Applications and Abstractions

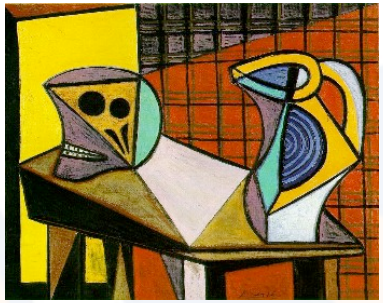
A Cautionary Tale

David S. Rosenblum

Felicitous Computing Institute
School of Computing
National University of Singapore

My Net Cred

- SIENA Internet-scale publish/subscribe system
  - *Collaboration with Alex Wolf & Antonio Carzaniga*
- Formerly Principal Architect and CTO of precache
- Confidentiality in Internet-scale publish/subscribe
- ROAR: Rendezvous on a Ring
  - *PhD of Costin Raiciu, collaboration with Mark Handley*
- Some papers in ACM TOCS, PODC, SIGCOMM, ICNP
- Ten patents for work at precache



Question 0

What is (an) abstraction?

"the process of considering something independently of its associations, attributes, or concrete accompaniments"
[Oxford American Dictionary]

- Implementation independence
- Widespread applicability and reusability

Question 1

Why are abstractions needed?

- for understanding and reasoning
- for designing and implementing

My focus in this talk is on abstractions for building applications that are to be deployed on the Internet

Question 2

What abstractions are needed?

- Communication paradigms
- Storage paradigms
- Structuring and coordination paradigms
- Formal logical models of these
- Formal quantitative models of these

My own interests are in communication paradigms and probabilistic models

NUS National University of Singapore

The Thesis of This Talk

General-purpose abstractions for building applications can lose their generality and/or abstractness once realized at Internet scale.

There may be many approaches for realizing an abstraction, but each one employs its own assumptions, algorithms, protocols, optimizations and heuristics.

Those choices can strongly constrain the set of applications able to use the realization naturally, effectively and efficiently.

Motivating Example

Publish/Subscribe

Applications	*notifications, alerts, updates*
Components	*events*
Objects	*events*
OS	*signals, interrupts*

- Natural abstraction for multi-way, asynchronous dissemination of data
- At application level, middleware or brokers provide decoupling, anonymity, matching, caching, authentication, and many other services
- Many conceivable applications at Internet scale

Internet-Scale Pub/Sub

Applications

symbol = "AAPL" and price > 700.00

symbol = "AAPL", price = 701.23, shares = 5000, [etc.]

Stock Quotes

Internet-Scale Pub/Sub

Applications

bus = (10 or 30 or 51 or 143 or 188) and nextnextstop = 16069

bus = 143, capacity = 0.9, stop = 16089, nextstop = 16079, nextnextstop=16069

Location-Dependent Travel Alerts

bus arrivals, taxi dispatching, traffic incidents, etc.

