PATTERNS IN NETWORK ARCHITECTURE:

DATA-CENTRIC NETWORKING

DATA-CENTRIC NETWORKING

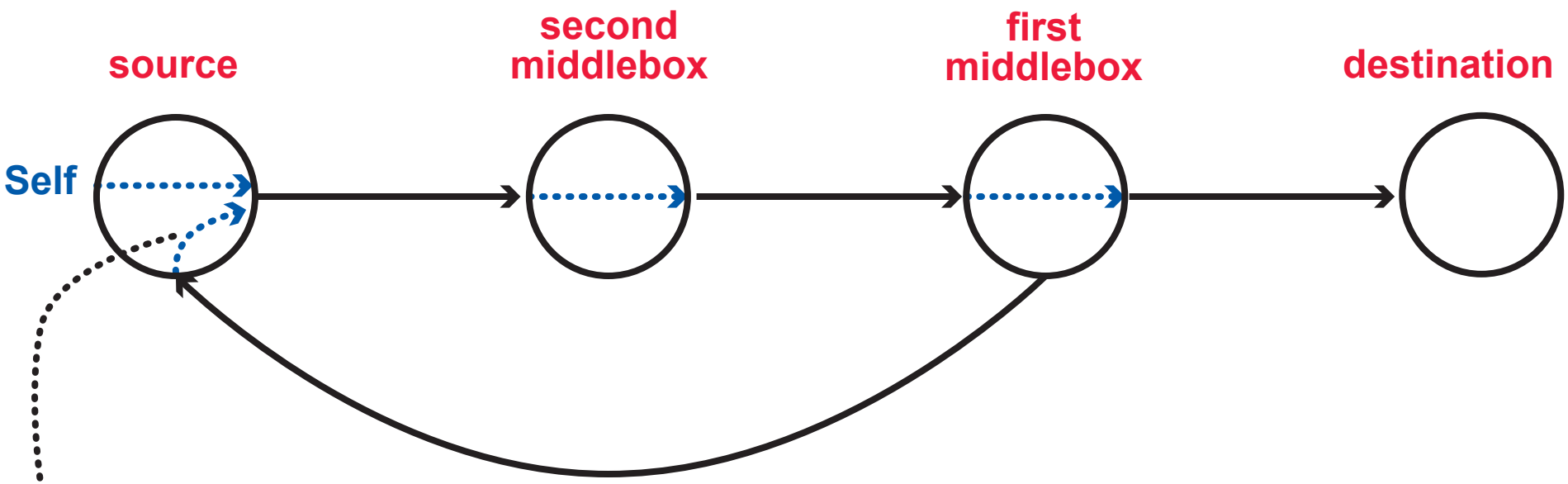
OLD TOPICS

1. A short topic on modeling in Alloy
2. A short topic on principles for layering

NEW TOPICS

3. An overview of data-centric networking (4 proposals)
4. David Rosenblum's "Cautionary Tale"
5. Discussion of the "narrow waist" of the Internet

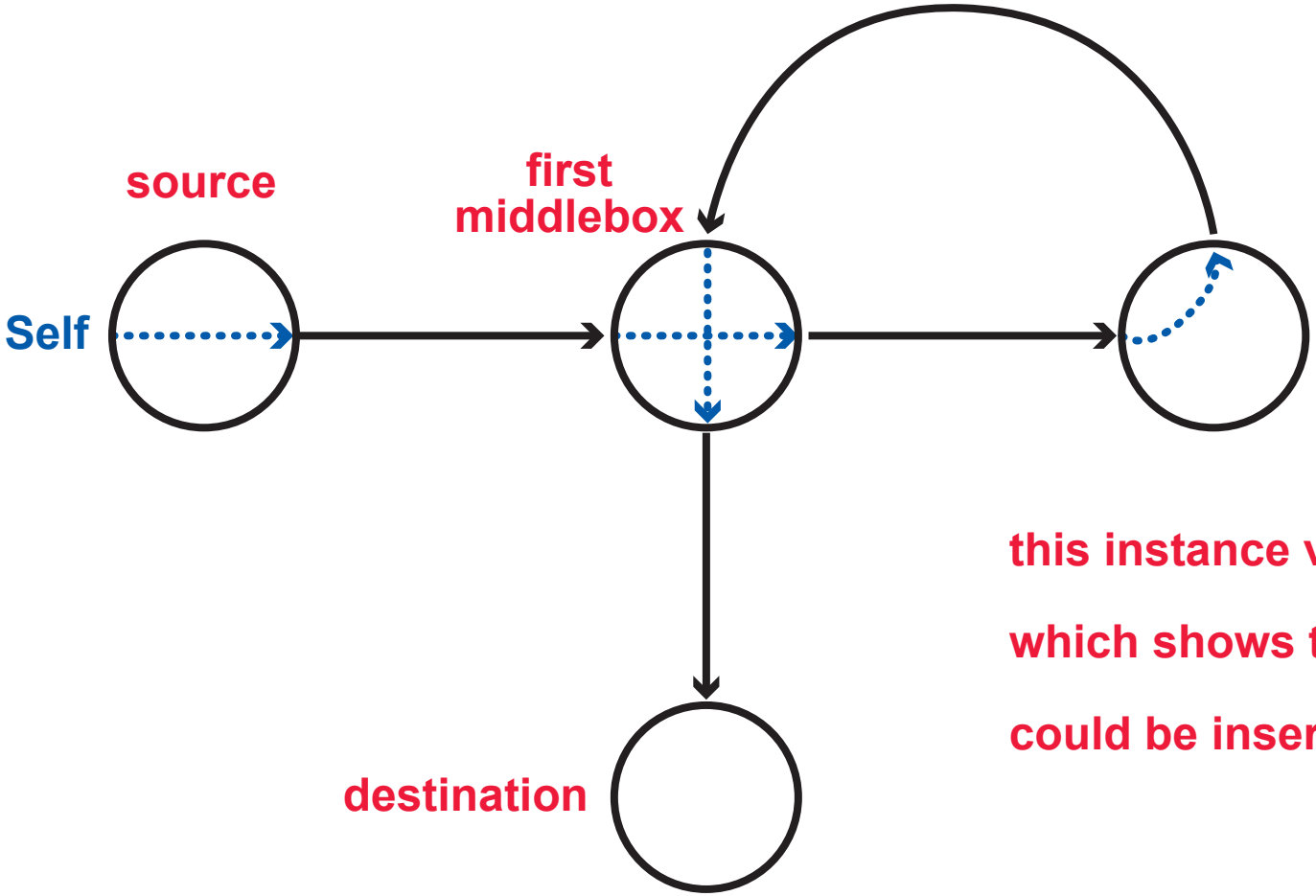
all forwarding rules for a single header



