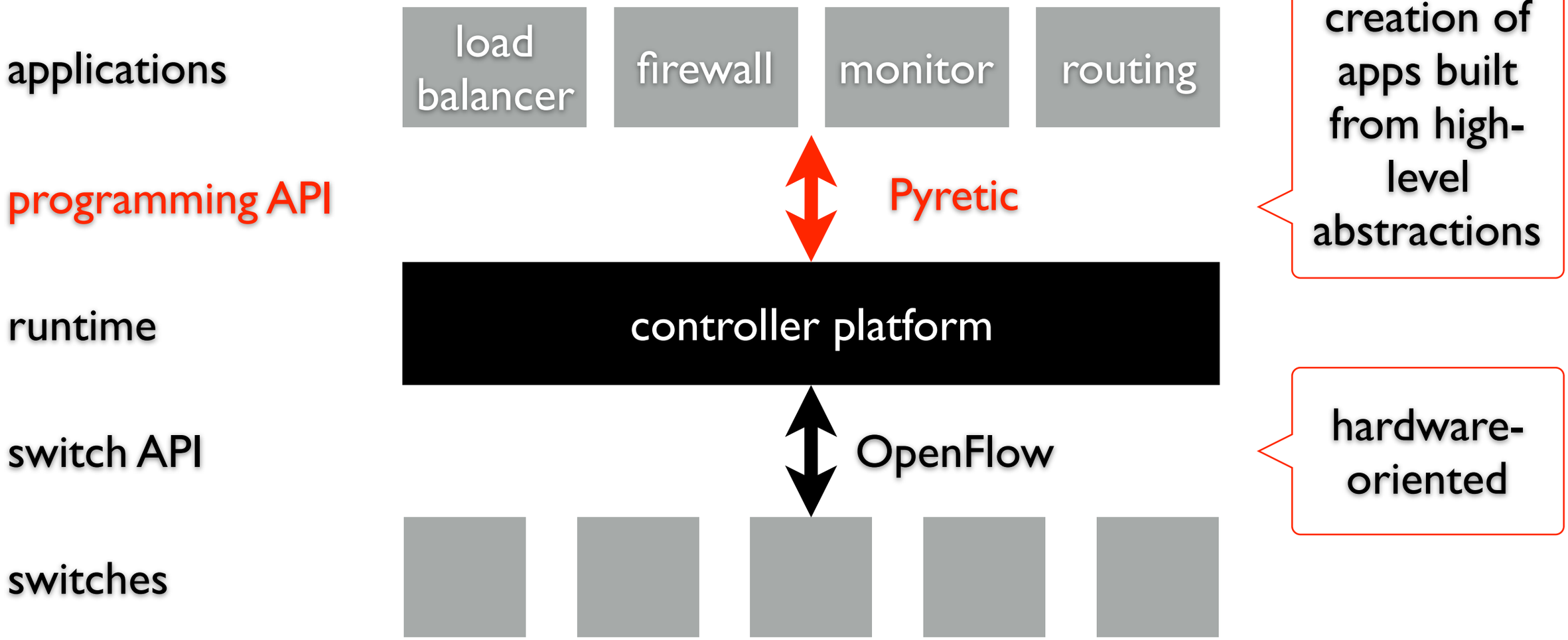
lecture 08:
state management

5590: software defined networking

anduo wang, Temple University

