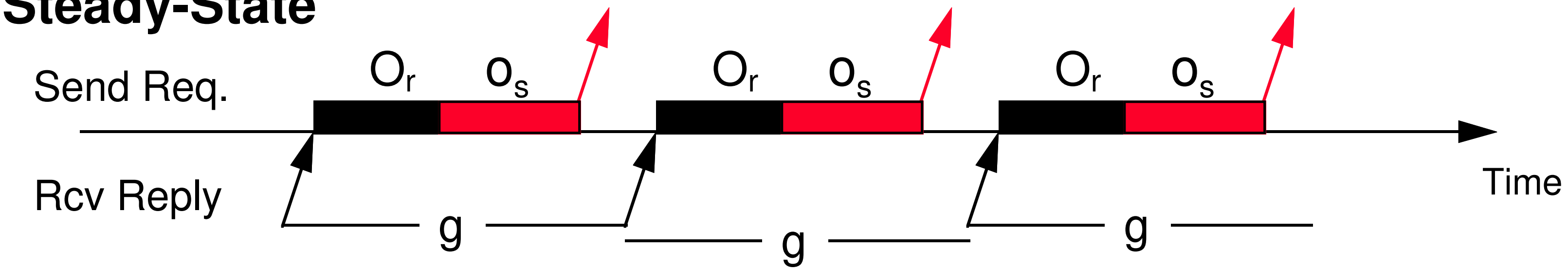
Send Only



Transition

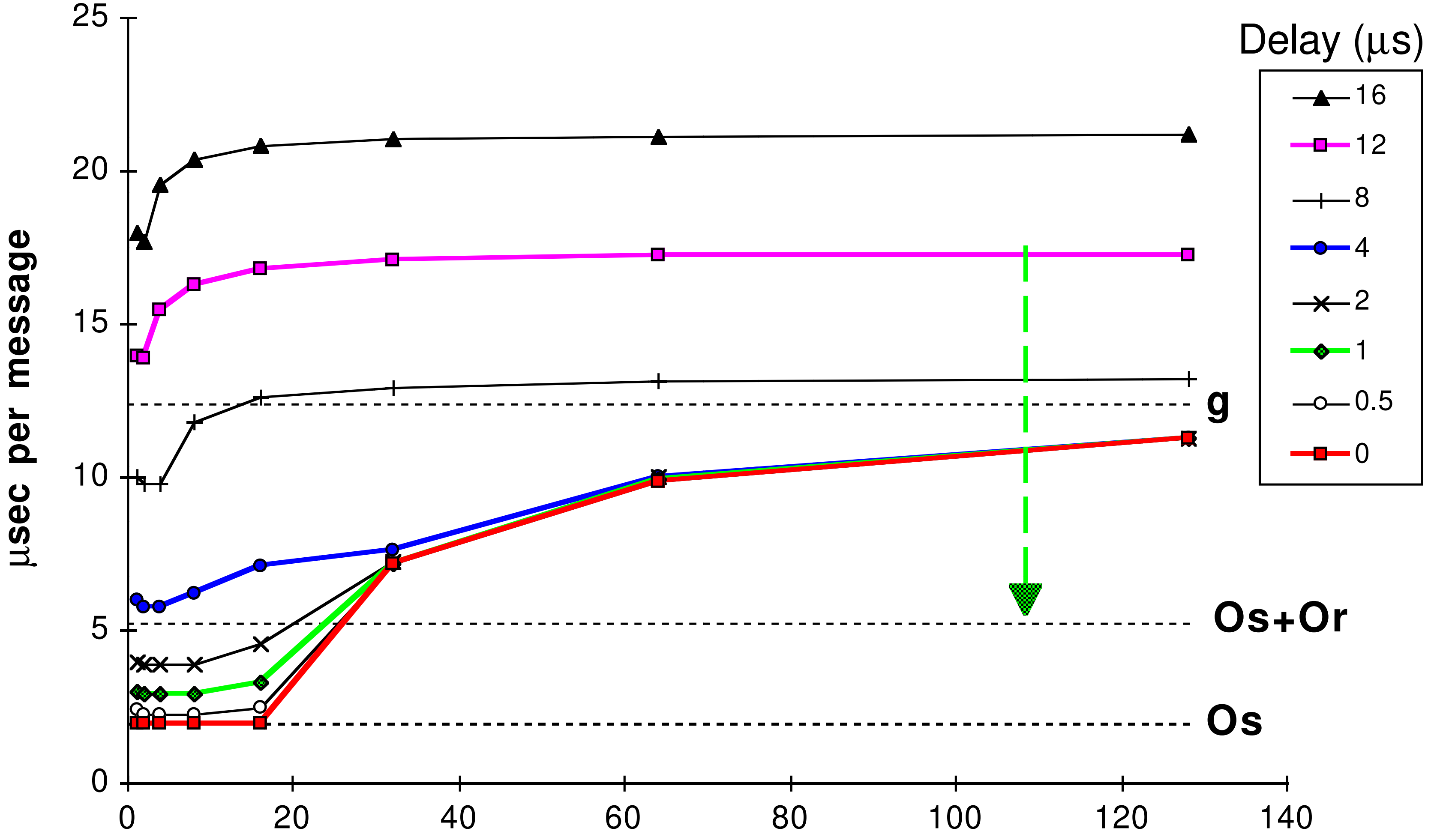


Steady-State



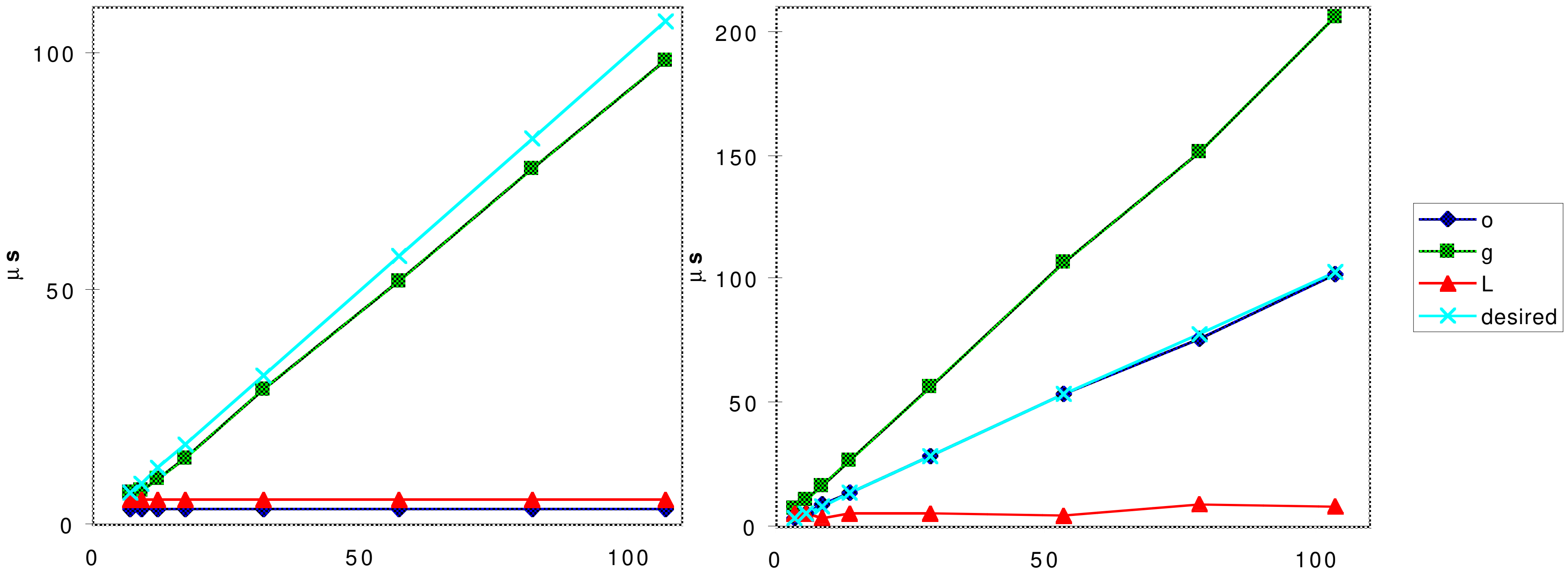
NOW-SS20 Microbenchmark Signature

L = 11.1 µs, Os = 2.0 µs, Or = 2.6 µs, g = 12.4 µs

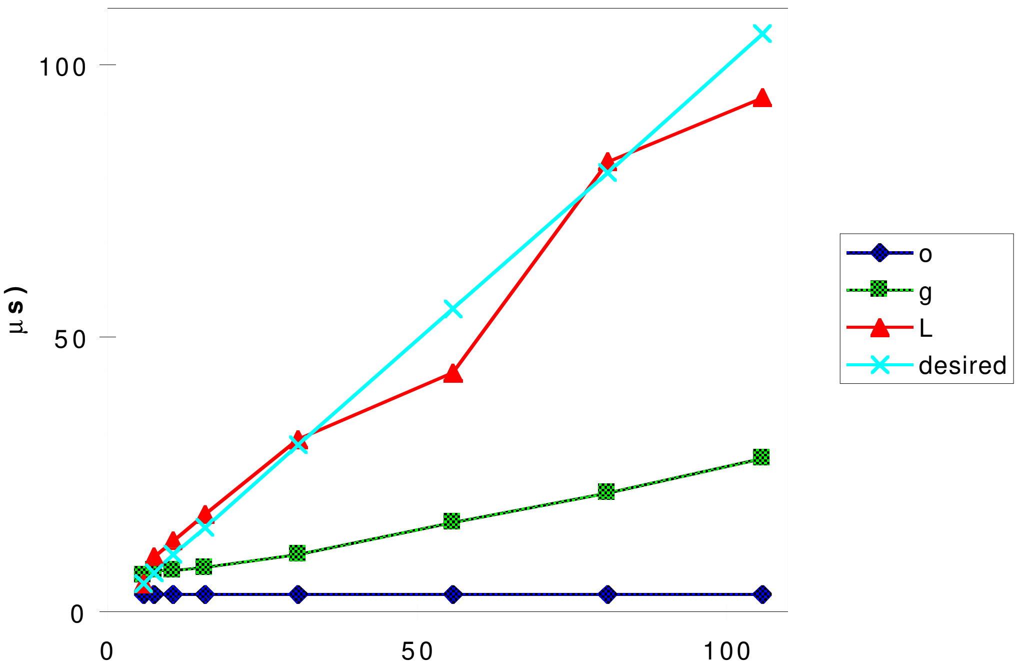


Credit: Richard Martin

Calibration



Calibration (cont)



Applications Characteristics

- Message Frequency
- Write-based vs. Read-based
- Short vs. Bulk Messages
- Synchronization
- Communication Balance

Applications used in the Study

Program	Description	Input	Time(16)		Msg Inte
radix	Int Radix Sort	16 M 32-bit keys	17.0 s	9.8 s	7.6
em3d	Int Sample Sort	32 M 32-bit keys	101 s	44.3 s	10.2
sample	EM Wave Prop.	8 K nodes, deg. 10, 100 steps	24.3 s	15.9 s	14.0
ebarnes	Hierarchical N-body	1 M bodies	81.2 s	46.6 s	60.1
p-ray	Ray Tracer	1 M pixel image, 16 k objs	25.8 s	17.8 s	108.1
murphi	Protocol Verifier	SCI, 2 procs, 1 line, 1 mem	71.3 s	37.9 s	219.4
connect	Connected Components	4 M nodes, 2D, 30 %	2.5 s	1.45 s	282.9
radb	Bulk Radix	16 M 32-bit keys	6.5 s	3.95 s	1260.0