fixed by prohibiting this forwarding rule, because it forwards through the header's sender

this instance satisfies the specification because the second middlebox can be reached by forwarding (according to the current definition) from the first middlebox

all forwarding rules for a single header



this instance violates my theorem,

which shows that middleboxes

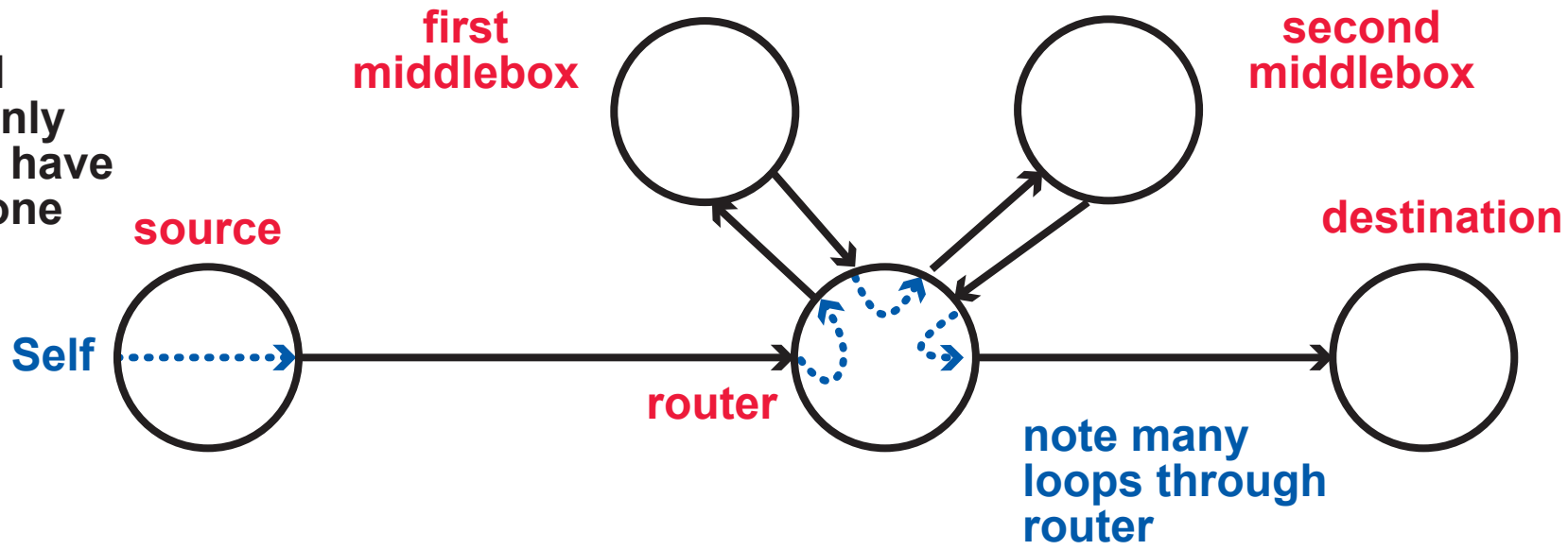
could be inserted in the wrong order

fixed by prohibiting middleboxes (which are “endpoints” as opposed to “routers”) from having more than one input link and one output link

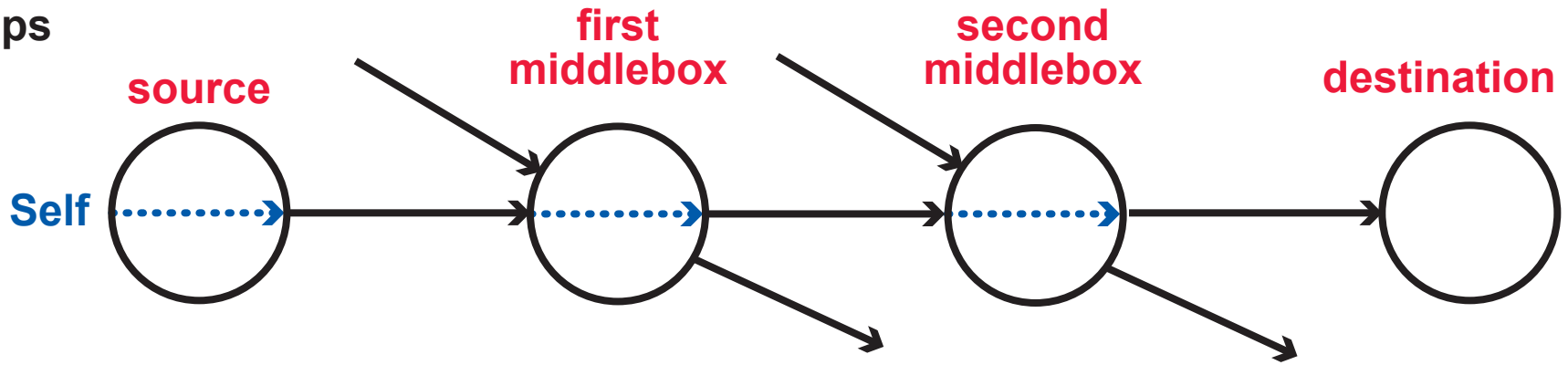
this is fine for some networks, too restrictive for others

