

Genesis Kernel on IXP1200

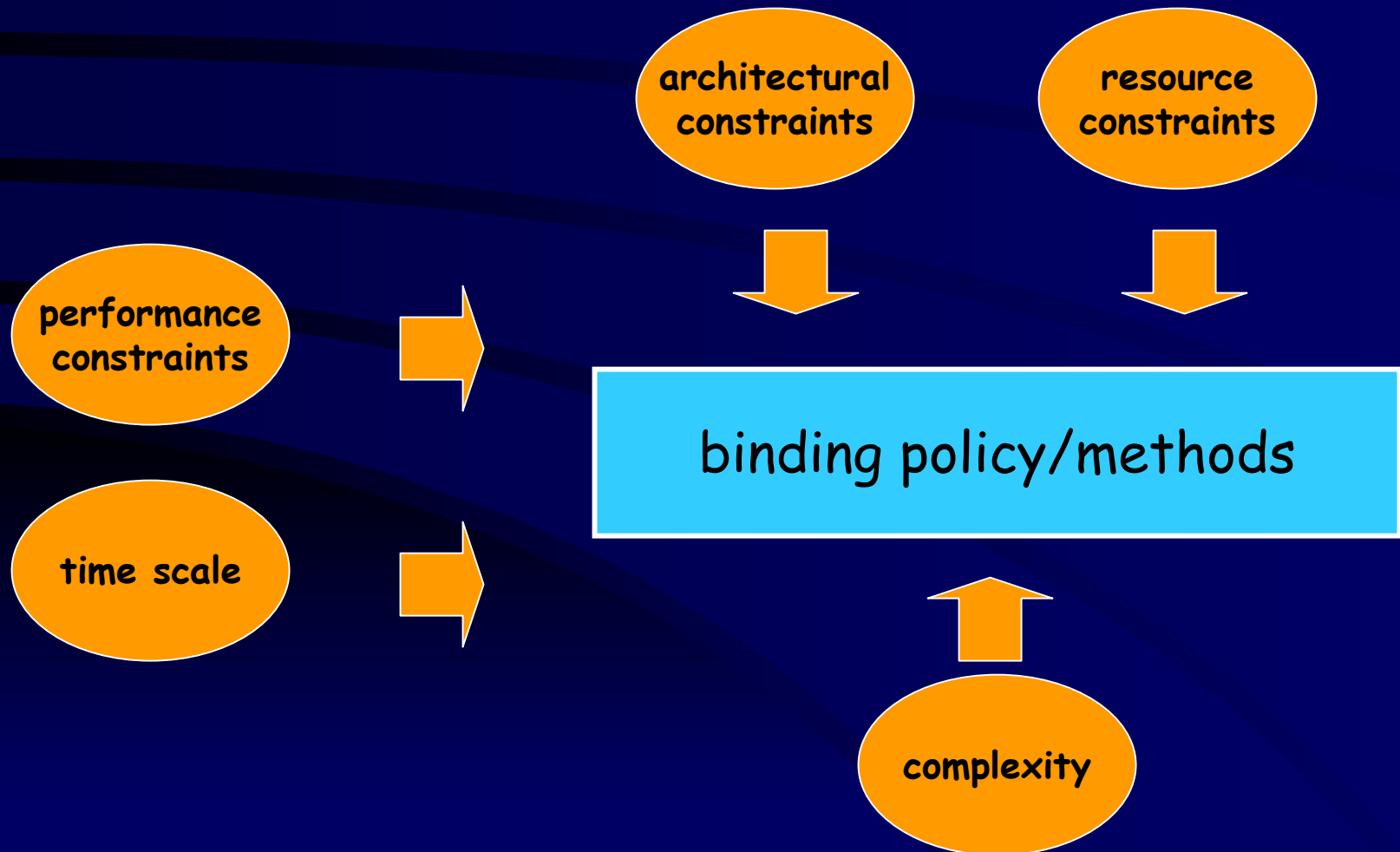


Stephen Chou
Michael Kounavis
Andrew Campbell
John Vicente
Columbia University

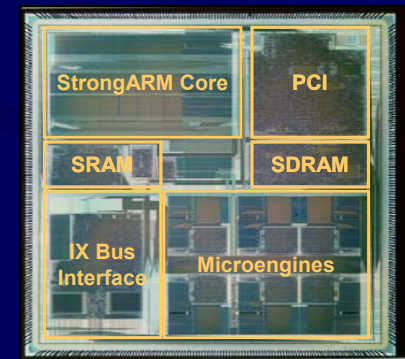
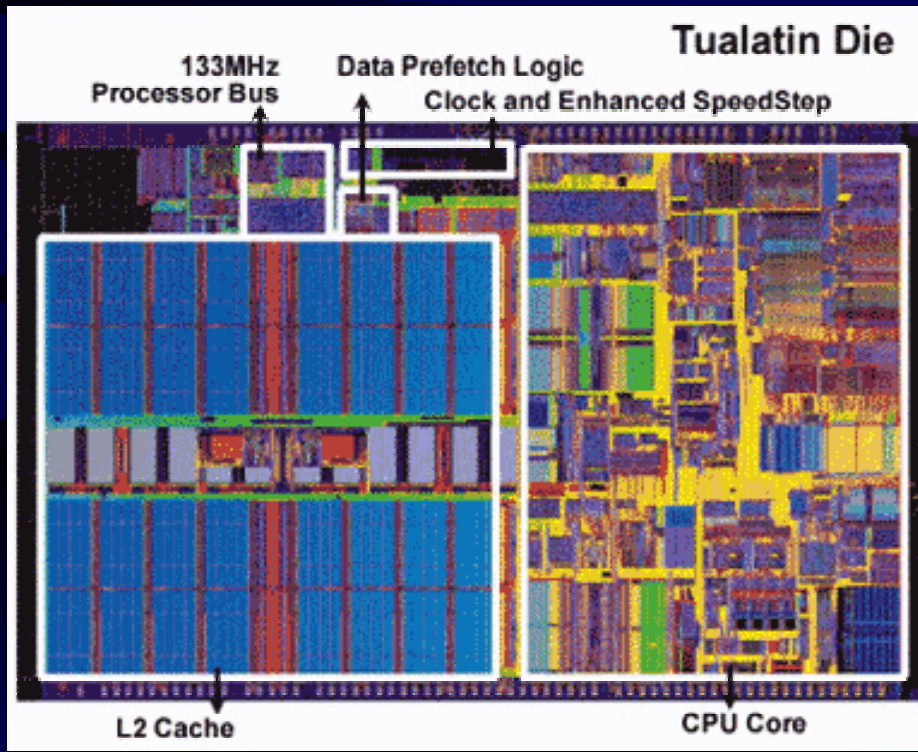
overview

- making network processor-based routers programmable.
- NetBind:
 - a binding tool for programming the data path in NPs
- Genesis Kernel
- "Genesis Box" testbed and status

