Architecture

Mark Allman
mallman@case.edu

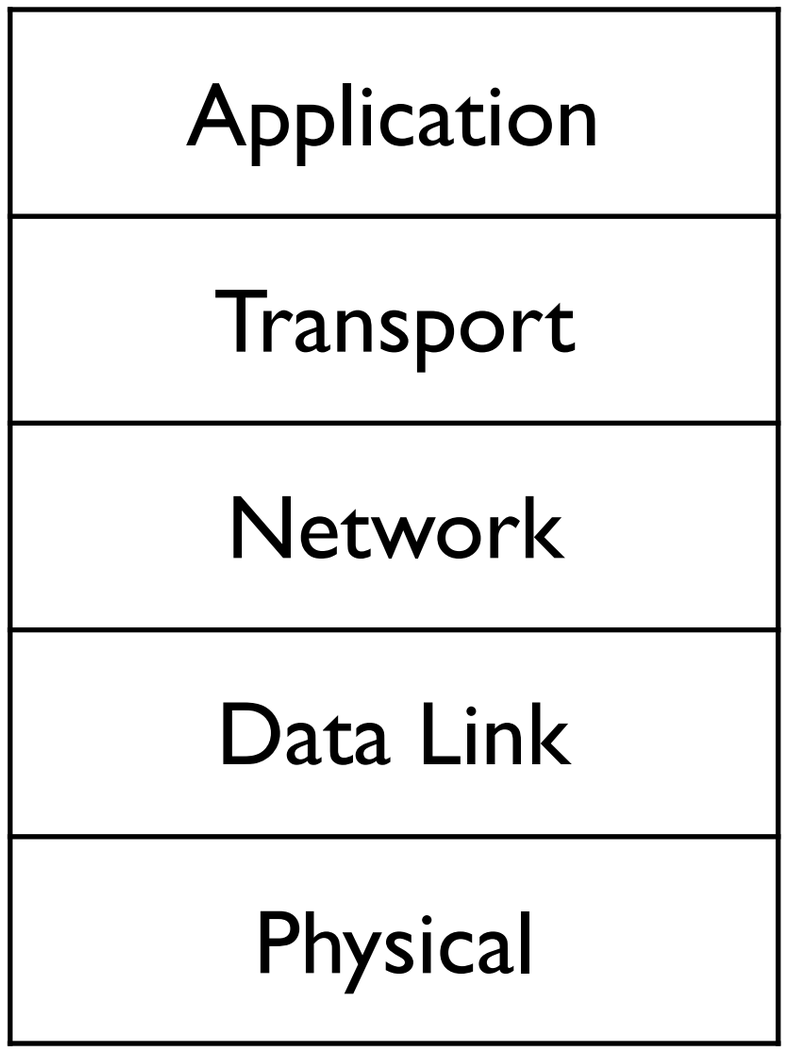
EECS 325/425
Fall 2018

Architecture

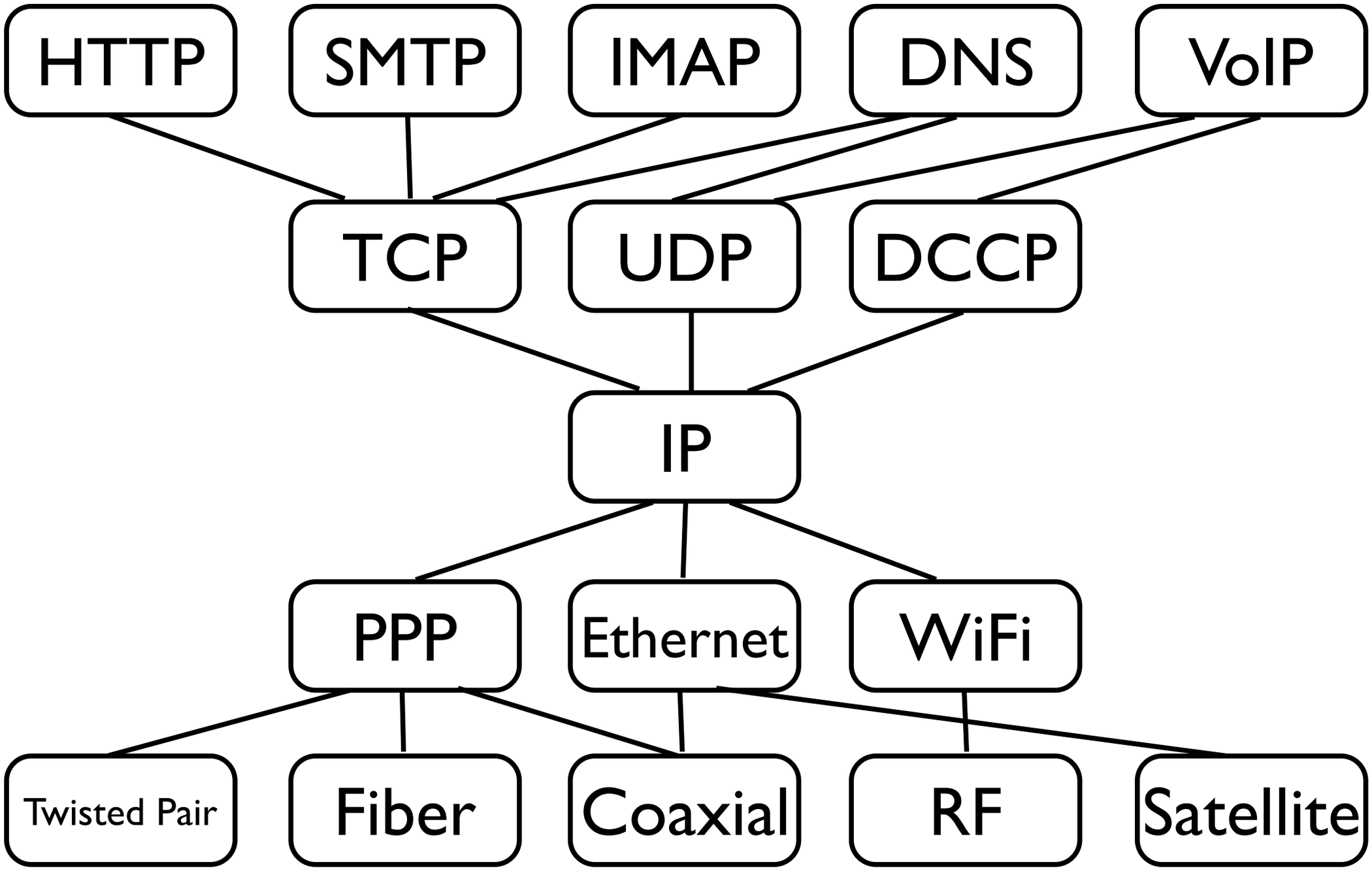
- How we organize and think about networks
- Including the key abstractions we use

E.g., Layering

E.g., Layering



E.g., The Thin Waist



E.g., End-to-End Principle

E.g., End-to-End Principle

- We can only ever really trust the endpoints
  - e.g., for reliability
  - e.g., for security

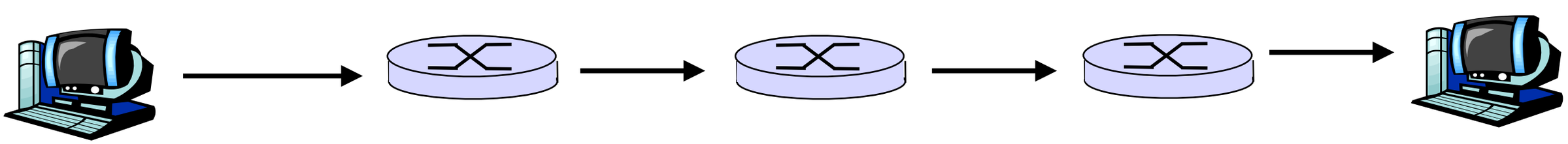
E.g., End-to-End Principle

- We can only ever really trust the endpoints
  - e.g., for reliability
  - e.g., for security

- Nodes in the middle—switches, routers, caches, firewalls, etc.—can help …
- … but do not obviate the need for end points to do these jobs

