ABSTRACTIONS

OF THE DATA PLANE

DIMACS Working Group on

Abstractions for Network

Services, Architecture, and Implementation

Pamela Zave

AT&T Laboratories—Research

Florham Park, New Jersey, USA

joint work with Jennifer Rexford, Princeton University

WHAT WE ATE FOR LUNCH

CHINESE FOOD, OF COURSE, AND LEARNED THAT . . .

"We find comfort among those who agree with us—

growth among those who don't."

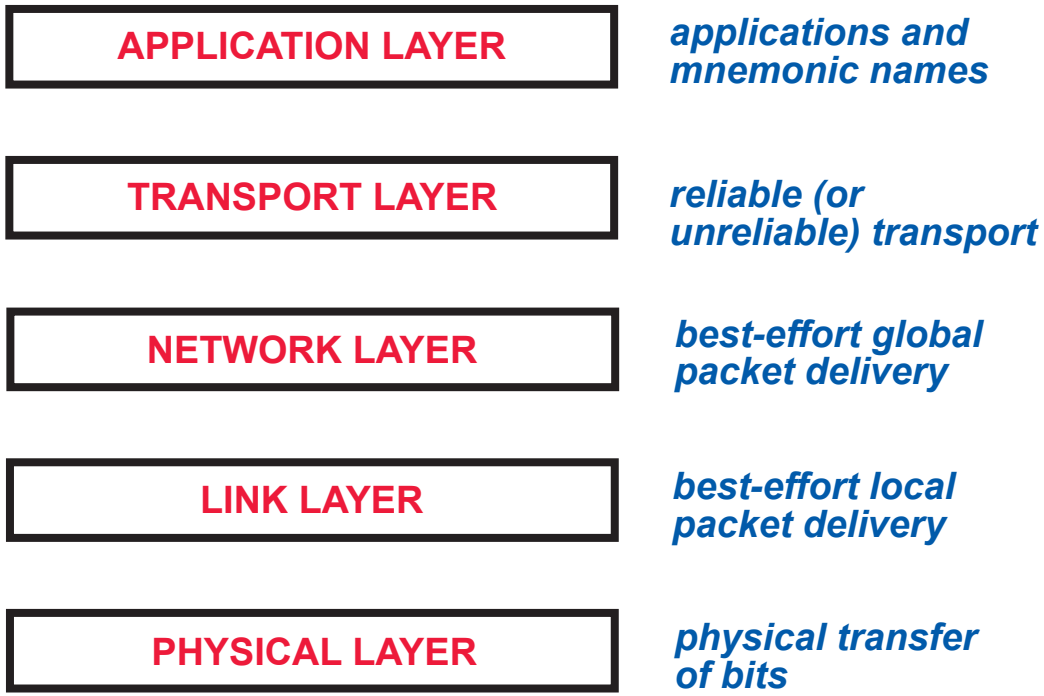
THE PREVAILING ABSTRACTION OF THE DATA PLANE

APPLICATION LAYER	*applications and mnemonic names*
TRANSPORT LAYER	*reliable (or unreliable) transport*
NETWORK LAYER	*best-effort global packet delivery*
LINK LAYER	*best-effort local packet delivery*
PHYSICAL LAYER	*physical transfer of bits*

abstractions from "The future of networking, and the past of protocols"

[Shenker 2011]

WHY SHOULD WE QUESTION THIS?



1 Because there are many serious problems with the current Internet, and we must look at all possible solutions.

2 Because the purpose of the control plane is to manage the data plane, so a well-structured data plane may be the key to a well-structured control plane.

For example, "An axiomatic basis for communication" is intended to formalize what routers do . . .

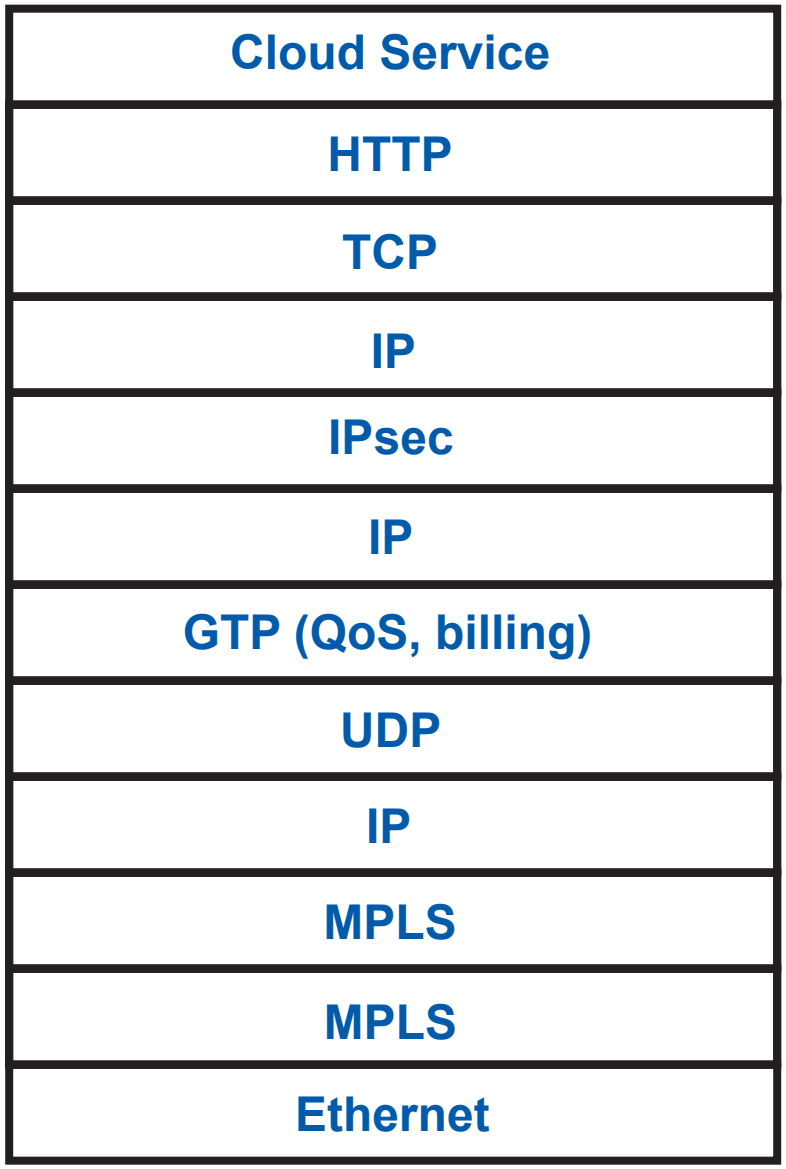
. . . but much of the space is devoted to a careful discussion of the behavior of the data plane.

[Karsten, Keshav, Prasad & Beg 2007]

WHY SHOULD WE QUESTION THIS?

3 Because it is not realistic.

headers in a typical AT&T packet



15+ load balancing / routing algorithms are involved in getting this packet to its destination . . .

. . . most with different goals in mind;

. . . most have been analyzed / designed in some state of isolation;

. . . all are getting more dynamic every day

from “Cloud computing and my worries about the network that enables it”

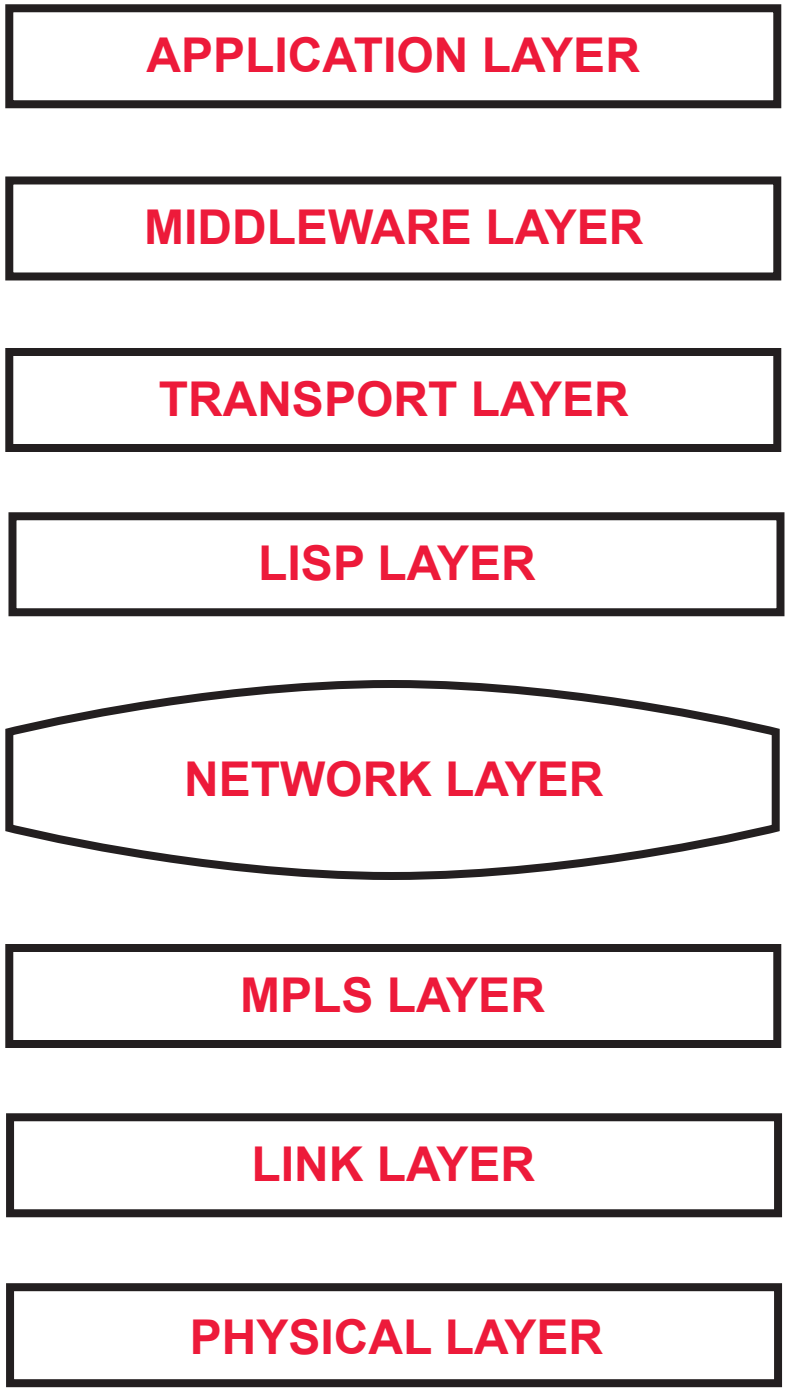
[Spatscheck 2010]

A BETTER ABSTRACTION OF THE DATA PLANE?

this is more realistic, . . .