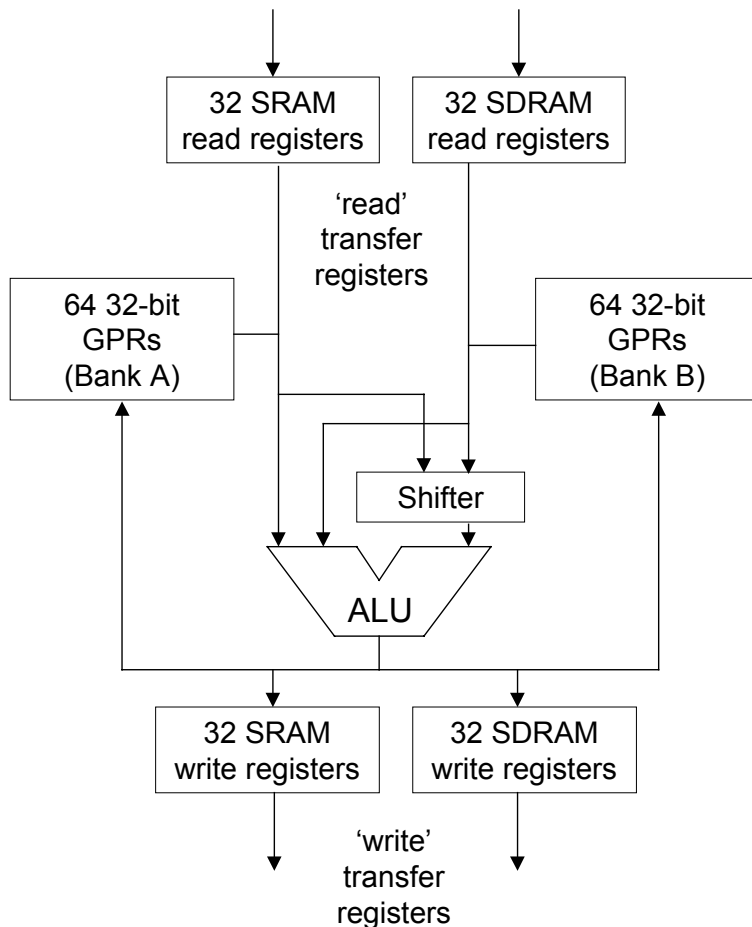
programming a network processor-based router



NPs and general-purpose processors



IXP1200 microengine



- 4 contexts
- 128 General Purpose Registers (GPRs) divided into Banks A and B
- 128 Address Registers
 - read and write transfer registers
 - SDRAM and SRAM
- addressing modes
 - absolute
 - context relative

programming the data path



Code Synthesis

Dynamic binding

Monolithic

Loader

Scripting



| | | | |
|--------------|------|------|-----|
| Complexity? | High | ?? | Low |
| Flexibility? | High | High | Low |
| Reusability? | High | High | Low |

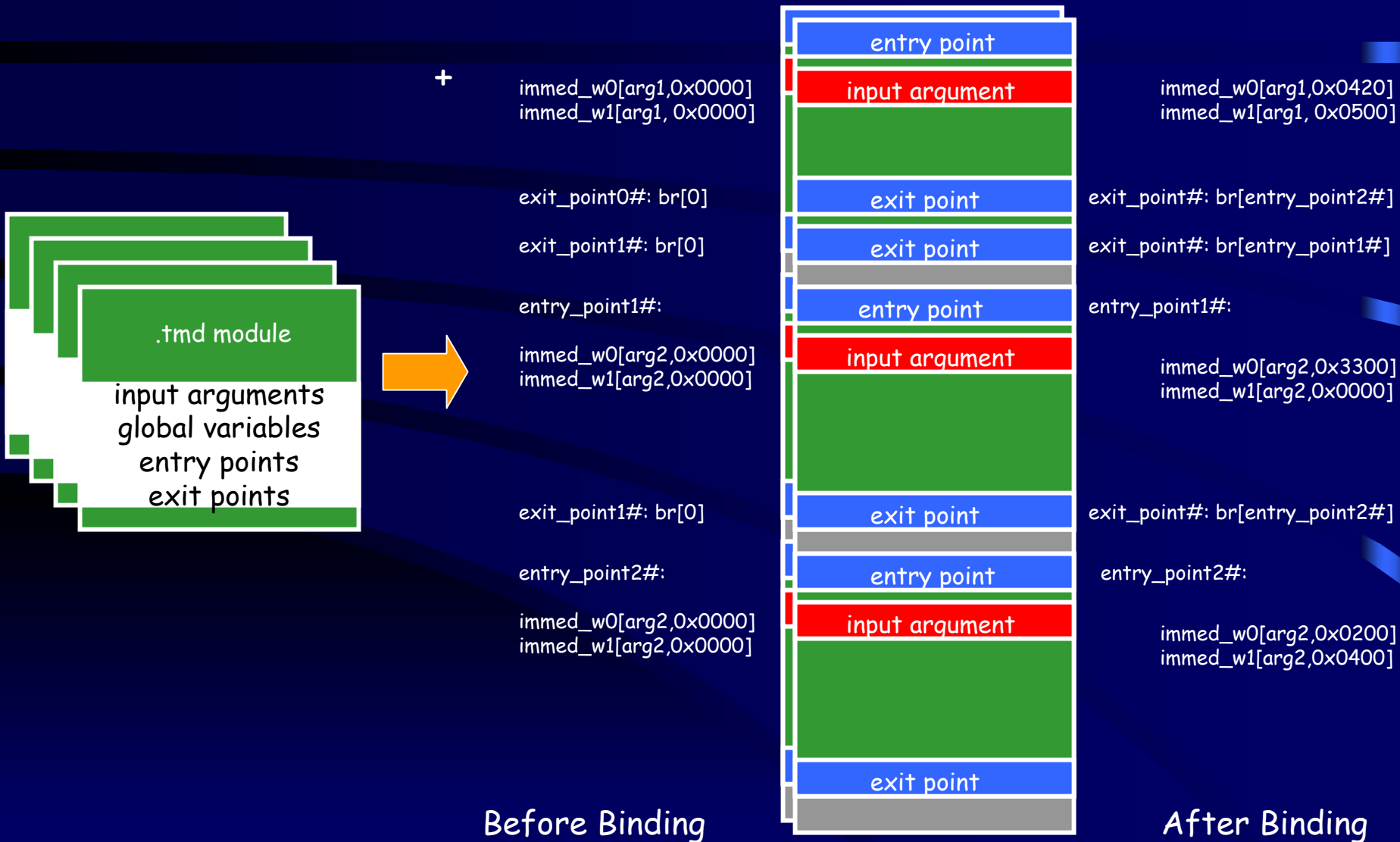
dynamic binding issues

- headroom limitations
- register space and state management
- choice of the binding method
- data path admission control
- processor handoffs
- instruction store limitation

