Active Pipes: Service Composition for Programmable Networks

ECE 697J

October 29th, 2002

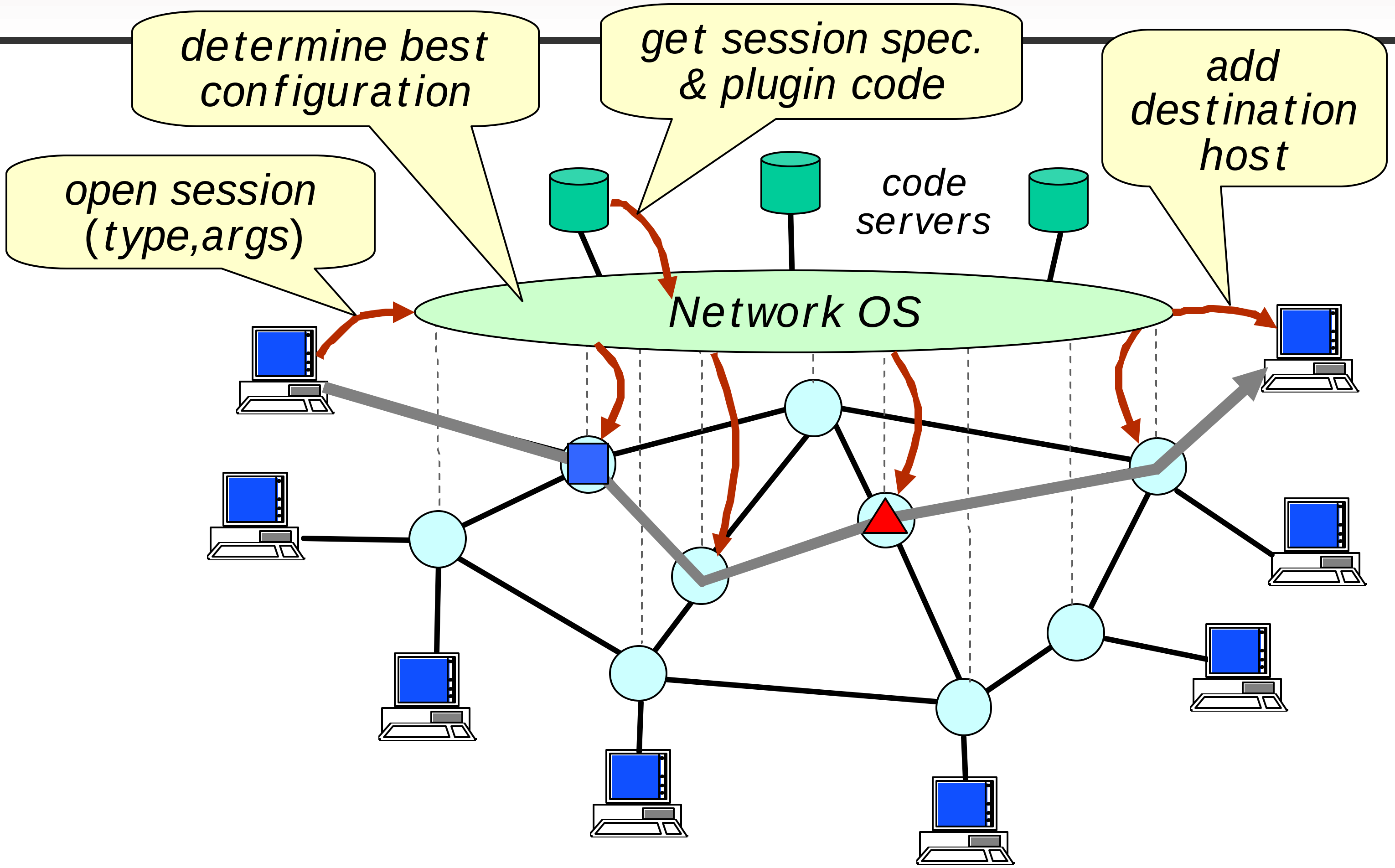
How to Program Active Network?