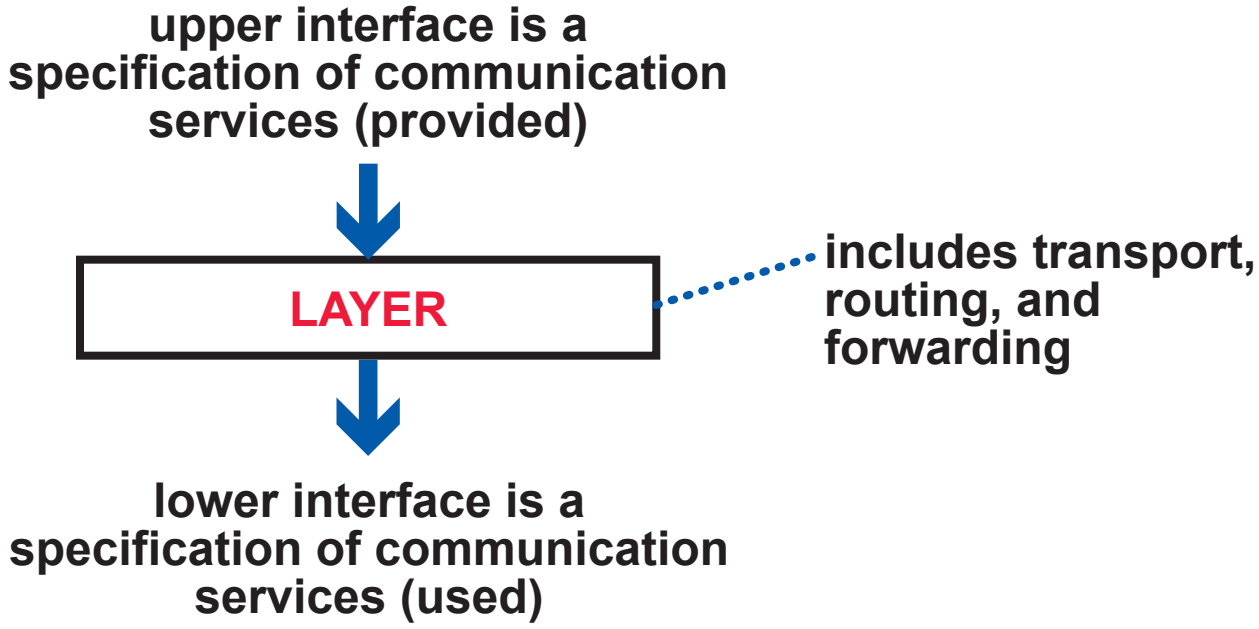
. . . but consensus would be difficult to achieve . . .

. . . and not long-lasting

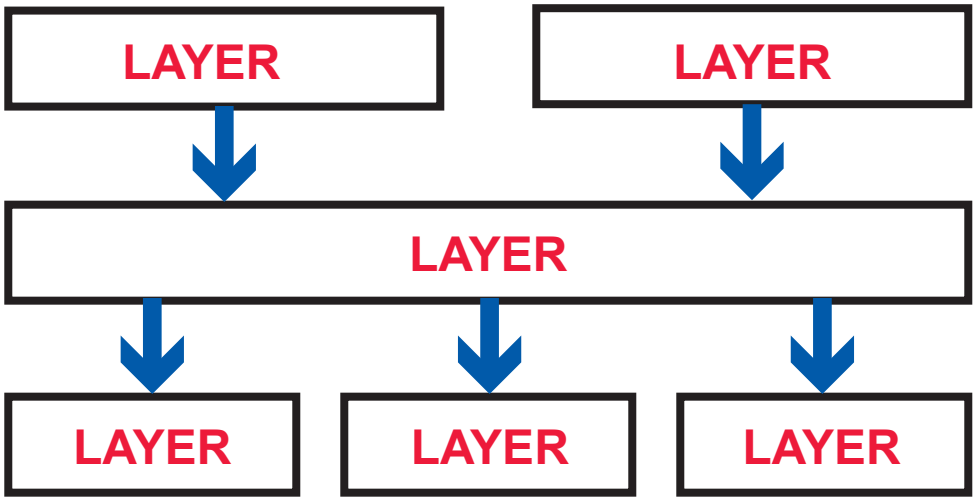


A DIFFERENT VIEW OF THE DATA PLANE

Each layer is a distributed system with the same abstract functionality and the same abstract state.



This pattern is instantiated many times in a network architecture, for many purposes, at many levels, and with many different scopes.

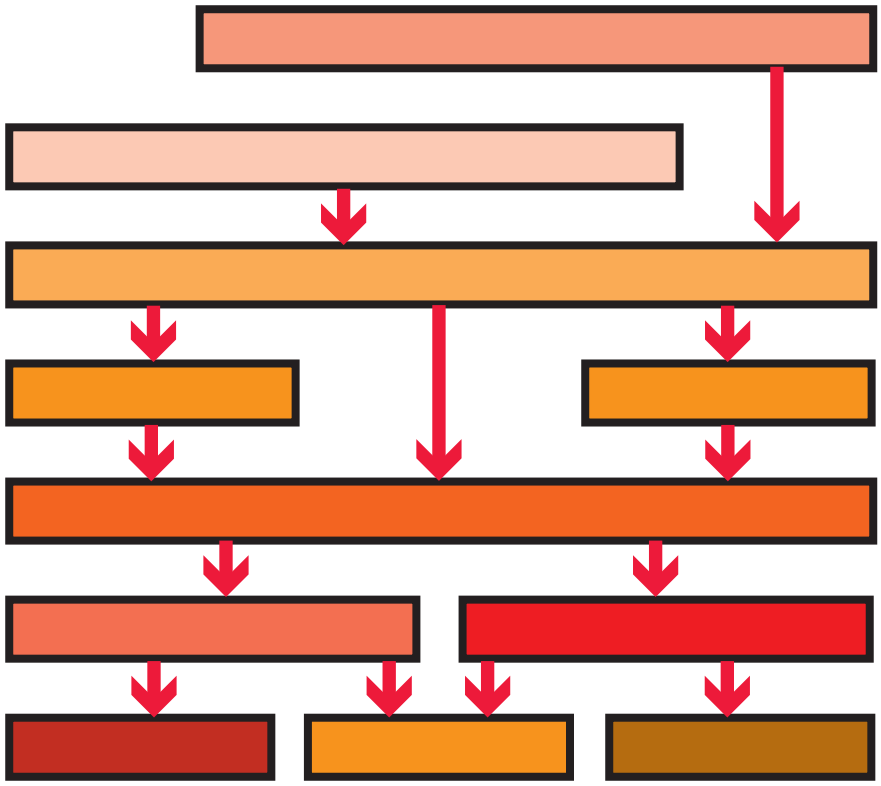


this hypothesis comes from *Patterns in Network Architecture*

[Day 2008]

WE CALL THIS THE "GEOMORPHIC VIEW" OF NETWORKS . . .

. . . BECAUSE THE ARRANGEMENT OF LAYERS RESEMBLES THE EARTH'S CRUST



it is inspired by Day's ideas, with many changes in terminology and (we hope) improvements

OUTLINE

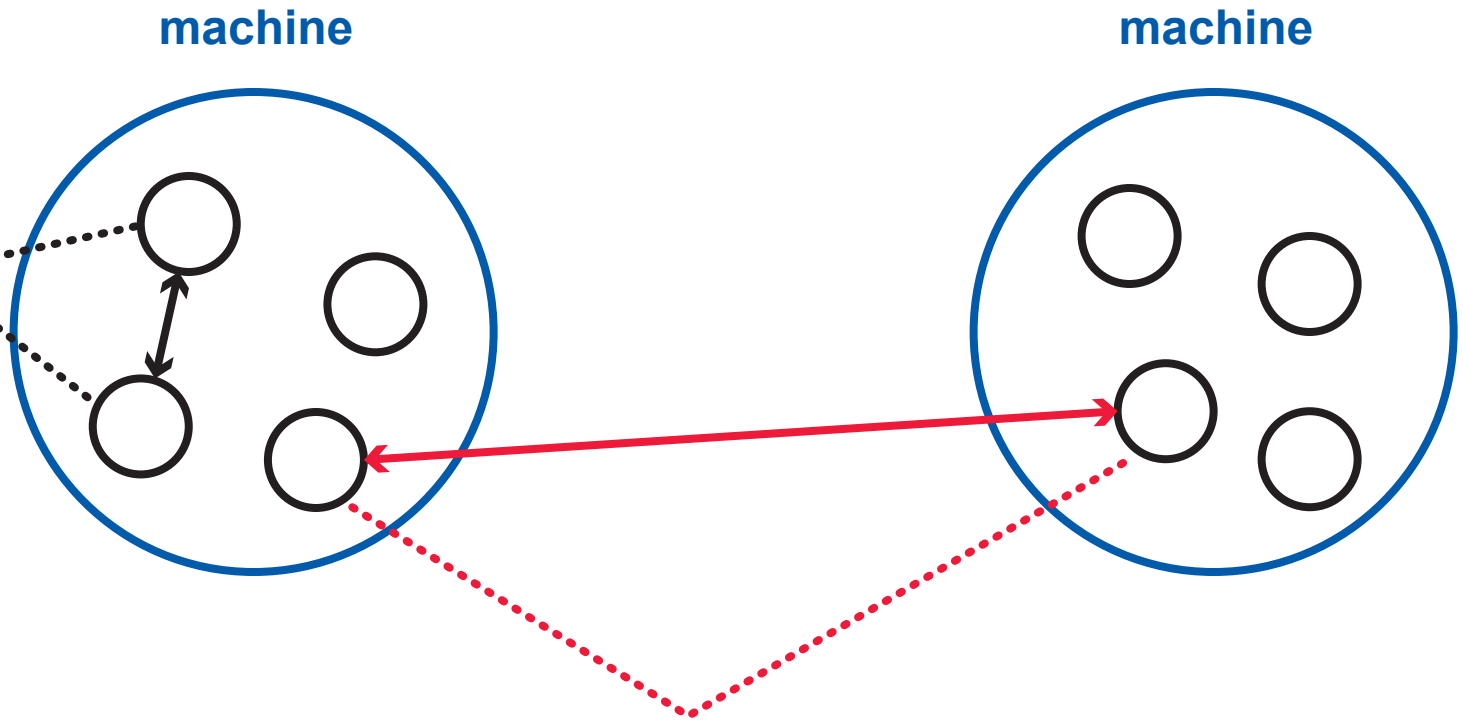
1 Basic information about layers

2 Frequently-asked questions

3 Examples

4 Summary and conclusions

LAYERS: MACHINES AND PROCESSES



the OPERATING SYSTEM creates these processes and enables them to communicate quickly and reliably with each other

this communication is assumed as a building block

these processes can only communicate through a NETWORK, with all of the challenges we know well

we can choose to regard a virtual machine as a machine . . .

