

MODELING THE INTERNET

Pamela Zave

AT&T Laboratories—Research

Florham Park, New Jersey, USA

MODELING THE INTERNET

OUTLINE

WHAT SHOULD BE MODELED?

overlays and aspects of overlays

covering functions across application, middleware, transport, and network levels of the current architecture

HOW SHOULD IT BE MODELED?

using "lightweight modeling" tools

using the right language for each purpose

WHY SHOULD IT BE MODELED?

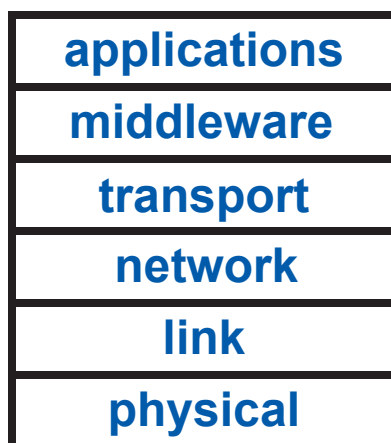
because functional modeling (as opposed to performance modeling) is almost unknown in networking . . .

. . . and is badly needed

THE STATE OF INTERNET ARCHITECTURE

THE "CLASSIC" INTERNET ARCHITECTURE

- defined in terms of layers with different functions



- designed to empower users and encourage innovation
- has succeeded beyond anyone's wildest dreams
- made obsolete by explosive growth in . . . users
 - . . . traffic
 - . . . applications
 - . . . security threats

THE REAL INTERNET

- does not meet current or future needs for . . .
 - . . . mobility,
 - . . . security,
 - . . . reliability,
 - . . . quality of service,
 - . . . scale,
 - . . . network management,
 - . . . balancing the interests of diverse stakeholders
- "classic" architecture is eroded badly by exceptions
- separation of concerns always loses to the desire for efficiency (or to tussles between stakeholders, or to anything else)
- as a result of all these factors, it is much too difficult to build, deploy, and maintain networked applications

THE IMPORTANCE OF OVERLAYS

Common definition: An *overlay* is a custom-built network layer deployed over existing layers.

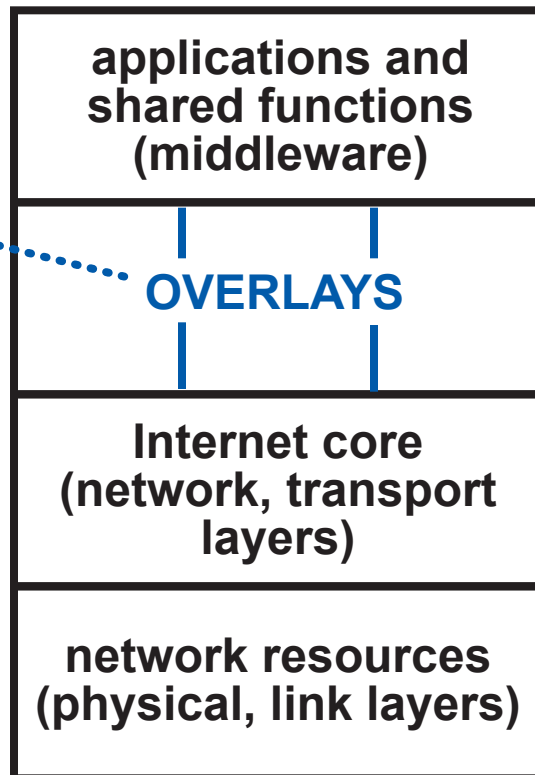
OVERLAYS ARE THE MODULES OF NETWORK ARCHITECTURE

AN OVERLAY IS A "CLEAN SLATE" FOR DESIGN

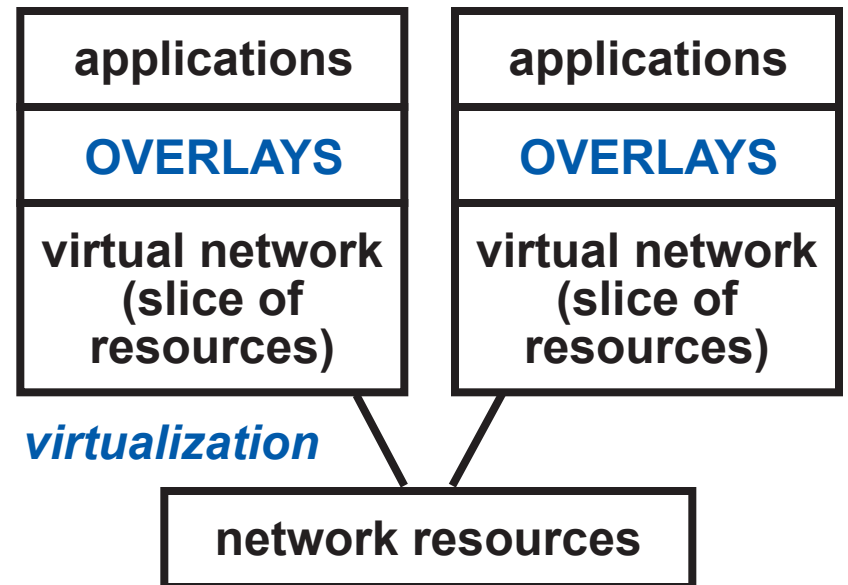
FOR TODAY:

FOR THE FUTURE:

frequently used to support applications better than the unvarnished Internet does



can experiment with new architectural ideas



in the end, there may be no universal Internet layer

A BETTER DEFINITION OF OVERLAYS

An overlay contains (potentially) all network functions.