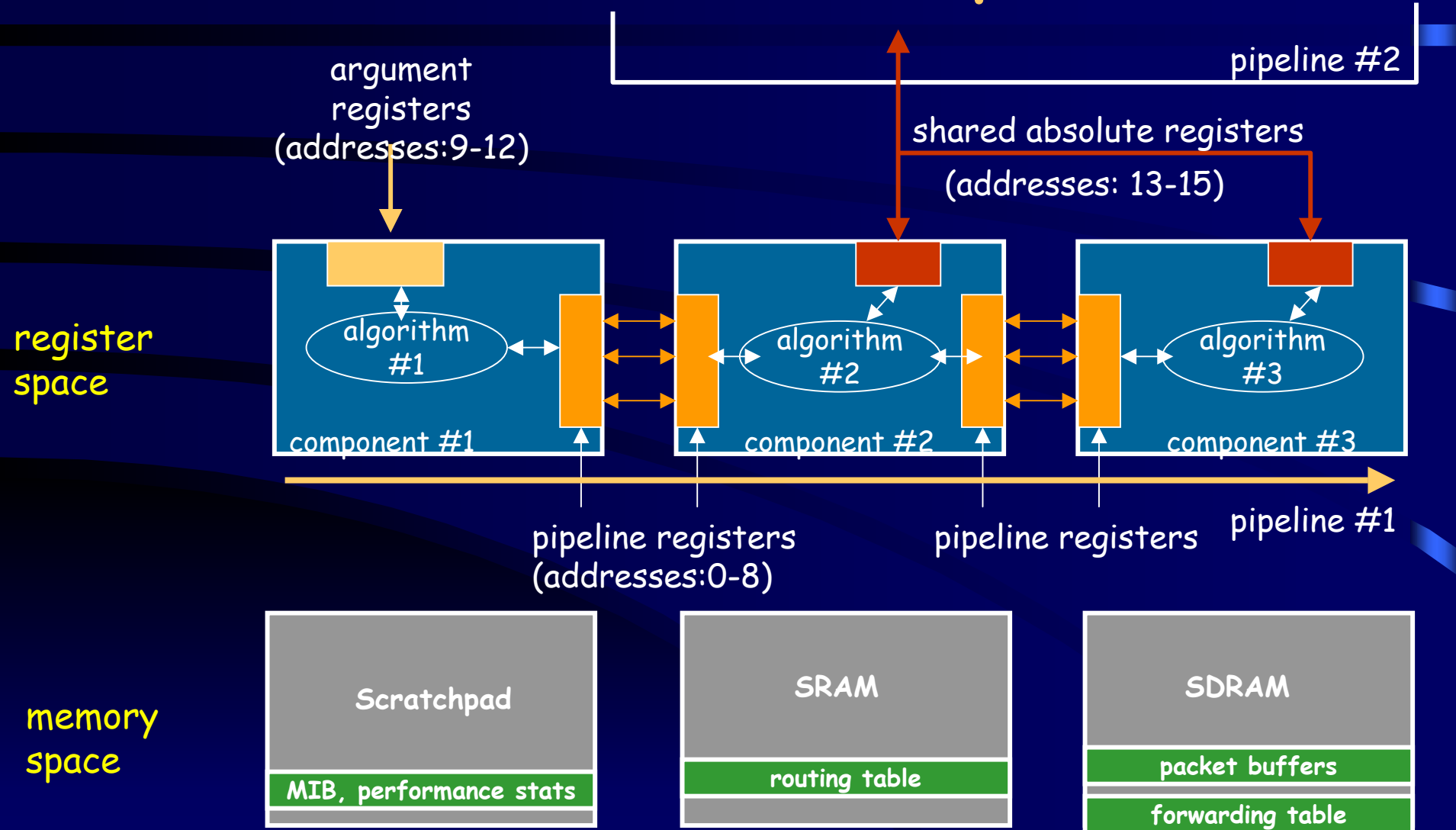
binding methods

- dispatch loop
 - MicroACE Extensions of the ACE framework in Intel IXP SDK 2.0
 - nested if-then-else on global binding state for determining the next code module
- vector table
 - table (stored in fast memory) of pointers to code modules
- code morphing
 - used by NetBind