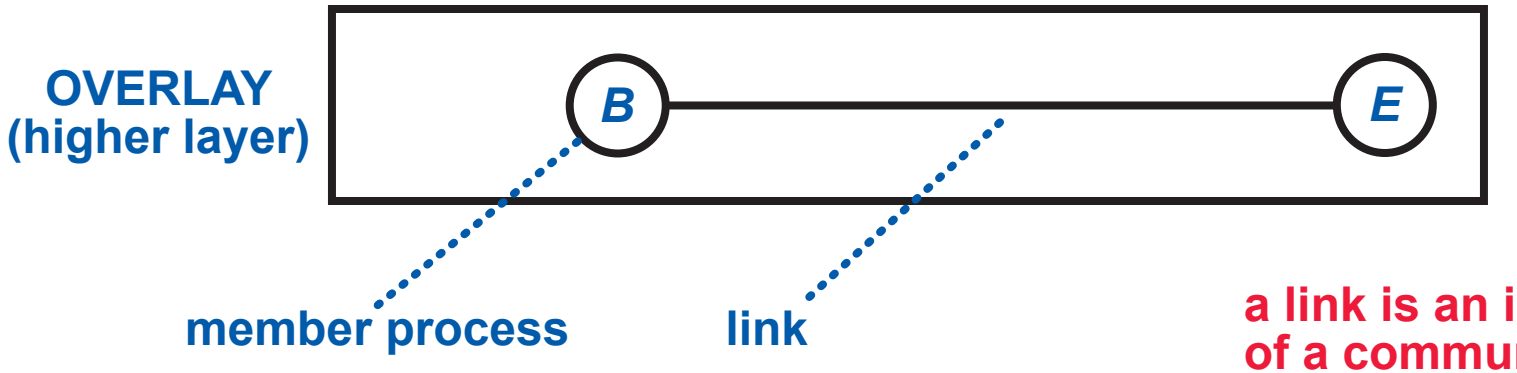
. . . and to regard communication through the hypervisor and softswitch of a physical machine as networked communication, and an object of study

LAYERS: MEMBERS, NAMES, AND ROUTING

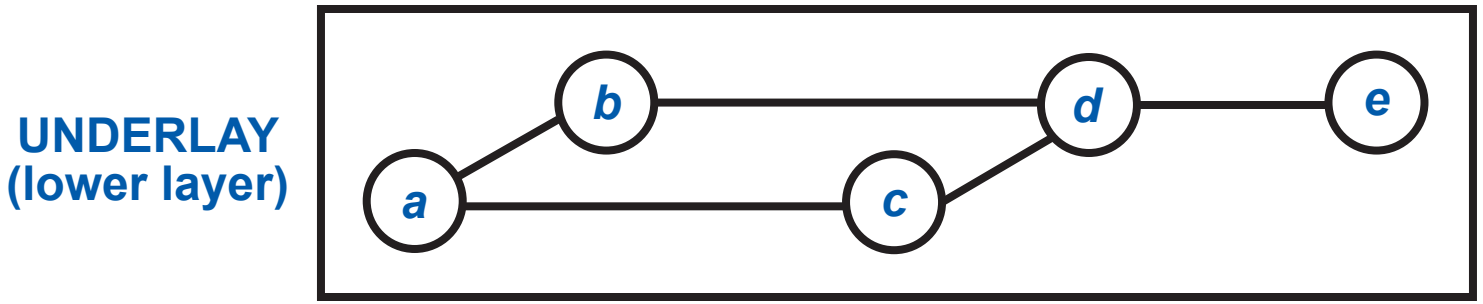
a member is a process that represents its machine in that layer

each layer has its own name space

a member has a name that is unique and permanent (although re-usable)



a link is an instance or usage of a communication service



members are connected to each other by links

because there is usually not a link between each pair of members, routes tell members how to reach each other

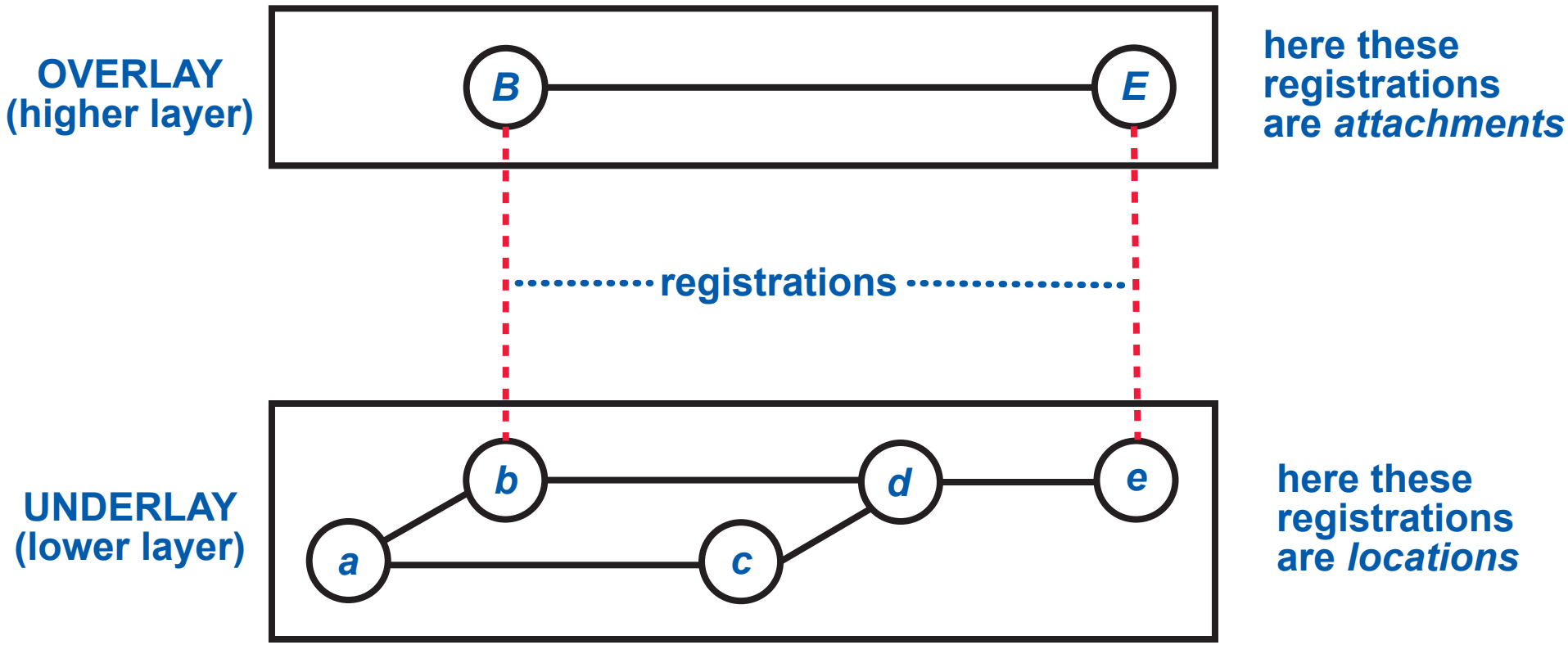
routing protocol maintains routes as links change

LAYERS: REGISTRATIONS

a registration maps an overlay process to an underlay process

both processes are on the same machine

the underlay process is a process in the lower layer that represents the overlay process to the network