Internet-Scale Pub/Sub

Application Characteristics

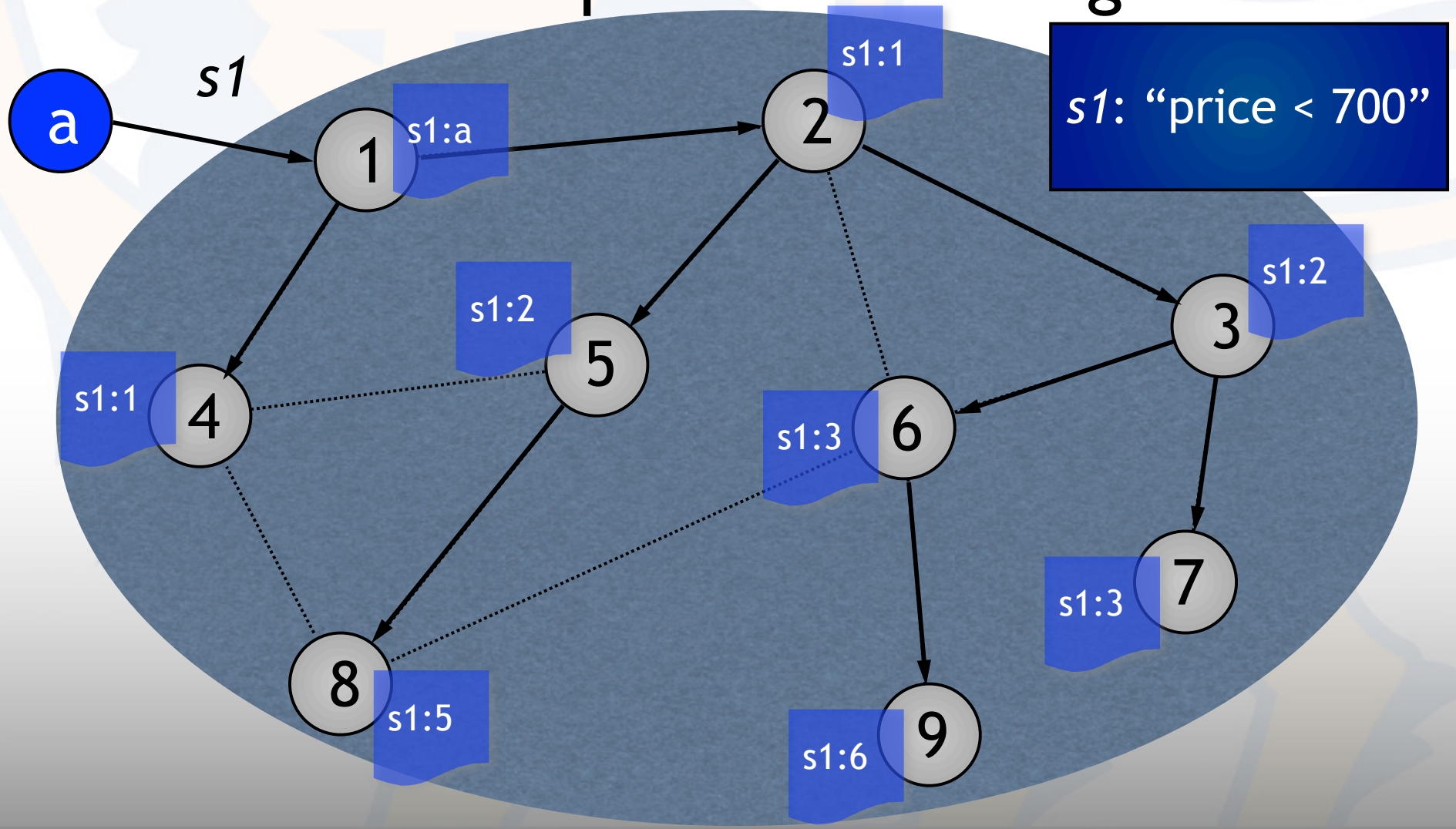
Subscriptions			Application	Notifications		
…	Selectivity	Churn		Frequency	Uniqueness	…
…	↑	↓	Stock Quotes	↑	↑	…
…	↓	↑	Software Updates	↓	↓	…
…	↑	↑	Travel Alerts	↑	↑	…
…	↑ ↑	↓	News Alerts	↑	↑	…
…	↑	↑	MMOGs	↑	↑	…
…	↑	↑	Battlefield Awareness	↓ ↑	↑	…
…	↑	↓	Location Updates	↓ ↑	↑	…
…	↓	↓	Social Network Alerts	↑	↓	…
…	↑	↑	Context Awareness	↑	↑	…
…	…	…	…	…	…	…

SIENA

- *General-purpose* realization of publish/subscribe at *Internet scale*
- Designed as a *decentralized overlay* of brokers
- Full *content-based matching* of notifications to subscriptions with *best-effort delivery*
- *Self-describing* notifications—no notification types, predefined topic hierarchies, etc.

