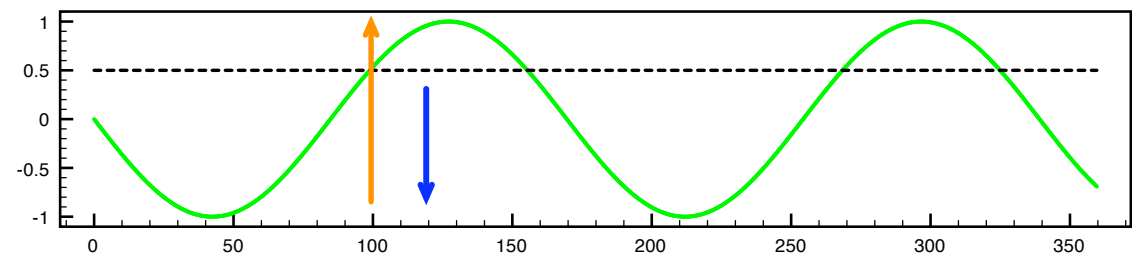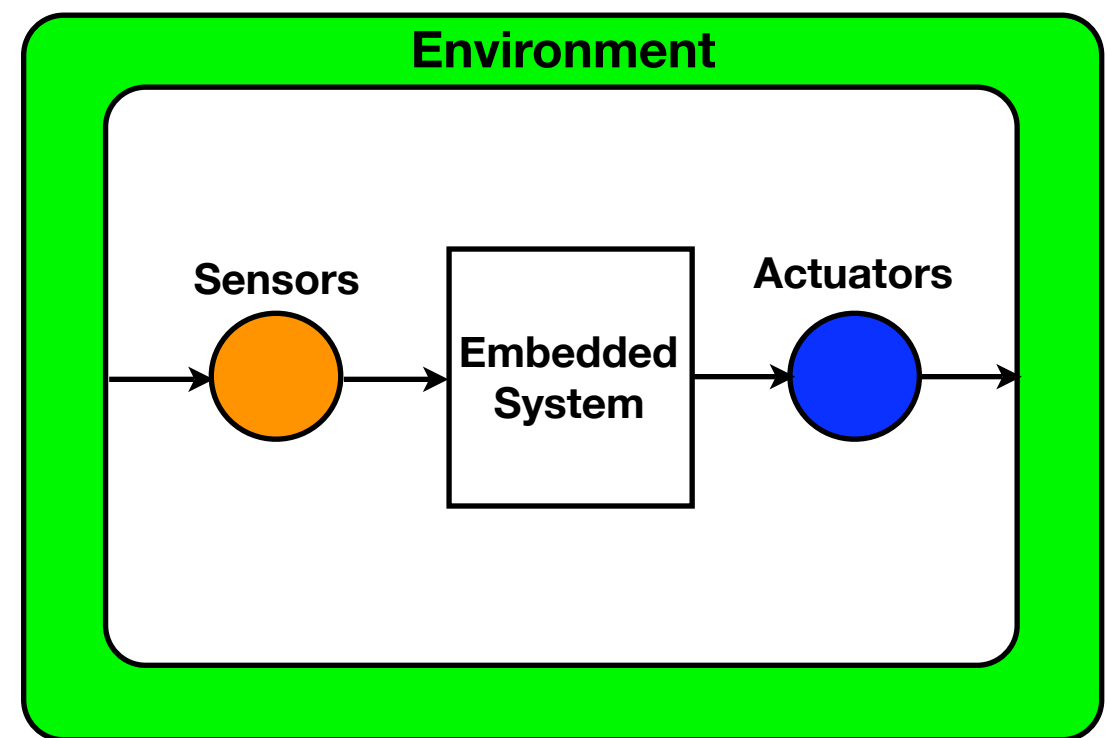# Communication-Based, Embedded System Design

Alessandro Pinto
University of California, Berkeley

apinto@eecs.berkeley.edu
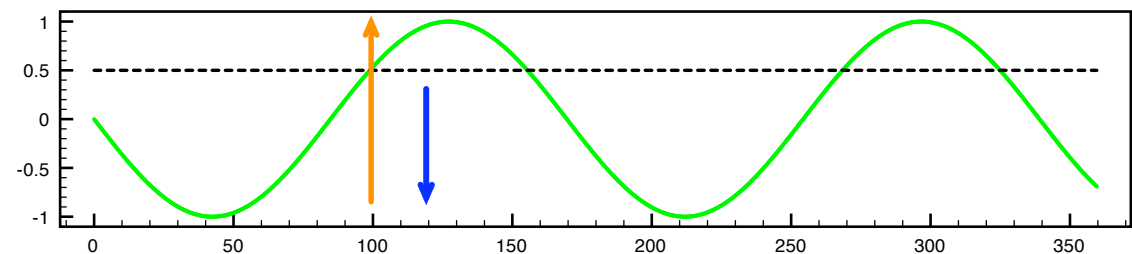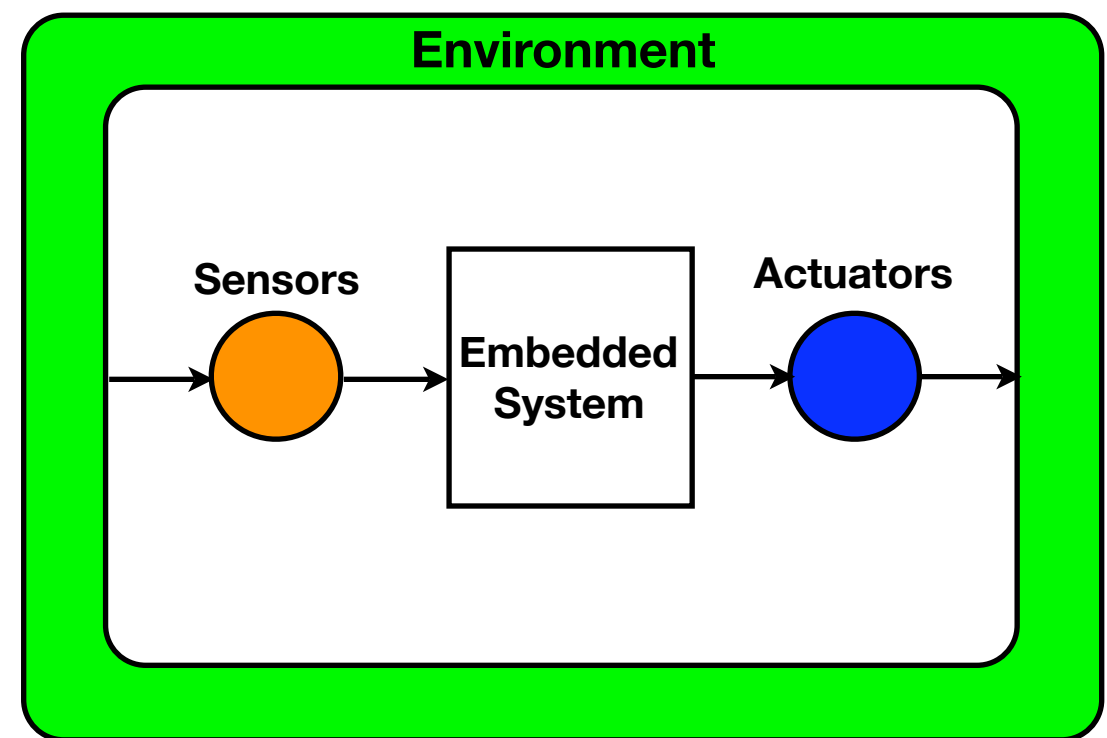
Dissertation Talk, Berkeley, 11/27/2007
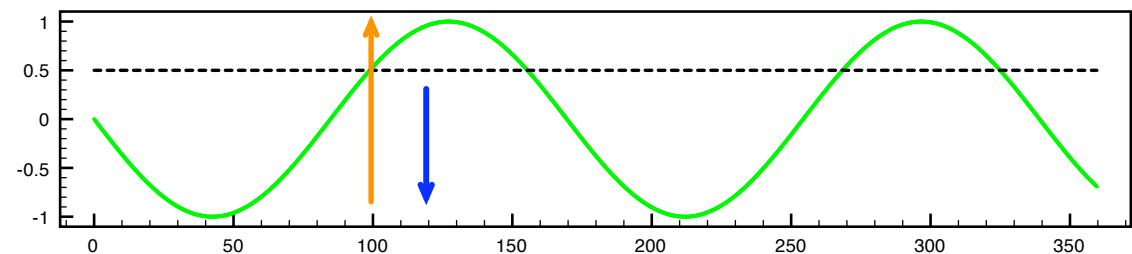
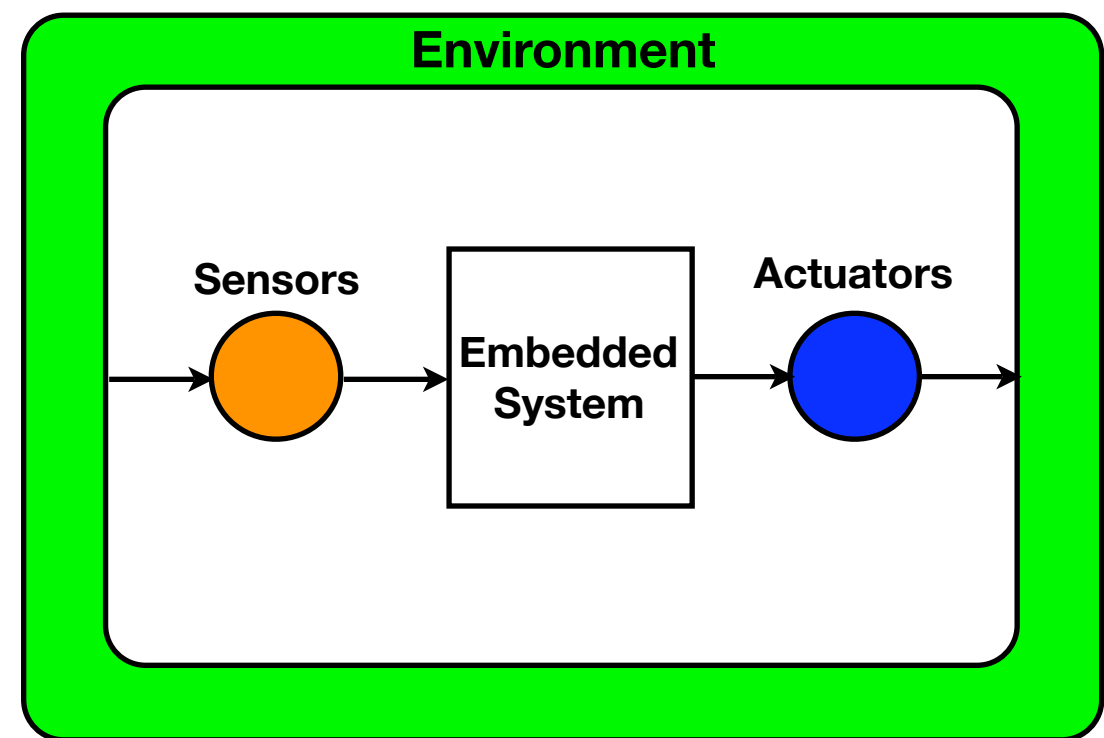# Embedded Systems