TTLMAN 401B, R 17:30-20:00

OpenFlow, Pyretic

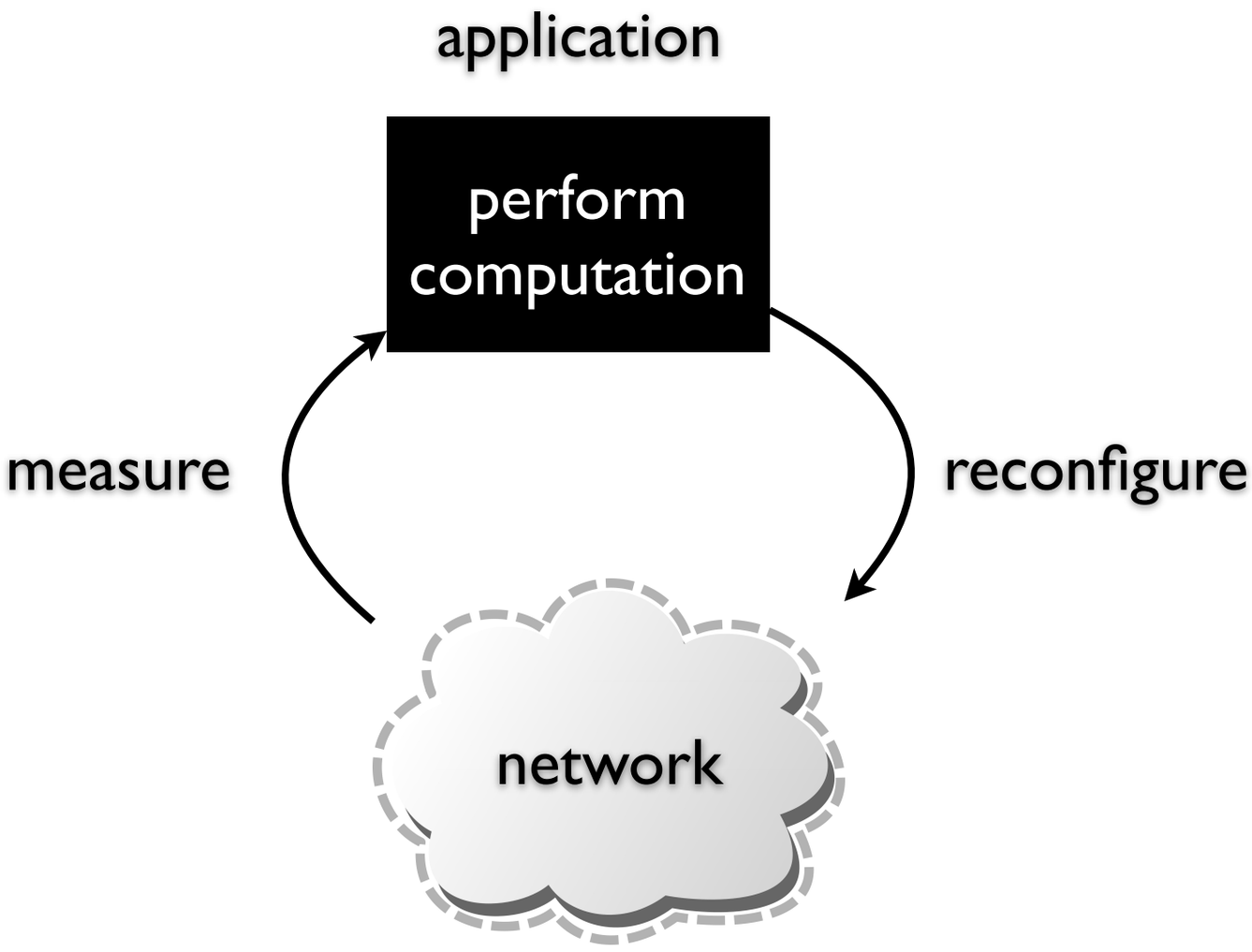


datacenter network (DCN)