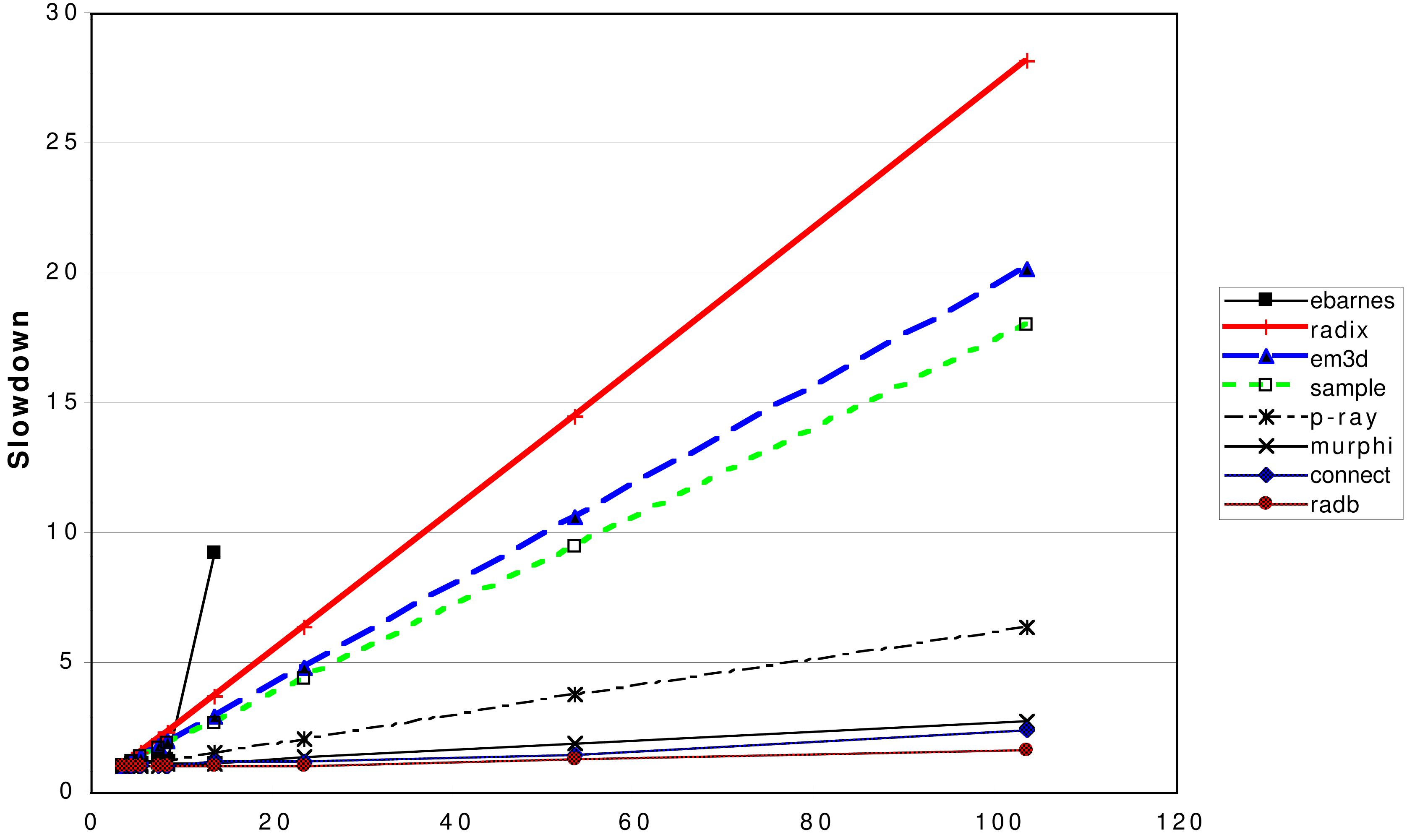
Baseline Communication

Program	µs/msg		Avg Msg/Proc	Max Msg./Proc	Reads	Bulk Msg
radix	7.6	895	2,228,364	2,229,106	0.0%	0.0%
em3d	10.2	324	9,953,384	9,974,265	0.0%	0.0%
sample	14.0	2499	1,966,199	2,319,362	0.0%	0.0%
ebarnes	60.1	233	1,351,194	1,400,601	9.6%	23.5%
p-ray	108.1	1465	216,869	353,640	47.7%	47.9%
murphi	219.4	36054	328,699	332,054	0.0%	51.2%
connect	282.9	101	8,912	9,221	34.1%	0.1%
radb	1260.0	80	5,520	5,927	0.0%	43.7%