- Active network nodes provide programmable interfaces and active applications
- End-user / system administrator should be able to use active network easily
- Traditional network: socket (choice: UDP or TCP)
- Active network: also processing ⇒ lots of choices
- Challenge: How can user specify active network behavior (= “service composition”)?
- Idea: Use “pipe abstraction” from UNIX (= “active pipe”)

Network OS

- Application developers should not deal directly with all details
  - hide network specific details and topology from applications
  - uniform programming model should abstract from network heterogeneity
- Need something like “Network OS”, not just Node OS
- Automate configuration of application sessions
  - allocate network resources efficiently
  - provide protection and performance isolation
  - facilitate policy-driven application configuration
- Simplify use of advanced network applications
  - hide network topology and implementation details from users
  - minimize configuration requirements on end-users

Application-Centric Networking

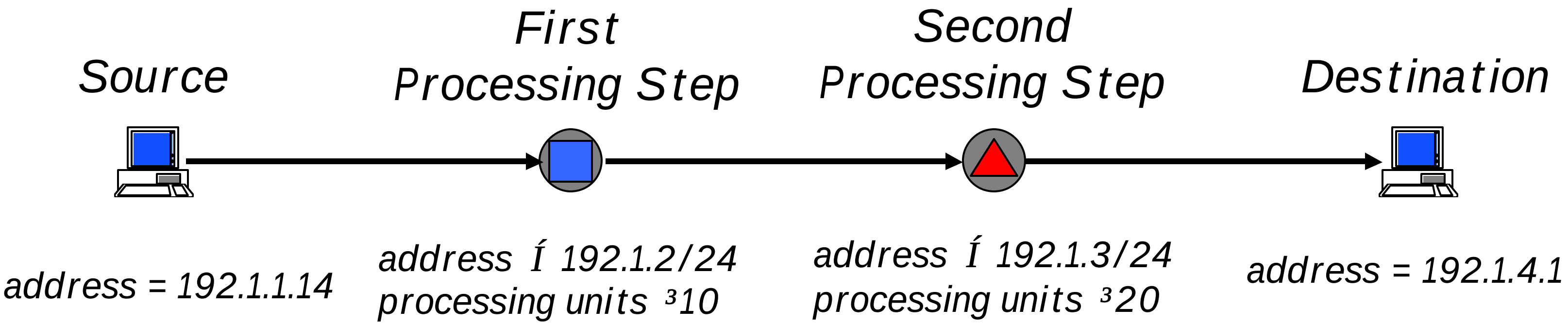


Requirements

- Programming abstraction
  - Express processing requirements of applications
- Session configuration algorithm
  - Quickly map application requirements onto available physical network resources
- Distribution of information
  - Use routing protocols to distribute the availability and usage of network resources
- Signaling mechanisms
  - Explicitly route and deploy processing functions

Active Pipe as Programming Abstraction

- Compose services out of reusable building blocks analogous to UNIX pipes:
  - cat | sort | a2ps | lp
- Active pipe describes sequence of processing functions to be executed
  - In networking context functions distributed on various nodes
  - Constraints define nodes suitable for processing