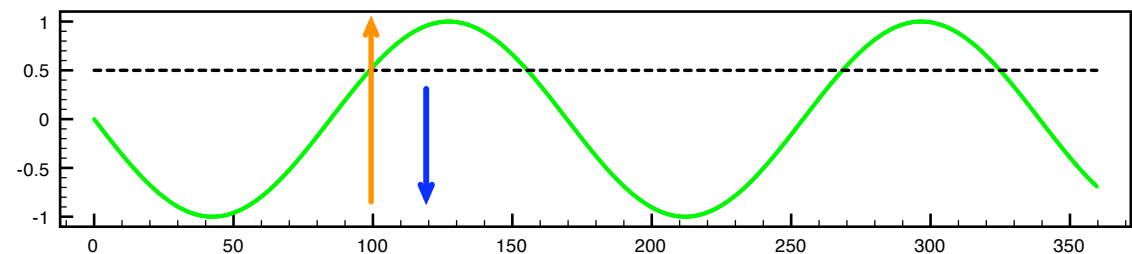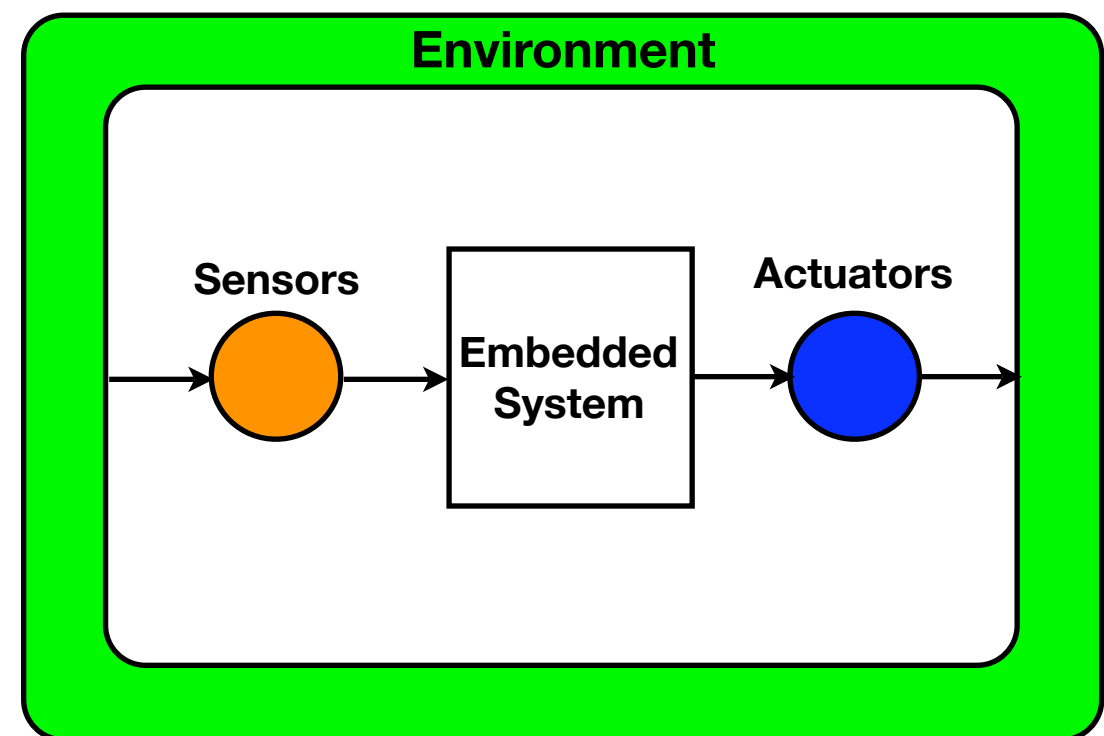# Embedded Systems

- ***Embedded*** in a physical environment

Embedded Systems

- ***Embedded*** in a physical environment

- ***Reacting*** at the speed of the environment

Embedded Systems

- *Embedded* in a physical environment

- *Reacting* at the speed of the environment

- *Heterogeneous* composition of subsystems

- *Networked*
  - Spatially distributed
  - Cooperative

Lifts

Security

2 CCTV

Power

3 UPS System

4 Power distribution

Fire and security monitoring

Elevator

Fixtures

Switch

Server room

Electricity

BMS

HVAC

6 Roof top unit

7 Equipment controller

8 Control board

9 Air terminal

10 Thermostat

Lighting

Fire

12 Fire Panel

13 Sensor device

11 BMS

Examples of Embedded Systems

Building Automation (Source: Clas Jacobson, UTRC)