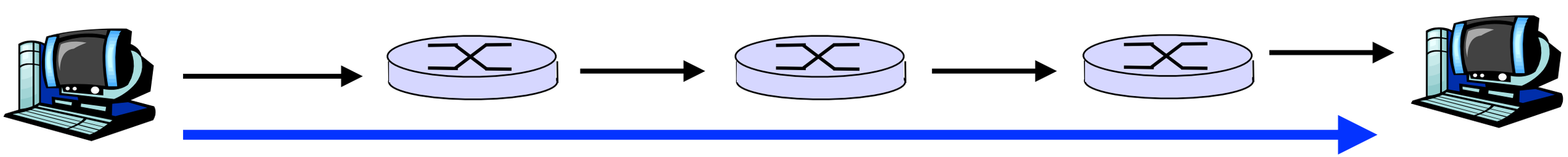
E2E Reliability

Goal: reliable transmission



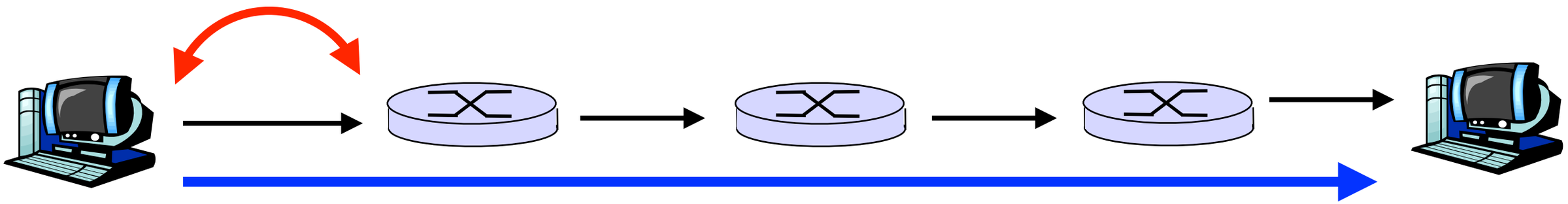
E2E Reliability

Goal: reliable transmission



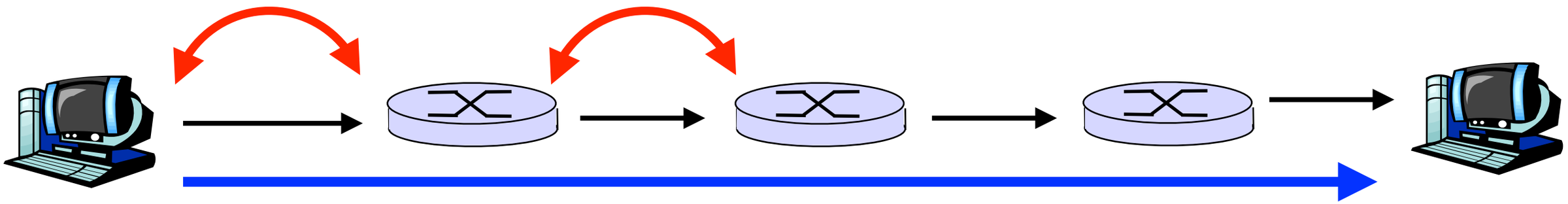
E2E Reliability

Goal: reliable transmission



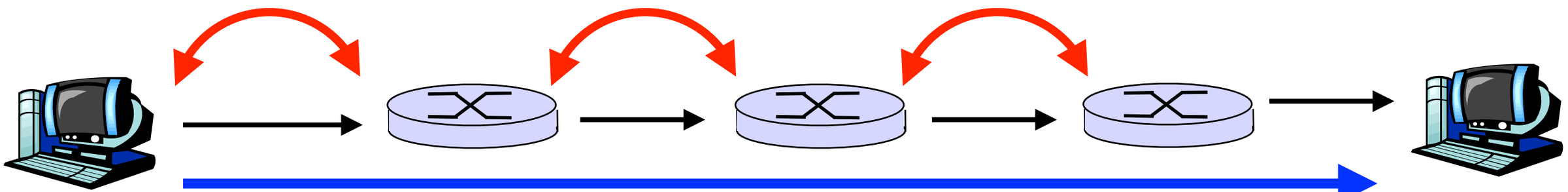
E2E Reliability

Goal: reliable transmission



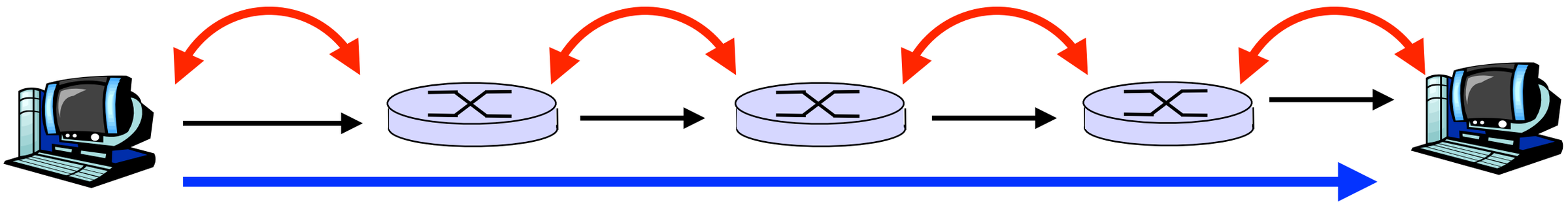