Source

First Processing Step

Second Processing Step

Destination

address = 192.1.1.14

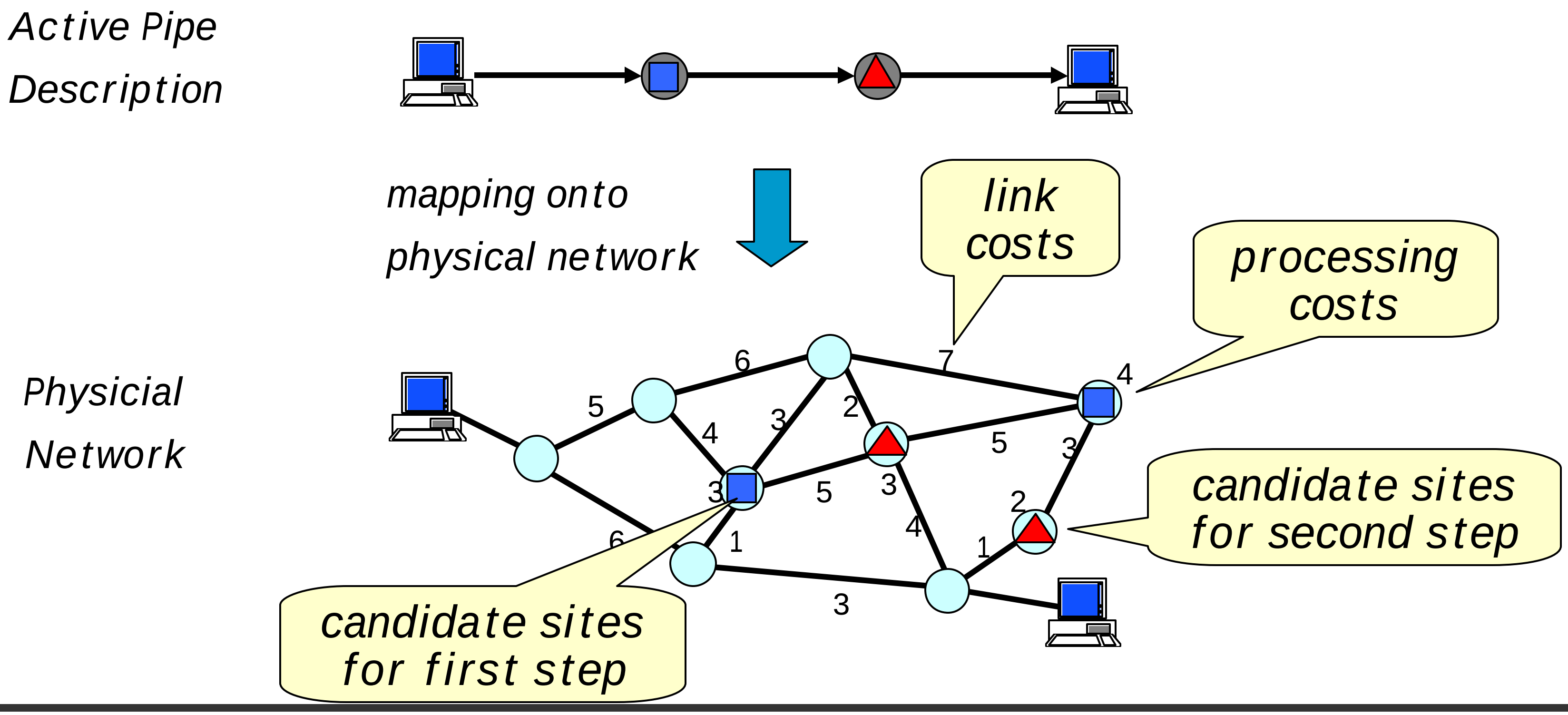
address Í 192.1.2/24
processing units ³10

address Í 192.1.3/24
processing units ³20

address = 192.1.4.1

Mapping Active Pipe to Physical Network

- Need to select processing sites and end-to-end path
  - Mapping Algorithm



Active Pipe Description

mapping onto physical network

Physical Network

link costs

processing costs

candidate sites for second step

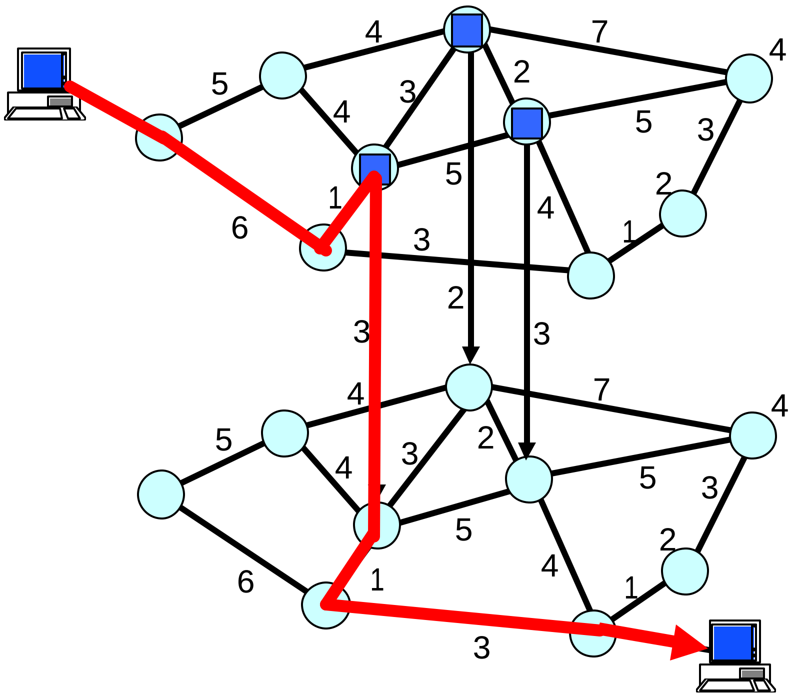
candidate sites for first step

Mapping Algorithm

- Constraint-based optimization problems with more than one cost metric are known to be intractable
- NP-complete algorithms do not scale for large networks
- Active Pipes approach:
  - One cost metric for both link and processing costs
  - Constraint-based routing problem can be mapped to shortest-path problem and solved with Dijkstra algorithm