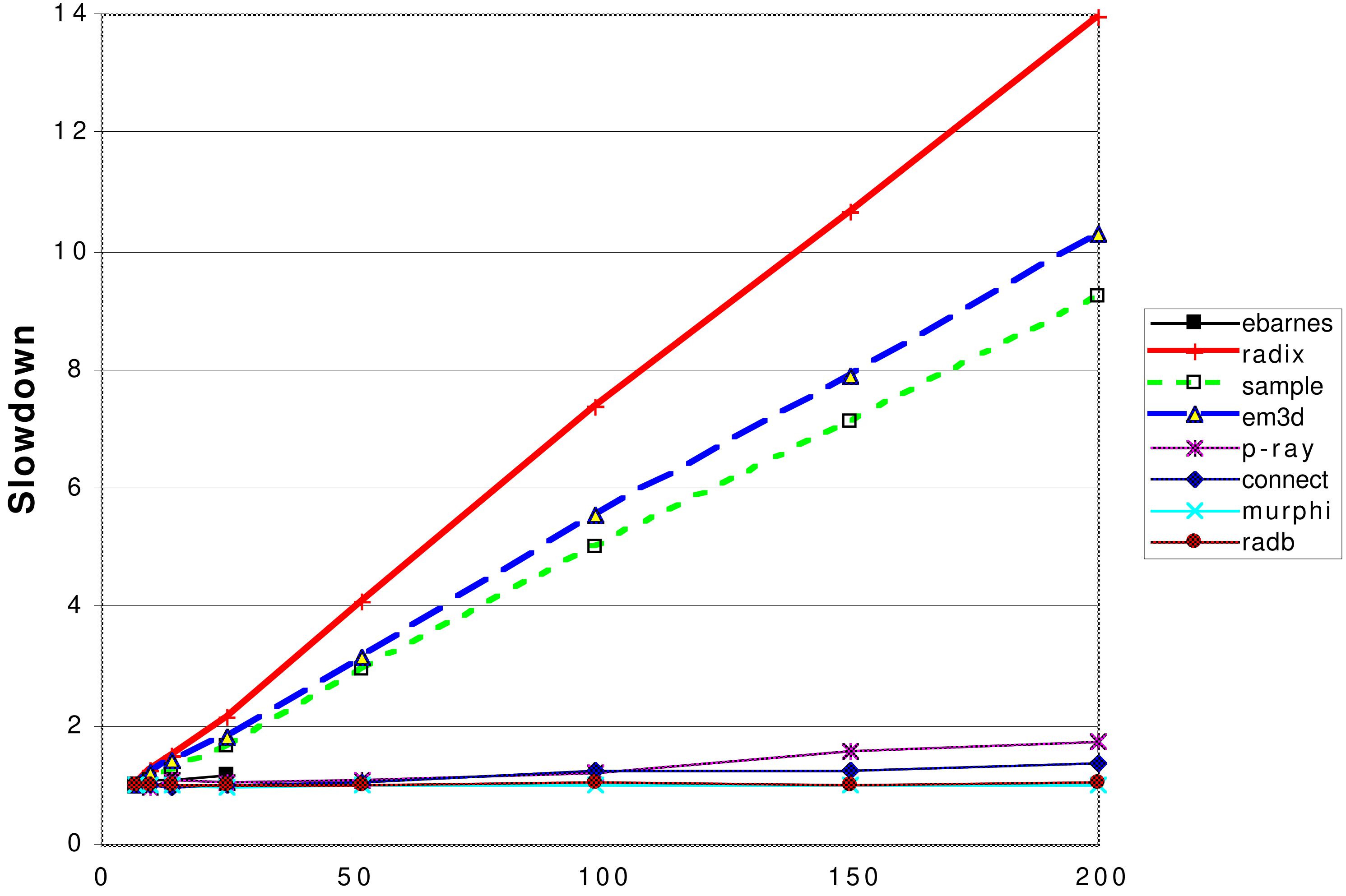
Communication Load Balance

Application Sensitivity to Communication Performance

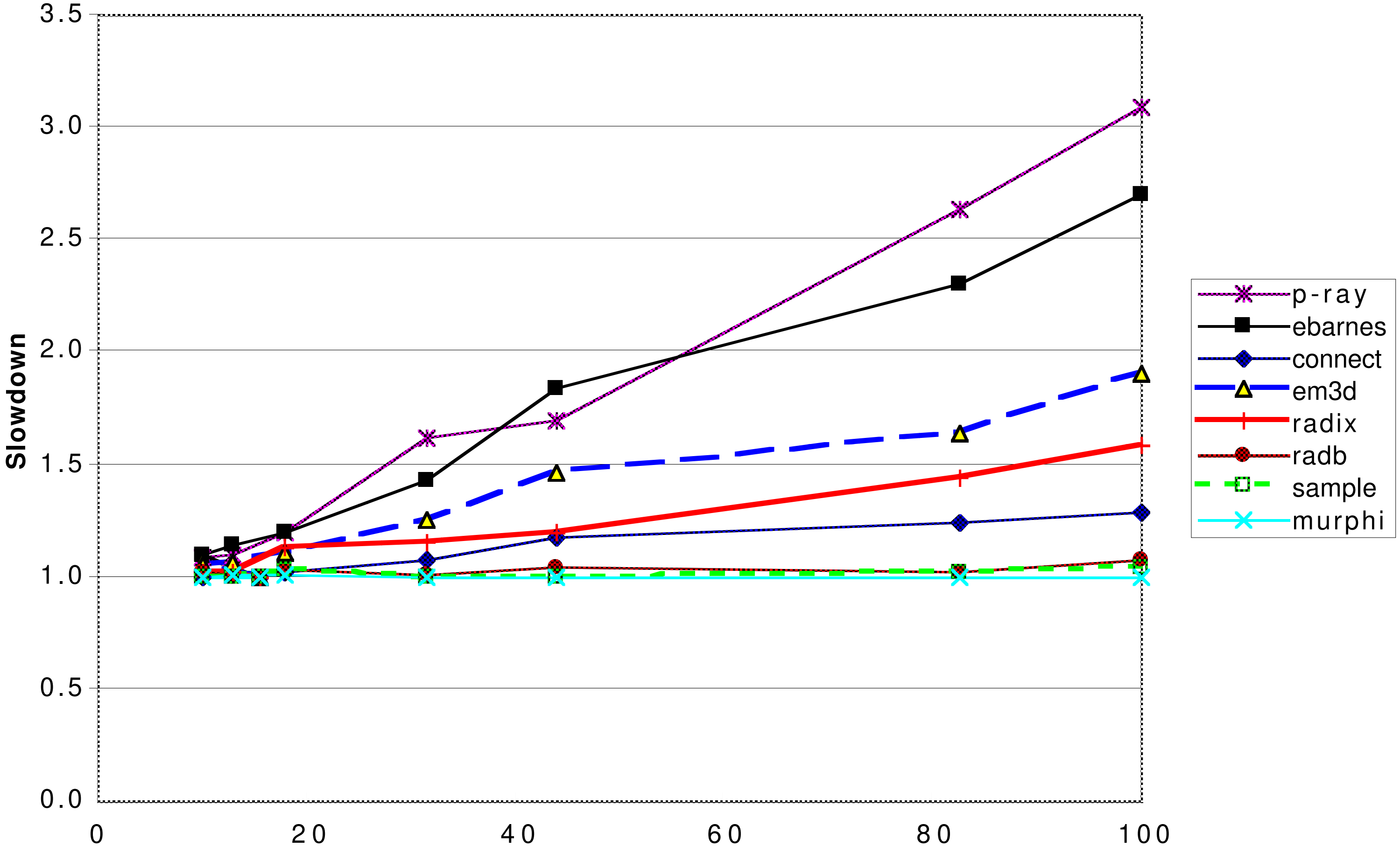
Sensitivity to Overhead



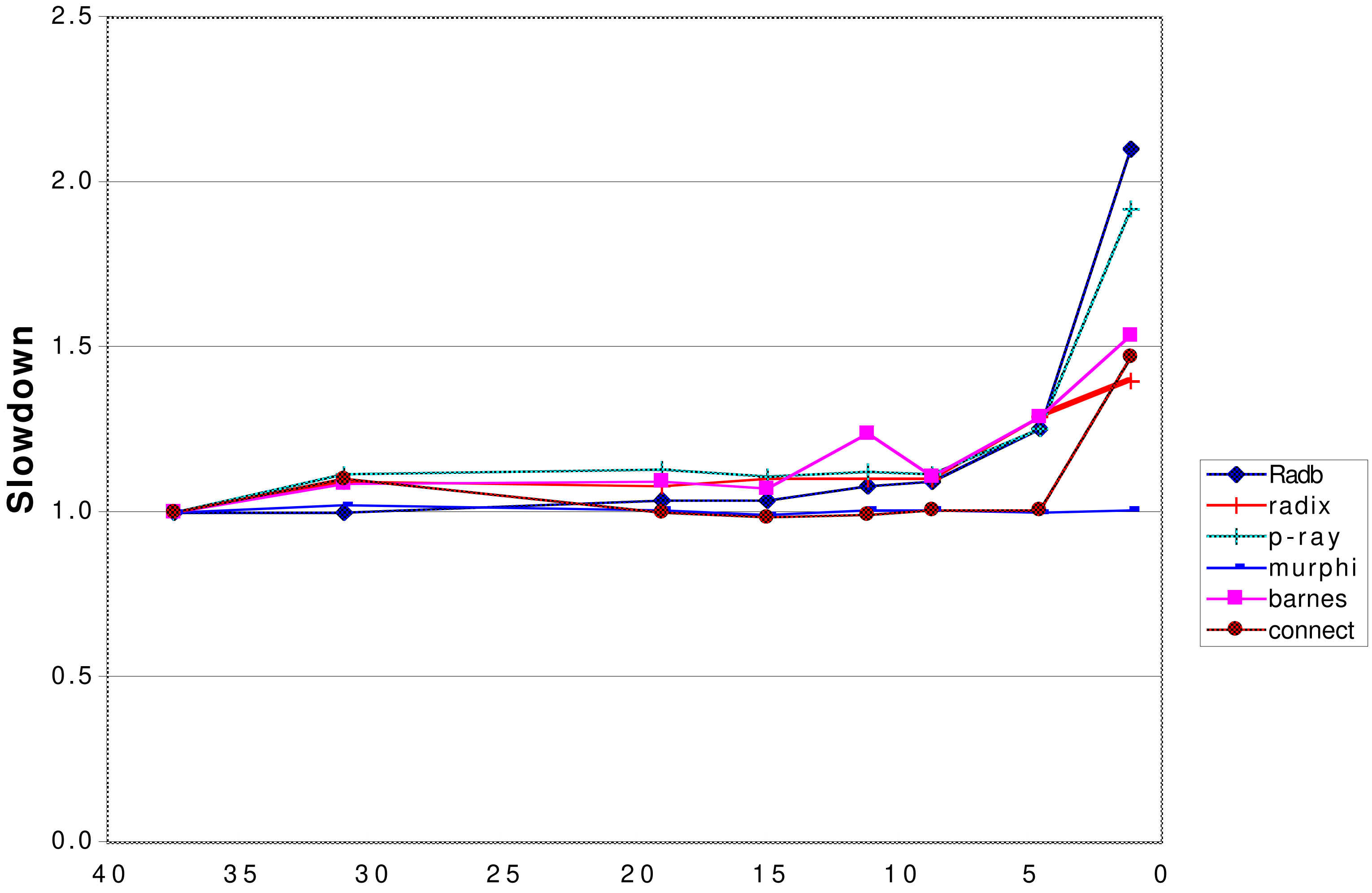
Sensitivity to gap (1/msg rate)



Sensitivity to Latency



Sensitivity to bulk BW (1/G)



Modeling Effects of Overhead

- $T_{pred} = T_{orig} + 2 \times \text{max\#msgs} \times \Delta o$
  - request / response
  - proc with most msgs limits overall time

Δ Overhead (µs)	radix	sample	em3d	ebarnes	p-ray	connect	murphi	radb
0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
1	1.03	1.00	1.14	1.02	1.03	1.02	1.00	1.01
2	1.02	1.00	1.26	1.00	1.04	1.04	1.02	1.00
4	1.02	1.00	1.50	0.90	1.07	1.07	1.04	1.00
5	1.01	0.99	1.59	0.88	1.06	1.12	1.03	1.00
10	1.01	0.99	2.01	4.83	1.21	1.30	1.09	0.99
20	1.02	0.98	2.53	N/A	1.31	1.32	1.16	0.99
50	1.03	0.98	3.22	N/A	1.50	1.60	1.30	1.02
100	1.04	0.98	3.61	N/A	1.58	1.85	1.46	1.07

- Why does this model under-predict?