SPECIALIZE!

in a "routed network," only routers can have more than one inlink or outlink

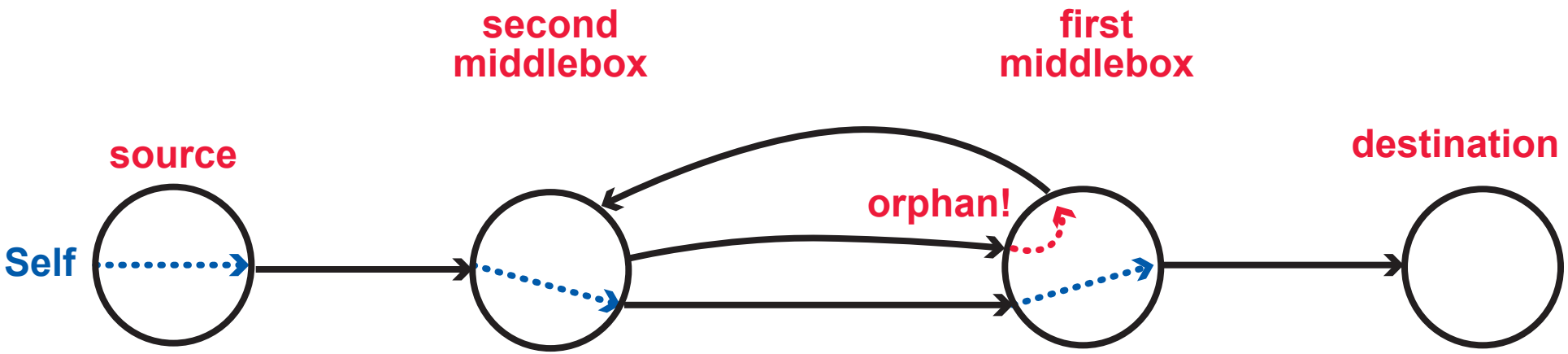


in a "peer network," there can be no node loops



WE'RE ALMOST THERE . . .