E2E Reliability

Goal: reliable transmission



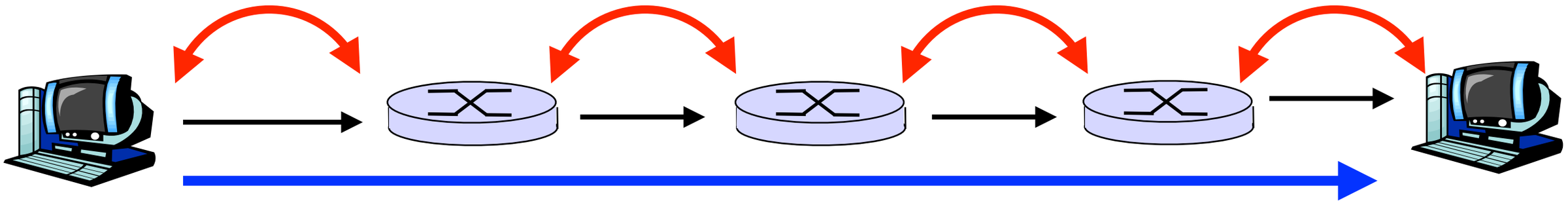
E2E Reliability

Goal: reliable transmission



E2E Reliability

Goal: reliable transmission



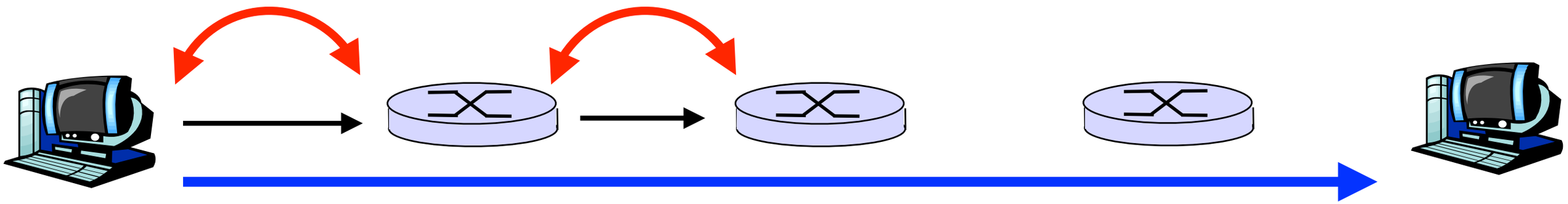
Hop-by-hop reliability

vs.

End-to-end reliability

E2E Reliability

Goal: reliable transmission

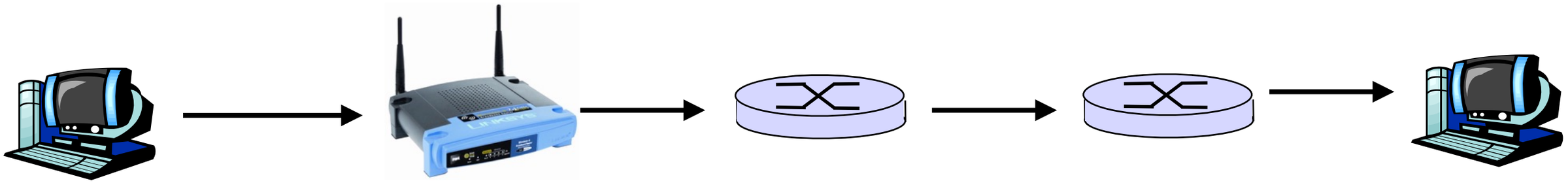


Hop-by-hop reliability

vs.