There are many instances of the overlay type, varying in their rank (level) and scope (membership).

Overlays are arranged in a "uses" hierarchy.

Registration:

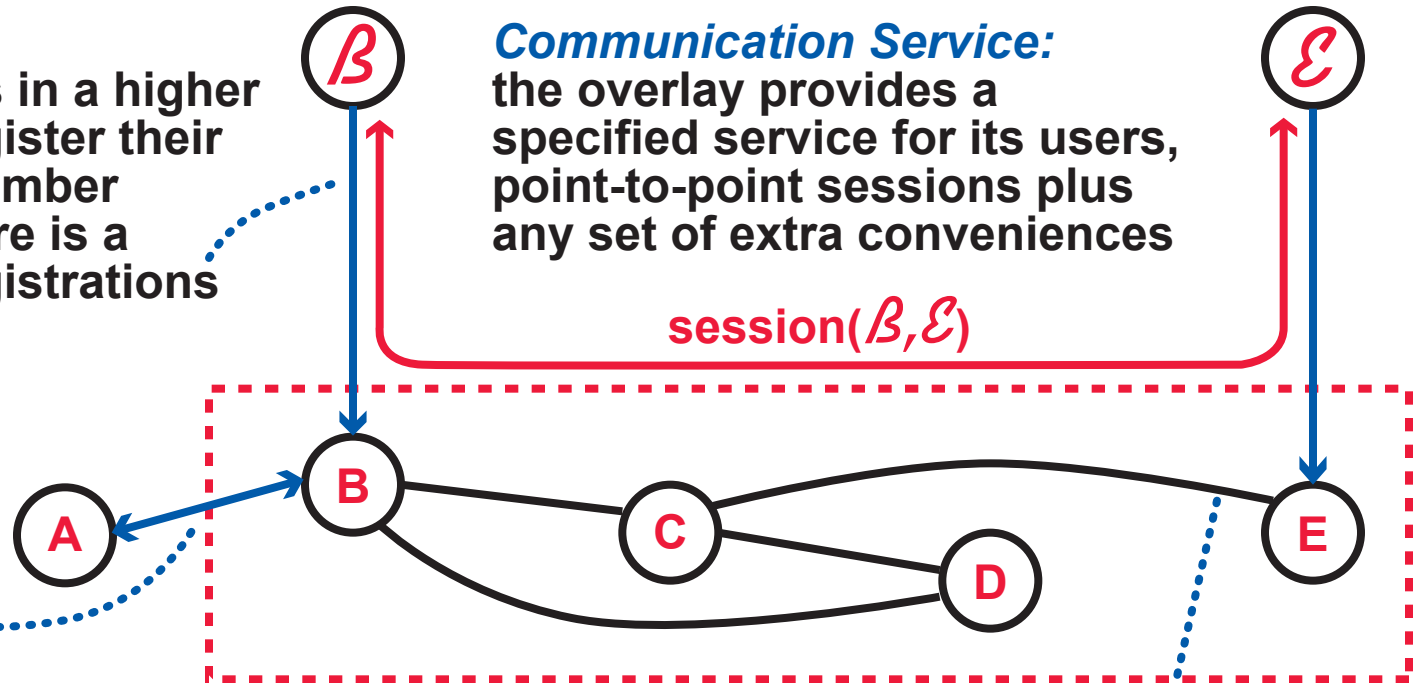
user processes in a higher overlay can register their locations at member processes; there is a directory of registrations

Communication Service:

the overlay provides a specified service for its users, point-to-point sessions plus any set of extra conveniences

Membership:

the members are processes; each has a unique and persistent name from the name space; enrollment protocol accepts and names new members



Routing:

any member can reach any other through a path in the overlay; routing protocol spreads knowledge of links and paths; forwarding protocol uses path knowledge

Links:

there is a link between two member processes if both are registered in the same lower overlay

**SECURITY
AND
RESOURCE
MANAGEMENT
THROUGHOUT**

MODELING THE INTERNET

OUTLINE

WHAT SHOULD BE MODELED?

overlays and aspects of overlays

covering functions across application, middleware, transport, and network levels of the current architecture

HOW SHOULD IT BE MODELED?

using "lightweight modeling" tools

using the right language for each purpose

WHY SHOULD IT BE MODELED?

because functional modeling (as opposed to performance modeling) is almost unknown in networking . . .

. . . and is badly needed

MODELING THE INTERNET

OUTLINE

WHAT SHOULD BE MODELED?

overlays and aspects of overlays

covering functions across application, middleware, transport, and network levels of the current architecture

HOW SHOULD IT BE MODELED?

using "lightweight modeling" tools

using the right language for each purpose

WHY SHOULD IT BE MODELED?

because functional modeling (as opposed to performance modeling) is almost unknown in networking . . .

. . . and is badly needed

FURTHER READING

[Day 2008]

John Day, *Patterns in Network Architecture*, Prentice Hall, 2008.

[Zave 2010]

Pamela Zave, Internet evolution and the role of software engineering, *Proceedings of the Symposium on the Future of Software Engineering*, Springer Lecture Notes in Computer Science, 2010.

[Zave & Rexford 2011]

Pamela Zave and Jennifer Rexford, *The architecture of mobility*, in preparation.