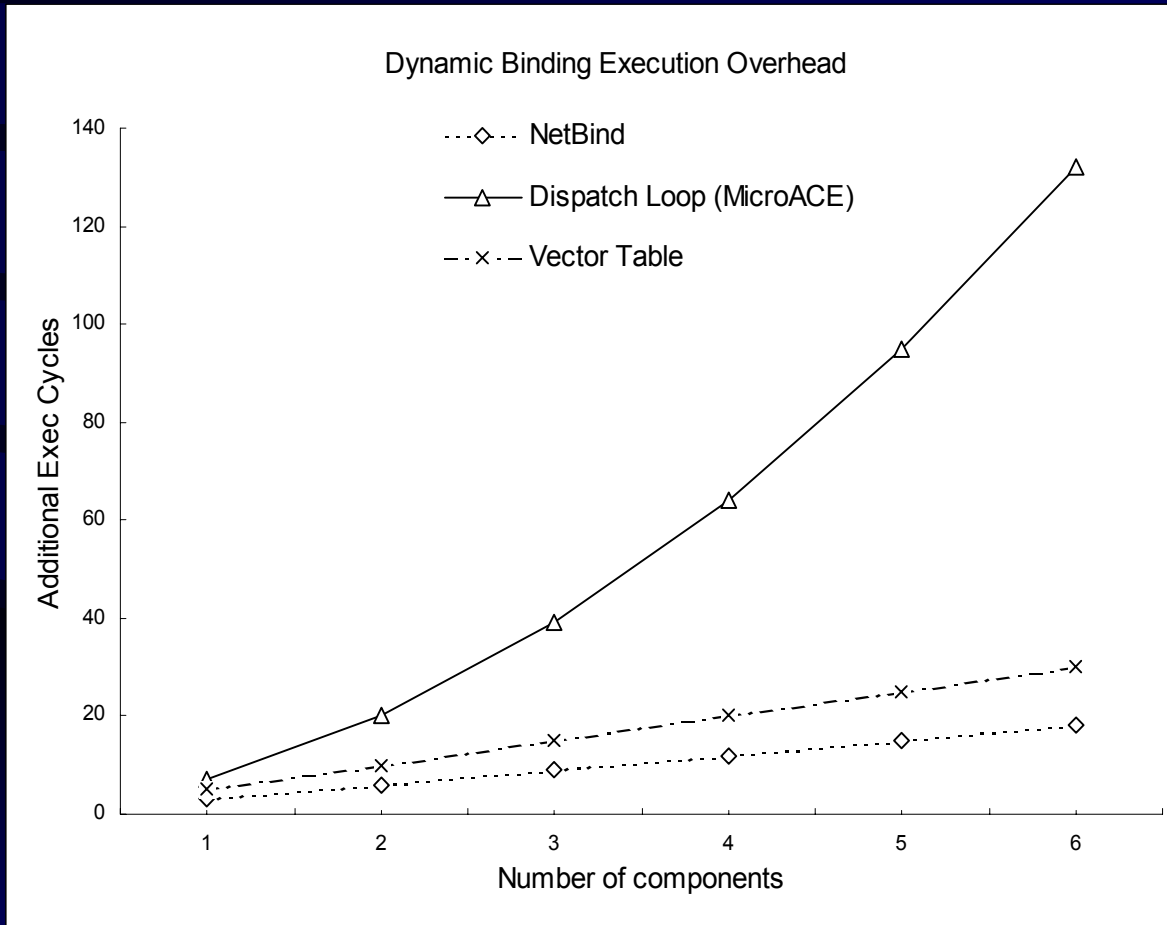
NetBind code space



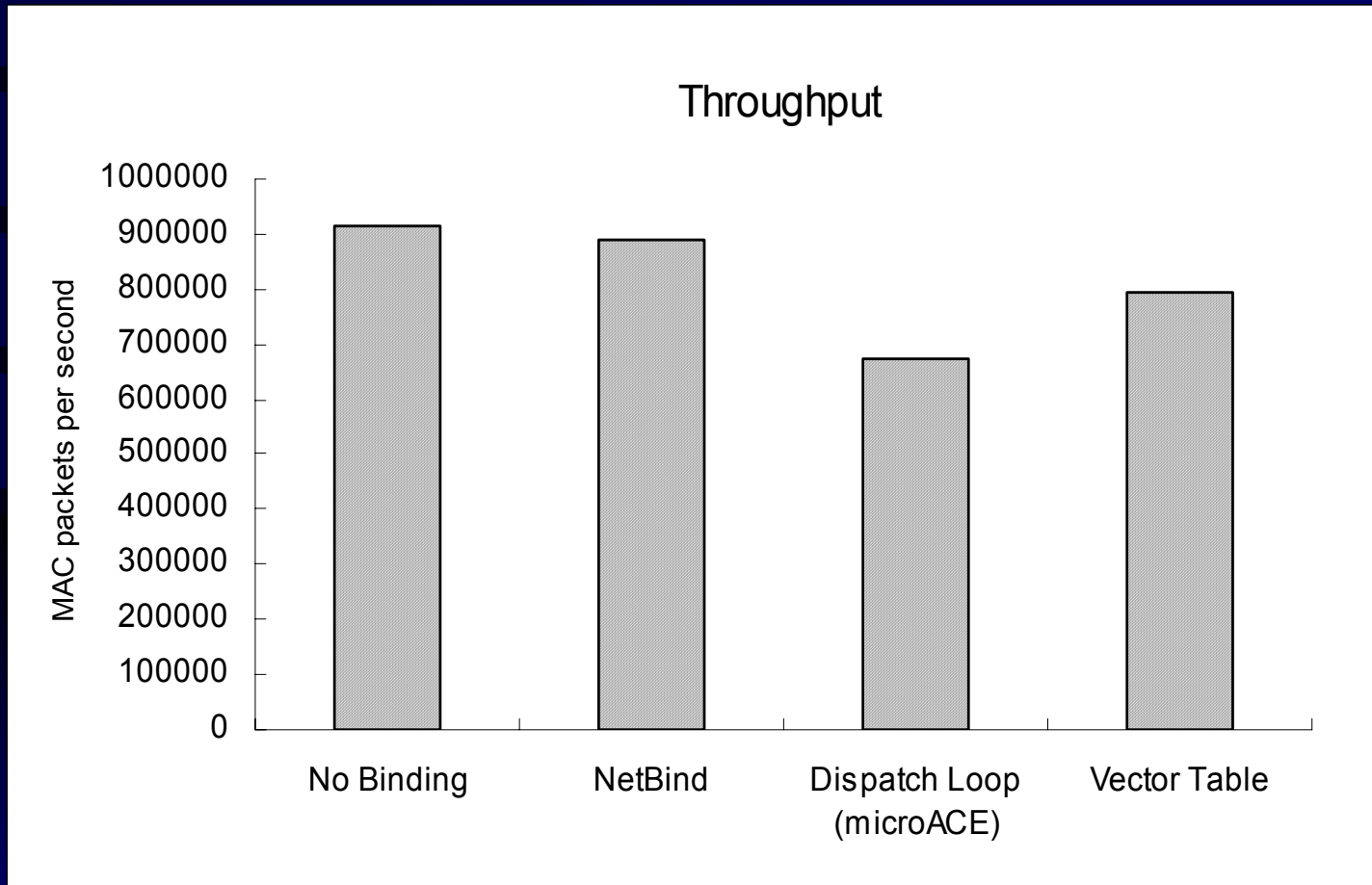
NetBind state space



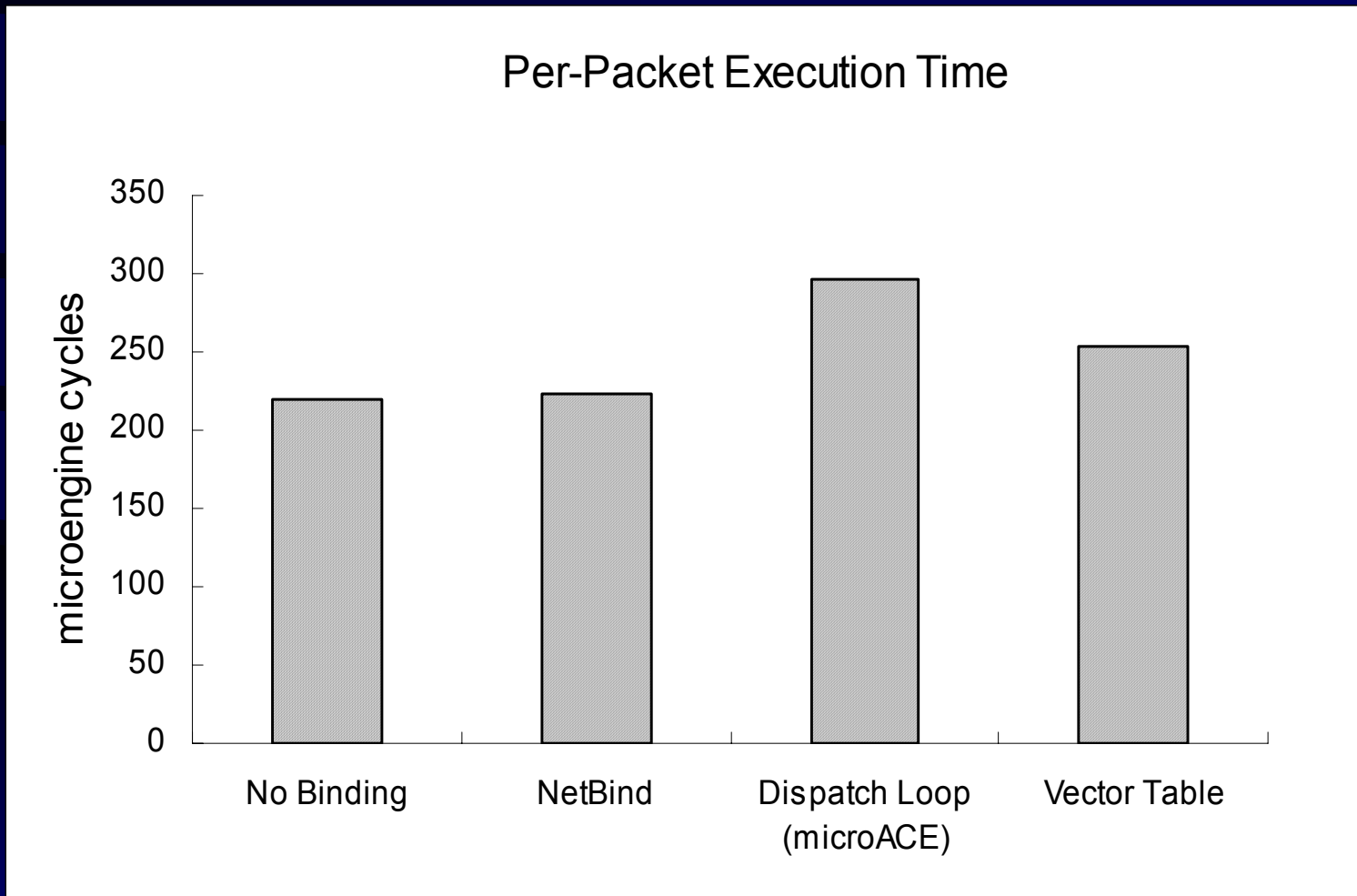
binding overhead



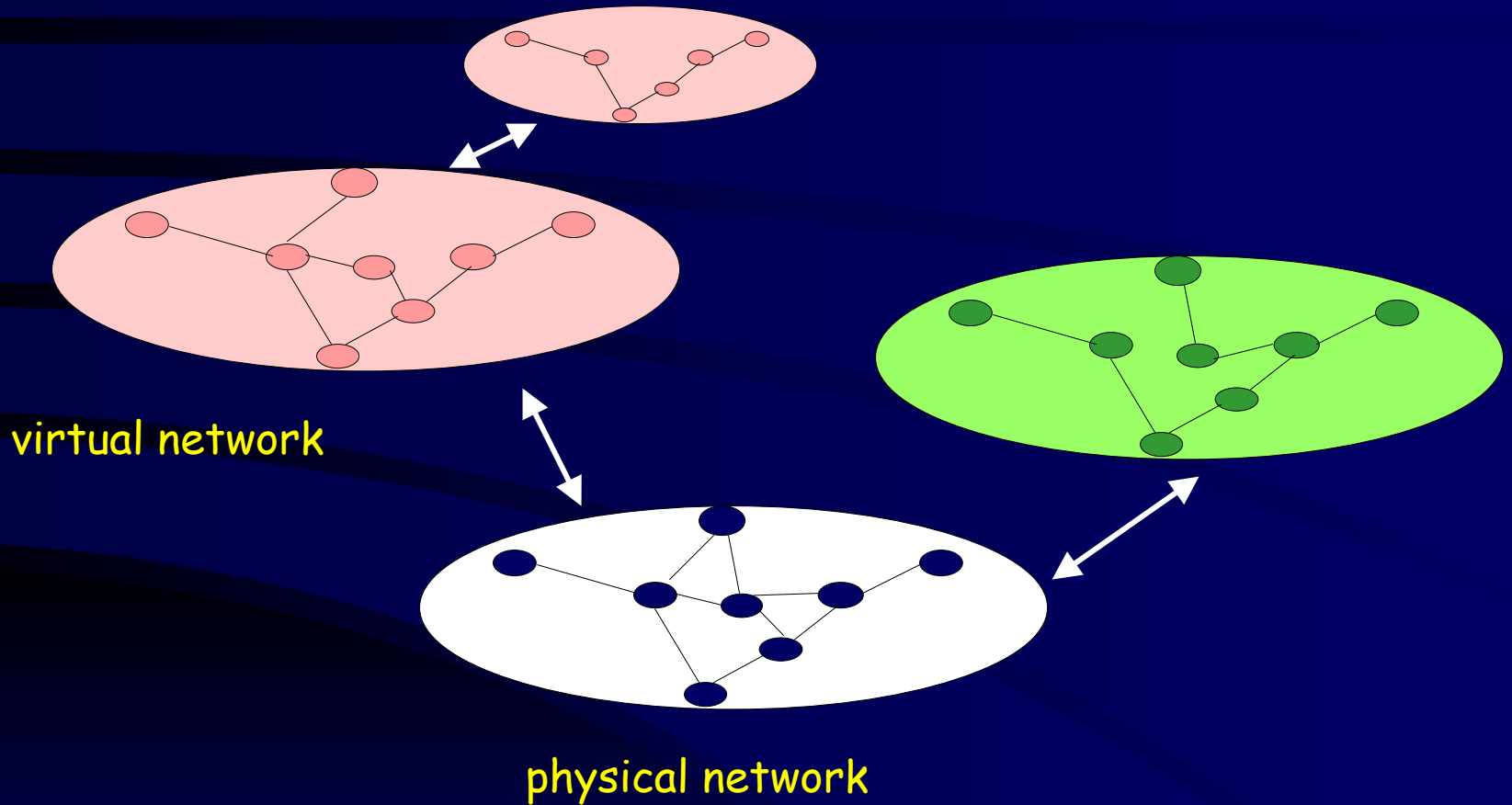
throughout analysis



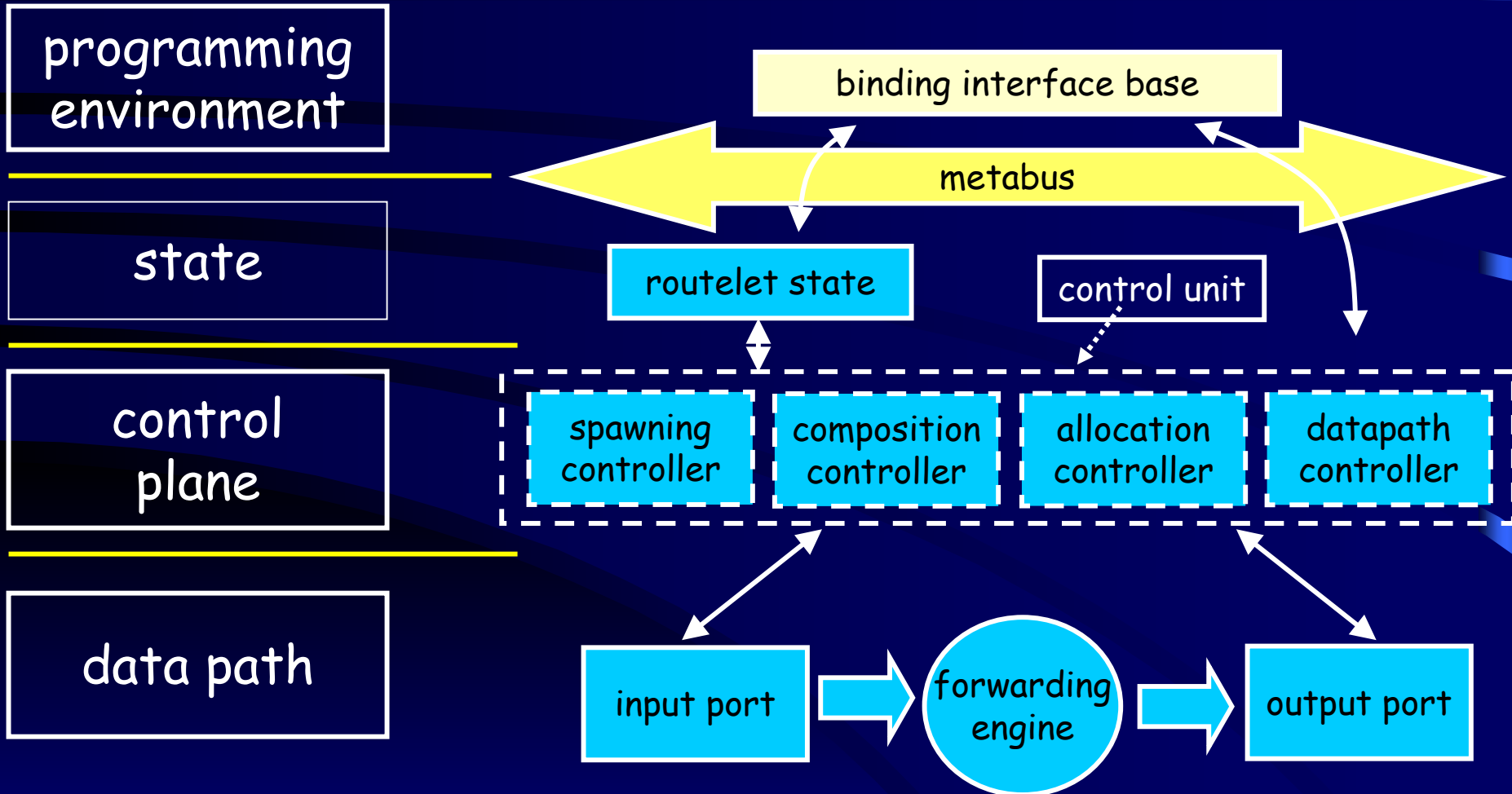
delay analysis



virtual networks on demand



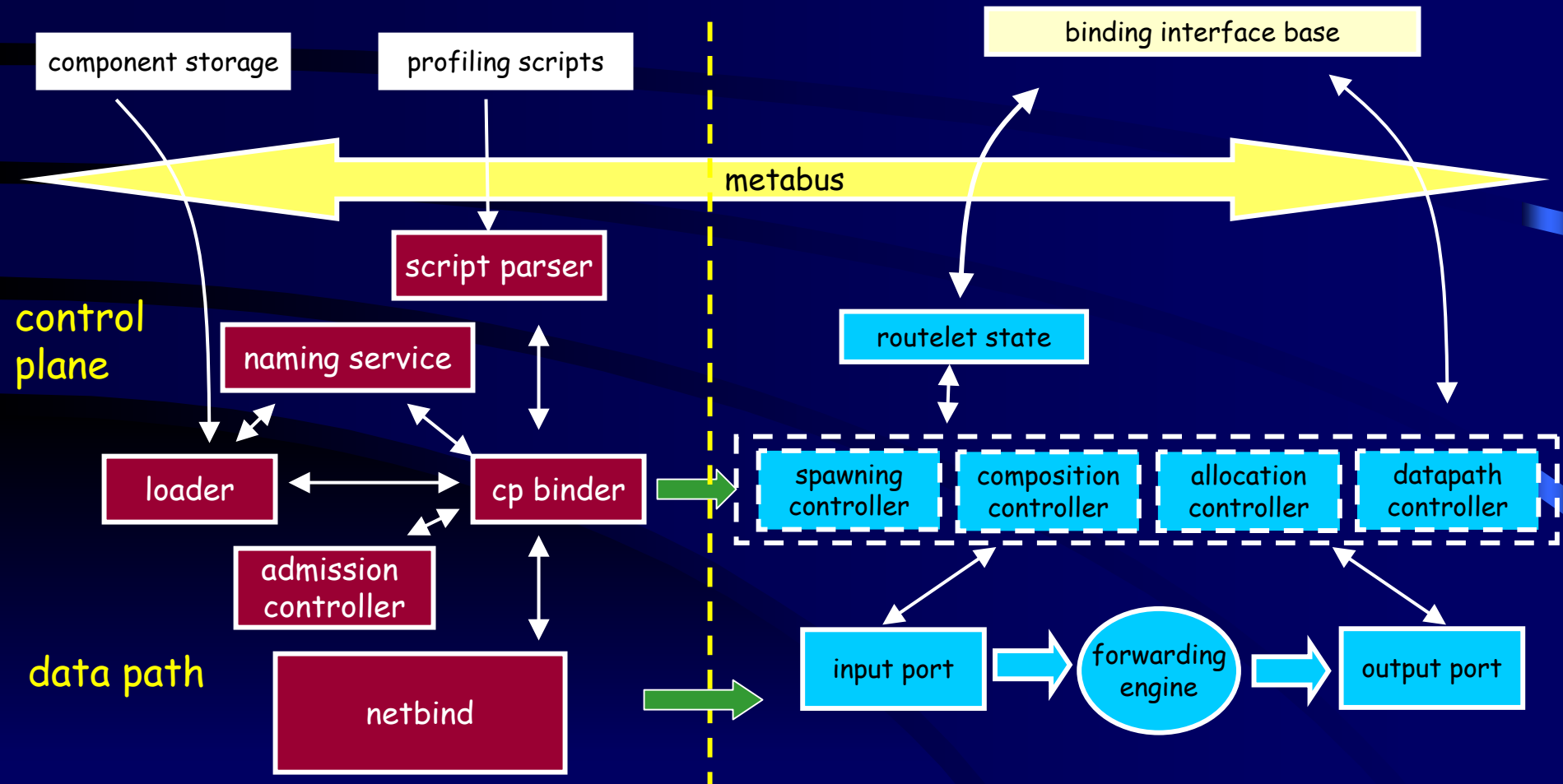
routelet: virtual router



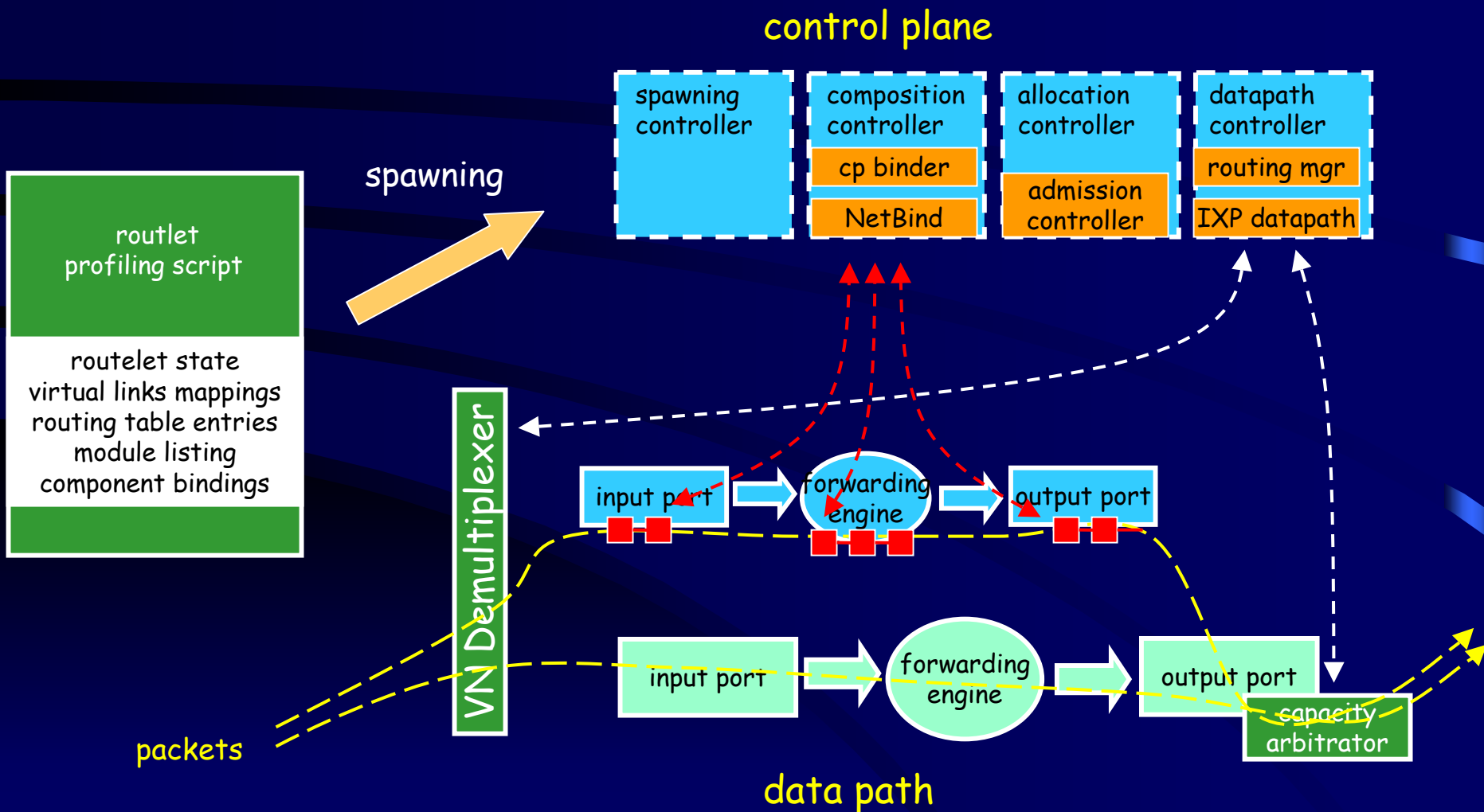
Genesis Kernel (GK) 1.0

algorithms

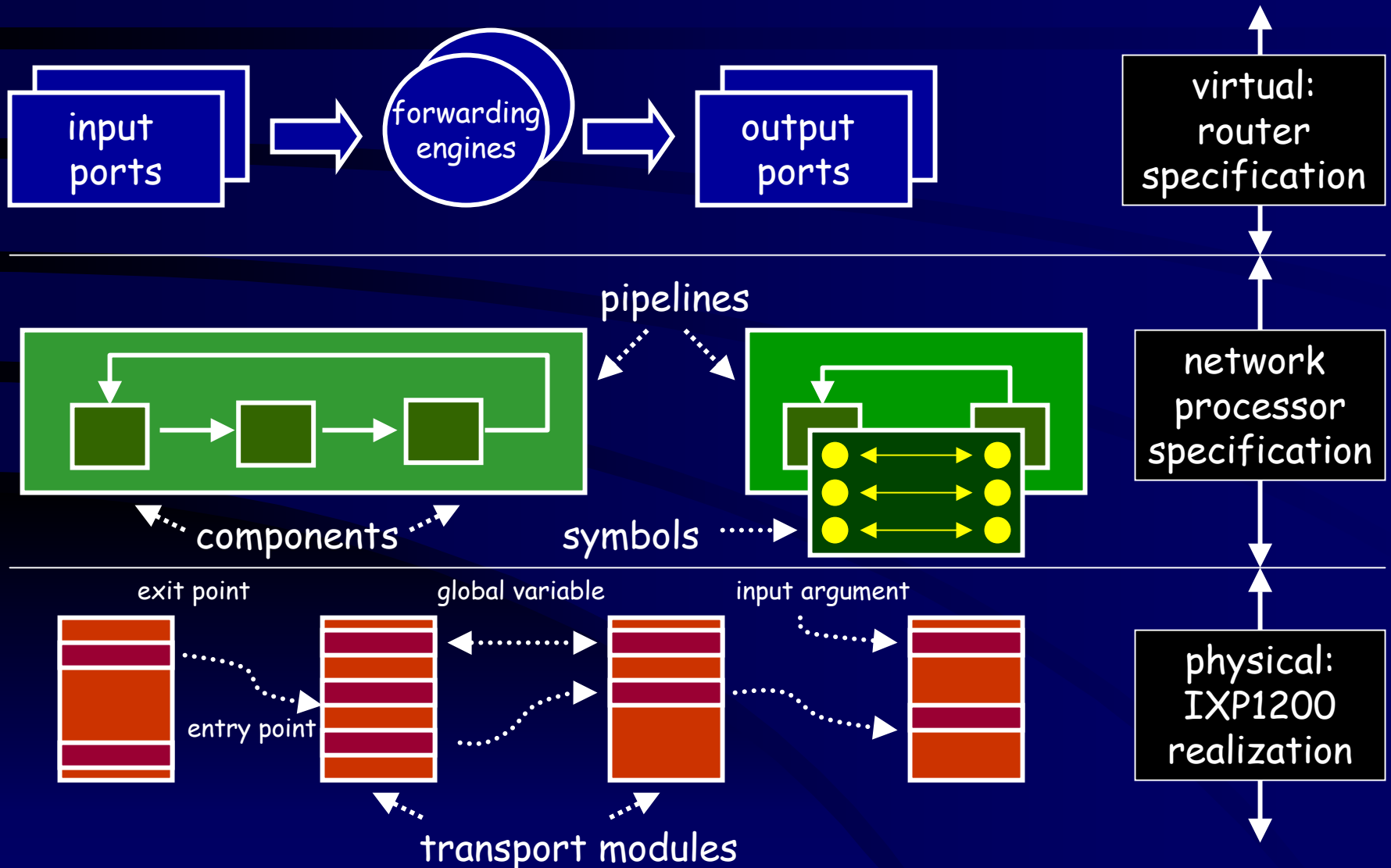
architecture



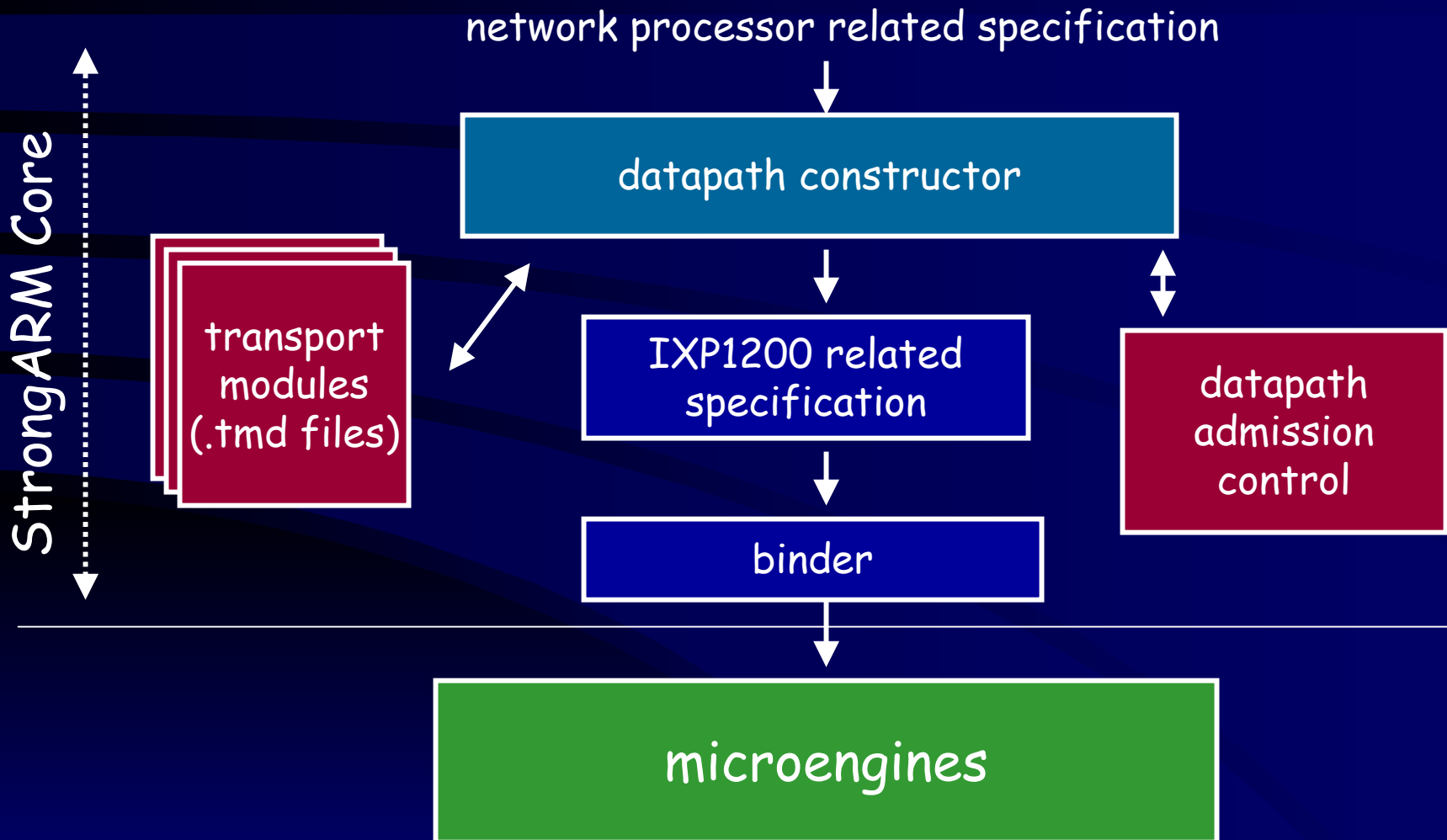