. . . except that this satisfies the specification



orphan forwarding rules seem to be
a problem no matter where they are

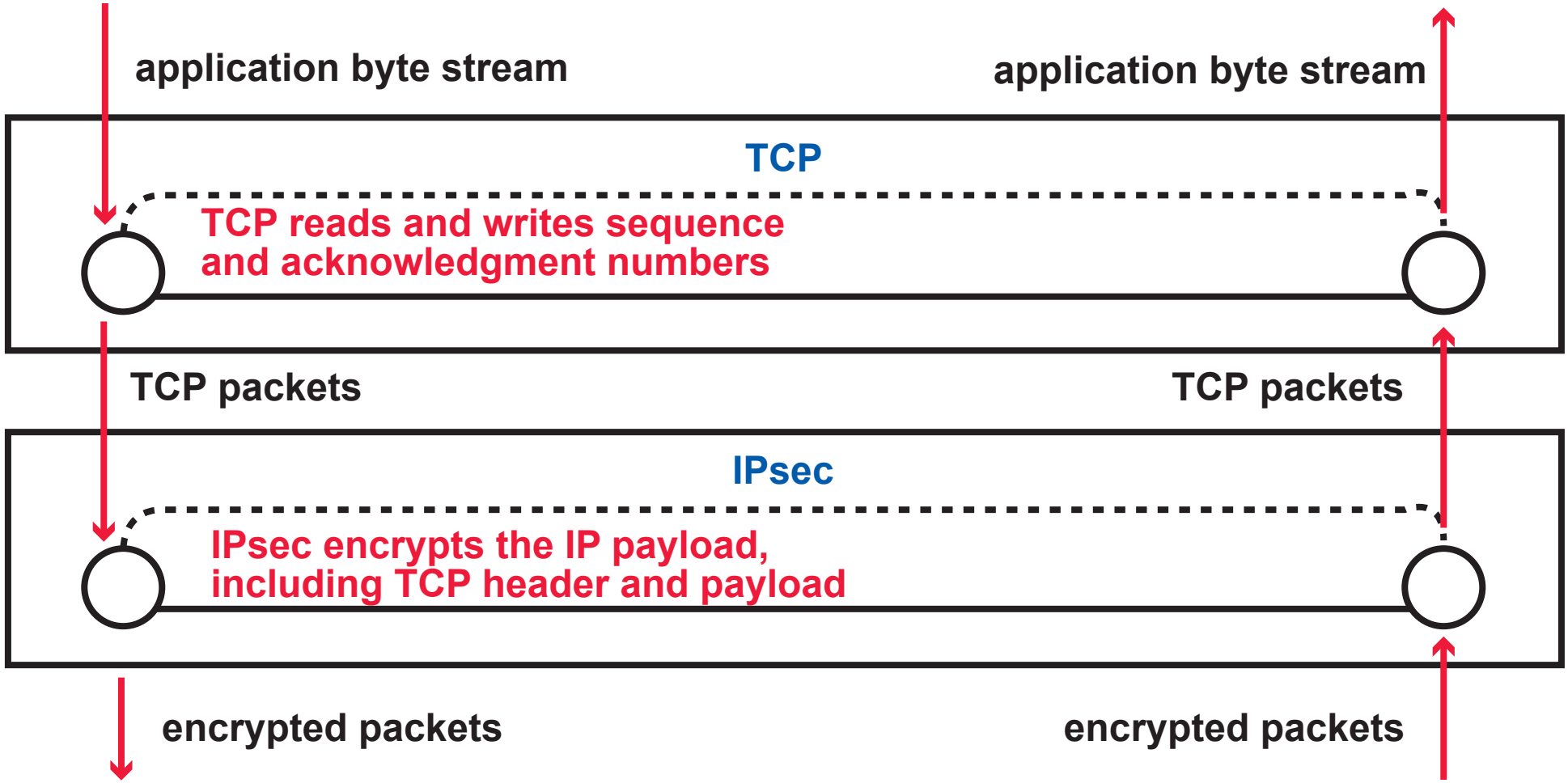
once we prohibit them, everything works

SERVICE INTERACTIONS

	QoS
RELIABILITY	reliability converts bandwidth, probability, latency to goodput, which also propagates piecewise
INTEGRITY	encryption decreases bandwidth and increases latency, both bad
MOBILITY	
PACKET FILTERING	filtering increases latency (bad) and bandwidth (good)

	RELIABILITY	INTEGRITY (ENCRYPTION)	MOBILITY
INTEGRITY	reliability above encryption		
MOBILITY	reliability above mobility	encryption above mobility	
PACKET FILTERING		filtering and encryption are independent only if session initiations are not encrypted	filtering above mobility or far from endpoint

RELIABILITY ABOVE ENCRYPTION



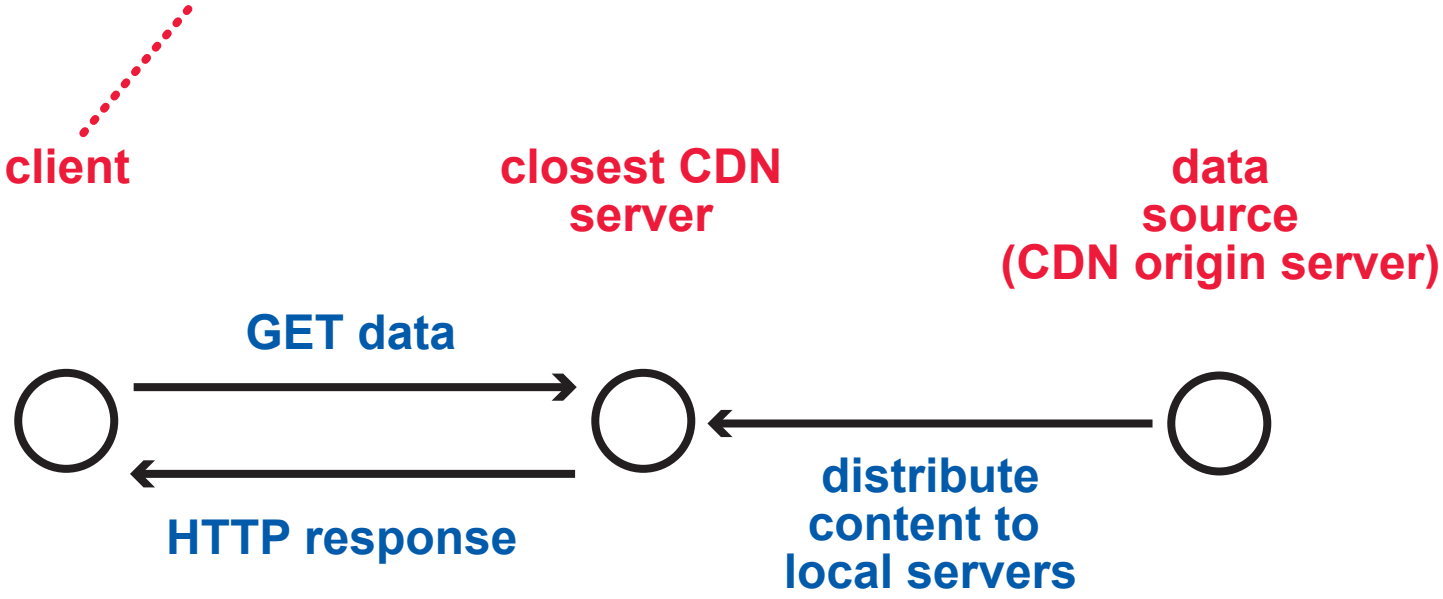
HOWEVER, both reliability and encryption are implemented in an end-to-end session protocol