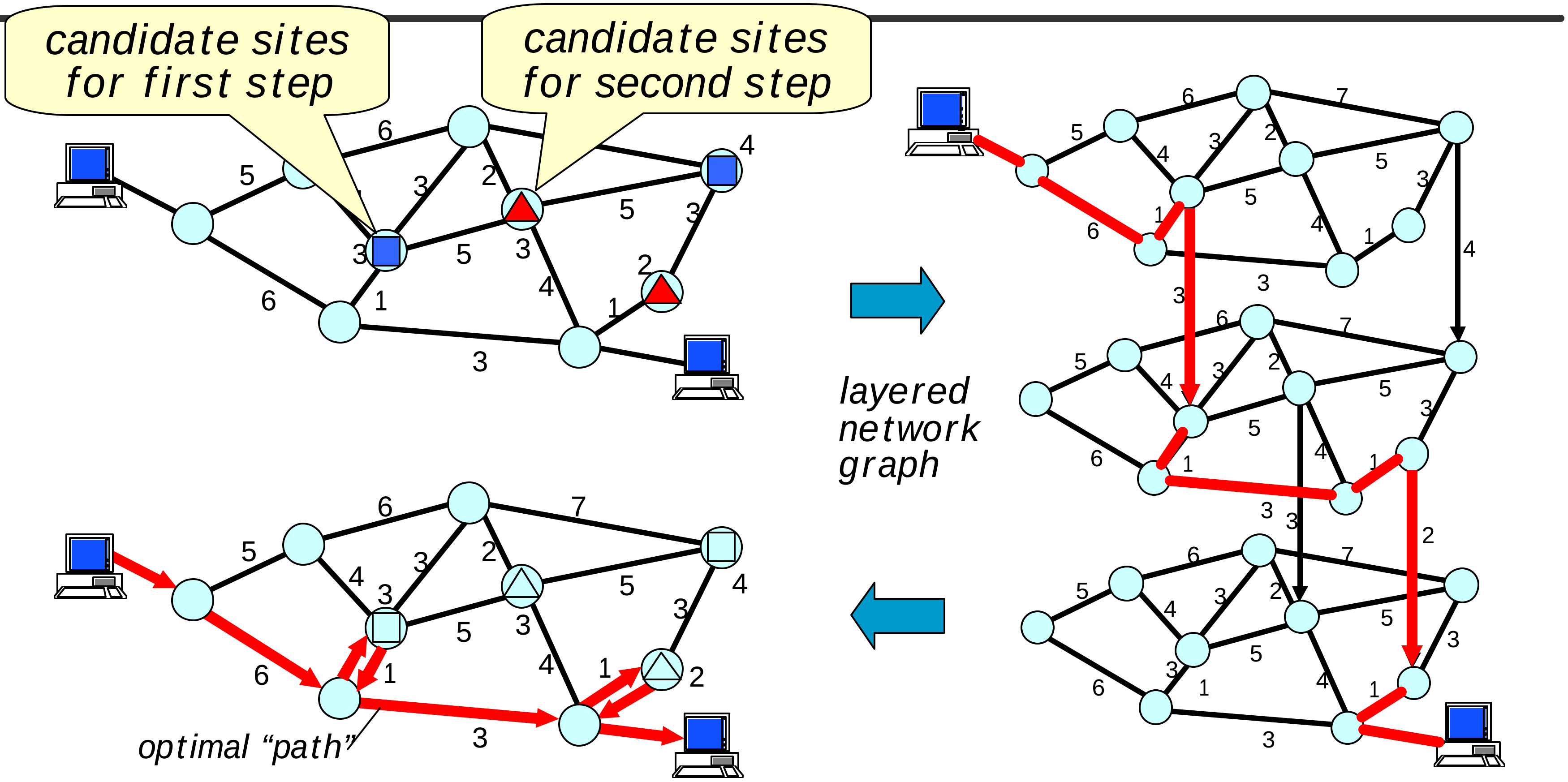
Constraint-based Routing: Single Processing Site

- Layered graph method:
  - build 2 layer graph:
    - processing nodes become inter-layer edges
  - find shortest path in graph
  - project two layers onto single layer, optimal processing where path *crosses layers*
  - extends nicely to k processing steps