Examples of Embedded Systems

Automotive

Examples of Embedded Systems
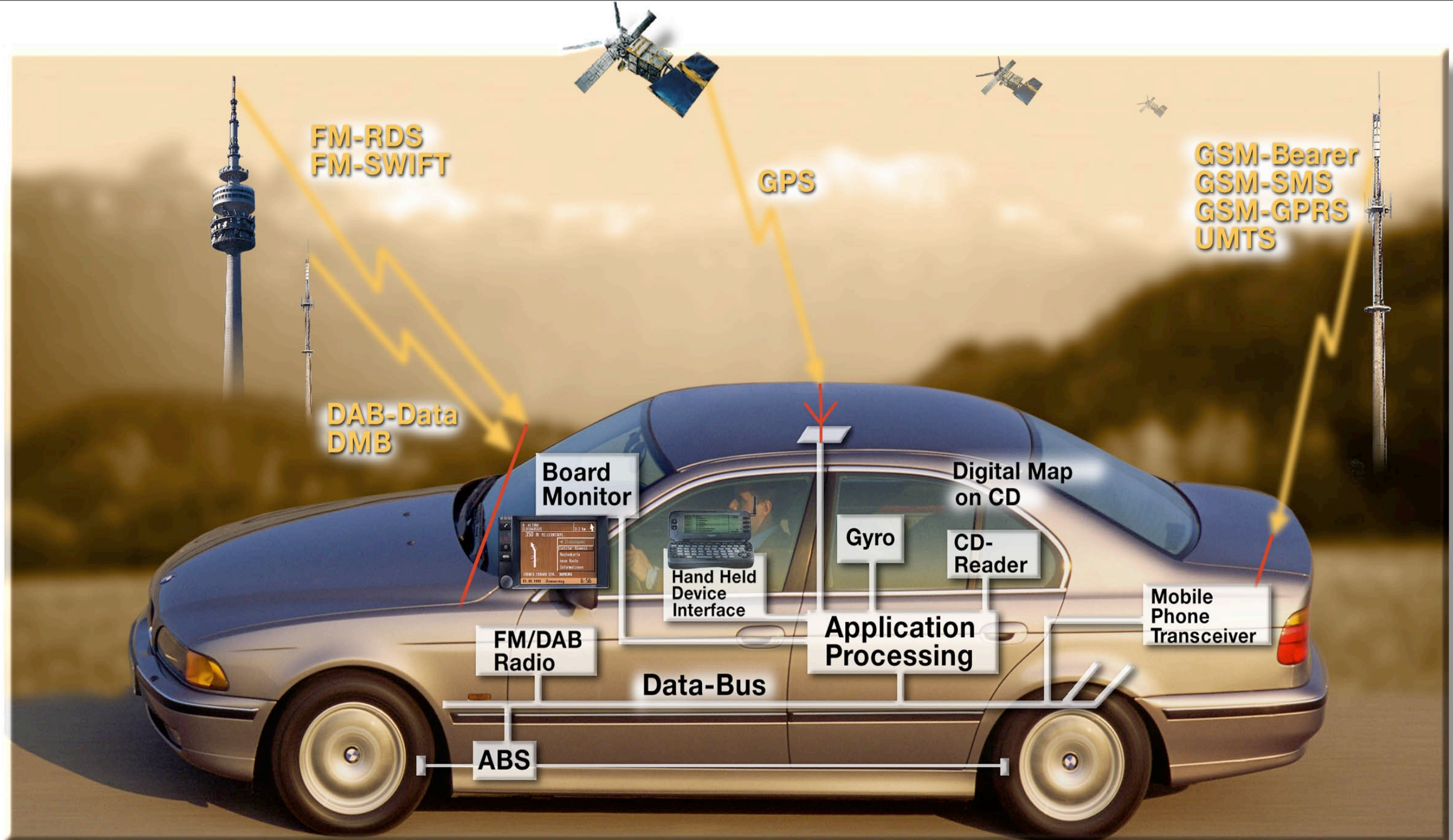