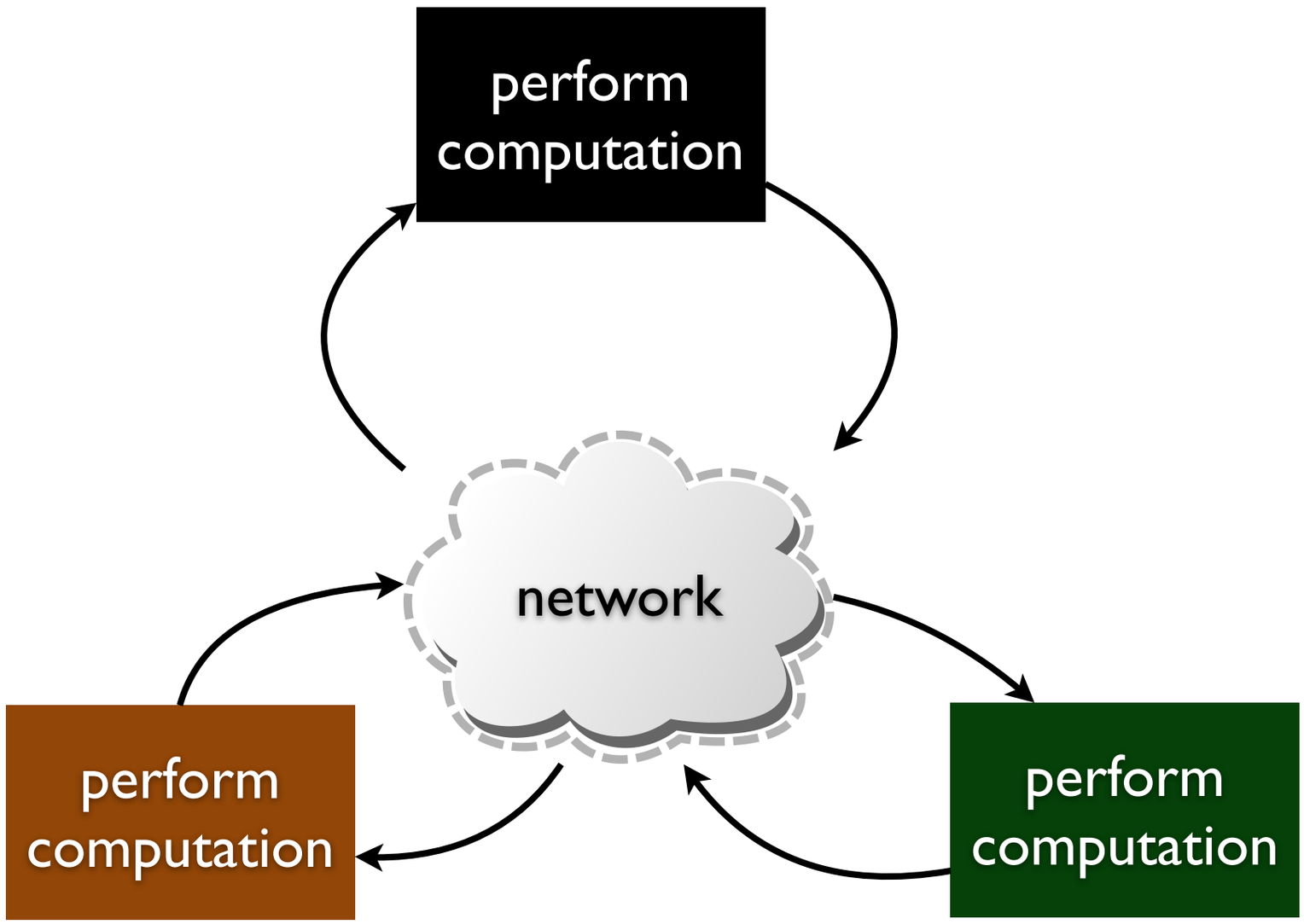
runs multiple management applications

- traffic engineering
- server load balancing
- network virtualization
- infrastructure
- failure recovery *NetPilot [SIGCOMM'12]*
- energy saving *Elastic tree [NSDI'10]*
- switch configuration