University of Massachusetts Amherst

Layering Model for Two Processing Sites



candidate sites for first step

candidate sites for second step

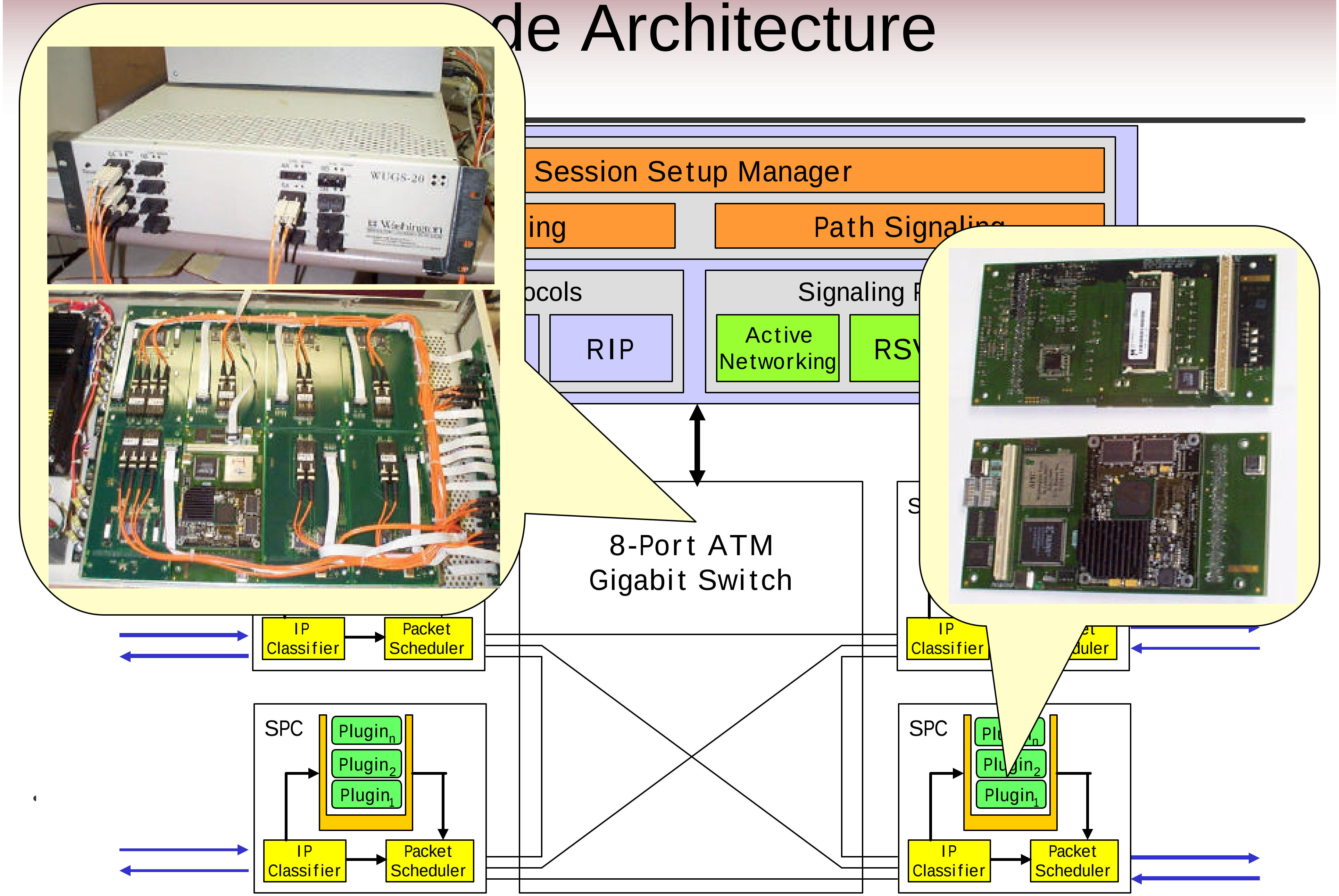
layered network graph

optimal “path”

Implementation Requirements

- Requirements for mapping algorithm
  - Topology information to build network graph
  - Location and costs of processing sites
  - Network attributes to express constraints
- Idea:
  - Existing routing protocols already have some topology information (link state database)
  - Routing protocols can be extended to include information about processing sites and attributes
  - Use of OSPF Opaque LSA options (RFC 2370) to carry transparent information