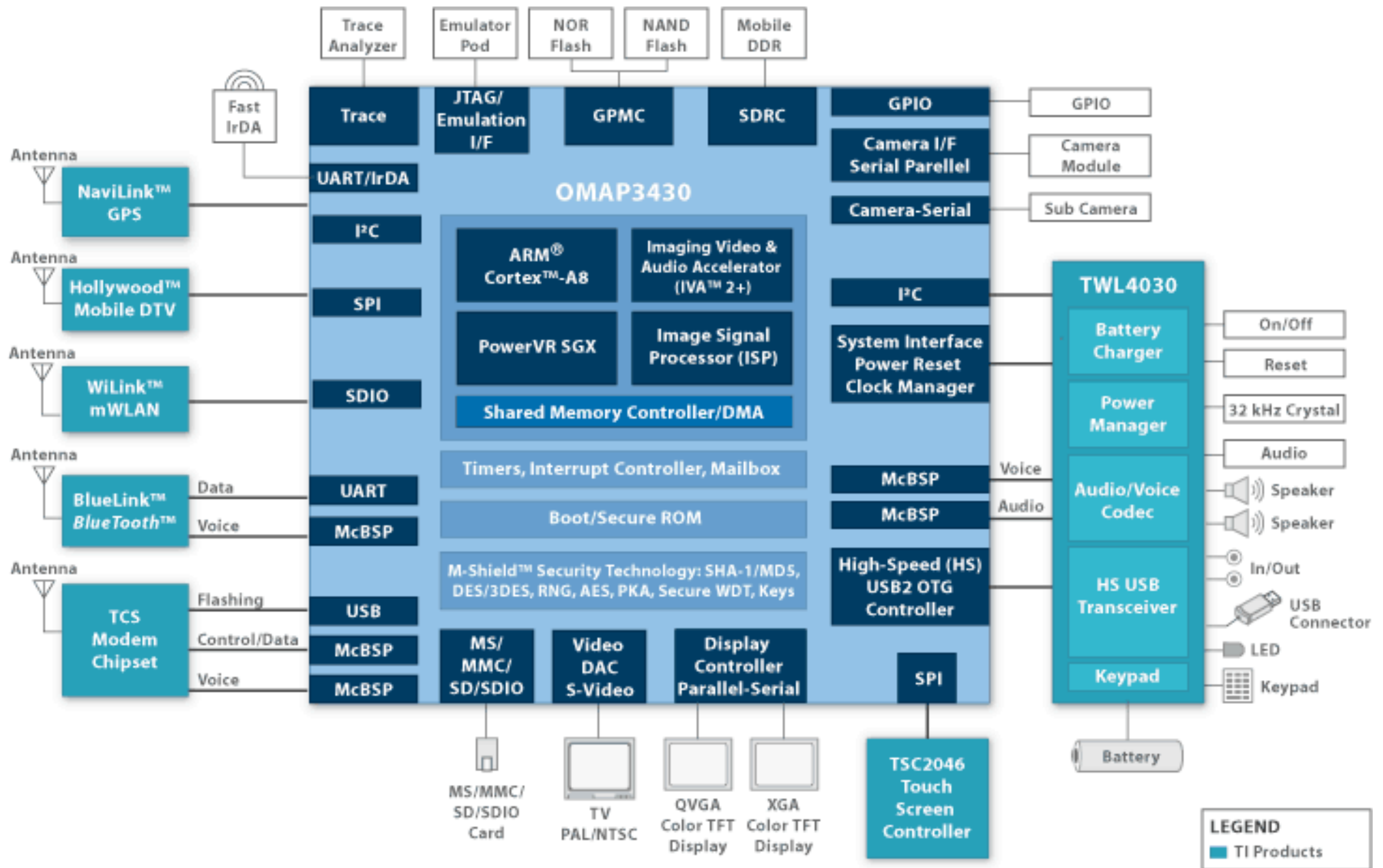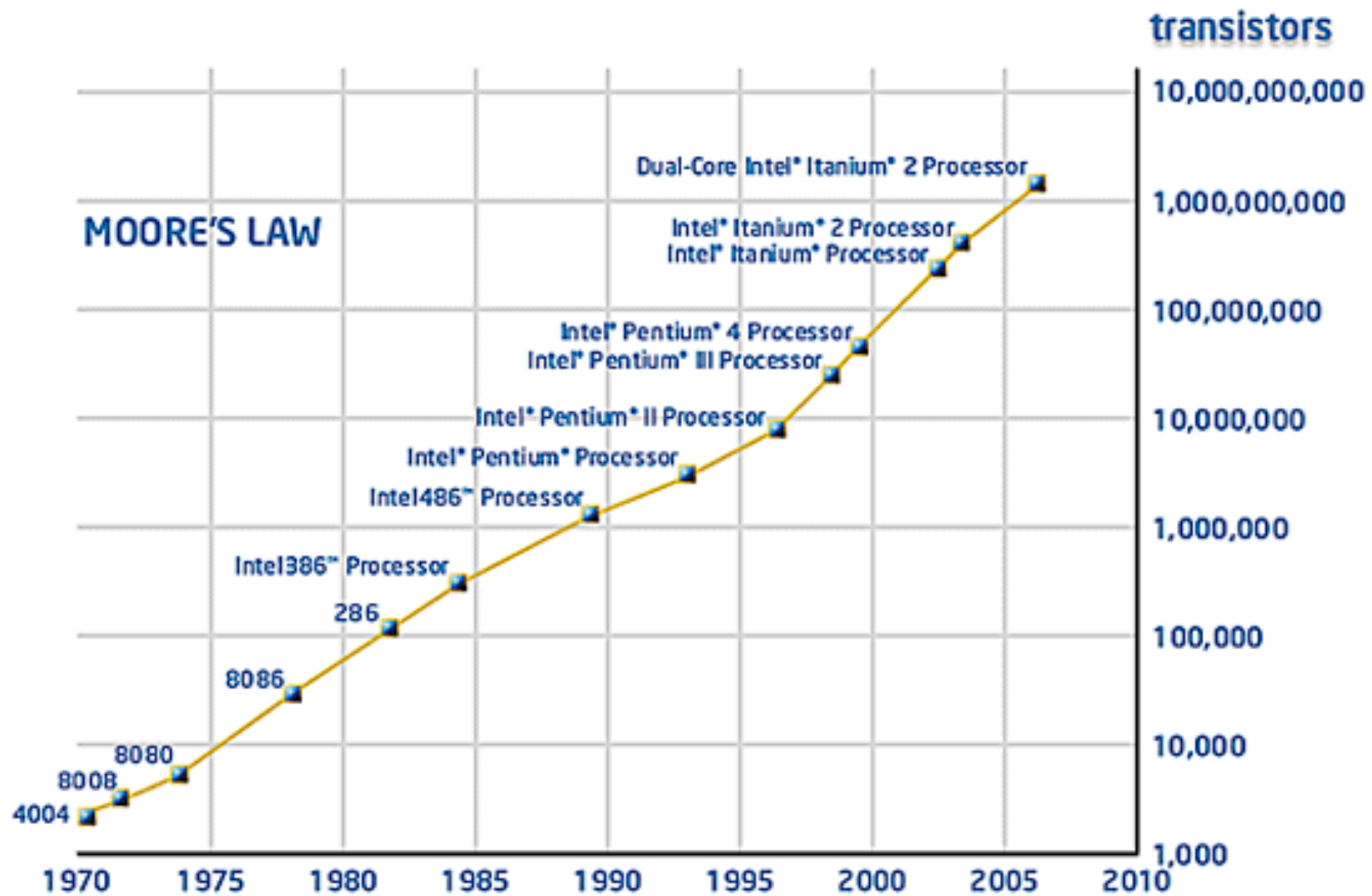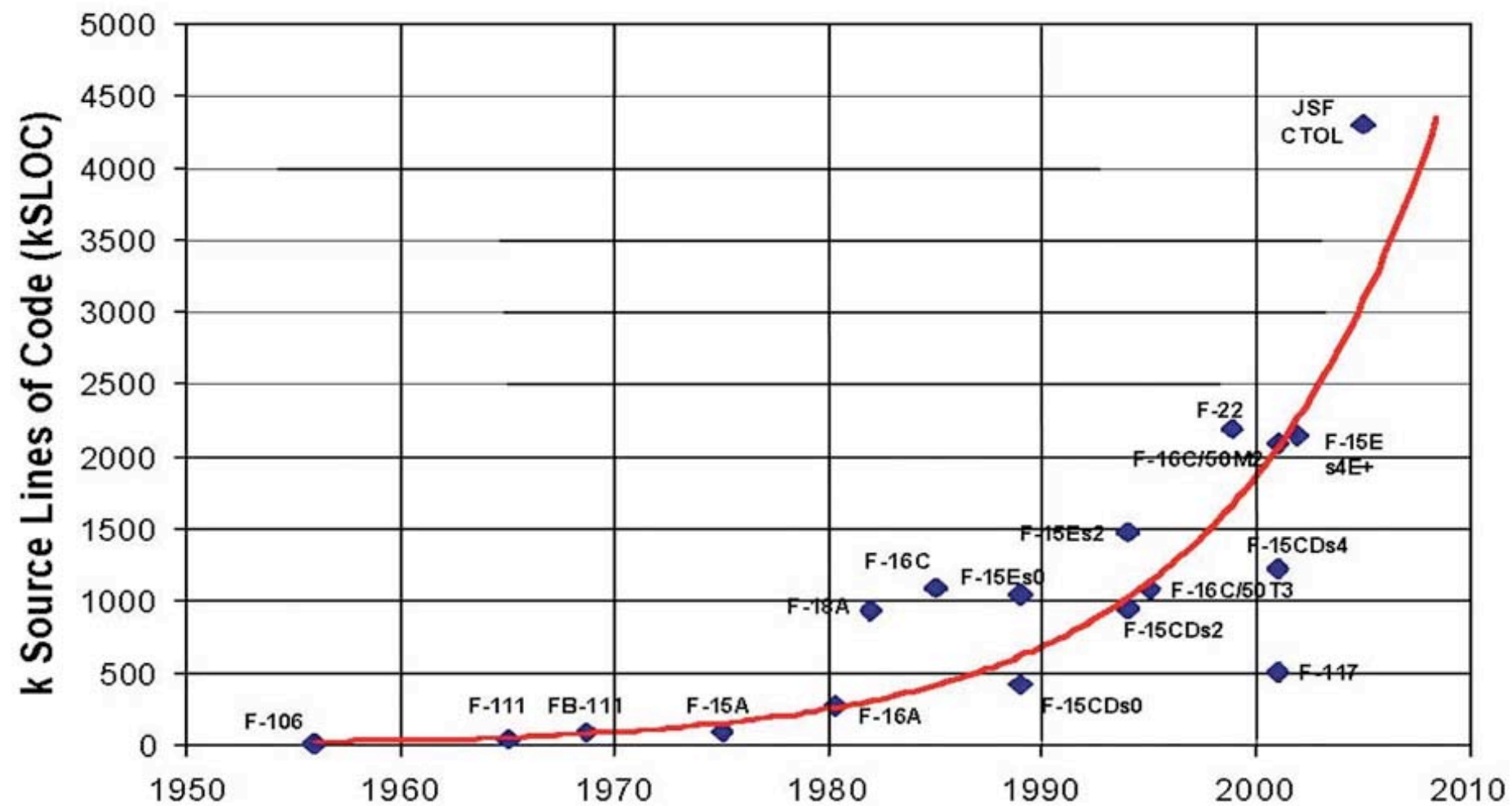
System-On-Chip