TLS combines them into one protocol

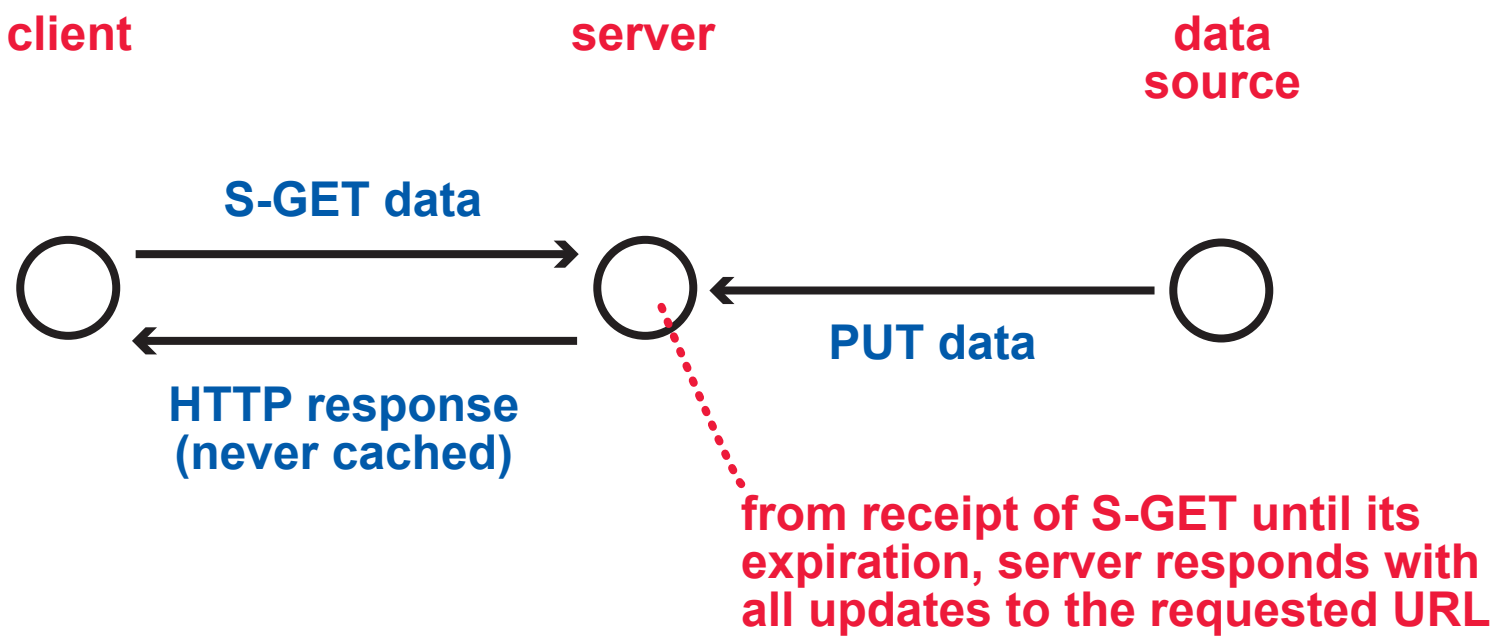
mcTLS extends it to support middleboxes —but they must be trusted!

CONTENT-DELIVERY NETWORK

client's DNS lookup goes to DNS server of CDN,
returns IP address of closest CDN server



HTTP SUBSCRIBE-GET



THIS IS A KIND OF

PUBLISH-SUBSCRIBE

SYSTEM

NAMED DATA NETWORKING

MEMBERS

- members are producers of data, consumers of data (or both), routers

LINKS

- there is a fixed set of links—this is very important

NAMES

- names of unique, immutable data packets are also names of their producers or repositories
- so a member can have no names or many names, and a name can belong to more than one member

ROUTING

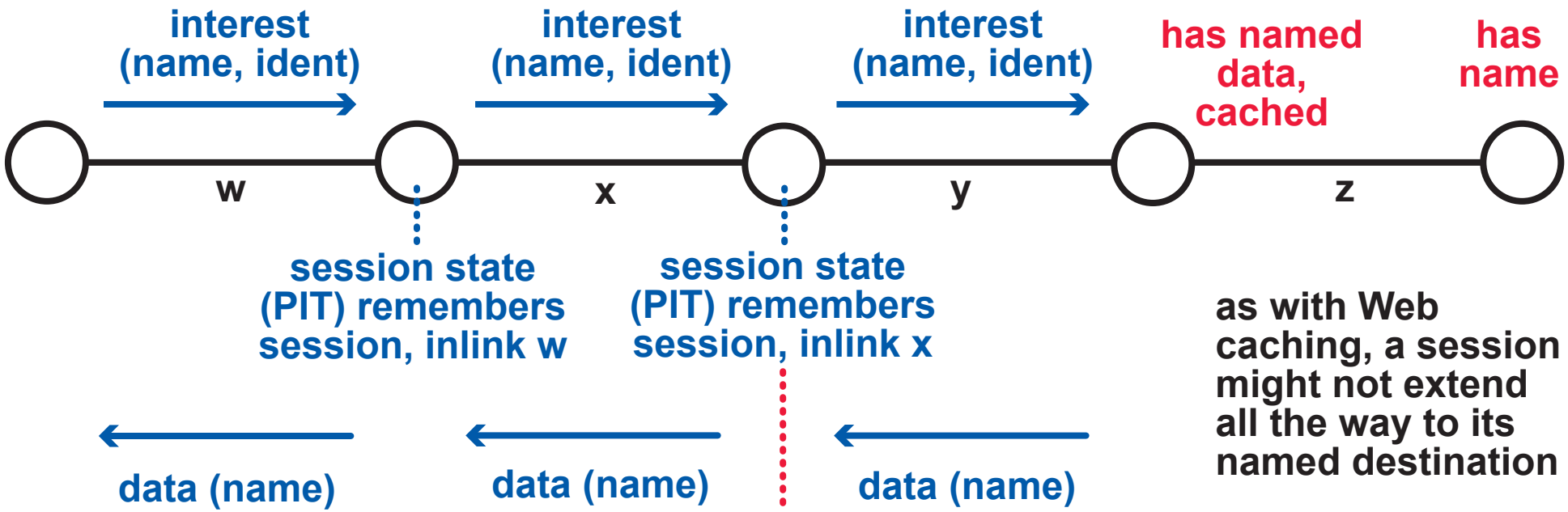
- routing is fairly normal, even using existing protocols
- route to name prefixes

SESSIONS

- these are interesting!