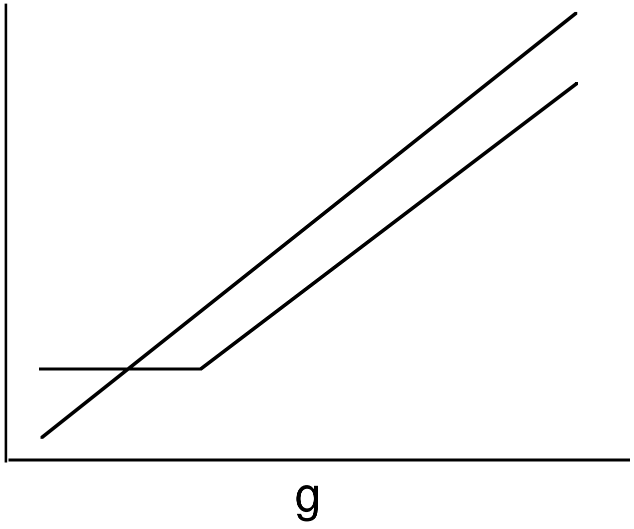
Modeling Effects of gap

- **Uniform communication model**

$$T_{pred} = T_{orig}, \quad \text{if } g < l, \text{ average msg interval}$$
$$= T_{orig} + m(g - l), \text{ otherwise}$$