Design Complexity Trends

Design Complexity Trends

Design Complexity Trends

Design Complexity Trends

Electronics, Controls & Software Shifting the Basis of Competition in Vehicles

-More functions & features
-Less hardware
-Faster

Potential inflection point. Now!

Software $ 2%
Other $ 9%
Electronics $ 13%
Mechanical $ 76%

Other $ 8%
Software $ 13%
Electronics $ 24%
Mechanical $ 55%

Electric Ignition
...
Electric Fan

ACC
Rear Vision
Passive Entry
Side Airbag
Head Airbag
...

OnStar
OBD II
Data
rad/vid

Hybrid PT
Electric Brake
Door
GDI

Fuel Cell
Wheel Motor
...

$400 20 ECUs 1M Lines
$1182 (+196%) 50 ECUs (+150%) 100M Lines of Code (+9900%)

1970s   1980s   1990s AVG 2000s AVG 2010s   2020s

Vehicle Integration
System Connection
Subsystem Controls & Features
Forefront of Innovation

Value from Electronics & Software

Source: Matt Tsien, GM

EE249Fall07

Productivity (Synopsys 2004)

600000
450000
300000
150000
0

4000 · 5550 · 9090 · 40000 · 56000 · 91000 · 125000 · 200000 · 600000

1990 1993 1995 1997 1999 2001 2003 2005 2007

Productivity (ITRS 2006)

11,000,000
8,250,000
5,500,000
2,750,000
0

2,000,000 · 2,625,000 · 3,425,000 · 4,500,000 · 5,900,000 · 7,725,000 · 10,150,000

2007 2008 2009 2010 2011 2012 2013

Productivity | Productivity in number of gates per designer per year

Productivity | Productivity in number of gates per designer per year

Productivity (Synopsys 2004)

600000
450000
300000
150000
0

1990 · 4000
1993 · 5550
1995 · 9090
1997 · 40000
1999 · 56000
2001 · 91000
2003 · 125000
2005 · 200000
2007 · 600000

ESL

Re-use

Productivity (ITRS 2006)

11,000,000
8,250,000
5,500,000
2,750,000
0

2007 · 2,000,000
2008 · 2,625,000
2009 · 3,425,000
2010 · 4,500,000
2011 · 5,900,000
2012 · 7,725,000
2013 · 10,150,000
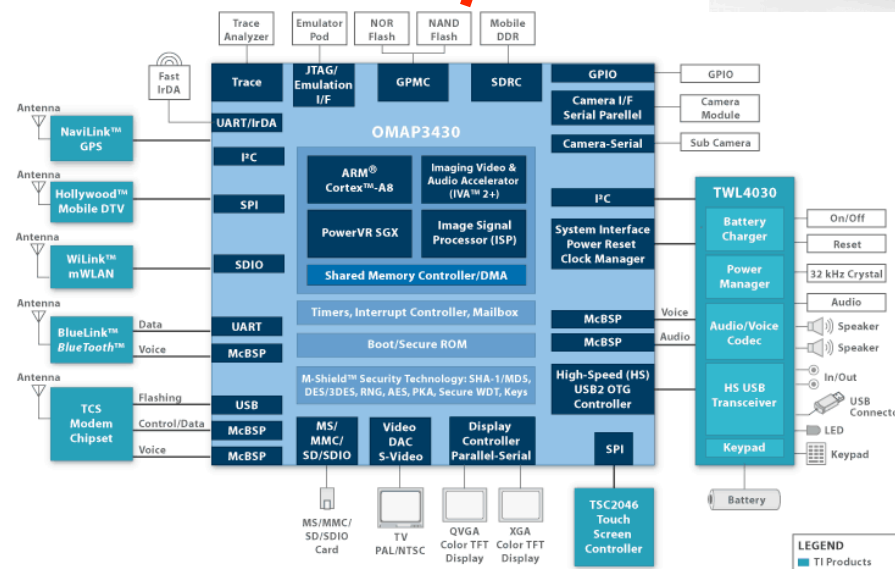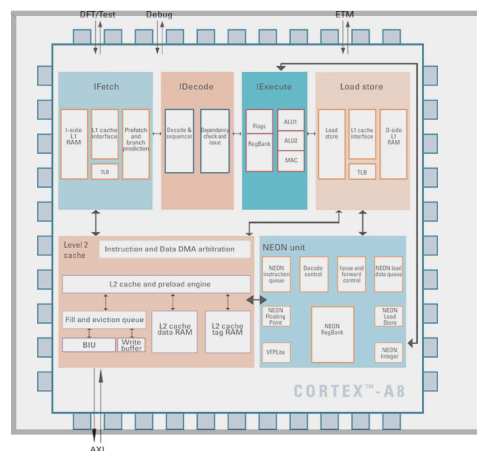
Productivity

Productivity in number of gates per designer per year
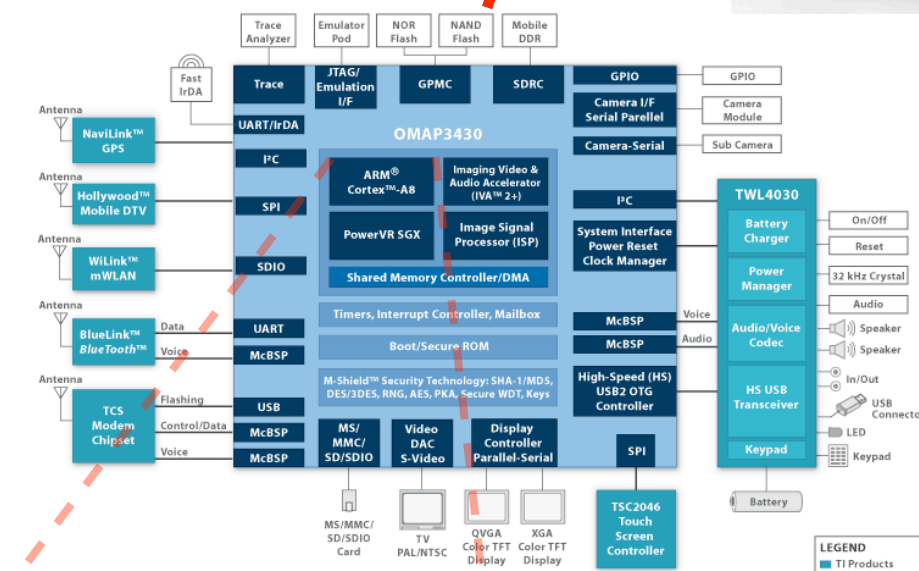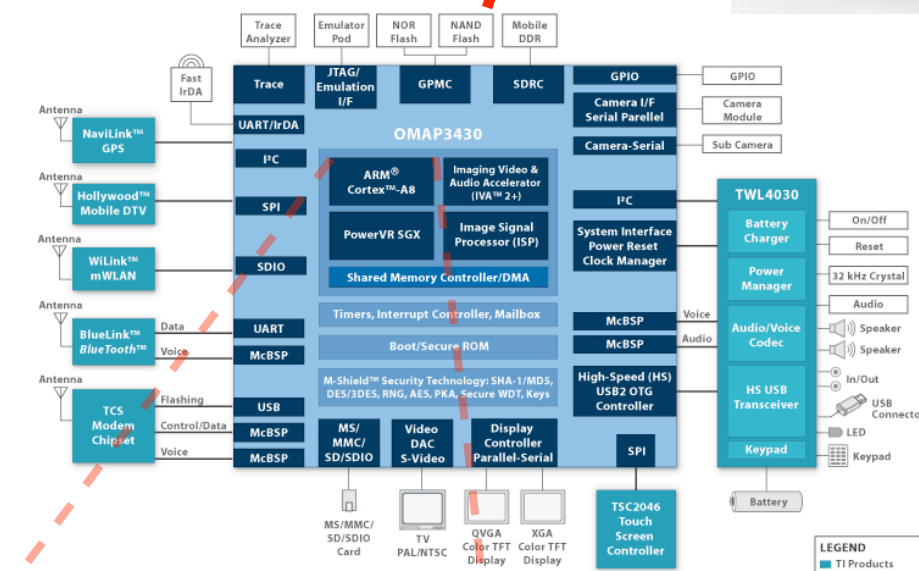
Reuse