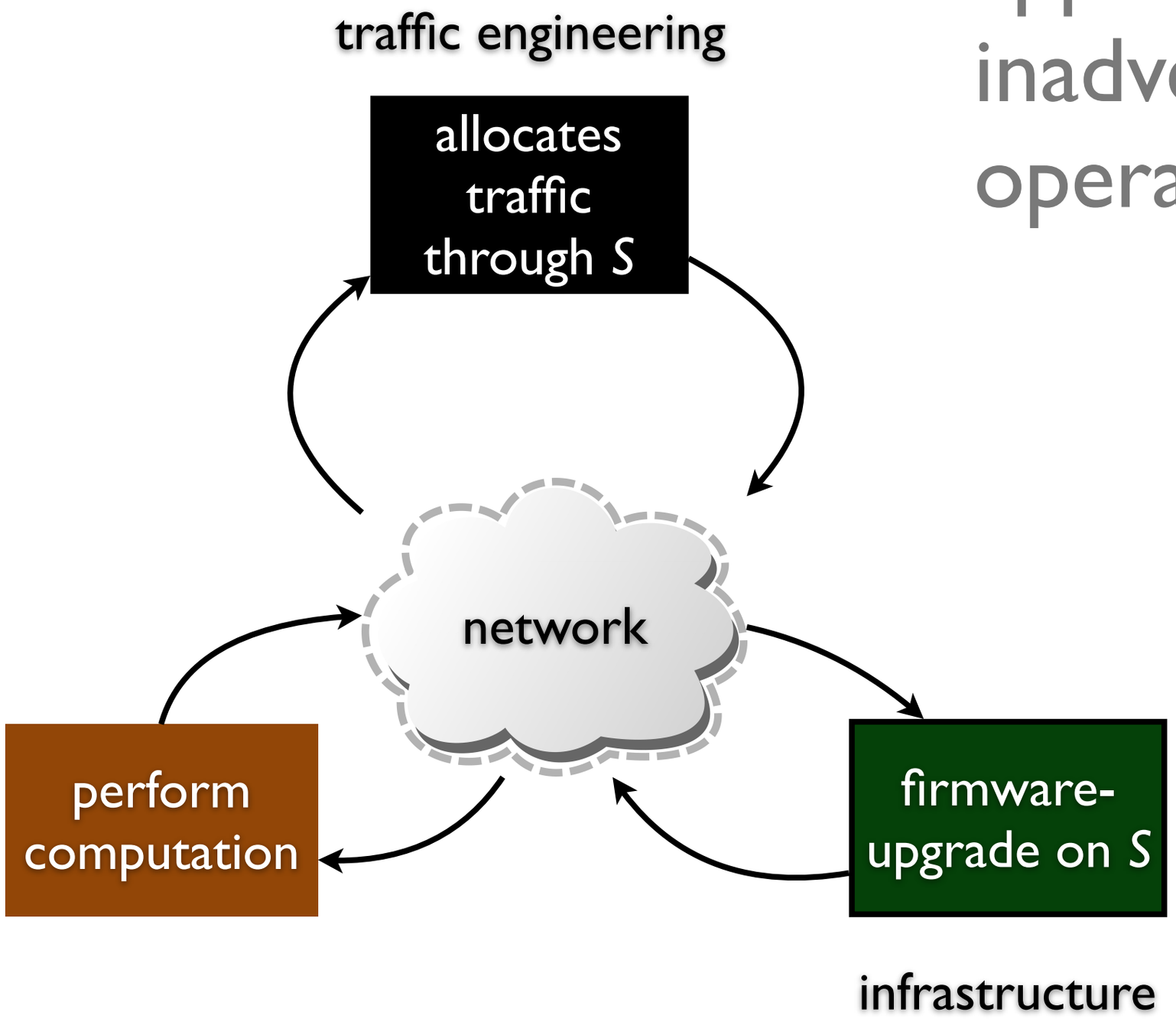
management applications



running multiple applications

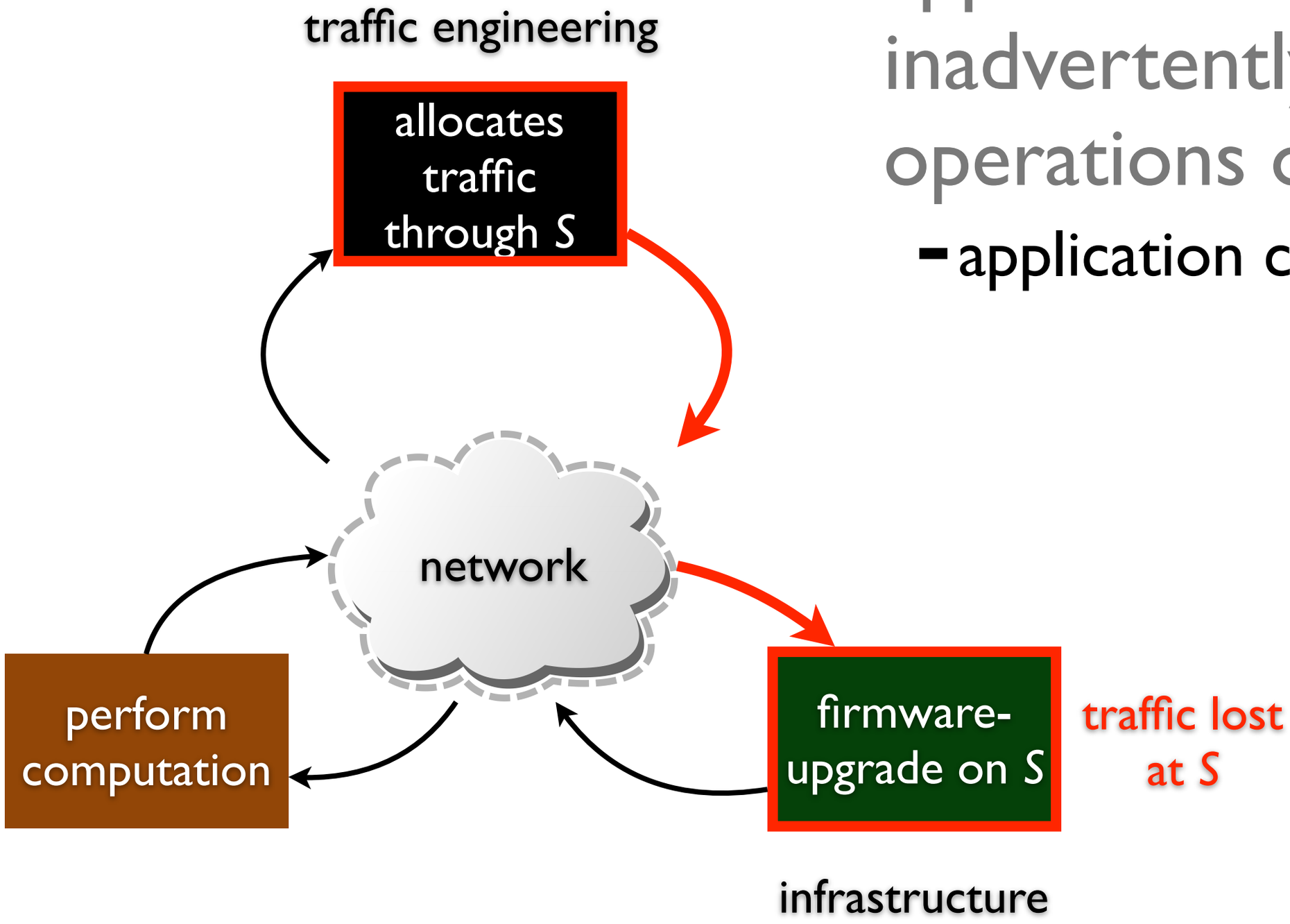


running multiple applications



applications can inadvertently affect the operations of another

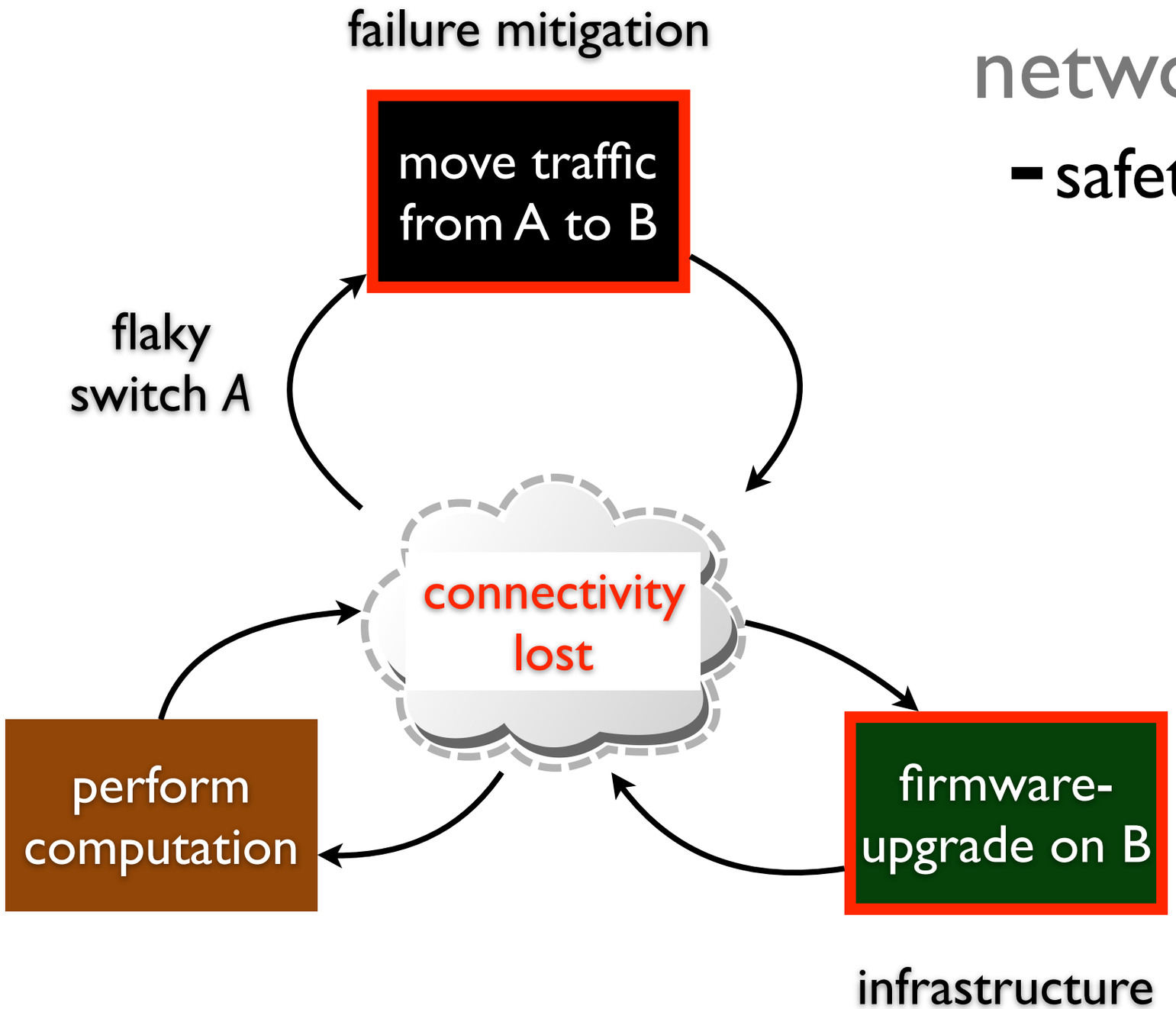
running multiple applications