- **Bursty Communication**

$$T_{pred} = T_{orig} + m\,\Delta g$$

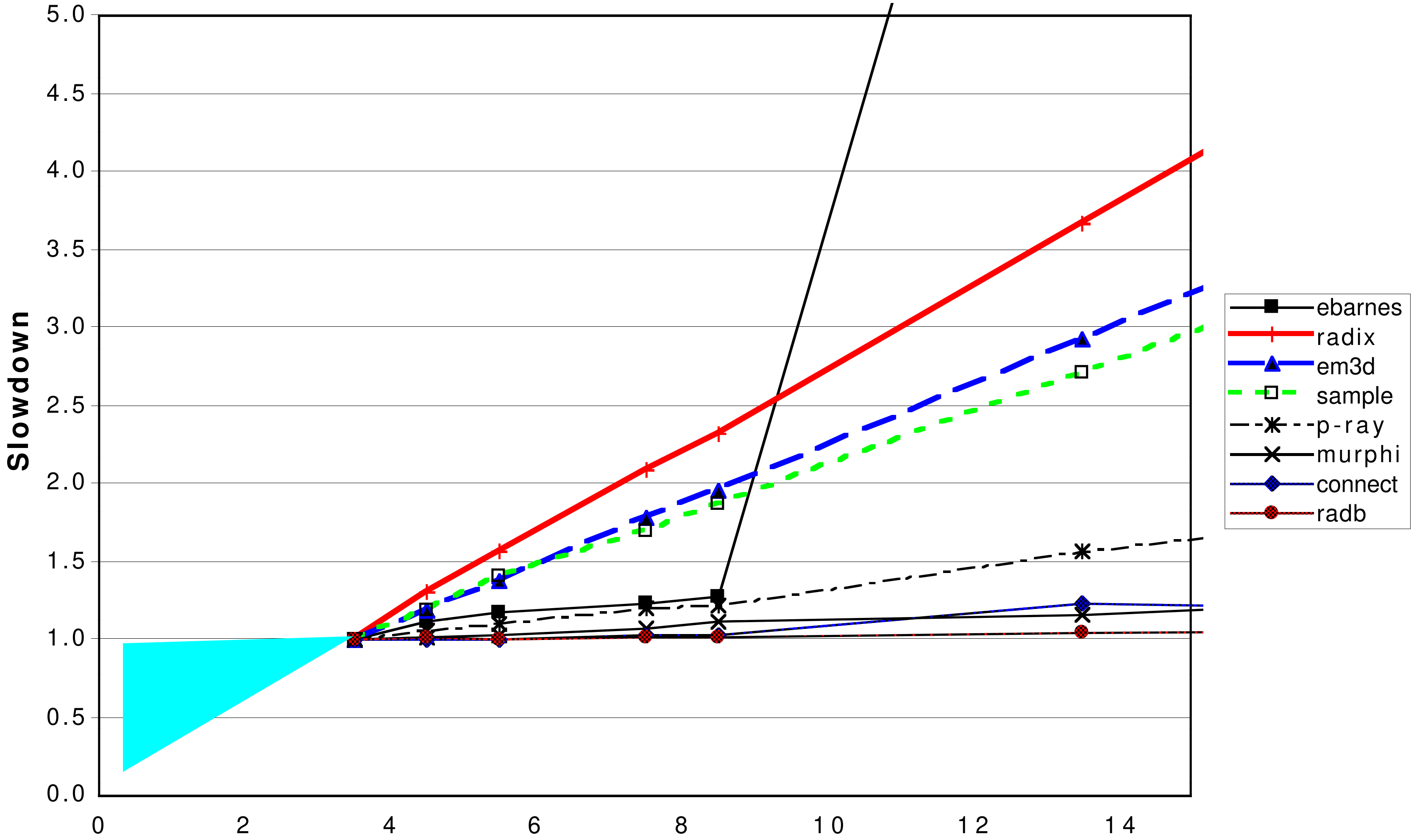


Evaluating BW Models

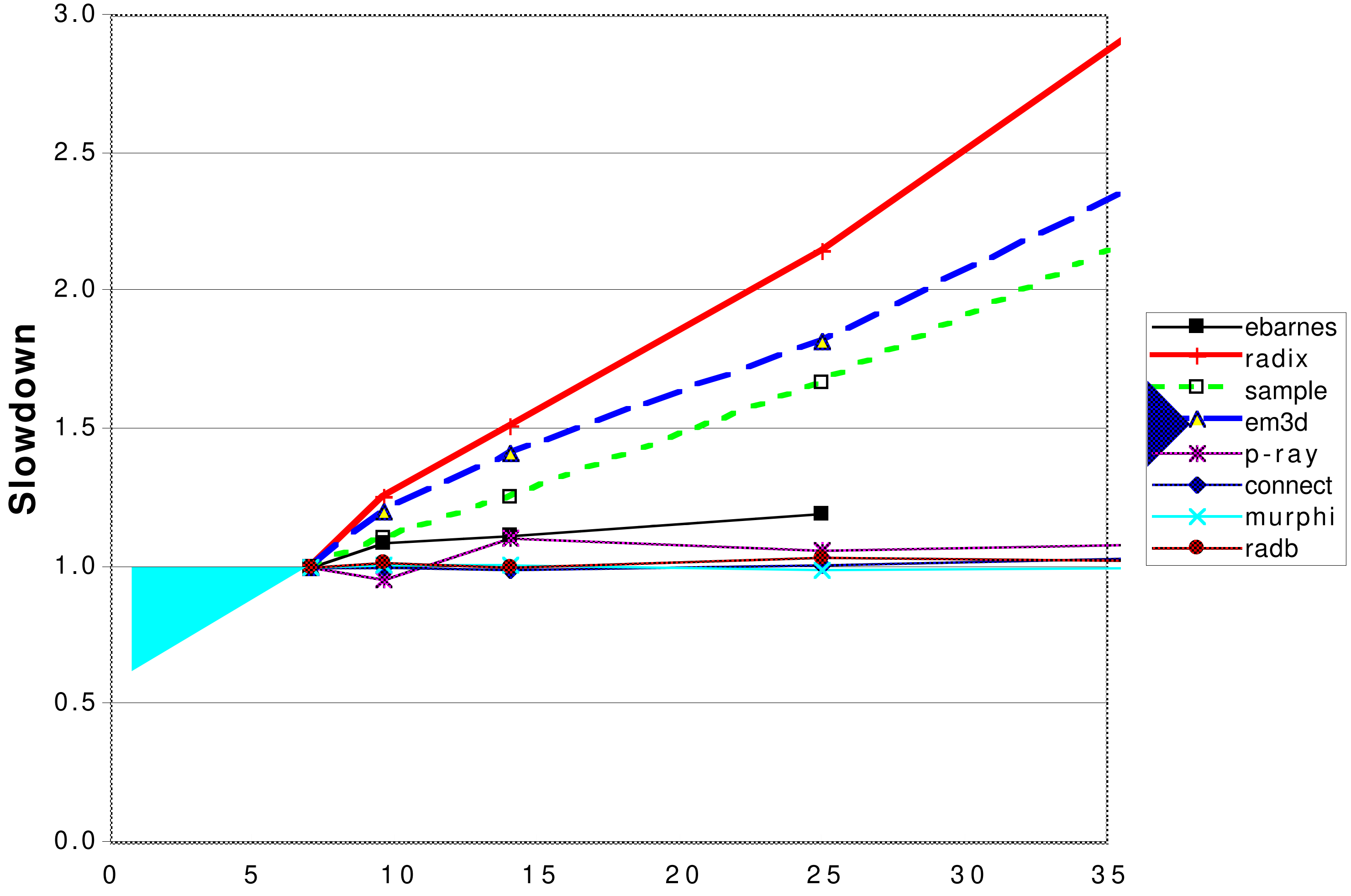
- Do not see the behavior of “uniform” model in any of the applications
- Pure bursty model gives good prediction
  - slightly over estimates in moderate cases

Do we need to push further?

Extrapolating to Low Overhead



Extrapolating to high msg rate



Conclusions

- **Applications exhibit a vast range of communication requirements**
- **Effort to improve communication performance is well rewarded**
- **Most sensitive to overhead**
  - **“does not parallelize well”**
  - **exasperbates serialization**
- **Surprisingly tolerant to latency**
- **Bulk BW not very critical**
- **Need to keep working on overhead!**
-
- **Valuable to use real experimental systems to explore relative to a model framework**