NAMED DATA NETWORKING: SESSIONS

A SESSION HAS (AS ALWAYS) A DESTINATION NAME AND IDENTIFIER (NONCE)



as with Web caching, a session might not extend all the way to its named destination

a session consists of one interest packet and one data packet . . .

. . . although a data packet on a link can be shared among sessions

the session identifier prevents routing loops

Applications and Abstractions

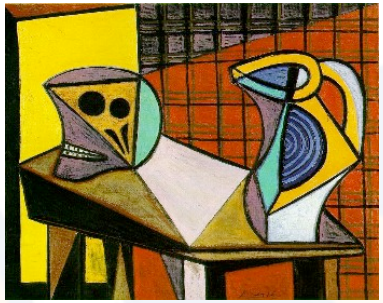
A Cautionary Tale

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

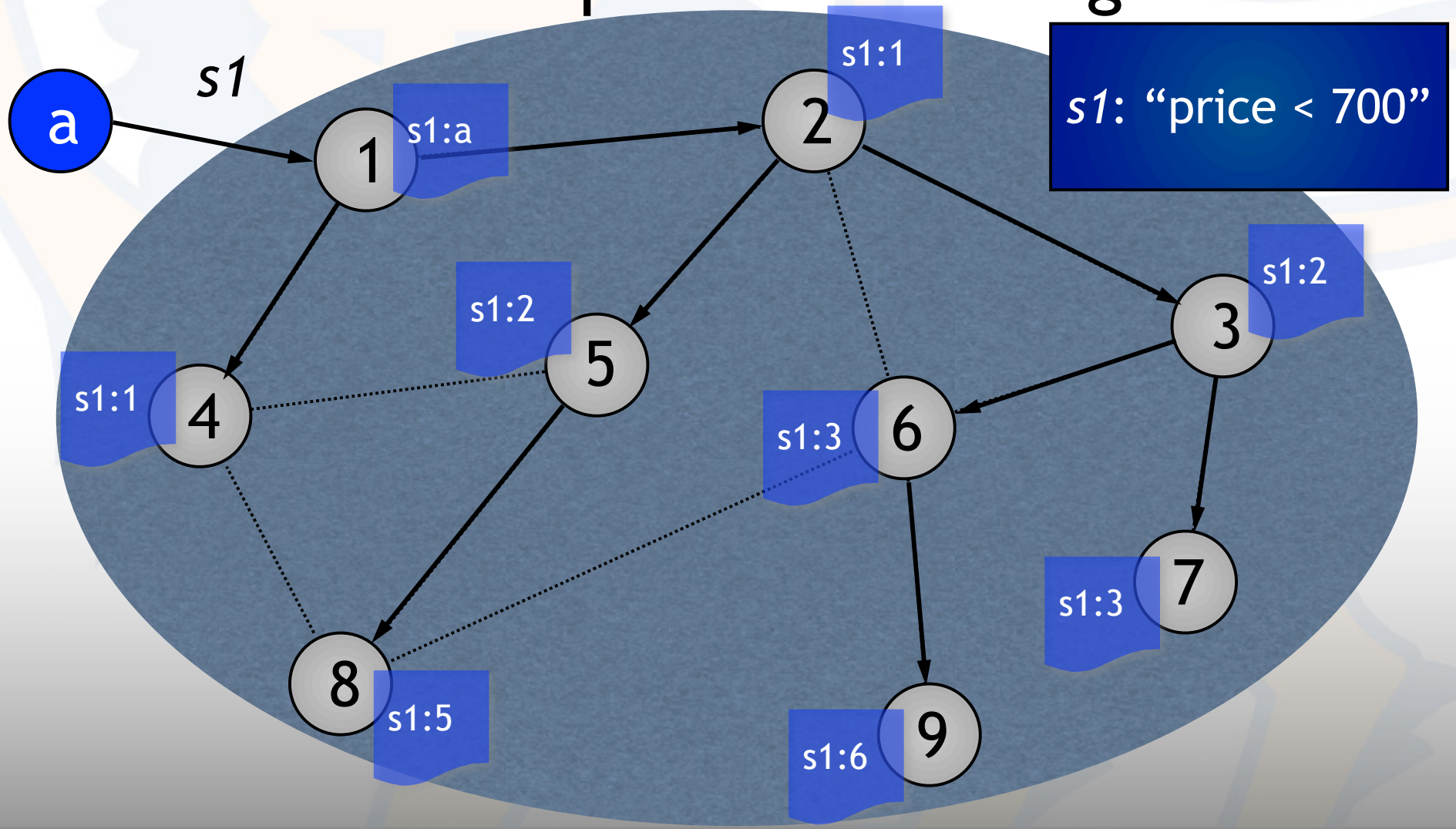
Subscriptions			Application	Notifications		
…	Selectivity	Churn		Frequency	Uniqueness	…
…	↑	↓	Stock Quotes	↑	↑	…
…	↓	↑	Software Updates	↓	↓	…
…	↑	↑	Travel Alerts	↑	↑	…
…	↑ ↑	↓	News Alerts	↑	↑	…
…	↑	↑	MMOGs	↑	↑	…
…	↑	↑	Battlefield Awareness	↓ ↑	↑	…
…	↑	↓	Location Updates	↓ ↑	↑	…
…	↓	↓	Social Network Alerts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