applications can inadvertently affect the operations of another

- application conflict

running multiple applications



combined effects lead to network-wide failures

- safety failure

alternative to running multiple applications

one single monolithic application

- complex
- explicit coordination
- high overhead on applications

alternative to running multiple applications

one single monolithic application

- complex
- explicit coordination
- high overhead on applications

tightly coupled, repeated extension

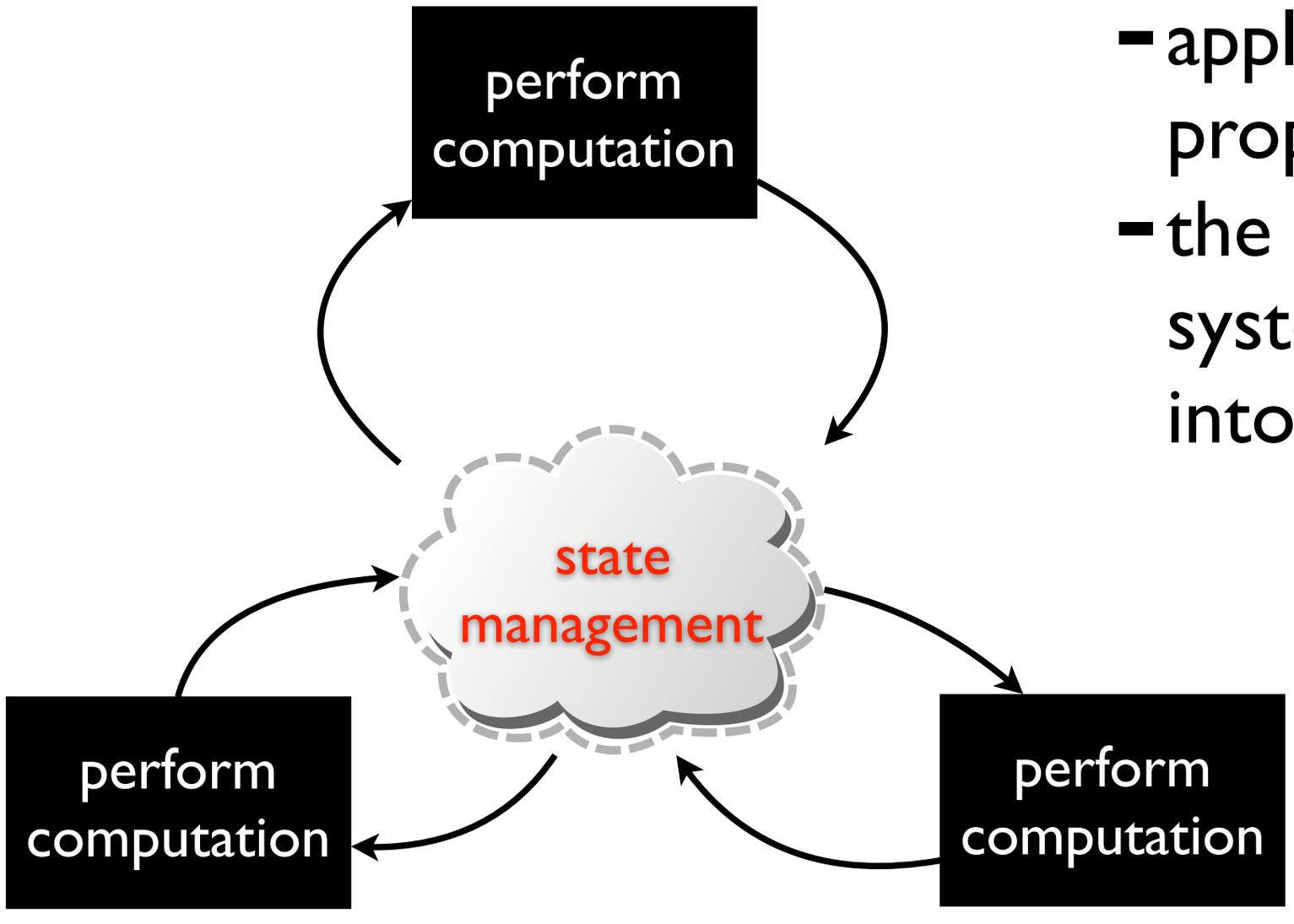
statesman approach

build and run applications in a loosely coupled manner

introduce a separate (state) management system