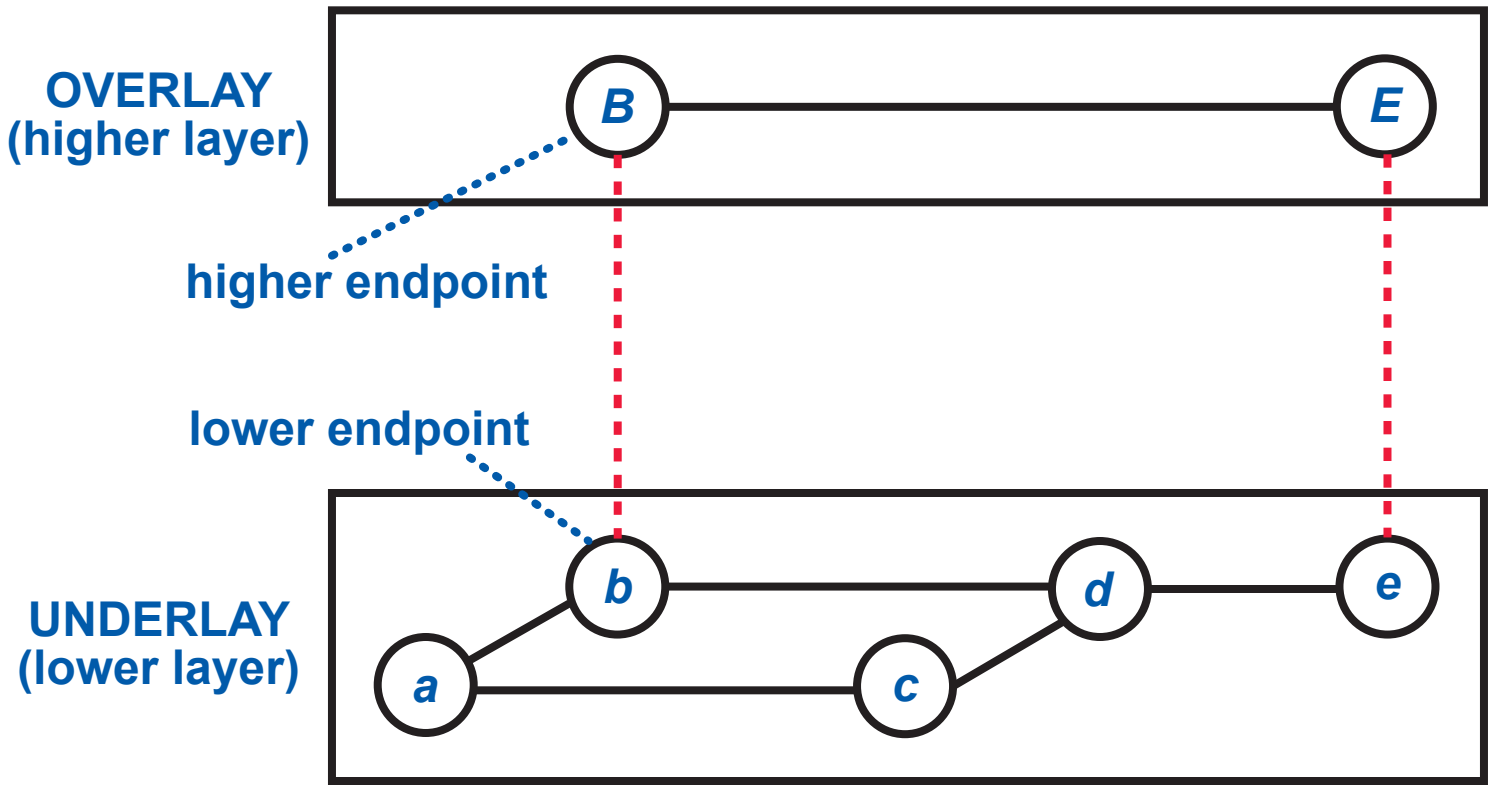


registrations can be created or destroyed by either layer

LAYERS: CHANNELS

a channel is an instance or usage of a communication service

a channel can be implemented as a service by an underlay for an overlay



in the overlay, the channel is called a *link*

in the underlay, the channel is called a *session*

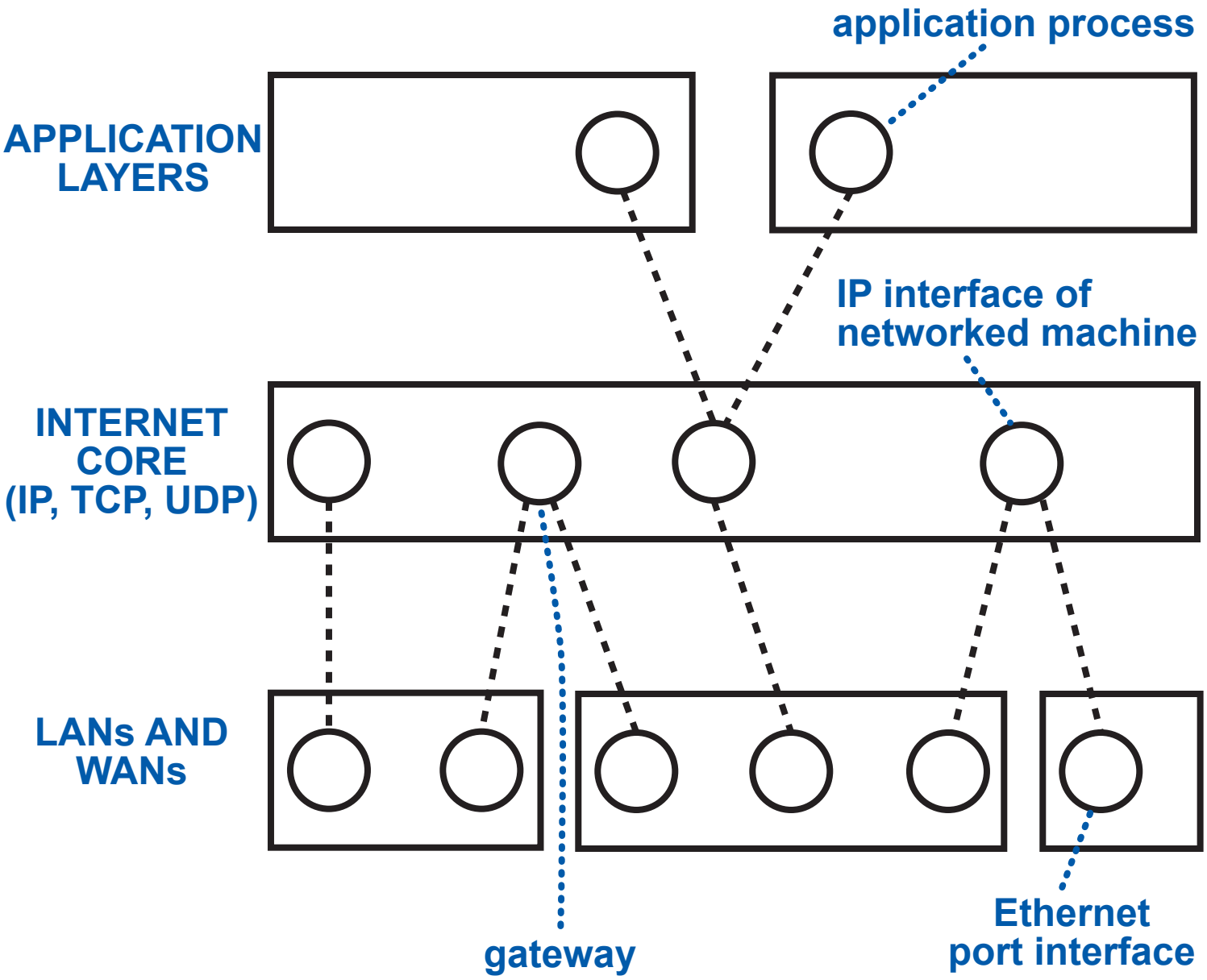
when *b* receives a channel request from *B* for *E*, it uses *locations* to find that *E* is located at *e*

underlay includes a transport protocol that enforces the service specification

LAYERS: SCOPE AND LEVEL

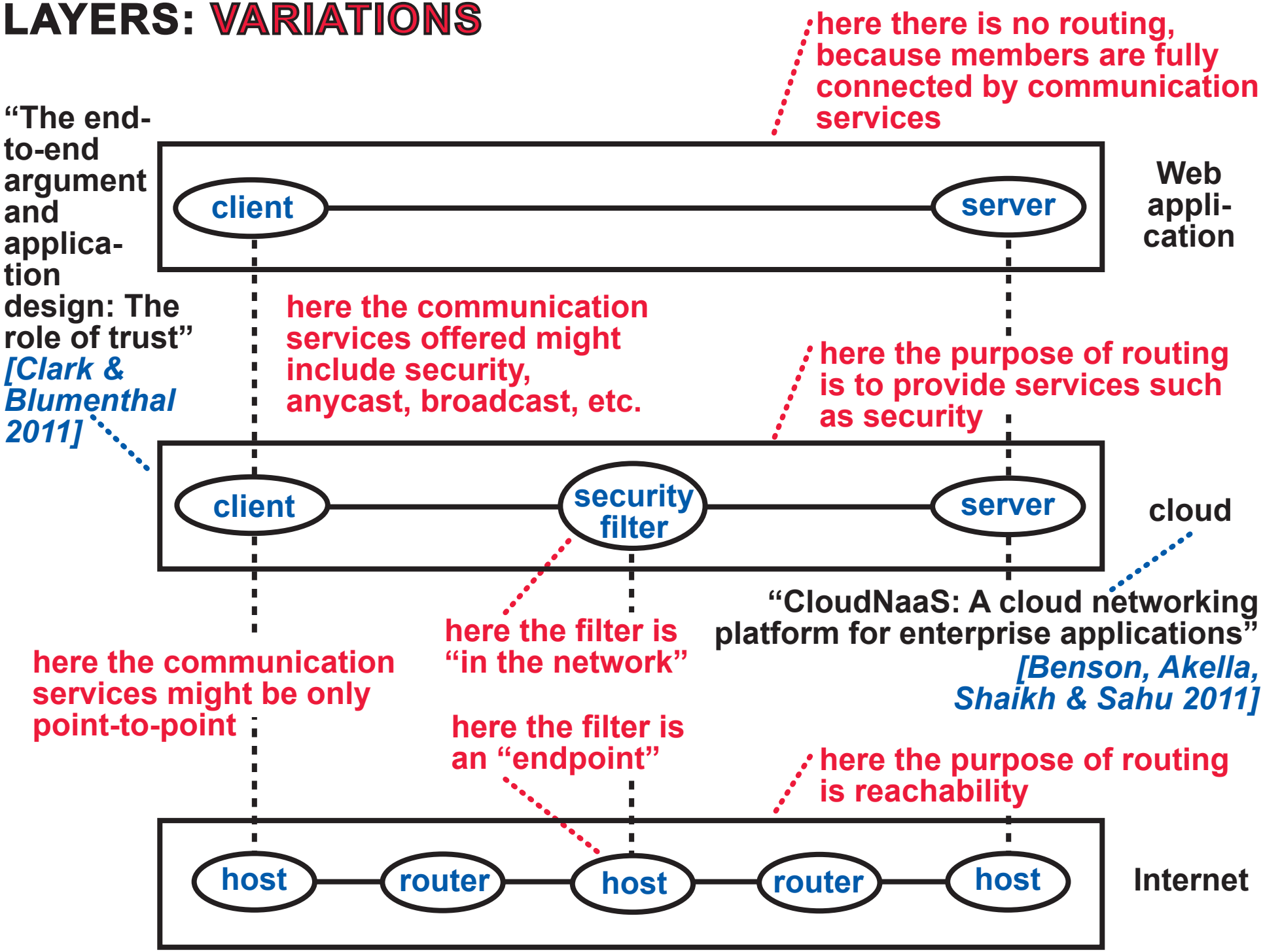
layers are arranged in a usage hierarchy, which defines levels