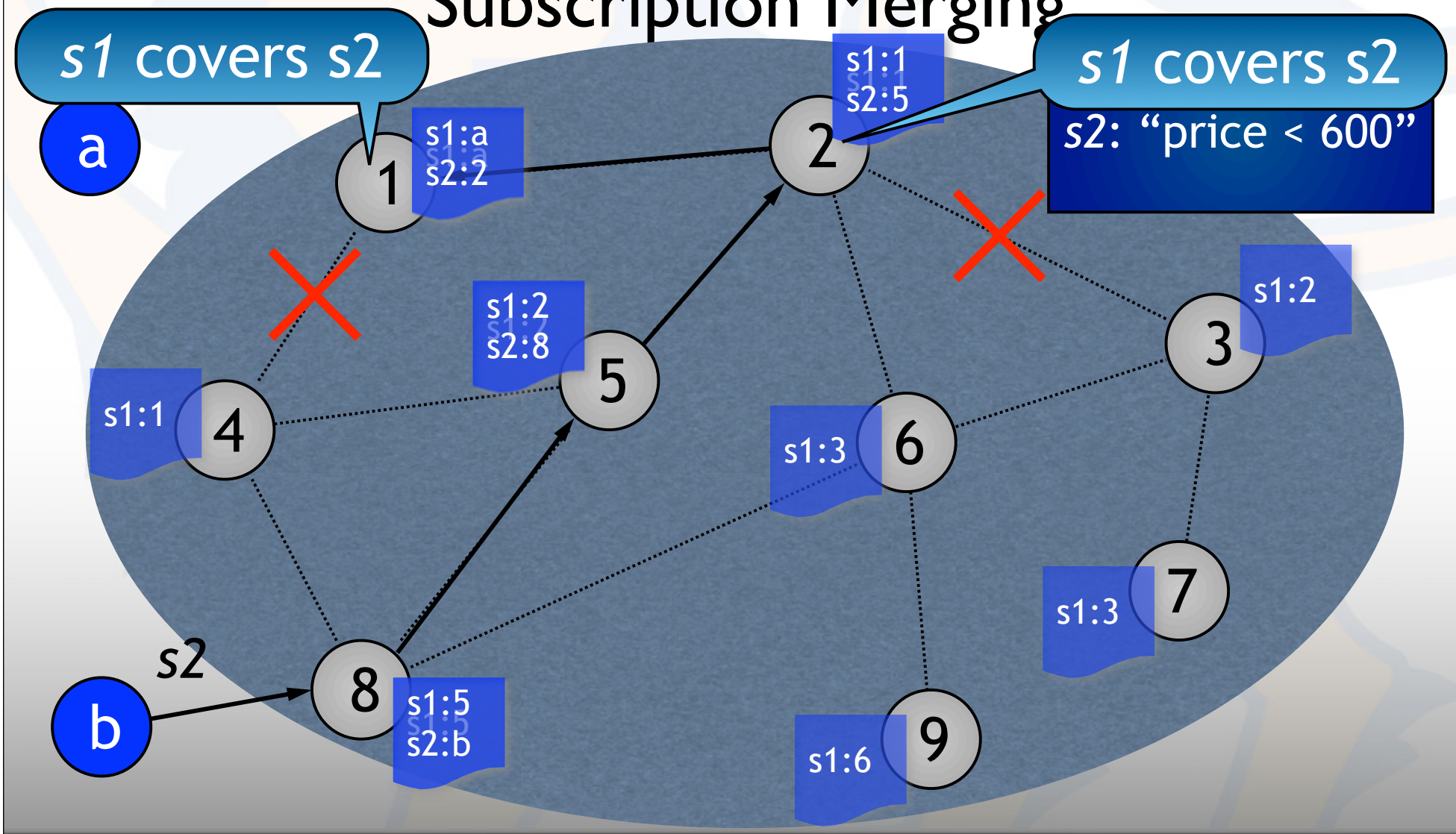
SIENA

Subscription Forwarding



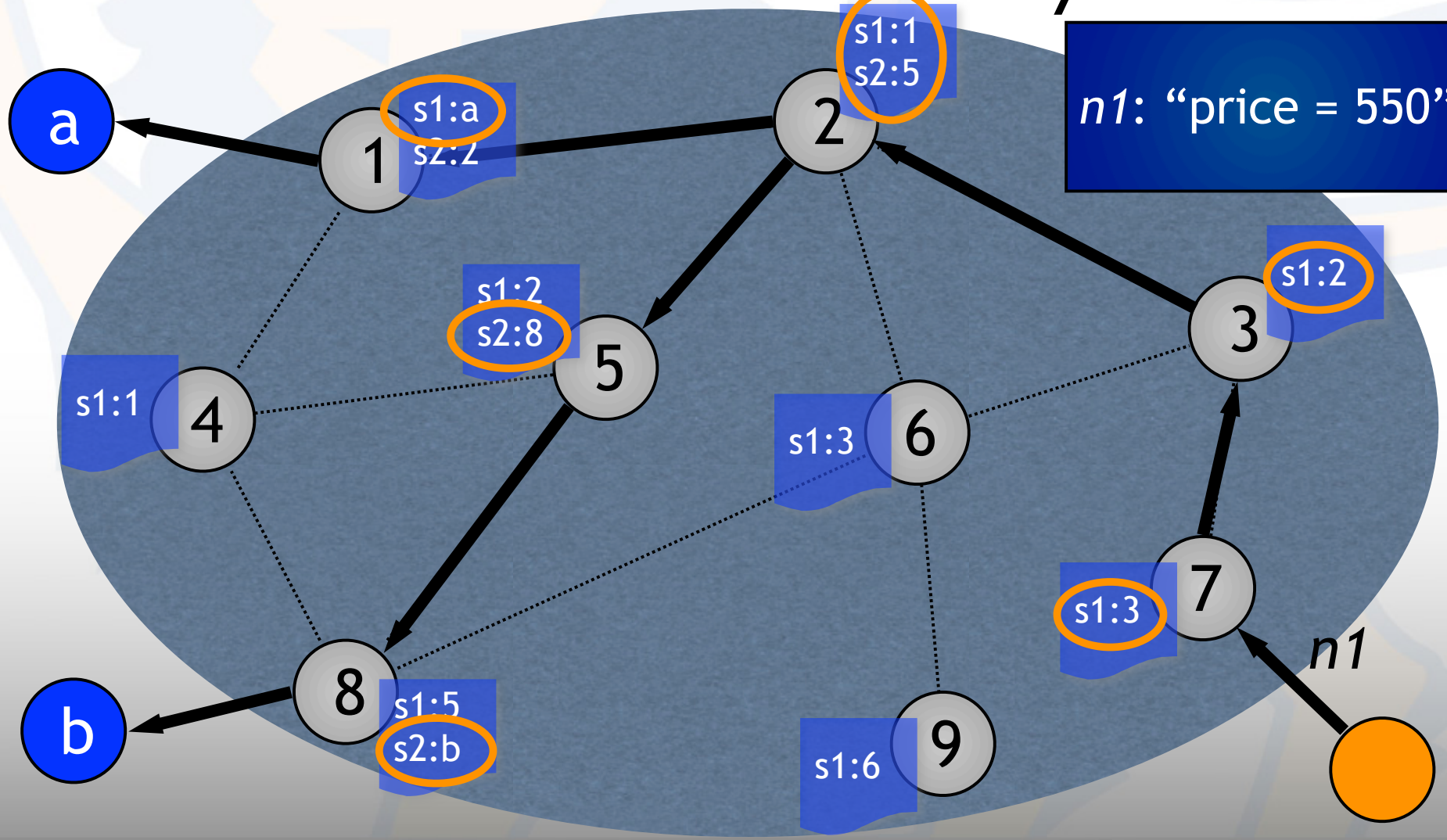
SIENA

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.

WHAT IS THE “NARROW WAIST” OF THE INTERNET?

WHAT IS THE "NARROW WAIST" OF THE INTERNET?

THE UNVARYING PART OF THE PROTOCOL STACK

THE MOST POPULAR PROTOCOL, WHERE MOST INVESTMENT IS NOW

application protocols

this appears to be the justification for HTTP

IP, TCP, UDP

link protocols

THE PROTOCOLS THAT EVERY NETWORKED MACHINE MUST UNDERSTAND

a new (to me) example involving ssh:

application
IP
link
application
IP
link

seems like a good definition

THE "NARROW WAIST" OF THE INTERNET

WE HAVE READ CLAIMS THAT BOTH . . .

. . . HTTP

. . . AND NAMED DATA NETWORKING

. . . SHOULD BE THE NEW "NARROW WAIST" OF THE INTERNET.

What do you think of that?

NAMED DATA NETWORKING

LIKE PUBLISH/SUBSCRIBE, THIS IS A NICE ABSTRACTION

As David Rosenblum found for publish/subscribe, can you think of any aspect of the NDN proposal that might not work for every application?

NAMED DATA NETWORKING

LIKE PUBLISH/SUBSCRIBE, THIS IS A NICE ABSTRACTION

As David Rosenblum found for publish/subscribe, can you think of any aspect of the NDN proposal that might not work for every application?

(even for data-oriented applications)