Reuse

Reuse

TI OMAP 3

Reuse

TI OMAP 3
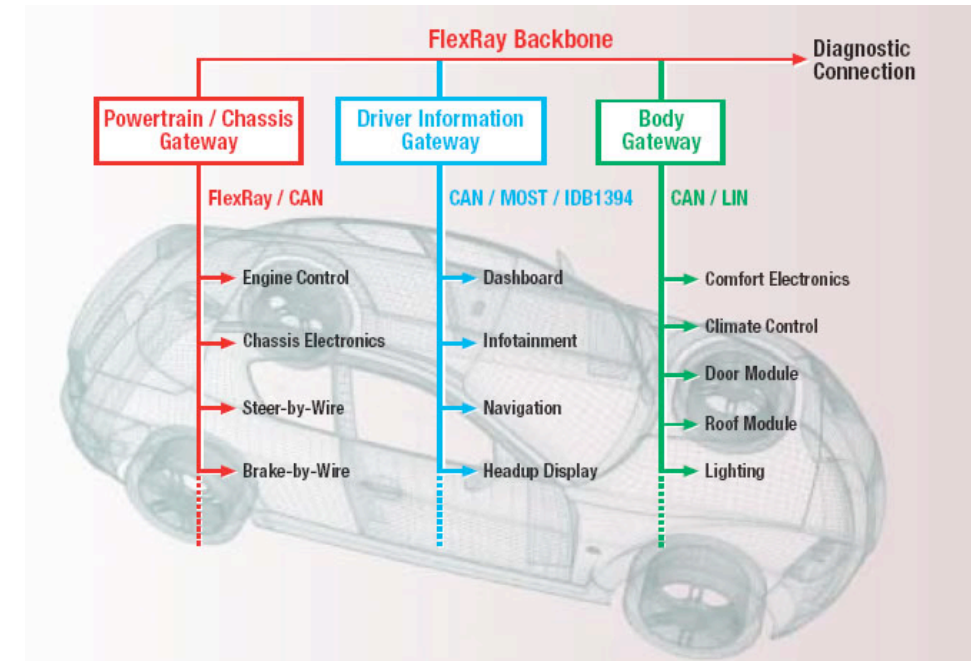
ARM Cortex-A8

Reuse

TI OMAP 3

ARM Cortex-A8

Reuse

TI OMAP 3

Freescale MPC5510

ARM Cortex-A8
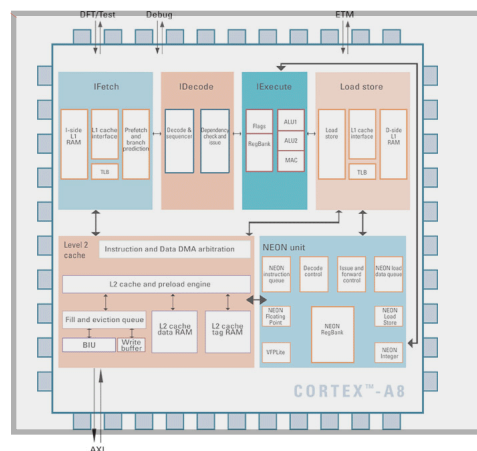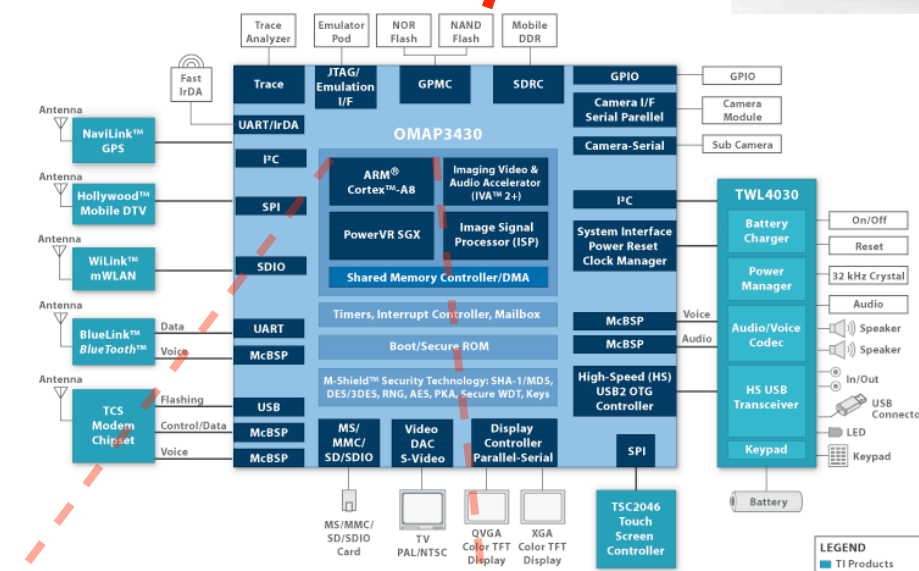
Reuse

TI OMAP 3

ARM Cortex-A8

Freescale MPC5510

e200

Designed for Windows Mobile™

Advantages of Re-Using Components

Advantages of Re-Using Components

- Increased productivity

Advantages of Re-Using Components

- Increased productivity

- Shortened verification time

Advantages of Re-Using Components

• Increased productivity

• Shortened verification time

• Premise

Advantages of Re-Using Components

- Increased productivity

- Shortened verification time

- Premise

  - Compositionality with respect to correctness, i.e. if each component satisfies a property P, then the composition satisfies P

Advantages of Re-Using Components

- Increased productivity

- Shortened verification time

- Premise

  - Compositionality with respect to correctness, i.e. if each component satisfies a property P, then the composition satisfies P

- Approach: constraint the interactions to be such that compositionality holds for P

Outline

- Defining the Problem

  - Building Automation and On-Chip Communication

- A Formal Framework to Enable Synthesis

- Applications

  - Abstraction (Modeling)

  - Algorithms

  - Results

Defining the Problem

Distributed Control

Centralized Control

$$\mathbf{u} \longrightarrow \boxed{C} \longrightarrow \mathbf{y}$$

Distributed Control

Centralized Control

$$\mathbf{u} \rightarrow \boxed{C} \rightarrow \mathbf{y}$$

Distributed Control

Centralized Control

$$\mathbf{u} \longrightarrow \boxed{C} \longrightarrow \mathbf{y}$$

Distributed Control

Distributed Control

Centralized Control

Distributed Control

Distributed Control

Centralized Control

$$\mathbf{u} \longrightarrow \boxed{C} \longrightarrow \mathbf{y}$$

Distributed Control

Communication Specification

Distributed Control

Centralized Control

Distributed Control

Communication Specification

Preserve accuracy
and stability

Available Network Technologies

BacNet

Router

DDC

Sensor

etc.

LonWorks

Router

DDC

Sensor

etc.

ZigBee

Price Reduction

APOGEE

Channels

Tx_1 — $h_{1,1}$ → Rx_1

$h_{1,2}$

$h_{2,1}$

Tx_2 — $h_{2,2}$ → Rx_2

The Physical Aspect of the Problem

The Physical Aspect of the Problem

The Physical Aspect of the Problem

The Physical Aspect of the Problem

Platform-Based Design

Platform-Based Design

Platform-Based Design

Platform-Based Design

Application Space

Platform-Based Design

$(t, b, l, p, \ldots)$

Application
Space

Platform-Based Design

Application
Space

Implementation
Space (Platform)

Platform-Based Design

Application Space

Implementation Space (Platform)

Platform Instance

Platform-Based Design

Constraints
Propagation

Application
Space

$(t, b, l, p, \ldots)$

Implementation
Space (Platform)

Platform Instance

Platform-Based Design

Constraints Propagation

Performance Abstraction

Application Space

Implementation Space (Platform)

Platform Instance

Platform-Based Design

Constraints Propagation

Performance Abstraction

Application Space

Implementation Space (Platform)

Platform Instance

Platform-Based Design

Constraints Propagation

Performance Abstraction

$(t, b, l, p, \ldots)$

Application Space

Implementation Space (Platform)