the scope of a layer is the set or class of processes that could be members



this is the geomorphic view of the classic Internet architecture

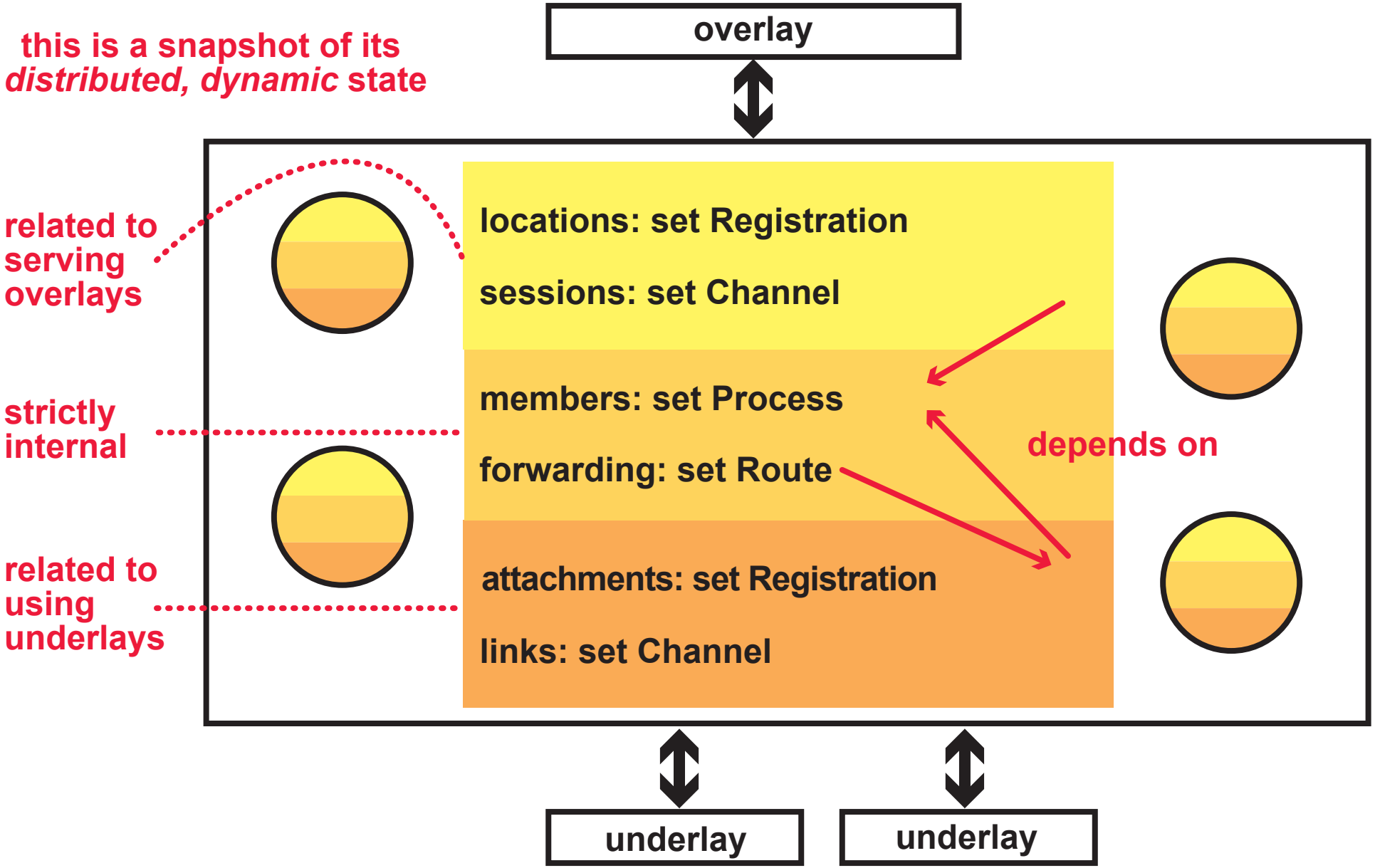
LAYERS: VARIATIONS



LAYERS: SOFTWARE STATE OF A LAYER

a layer is a distributed software system

this is a snapshot of its *distributed, dynamic* state



OUTLINE

1 Basic information about layers

2 Frequently-asked questions

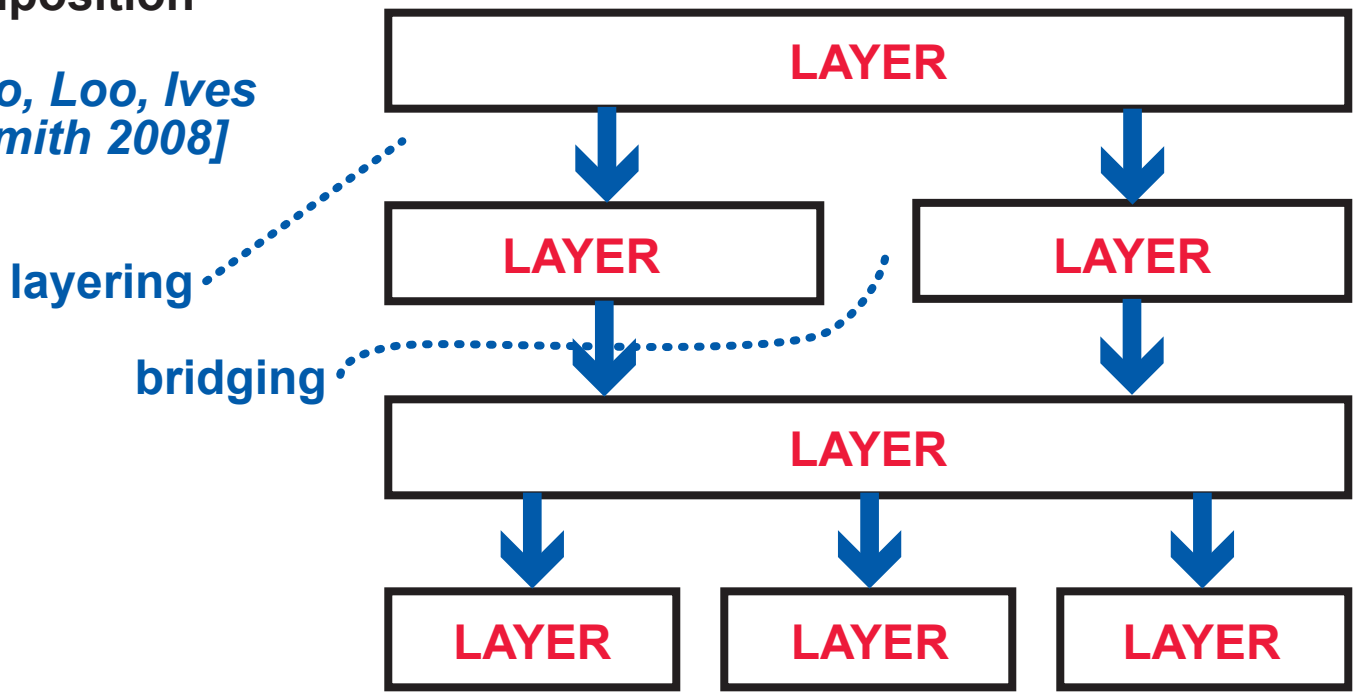
3 Examples

4 Summary and conclusions

FAQ: HOW IS THE GEOMORPHIC VIEW DIFFERENT FROM OVERLAYS?

"MOSAIC: Unified declarative platform for dynamic overlay composition"

[Mao, Loo, Ives & Smith 2008]



geomorphic view attempts to explain what is in each layer, as well as how they compose

geomorphic view has no unique reference point, so there is nothing for an "overlay" to be "over"

FAQ: IS THE GEOMORPHIC VIEW DESCRIPTIVE OR PRESCRIPTIVE?

FUNCTIONALLY, IT IS DESCRIPTIVE

- there should be no major function or design that cannot be described

HOWEVER, THERE ARE FEWER MECHANISMS THAN ARE FOUND "IN THE WILD"

- no arguing about names vs. identifiers vs. locators vs. addresses—each layer has one name space, designed and used for the purposes of the layer
- no tunneling used as an intra-layer exception to the routing system—just inter-layer interfaces

FEWER MECHANISMS COULD MEAN:

- each design has exactly one correct description
- designs can be compared easily
- it is possible to map out structured spaces of design trade-offs
- it is possible to get implementations by code generation and re-use

GOAL IS TO CHOOSE THE MECHANISMS THAT ARE THE BEST BECAUSE THEY FACILITATE . . .

. . . composition—of layers, mechanisms within a layer, or reasoning methods

. . . separation of concerns, so that diverse goals can be met without interfering with each other

OUTLINE

1 Basic information about layers

2 Frequently-asked questions

3 Examples

4 Summary and conclusions

EXAMPLE: COMPARING RESEARCH RESULTS

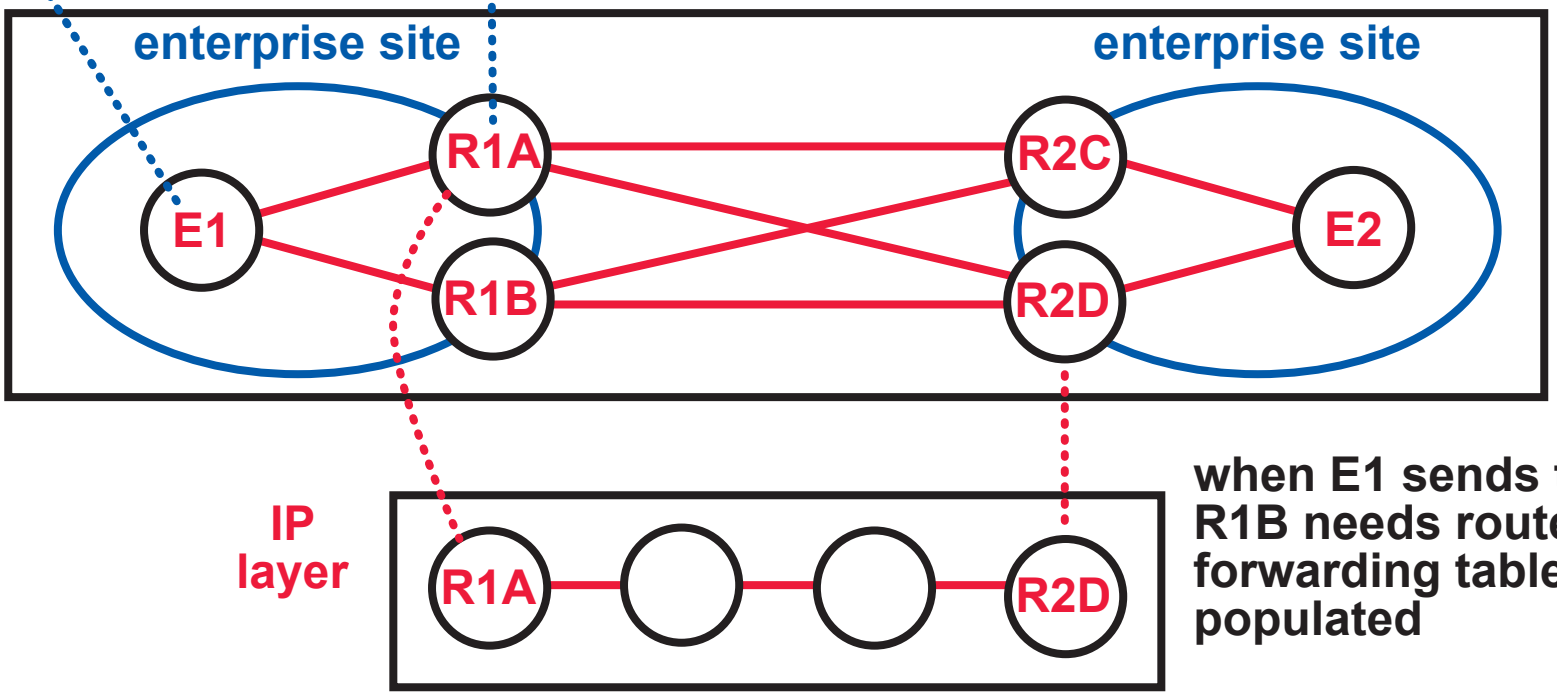
WHAT DO LISP AND SEATTLE HAVE IN COMMON?