spawning a routelet



data path specifications



dynamic binding in GK



Genesis Developer Workbench

The screenshot displays the Genesis Developer Workbench interface, which is used for configuring and managing network topologies. The interface is divided into several main sections:

- XML Explorer:** Shows the project structure for 'ipv4' and the XML configuration for a routelet. The XML includes parameters for VNI, bandwidth, capacity, mean rate, peak rate, script name, and table size. It also defines physical and virtual links for the routelet.
- Network Node Monitor:** Displays a list of routelets, including their addresses and physical router IDs. A specific node, '26472 ipv4 0', is highlighted.
- Demultiplexer:** A vertical component that routes traffic from multiple virtual networks (VNs) to a specific routelet.
- Control Unit:** Consists of four controllers: spawning, composition, allocation, and datapath. These controllers manage the network's state and resources.
- Forwarding Engines:** A central component that receives traffic from input software ports and forwards it to output software ports.
- Topology Tool:** Provides a visual representation of the network topology. It shows a hierarchical structure of routelets connected by physical and virtual links. The nodes are labeled with their IP addresses, such as 128.59.67.175, 128.59.67.174, 128.59.67.85, 128.59.67.173, 128.59.67.185, 128.59.67.189, 128.59.67.197, 128.59.67.177, 128.59.67.176, 128.59.67.181, 128.59.67.180, 128.59.67.178, 128.59.67.179, 128.59.67.182, and 128.59.67.183.
- 3D Topology:** A 3D visualization of the network topology, showing the routelets as interconnected nodes in a 3D space.

Genesis "Box" testbed



Genesis Kernel status

- NetBind Code release
 - www.comet.columbia.edu/genesis/netbind
- Genesis Kernel (GK) v1.0 developed
 - datapaths implemented on IXP1200
 - Genesis Developer Workbench front-end
 - Genesis testbed consisting of 15 IXP1200 evaluation platforms
 - Release March 2002

Genesis publications

- "The Genesis Kernel: A Programming System for Spawning Network Architectures", *JSAC*, March 2001.
- "Sphere: A Binding Model and Middleware for Routing Protocols ", *OPENARCH' 01*, April 2001.
- "Netbind: A Binding Tool for Constructing Data Paths in Network Processor-Based Routers", *OPENARCH'02*, June 2001.

thanks for listening