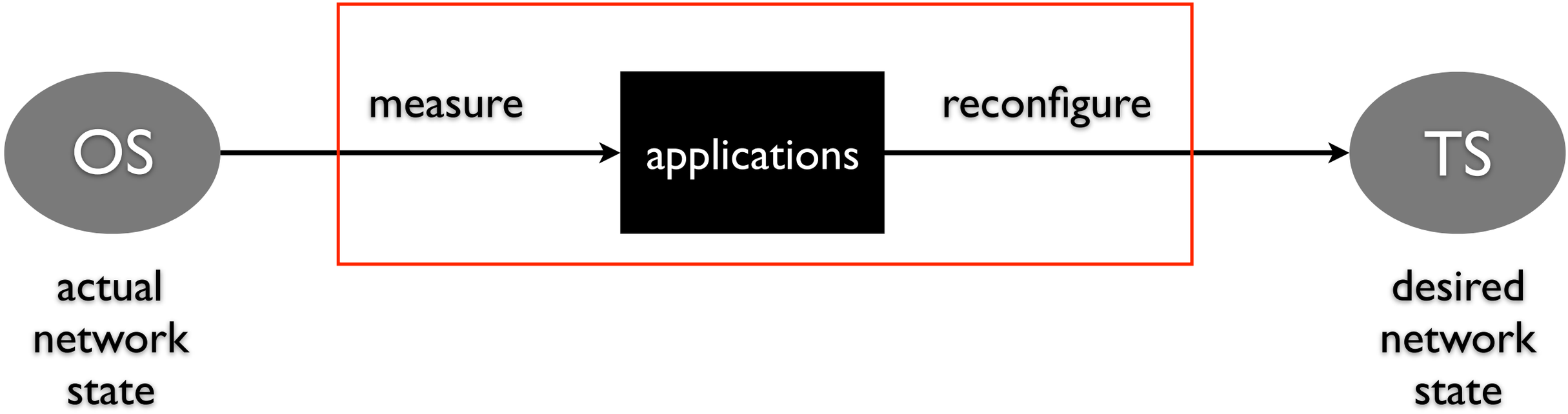
- conflict resolution
- invariant enforcement

statesman approach

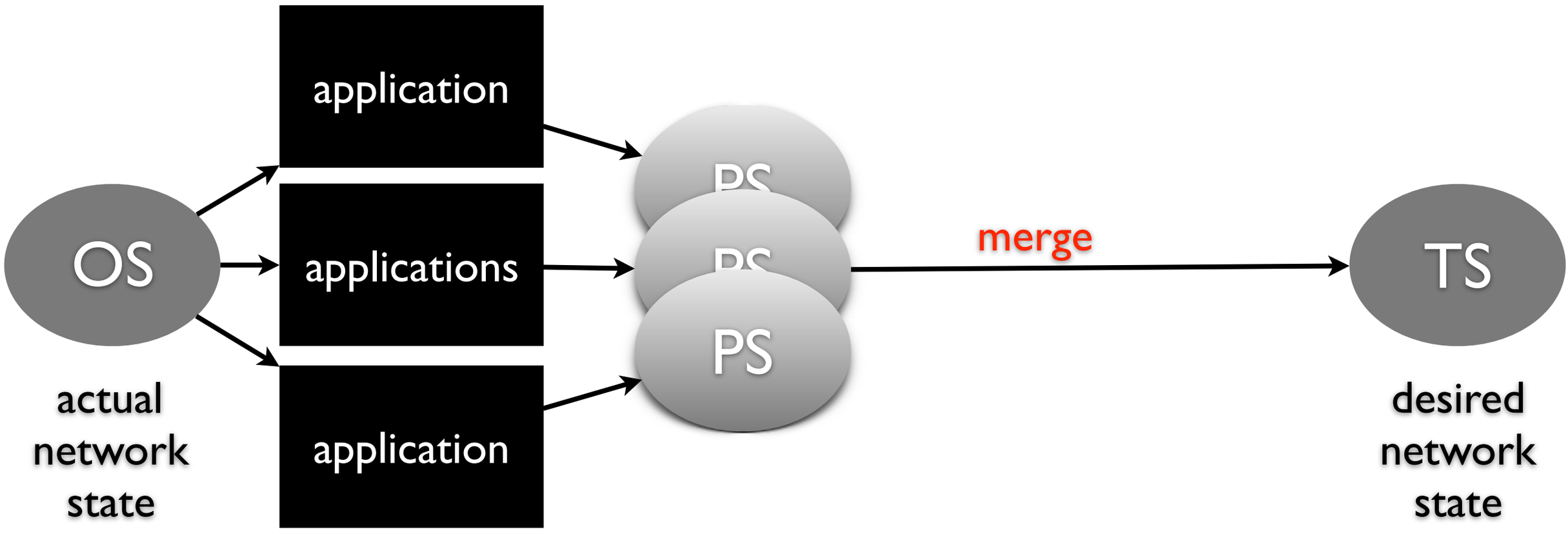


- applications "pull" observed states (OS)
- applications "push" proposed states (PS)
- the separate statesman system "merges" the states into target states (TS)