Locator/Identifier
Separation Protocol

edge routers are
fully connected
by IP links

Endpoint
Identifier (EID)

RLOC (IP
address)

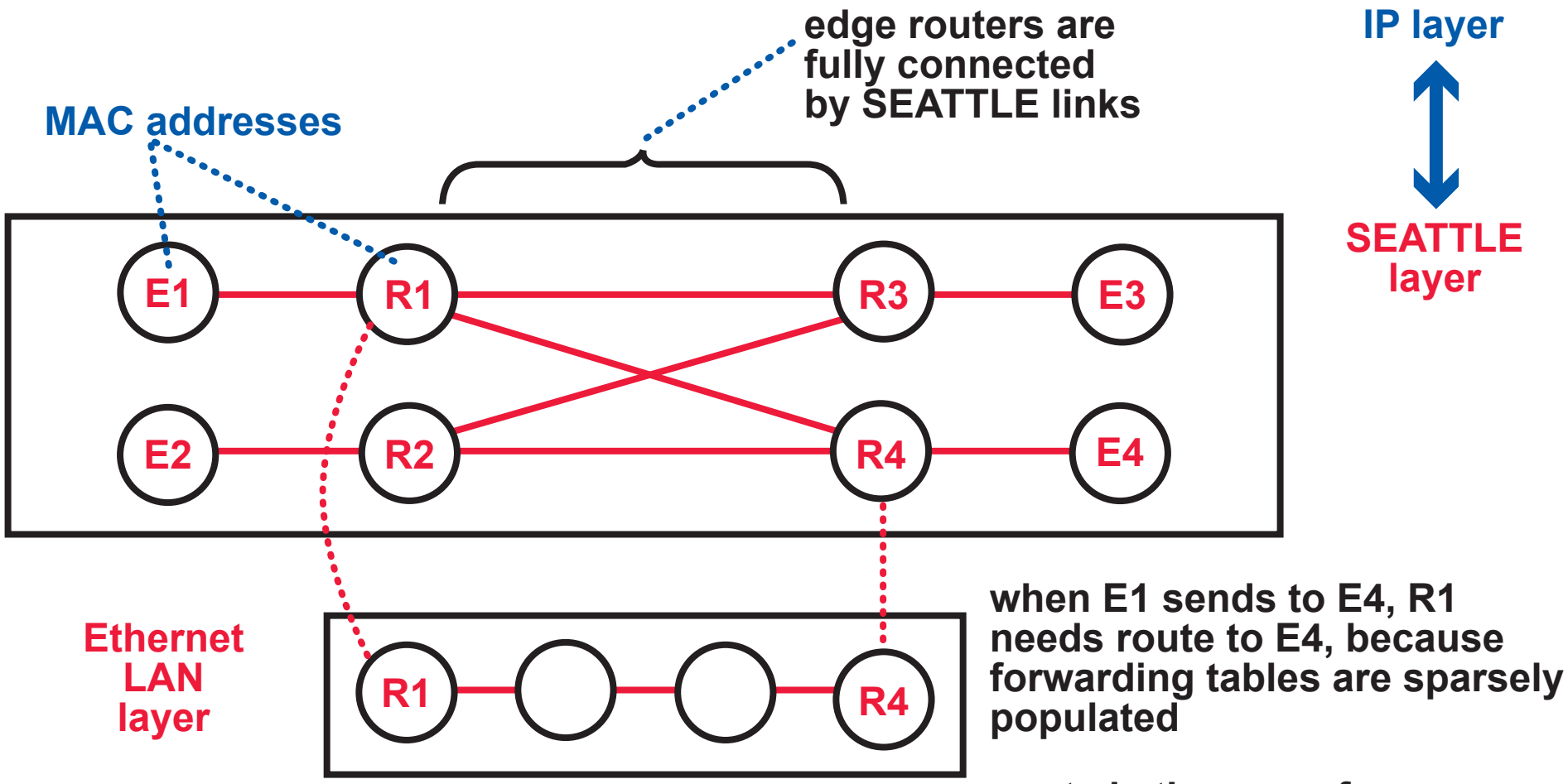


when E1 sends to E2, R1A or R1B needs route to E2, because forwarding tables are sparsely populated

route is (routes are) the same from every router, so the route can be obtained by directory lookup

WHAT DO LISP AND SEATTLE HAVE IN COMMON?

"Floodless in SEATTLE: A scalable Ethernet architecture for large enterprises"
[Kim, Caesar & Rexford 2008]



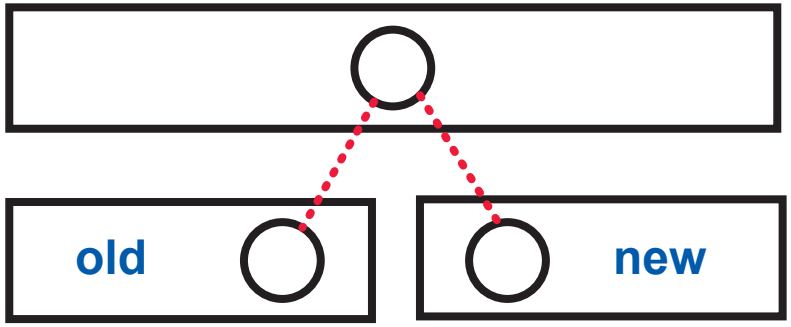
when E1 sends to E4, R1 needs route to E4, because forwarding tables are sparsely populated

route is the same from every router, so the route can be obtained by directory lookup

Nicira networks also have this structure; comparison focuses attention on the difference, which is how directories are implemented

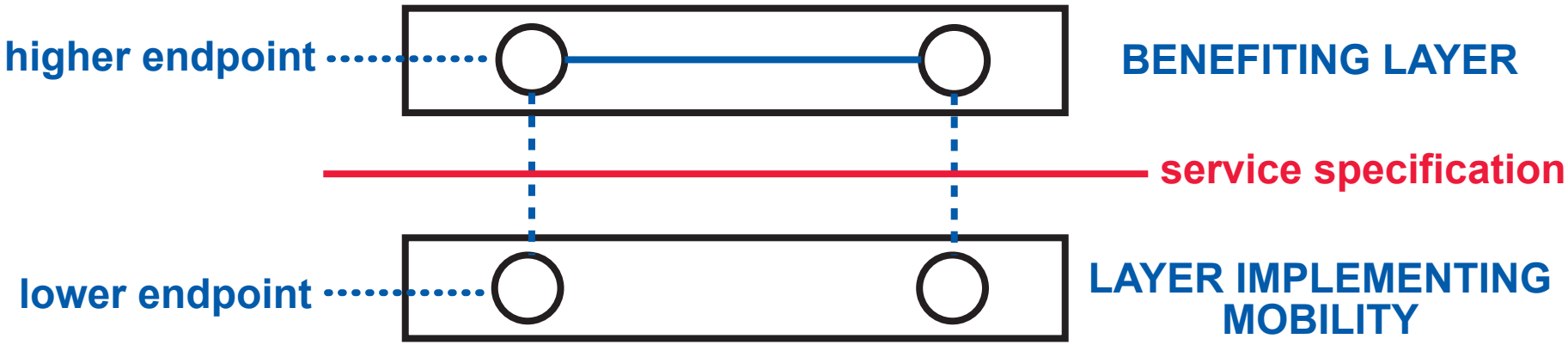
EXAMPLE: COMPOSITION OF MOBILITY MECHANISMS

AS A PROBLEM, NETWORK MOBILITY
IS A CHANGE IN REGISTRATION . . .



. . . of a process, while it is participating
in a channel

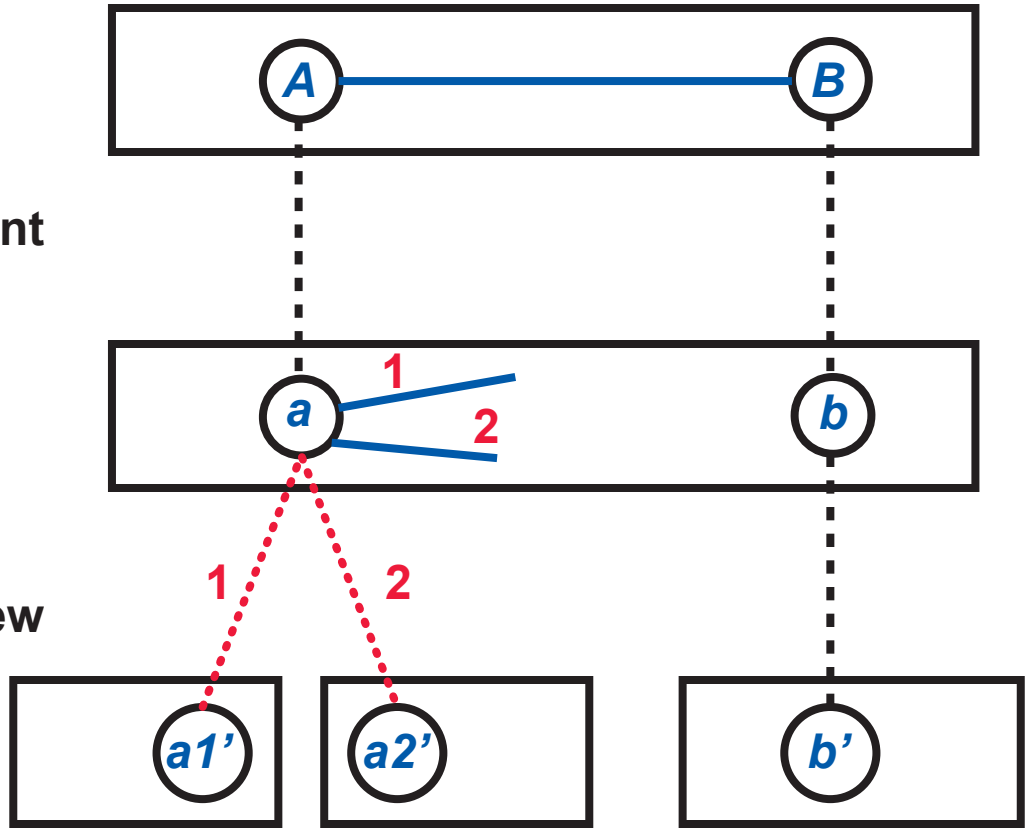
AS A SOLUTION, NETWORK MOBILITY
MAINTAINS CHANNELS, DESPITE THE MOBILITY
OF THE PROCESSES PARTICIPATING IN THEM



MOBILITY IMPLEMENTATION: ATTACHMENT MOBILITY

1
a becomes disconnected because attachment to *a1'* and links implemented through *a1'* fail

2
a forms a new attachment and new links through *a2'*



BENEFITING LAYER

LAYER IMPLEMENTING MOBILITY

the hard work of implementing attachment mobility is re-routing so that other processes can reach *a* through new links