de Architecture



Session Setup Manager

ing

Path Signaling

ocols

RIP

Signaling P

Active Networking

RSV

8-Port ATM Gigabit Switch

S

IP Classifier

Packet Scheduler

IP Classifier

et duler

SPC

Plugin$_n$

Plugin$_2$

Plugin$_1$

IP Classifier

Packet Scheduler

SPC

Plugin$_n$

Plugin$_2$

Plugin$_1$

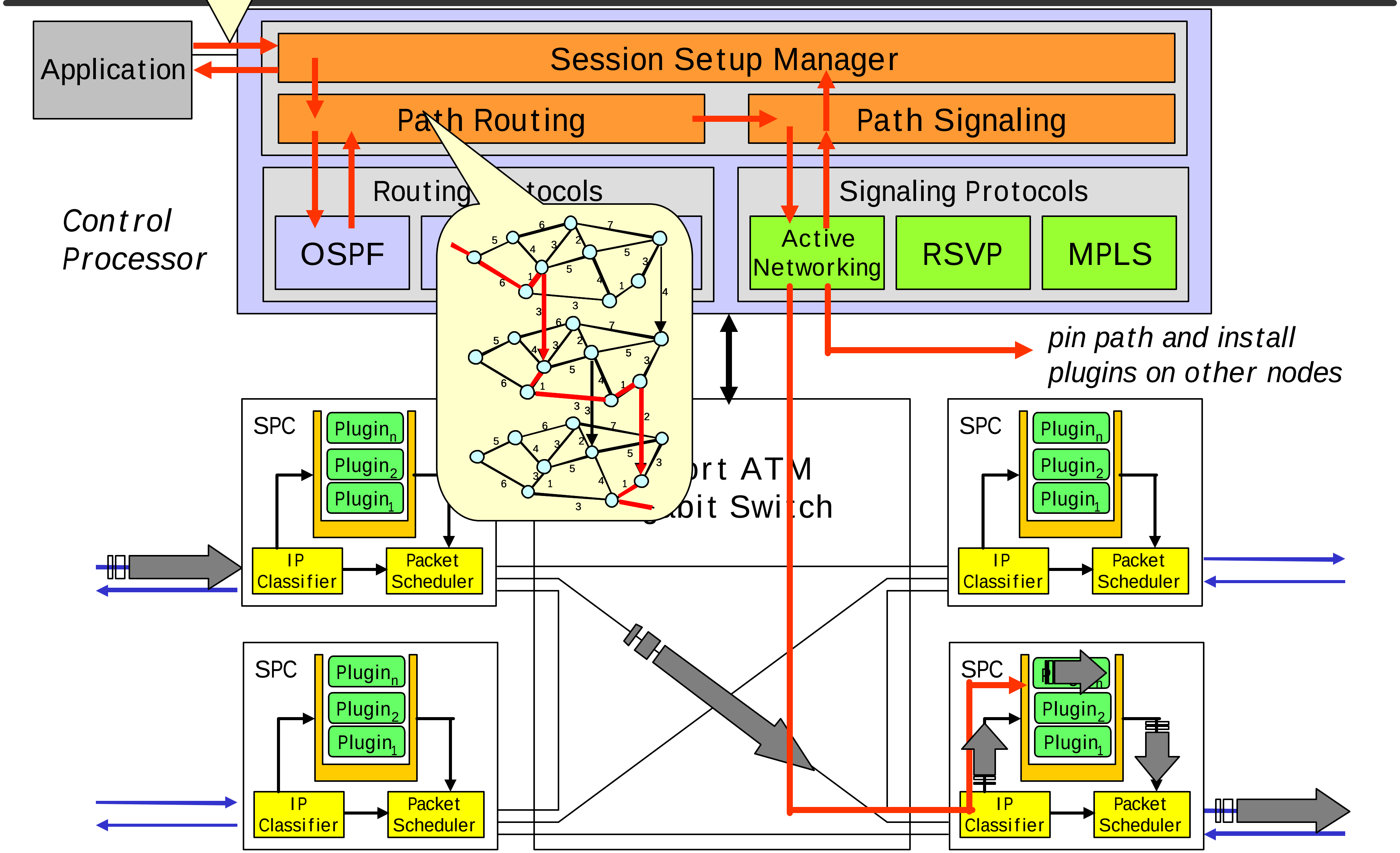
IP Classifier

Packet Scheduler

ession Setup

…etermine
…d plugins

open session
(type,args)



pin path and install plugins on other nodes

Conclusions

- Need for automated mechanisms to configure application sessions
  - free application developers from low level concerns
  - enable effective resource allocation
  - to give network administrators control over network usage
  - simplify use of advanced applications by end users
- Active pipe paradigm for specifying processing and transmission requirements
- Layered network graph for session mapping
- Goal is a “Distributed Network OS”