Platform Instance

Level 2: Data link layer

Platform-Based Design

Constraints Propagation

Performance Abstraction

Application Space

Implementation Space (Platform)

Platform Instance

Level 2: Data link layer

$(t, b, l, p, \ldots)$

Platform-Based Design

Platform-Based Design

Constraints Propagation

Performance Abstraction

Application Space

Implementation Space (Platform)

Platform Instance

Level 2: Data link layer

Level1: Physical layer

Building structure

The On-Chip Communication Specification

The On-Chip Communication Platform

Energy per flit: $35.2pJ$
Leakage @ $1GHz$: $5.1mW$
Area: $31488\mu m^2$

Energy per flit: $8.2pJ$
Leakage @ $1GHz$: $0.85mW$
Area: $5888\mu m^2$

G_1

G_2

Can be placed only on chip boundaries

Can be placed anywhere

Distance $\leq l_{st}$

Bandwidth $\leq b_{max}$

Example of Platform Instances

Platform Instance G_P^1

Platform Instance G_P^2

Compositional Framework

- As in the case of MoC we need to define

  - Agents -> Communication Structures

  - Properties -> Quantities

  - Composition

- A synthesis method selects a composition that implements a function and minimizes total cost

A.Pinto et. al. "A communication synthesis infrastructure for heterogeneous networked control systems and its application to building automation and control", In EMSOFT 2007, October 2007.
A.Pinto et. al. "A Methodology and an Open Software Infrastructure for Constraint-Driven Synthesis of On-Chip Communications"

Communication Structures

$$N(\mathcal{C}, Q, L)$$

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$
Components

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$

Components

v_2

e_1

v_1

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$
Components

$$\{x, y, b\}$$
Quantities

v_1 e_1 v_2

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$
Components

$$\{x, y, b\}$$
Quantities

$$D_Q = D_x \times D_y \times D_b$$
Domain

$v_1 \xrightarrow{e_1} v_2$

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$
Components

$$\{x, y, b\}$$
Quantities

$$D_Q = D_x \times D_y \times D_b$$
Domain

D_b

100$[Mb/s]$

10$[Mb/s]$

$\bot$

v_2

e_1

v_1

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$

Components

$$\{x, y, b\}$$

Quantities

$$D_Q = D_x \times D_y \times D_b$$

Domain

D_b

D_t

100[Mb/s]

10[ns]

10[Mb/s]

100[ns]

$\bot$

$\bot$

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$V \cup E$
Components

$\{x, y, b\}$
Quantities

$$D_Q = D_x \times D_y \times D_b$$
Domain

$D_{\{b,t\}}$

D_b

D_t

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$
Components

$$\{x, y, b\}$$
Quantities

$$\{l | l : \mathcal{C} \to D_Q\}$$
Configurations

$$D_Q = D_x \times D_y \times D_b$$
Domain

Communication Structures

$$N(\mathcal{C}, Q, L)$$

$$V \cup E$$
Components

$$\{x, y, b\}$$
Quantities

$$D_Q = D_x \times D_y \times D_b$$
Domain

$$\{l \,|\, l : \mathcal{C} \to D_Q\}$$
Configurations

$$D_{\{b,t\}}$$

$(1, 1, \perp)$

v_2

e_1

$(\perp, \perp, 10)$

v_1

$(0, 0, \perp)$

D_b

$100[Mb/s]$

$\uparrow$

$10[Mb/s]$

$\uparrow$

$\perp$

D_t

$10[ns]$

$\uparrow$

$100[ns]$

$\uparrow$

$\perp$

$(100[Mb/s], 10[ns])$

$(10[Mb/s], 10[ns])$ $(100[Mb/s], 100[ns])$

$(10[Mb/s], 100[ns])$

$(\perp, 10[ns])$ $(100[Mb/s], \perp)$

$(\perp, 100[ns])$ $(10[Mb/s], \perp)$

$\perp$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$\mathcal{C}_1 \cup \mathcal{C}_2$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad\qquad L_1 \oplus_Q L_2$$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad\qquad L_1 \oplus_Q L_2$$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad\qquad L_1 \oplus_Q L_2$$

Composition

$$|E| = |V| - 1$$

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad\qquad L_1 \oplus_Q L_2$$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1)\|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$|E| = |V| - 1$

$\mathcal{C}_1 \cup \mathcal{C}_2$

$L_1 \oplus_Q L_2$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$$|E| = |V| - 1$$

$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad\qquad L_1 \oplus_Q L_2$$

Composition

$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1)\|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

$$|E| = |V| - 1$$

$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad L_1 \oplus_Q L_2$$

Definition of a Platform

Definition of a Platform

$$r_1(v_1) = sens_1$$
$$r_1(v_2) = router$$

Definition of a Platform

Definition of a Platform

$r_1(N_1)\|_Q^{\mathcal{R}} r_2(N_1)\|_Q^{\mathcal{R}} r_3(N_1)$

$$r_3(v_1) = router$$
$$r_3(v_2) = ctrl$$

$$r_2(v_1) = sens_2$$
$$r_2(v_2) = router$$

$$r_1(v_1) = sens_1$$
$$r_1(v_2) = router$$

Definition of a Platform

Definition of a Platform

$$r_1(N_1) \|_Q^{\mathcal{R}} r_2(N_1) \|_Q^{\mathcal{R}} r_3(N_1)$$

$$\begin{aligned} r_3(v_1) &= \text{router} \\ r_3(v_2) &= \text{ctrl} \end{aligned}$$

$$\begin{aligned} r_2(v_1) &= sens_2 \\ r_2(v_2) &= \text{router} \end{aligned}$$

$$\begin{aligned} r_1(v_1) &= sens_1 \\ r_1(v_2) &= \text{router} \end{aligned}$$

$$\langle \mathcal{L} \rangle = \mathcal{L} \,\cup\, \{N = N' \|_Q^{\mathcal{R}} N'_L : N'_L \subseteq r(N_L), r \in R', \; N_L \in \mathcal{L}, \; N' \in \langle \mathcal{L} \rangle \}$$

Specification, Platform, Mapping

Specification $N_C(C_C, Q_C, L_C)$

Platform Instance $N_P(C_P, Q_P, L_P)$

Specification, Platform, Mapping

Specification $N_C(C_C, Q_C, L_C)$

Platform Instance $N_P(C_P, Q_P, L_P)$

Implementation $N_I(C_I, Q_I, L_I)$

Specification, Platform, Mapping

Specification $N_C(C_C, Q_C, L_C)$

Platform Instance $N_P(C_P, Q_P, L_P)$

Implementation $N_I(C_I, Q_I, L_I)$

Specification, Platform, Mapping

Specification $N_C(C_C, Q_C, L_C)$

Platform Instance $N_P(C_P, Q_P, L_P)$

Implementation $N_I(C_I, Q_I, L_I)$

$$(sens_1, ctrl) \rightarrow ctrl$$
$$(sens_2, ctrl) \rightarrow ctrl$$

Specification, Platform, Mapping

Specification $N_C(C_C, Q_C, L_C)$

Platform Instance $N_P(C_P, Q_P, L_P)$

Implementation $N_I(C_I, Q_I, L_I)$

ψ

$(sens_1, ctrl) \rightarrow ctrl$

$(sens_2, ctrl) \rightarrow ctrl$

Specification, Platform, Mapping

Specification $N_C(C_C, Q_C, L_C)$

Platform Instance $N_P(C_P, Q_P, L_P)$

Implementation $N_I(C_I, Q_I, L_I)$

$(sens_1, ctrl) \rightarrow ctrl$

$(sens_2, ctrl) \rightarrow ctrl$

The General Synthesis Problem

The General Synthesis Problem

Specification
domain

$\mathcal{G}_{Q_C}$

The General Synthesis Problem

Specification
domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

The General Synthesis Problem

Specification
domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$\mathcal{G}_{Q_I}$