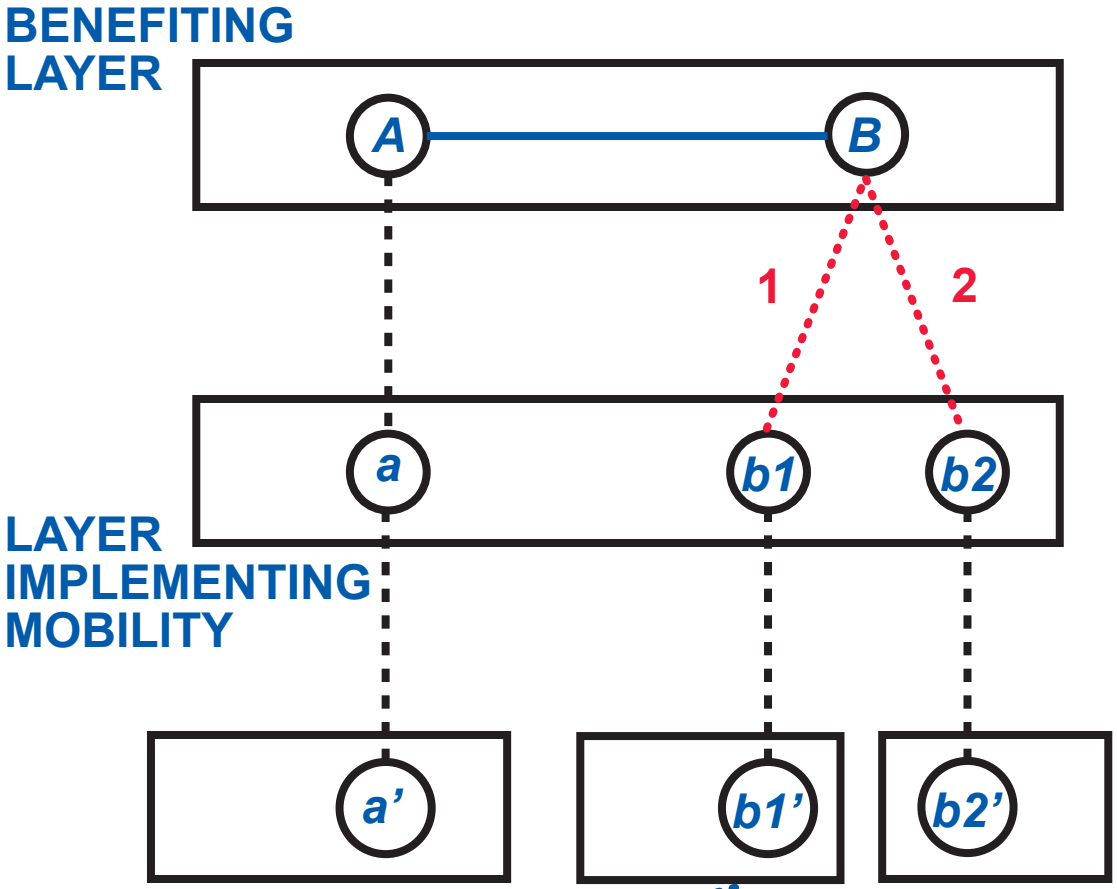
examples:

LANs/VLANs

Mobile IP

MSM-IP (uses IP multicast)

MOBILITY IMPLEMENTATION: LOCATION MOBILITY



1 *B* or *b1* destroys their registration

2 *B* or *b2* creates a new registration, session state transferred from *b1* to *b2*

unless this is a case of process migration, *b1* and *b2* are on same machine

the hard work of implementing location mobility is updating locations and *a*'s session state so that *B* can be found at *b2*

disconnection may come from this layer, but there is no mobility involving this layer

examples:

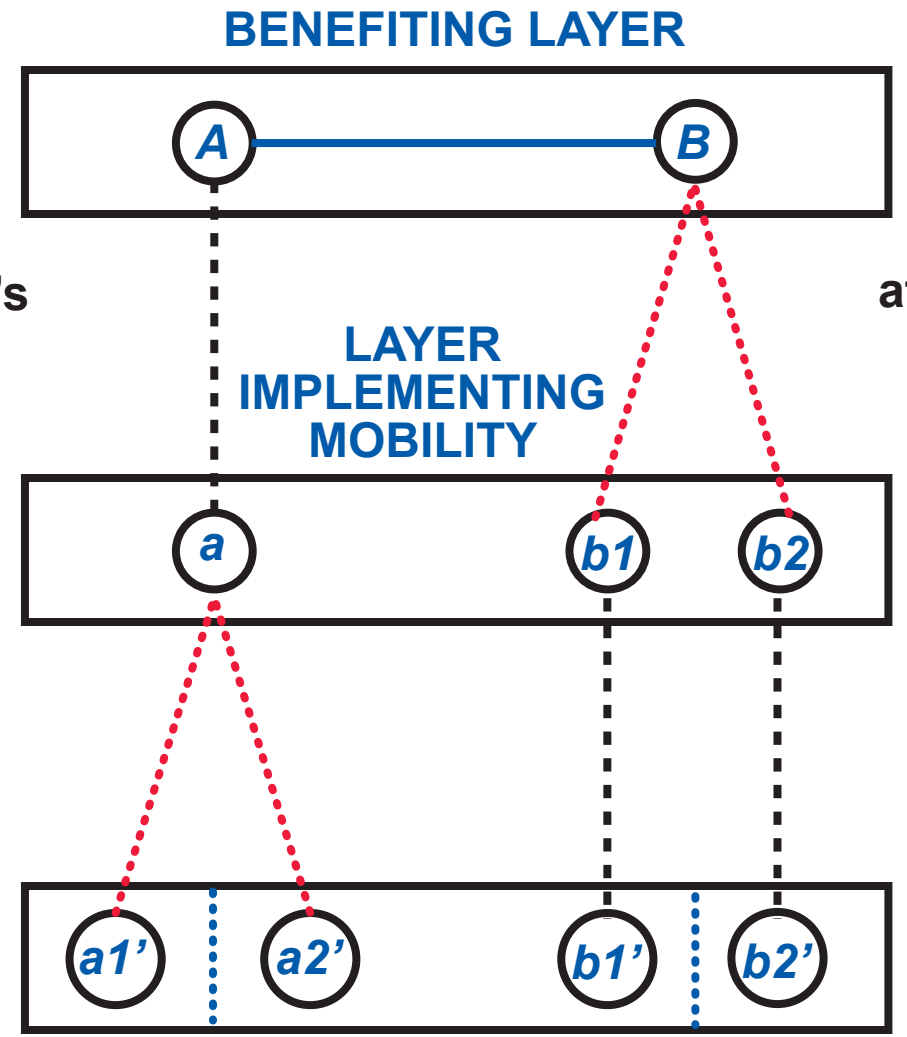
TCP Migrate

"Serval: An end-host stack for service-centric networking" *[Nordstrom, Shue, Gopalan, Kiefer, Arye, Ko, Rexford & Freedman 2012]*

ARE THEY REALLY DIFFERENT?

ATTACHMENT MOBILITY

LOCATION MOBILITY



registration change at implementing layer's lower interface

registration change at implementing layer's upper interface

distributed layer state affected:
attachments
links
forwarding

distributed layer state affected:
locations
sessions

EXAMPLE: COMPOSITION OF MOBILITY MECHANISMS

WITHIN A LAYER:

- mobility at one end of a session is independent of mobility at the other end—either one can be attachment or location mobility, even simultaneously
- at one end of a session, location mobility can take over if attachment mobility is failing

established by verification of a formal model of the session protocol, reasoning about the layer state

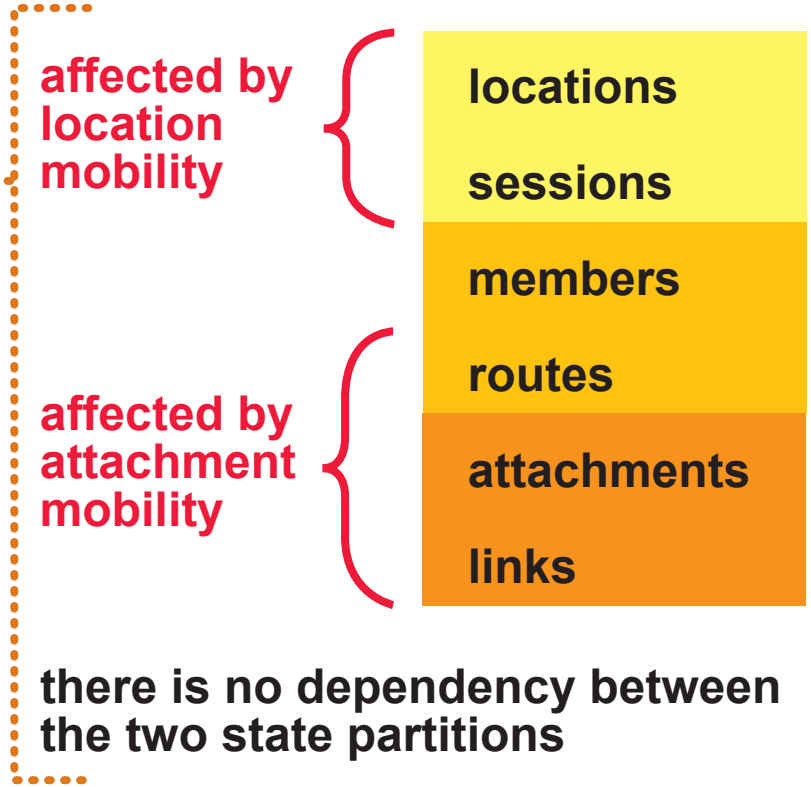
"Compositional network mobility"

[Zave & Rexford 2012]

ACROSS LAYER BOUNDARIES:

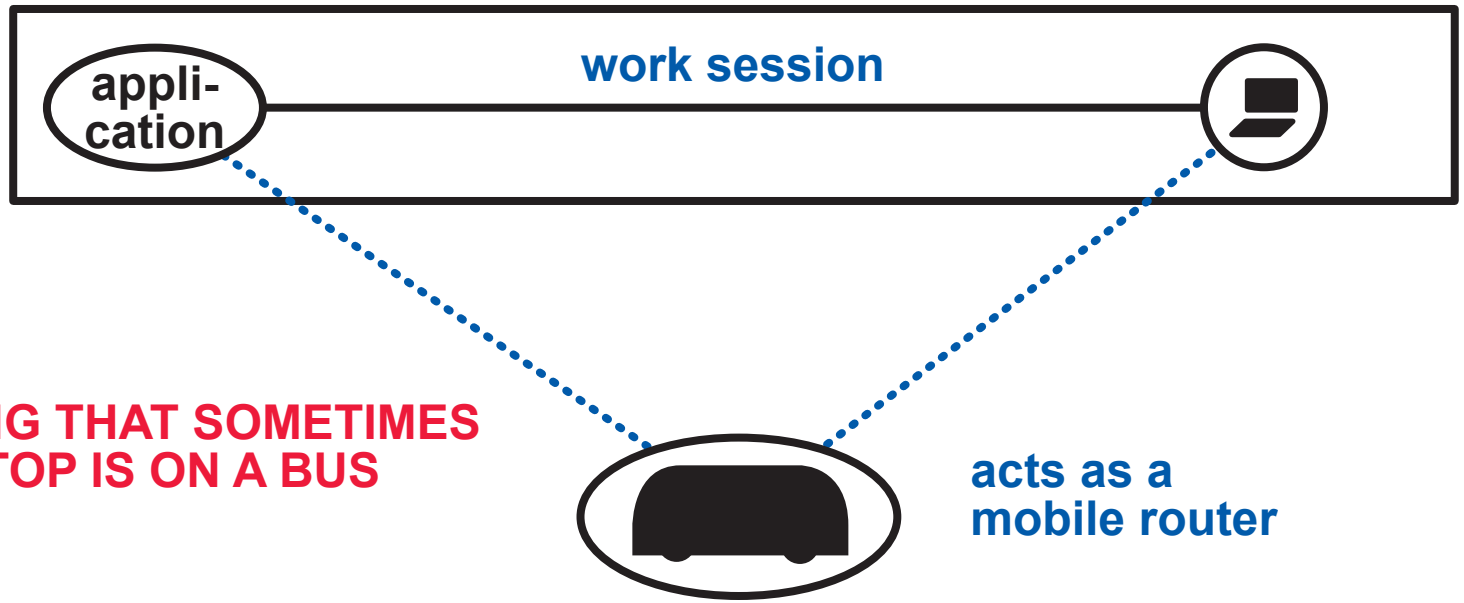
- mobility mechanisms in adjacent layers are logically independent, can co-exist and even operate simultaneously

established by reasoning about actions at the layer interface



EXAMPLE: NEW DESIGNS FROM THE MOBILITY SPACE

THE GOAL IS TO PROVIDE MOBILITY FOR THIS LAPTOP . . .



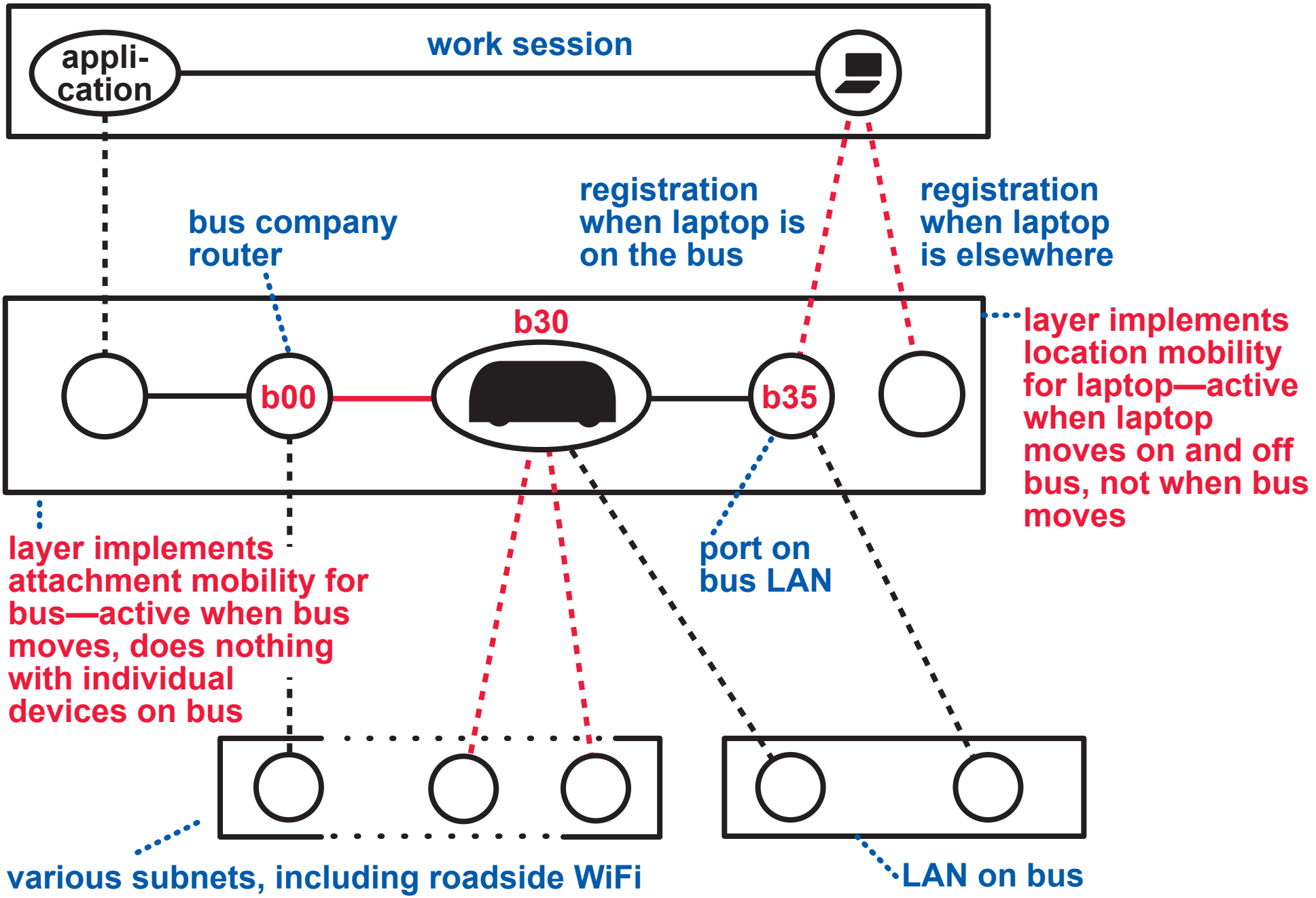
. . . NOTING THAT SOMETIMES THE LAPTOP IS ON A BUS

we want to avoid, *e.g.*, . . .

. . . solutions that require updates for every passenger when the bus moves

. . . solutions that require an update for the bus when a passenger gets on or off

EXAMPLE: NEW DESIGNS FROM THE MOBILITY SPACE



OUTLINE

1 Basic information about layers

2 Frequently-asked questions

3 Examples

4 Summary and conclusions

SUMMARY

SOME PURPOSES OF ABSTRACTIONS

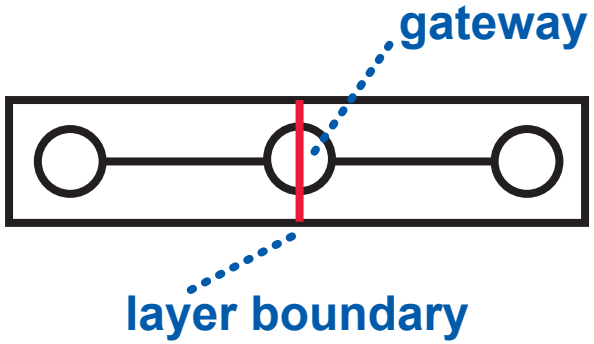
- to provide applications with richer communication services and cleaner interfaces to them
- to manage complexity with separation of concerns and use of general-purpose theories
- to compare and relate research results
- to map out structured spaces of design trade-offs
- to compose successful solutions to diverse problems
- to implement layers by means of code generation and code re-use

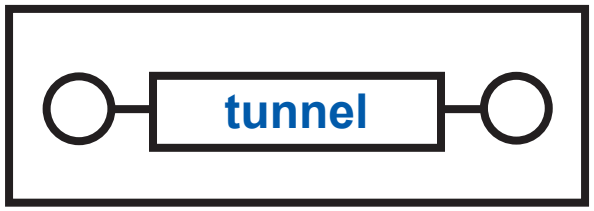
SOME CHARACTERISTICS (OR, MORE ACCURATELY, GOALS) OF THE GEOMORPHIC VIEW

- basic structures such as members, channels, and routing can look very different at different levels of the stack
- the same basic pattern or template is instantiated many times, for many different purposes, at different levels and scopes
- each design has exactly one correct description
- by partitioning the control state of a layer, facilitates composition of mechanisms within a layer
- by solidifying layer interfaces, facilitates composition of layers in a network stack

A BIG TRADE-OFF

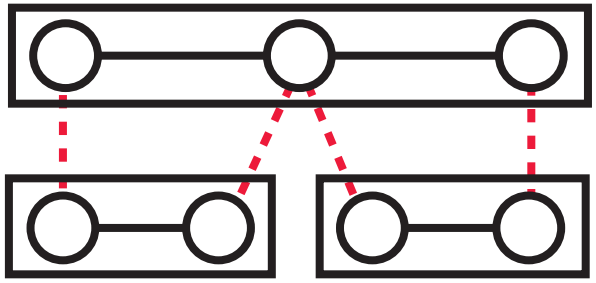
STRUCTURES YOU OFTEN SEE

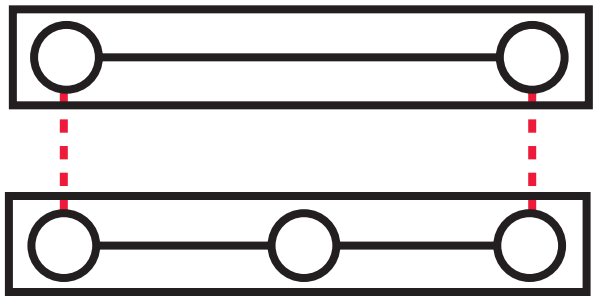




- each mechanism is *ad hoc*, and its interactions with other mechanisms are unpredictable

THE GEOMORPHIC VIEW OF THESE STRUCTURES





- interactions among mechanisms can be studied, will become well-known
- each layer can achieve multiple purposes

WHAT ABOUT THE REDUNDANCY AND OVERHEAD?

How much? Routers already have forwarding rules from multiple layers.

Redundancy and overhead can be removed by optimization—at the cost of less resilience to change.

THE GEOMORPHIC VIEW IS NEW AND IMMATURE

IN PARTICULAR, THE VIEW OF LAYER STATE (partitions, dependencies) IS VERY SIMPLISTIC . . .

. . . because the only issue we have studied in enough detail is mobility

OTHER ISSUES TO BE INVESTIGATED

- anycast, multicast, broadcast, etc.
- multihoming
- middleboxes
- enrollment
- authentication
- access control
- privacy
- failure recovery
- resource management

ON THE OTHER HAND, THESE TWO IDEAS NOW SEEM INTUITIVELY OBVIOUS:

- To understand the control plane, you must first understand the data plane.
- The geomorphic view of the data plane—
  consisting of "complete" layers that can be instantiated freely with different levels, scopes, and purposes—
  is a better abstraction of the data plane than the classic Internet architecture.