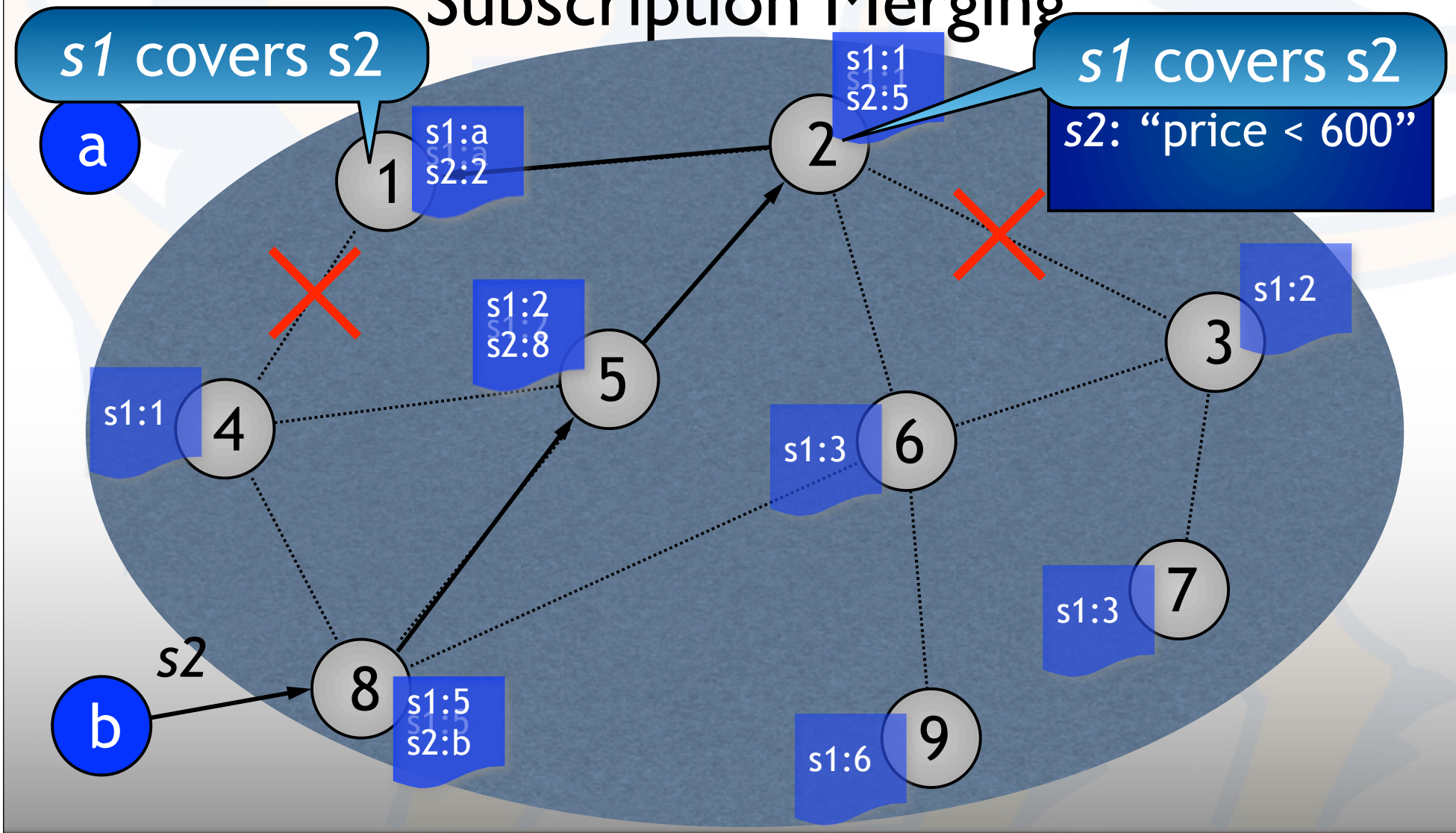
SIENA

Subscription Forwarding



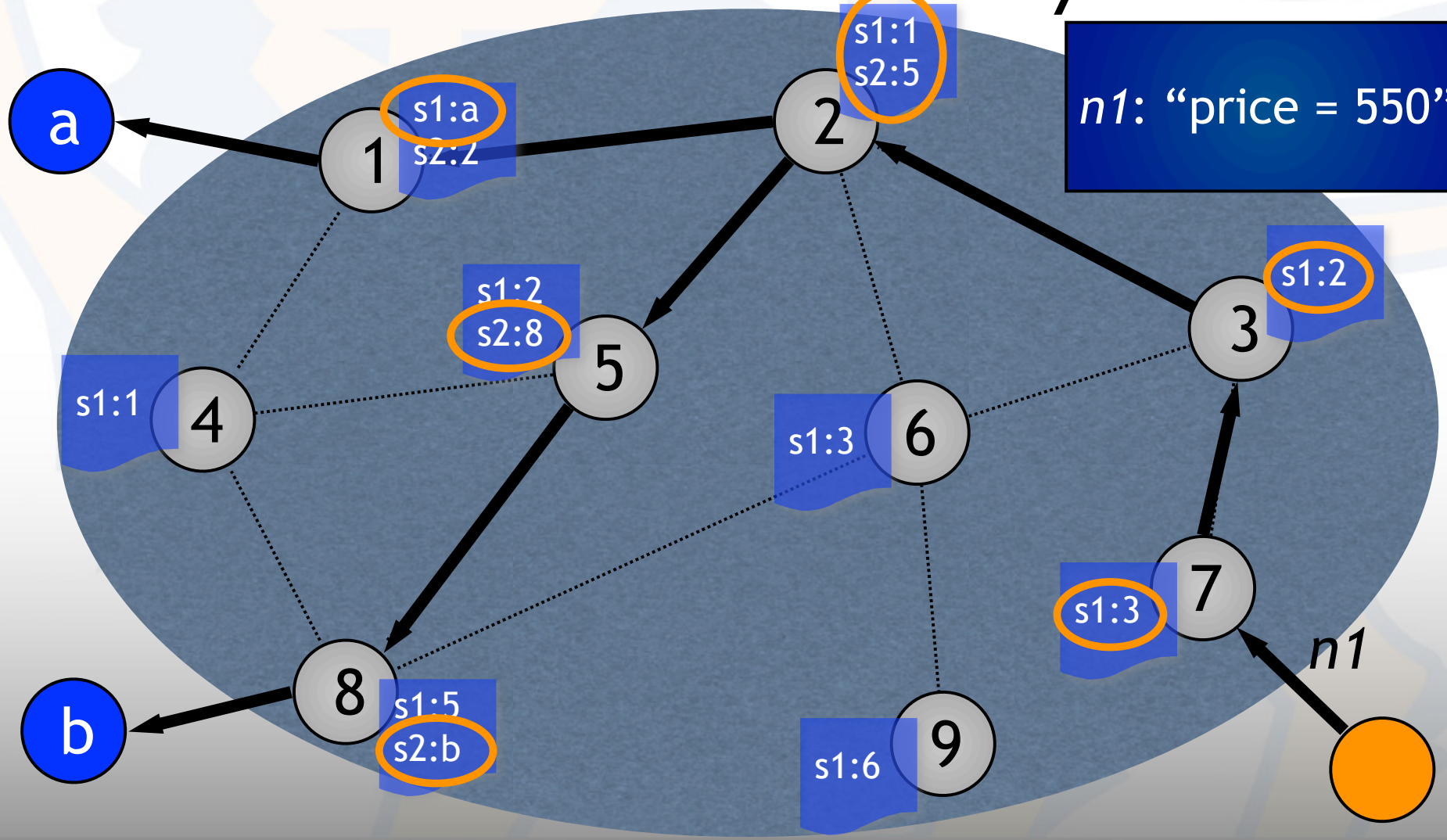
SIENA

Subscription Merging



SIENA

Notification Delivery



SIENA

Implied Ideal Application Characteristics

- Many publishers *and* many subscribers
  To justify decentralized implementation
- Notifications much more frequent than subscriptions
  To justify subscription forwarding
- Low subscription churn
  To justify subscription forwarding and merging
- High subscription selectivity
  To justify content-based matching in brokers
- Subscription similarity correlated with network locality
  To justify subscription merging

SIENA

Implied Ideal Application Characteristics

- Many publishers *and* many subscribers
  not Stock Quotes
- Notifications much more frequent than subscriptions
  not Software Updates
- Low subscription churn
  not location-dependent applications
- High subscription selectivity
  not Software Updates
- Subscription similarity correlated with network locality
  not Stock Quotes, Software Updates, MMOGs, etc.

SIENA

Implied Ideal Application Characteristics

☞ Few applications have *all* these characteristics

Traffic alerts

Social interaction alerts

others?

Internet-Scale Pub/Sub

Other Approaches

☞ Other approaches induce similar limitations

- Gryphon
  - Subscription flooding over tree of clusters
  - *Applicable if subscriptions are few and stable*
- Hermes
  - Rendezvous nodes allocated to content types
  - *Applicable if load is spread evenly by type*
- PreCache
  - Trie- and *kd*-tree-based subscription storage
  - *Applicable if subscription churn is very low*

Conclusion

- *Conceptually*, publish/subscribe is a very general abstraction
- But it loses generality once realized *at Internet scale*
- And it does so for reasons that have little to do with the peculiarities of the Internet
- *Adaptability* as a compromise
  - *ROAR's partitioning/replication tradeoff*
  - *Alex and Antonio's content-based networking (CBN)*

Question 3

How can research ... be fostered ... ?

- With respect to abstractions for *building* ...

  I would like to have better formal logical and probabilistic models ...

  ... for exploration of and reasoning about ...

  ... the design space induced by a network abstraction like publish/subscribe.