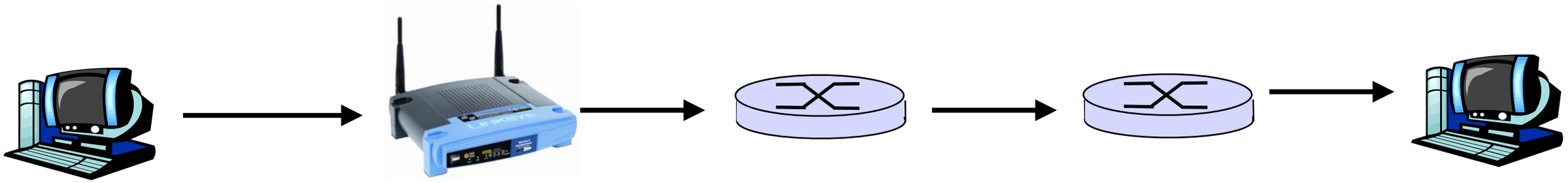
End-to-end reliability

E2E Security

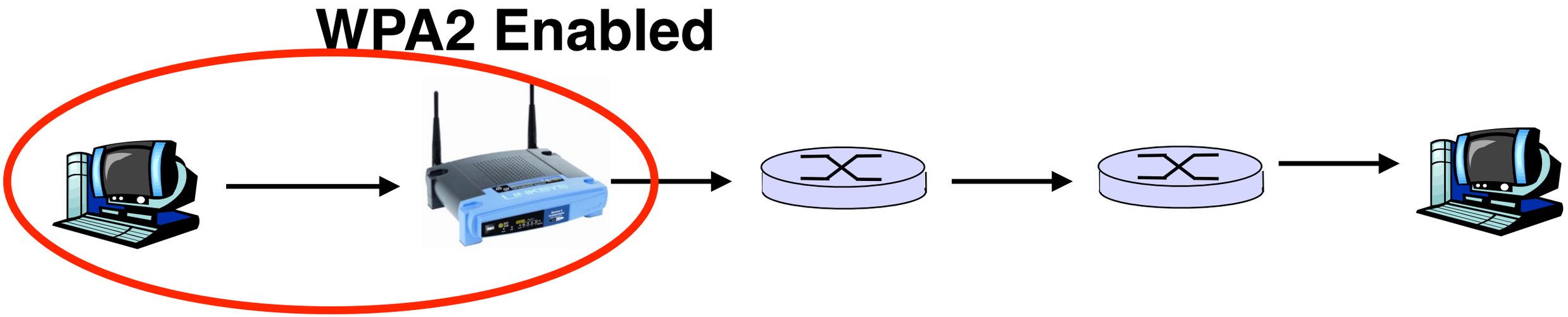


E2E Security

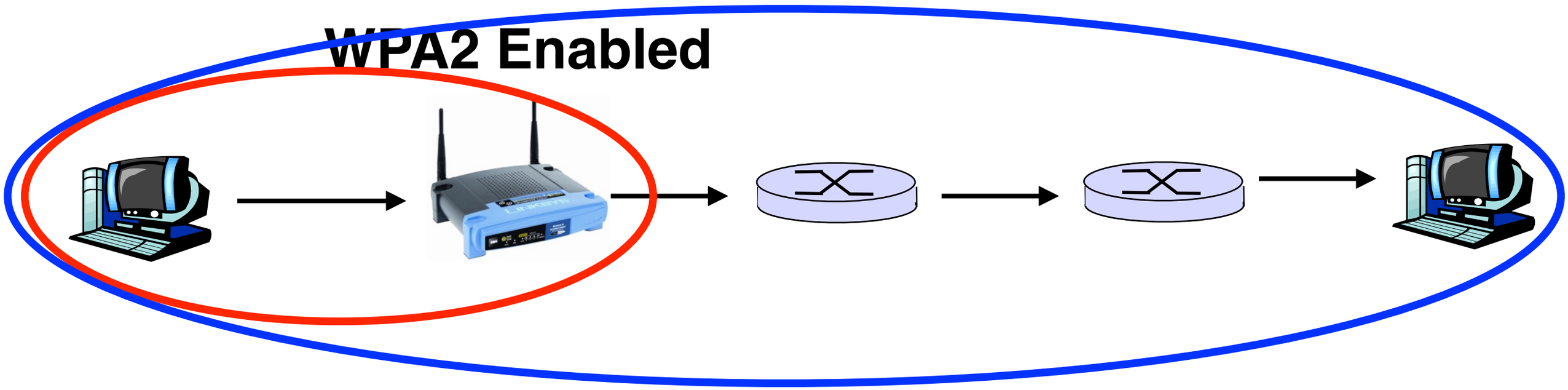
WPA2 Enabled



E2E Security



E2E Security



E.g., Soft State

- We prefer *soft state* to *hard state*
- To the extent possible we want the system to learn what it needs to operate on-the-fly rather than hard-code information into the system

Soft State: IP Addr. Assignment

- DHCP is built on soft state
  - we connect to a network and request an IP address (etc.)
  - we configure the end system's local state using information in the DHCP response
  - later, the state times out and is removed
    - (if not actively renewed)

E.g., Generality

E.g., Generality

- We prefer general solutions to point solutions

E.g., Generality

- We prefer general solutions to point solutions

- E.g., HTTP is not tuned to deliver text
  … or videos
  … or images

E.g., Generality

- We prefer general solutions to point solutions

- E.g., HTTP is not tuned to deliver text
  … or videos
  … or images

- We can nearly always "do better"—in some dimension—with point designs
  - but, then we add the complexity of a zillion designs for a zillion tasks