NDN requires an interest packet for every data packet. In some cases, this could double the number of packets transmitted over the Internet.

each session, retrieving one data packet, leaves session state in a number of routers—

—not very fault-tolerant

—seems to need a lot of state

CELLPHONES ARE PART OF THE INTERNET, TOO

" . . . smartphones alone far outnumber tethered Internet hosts"

cellular networks will inevitably become more integrated with the rest of the Internet, because of costs

WHAT CELLPHONE SERVICE (TEXTING, VOICE, VIDEO) NEEDS

- **the ability to FIND a private device and INITIATE communication to it**
- **long-distance, low-jitter, high-bandwidth communication, always less than 150 ms latency**
- **except for lawful intercept, voice is not stored anywhere**
- **conserve the battery of a mobile device, despite the fact that some communication is bursty**

WHAT THE INTERNET OF THINGS NEEDS

10 billion networked things in 2010, 50 billion networked things in 2020

- the ability to FIND a private device and INITIATE communication to it
- conserve the batteries of very small devices
- upload massive, unprecedented amounts of data
- monthly communication cost of $ 2 per device

CONCLUSION:

WHAT CELLPHONES AND THE INTERNET OF THINGS NEED . . .

. . . IS ROUGHLY THE OPPOSITE OF WHAT DATA-CENTRIC NETWORKING NEEDS