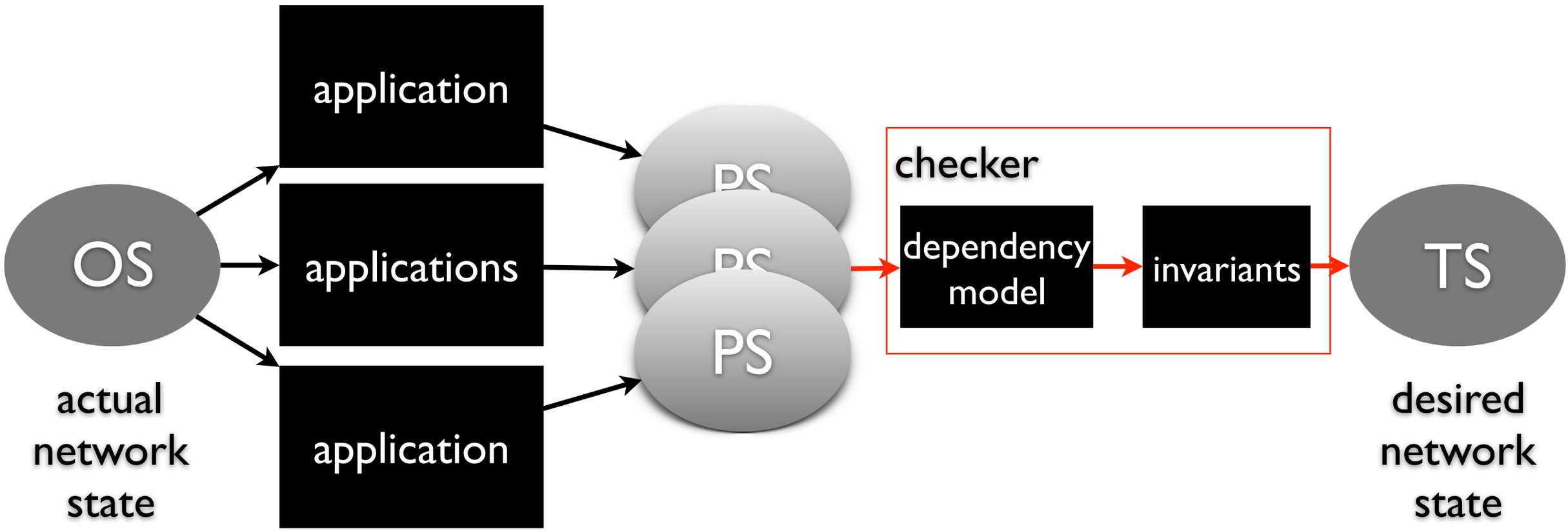
statesman approach



statesman approach



statesman approach



checker

use dependency model

- detect and resolve conflicts among PSes

use operator-specified invariants

- examine the TS

detecting conflicts

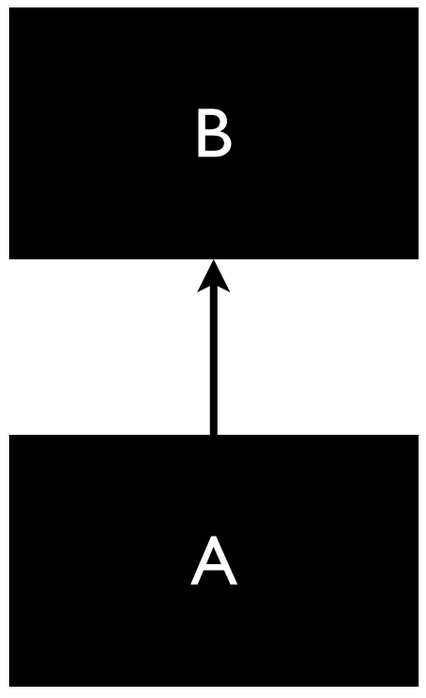
state variables in one application's PS can depend on state variables in another application's PS

B depends on A

- A is a prerequisite for managing B states
- B is controllable only if A value is appropriate

conflicts

- B is uncontrollable due to state (or state change) in A



resolving conflicts

TS-OS, PS-OS

- conflicts due to the changing OS
  - makes some variables in TS/PS uncontrollable

PS-TS

- a PS can conflict with the TS due to an accepted PS from another application

- last-write-wins
- priority-based locking
- at the level of individual switches and links

maintaining invariants

what

- invariants: infrastructure's operational stability (in the face of application bugs or undesirable effects of collective actions of multiple applications
  - safety & performance

how

- checking TS against invariants
  - difference operation f = (TS-OS)
  - new network state s = *f*(network state)
  - invariant checking on s

when

- TS+OS, TS+PS

discussion

making multiple applications coexist

- ONIX, NOX: no support
- Pyretic, Pane, Maple: compose target traffic management applications
- Corybantics: hosting multiple applications on isolated slices

course project

30% your score, start early

course project

proposal due

- 10/20, 5pm

what to submit

- no more than 1 page
- double-column, ACM guideline
- http://conferences.sigcomm.org/sigcomm/2017/submission.html#paper-formatting

project ideas

bring your own ideas

- talk to each other, talk to me

we suggest

- design and implement Ravel applications
  - extending Ravel with new applications
  - eg., VLAN, traffic engineering
- connect Ravel to other tools
  - interoperability
  - eg., using Ravel as a hypervisor to bridge PGA and Kinetic