The General Synthesis Problem

Specification
domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

The General Synthesis Problem

Specification
domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

Π

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

The General Synthesis Problem

The General Synthesis Problem

Specification domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_P(\mathcal{C}_P, Q_P, L_P)$

$N_C(\mathcal{C}_C, Q_C, L_C)$

ψ

Π

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

The General Synthesis Problem

Specification domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_P(\mathcal{C}_P, Q_P, L_P)$

$N_C(\mathcal{C}_C, Q_C, L_C)$

Ψ

Π

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

$$PR1: \quad \min_{L_I} \quad F(N_I)$$

$$subject\ to \quad N_C \leq_{Q_C} \Pi(N_I), \ \Psi(N_I) \subseteq N_P$$

The General Synthesis Problem

Specification domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_P(\mathcal{C}_P, Q_P, L_P)$

$N_C(\mathcal{C}_C, Q_C, L_C)$

Ψ

Π

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

PR1 : $\displaystyle\min_{L_I} \quad F(N_I)$

$subject\ to \quad N_C \leq_{Q_C} \Pi(N_I), \ \Psi(N_I) \subseteq N_P$

Lemma

$$N_P^1 \leq_{Q_I} N_P^2 \Rightarrow F(N_I^1) \leq F(N_I^2)$$

The General Synthesis Problem

Specification domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_P^{\langle \mathcal{L} \rangle}$

$N_P(\mathcal{C}_P, Q_P, L_P)$

$N_C(\mathcal{C}_C, Q_C, L_C)$

Ψ

Π

PR1 : $\quad \min_{L_I} \quad F(N_I)$

$\quad subject \ to \quad N_C \leq_{Q_C} \Pi(N_I), \ \Psi(N_I) \subseteq N_P$

Lemma

$$N_P^1 \leq_{Q_I} N_P^2 \Rightarrow F(N_I^1) \leq F(N_I^2)$$

$N_I(\mathcal{C}_I, Q_I, L_I) \quad \mathcal{G}_{Q_I}$

The General Synthesis Problem

PR1 :
$$\min_{L_I} \quad F(N_I)$$

$$subject \ to \quad N_C \leq_{Q_C} \Pi(N_I), \ \Psi(N_I) \subseteq N_P$$

Lemma

$$N_P^1 \leq_{Q_I} N_P^2 \Rightarrow F(N_I^1) \leq F(N_I^2)$$

PR2 :
$$\min_{L_I} \quad F(N_I)$$

$$subject \ to \quad N_C \leq \Pi(\Phi(N_I)), \ \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$$

$$\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$$

The General Synthesis Problem

Specification domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_P^{\langle \mathcal{L} \rangle}$

$N_P(\mathcal{C}_P, Q_P, L_P)$

$N_C(\mathcal{C}_C, Q_C, L_C)$

Ψ

Π

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

$\text{PR1}:\qquad \min_{L_I} \qquad F(N_I)$

$subject\ to\quad N_C \leq_{Q_C} \Pi(N_I),\ \ \Psi(N_I) \subseteq N_P$

Lemma

$$N_P^1 \leq_{Q_I} N_P^2 \Rightarrow F(N_I^1) \leq F(N_I^2)$$

$\text{PR2}:\qquad \min_{L_I} \qquad F(N_I)$

$subject\ to\quad N_C \leq \Pi(\Phi(N_I)),\ \ \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$

Depends on the composition rules $\Rightarrow$ $\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$

The General Synthesis Problem

Specification domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_P^{\langle \mathcal{L} \rangle}$

$N_P(\mathcal{C}_P, Q_P, L_P)$

$N_C(\mathcal{C}_C, Q_C, L_C)$

Ψ

Π

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

PR1 : $\min\limits_{L_I}$ $F(N_I)$

$subject\ to$ $N_C \leq_{Q_C} \Pi(N_I),\ \Psi(N_I) \subseteq N_P$

Lemma

$$N_P^1 \leq_{Q_I} N_P^2 \Rightarrow F(N_I^1) \leq F(N_I^2)$$

Depends on quantities and library

PR2 : $\min\limits_{L_I}$ $F(N_I)$

$subject\ to$ $N_C \leq \Pi(\Phi(N_I)),\ \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$

Depends on the composition rules

$\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$

The General Synthesis Problem

Specification domain

$\mathcal{G}_{Q_C}$

$\mathcal{G}_{Q_P}$ Platform

$N_P^{\langle \mathcal{L} \rangle}$

$N_P(\mathcal{C}_P, Q_P, L_P)$

$N_C(\mathcal{C}_C, Q_C, L_C)$

Ψ

Π

$N_I(\mathcal{C}_I, Q_I, L_I)$ $\mathcal{G}_{Q_I}$

Theorem

$N_P^{\langle \mathcal{L} \rangle}$ exists.

$$\text{PR1}: \quad \min_{L_I} \quad F(N_I)$$

$$subject\ to \quad N_C \leq_{Q_C} \Pi(N_I), \ \Psi(N_I) \subseteq N_P$$

Lemma

$$N_P^1 \leq_{Q_I} N_P^2 \Rightarrow F(N_I^1) \leq F(N_I^2)$$

$$\text{PR2}: \quad \min_{L_I} \quad F(N_I)$$

Depends on quantities and library

$$subject\ to \quad N_C \leq \Pi(\Phi(N_I)), \ \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$$

Depends on the composition rules

$$\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$$

A General Method for Communication Design

- Define the library L

- Define the rules R

- Define the models M

- Find the upper bound

- Formulate the specific problem: Given the properties to preserve and L, R, M

- Find an efficient algorithm

Communication Synthesis for Building Automation Systems

The Building Automation Segment

- The building segment consumes 40% of domestic energy produced in the US

- Estimated 40% reduction in energy consumption through advanced control

- Comfort and safety can also be improved: 95% false alarms

- Solution: Integrated design

    - Advance and reliable control software

    - Reliable communication

The Library of Communication Components

- Twisted-pair wires

- Daisy-chain connection

- ARCNET protocol (token ring bus)

- Wireless communication channels

- Tree topology

- ZigBee (802.15.4)

Capturing the Building Geometry

$$\mathbf{p} = \mathbf{p_1} + t_1 \mathbf{a}_1 + t_2 \mathbf{a}_2$$

$$0 \leq t_1 \leq ||\mathbf{p}_2 - \mathbf{p}_1||$$

$$0 \leq t_2 \leq ||\mathbf{p}_4 - \mathbf{p}_1||$$

- Capture any surface

- Walls as special cases (adding thickness and material)

- Cabling constrained on special surfaces

- Number of walls intersected by a line becomes a simple set of linear equations

Modeling the ARCNET Protocol

Modeling the ARCNET Protocol

Algorithms

Algorithms

- The idea is to cover each sensor and each actuator with exactly one chain

Algorithms

- The idea is to cover each sensor and each actuator with exactly one chain

- Suppose that there is an oracle that gives us the set of all possible valid daisy chain busses containing exactly one router

Algorithms

- The idea is to cover each sensor and each actuator with exactly one chain

- Suppose that there is an oracle that gives us the set of all possible valid daisy chain busses containing exactly one router

$$\min \quad \sum_{j=1}^{n} f_j z_j$$

$$s.t. \quad \sum_{j=1}^{n} x_{ij} z_j = 1, \forall i \quad \sum_{j=1}^{n} y_{jk} z_j = 1, \forall k$$

$$z_j, x_{ij}, y_{jk} \in \{0, 1\}$$

Algorithms

- The idea is to cover each sensor and each actuator with exactly one chain

- Suppose that there is an oracle that gives us the set of all possible valid daisy chain busses containing exactly one router

j-th chain

$$\min \quad \sum_{j=1}^{n} f_j z_j$$

$$s.t. \quad \sum_{j=1}^{n} x_{ij} z_j = 1, \forall i \quad \sum_{j=1}^{n} y_{jk} z_j = 1, \forall k$$

$$z_j, x_{ij}, y_{jk} \in \{0,1\}$$

Algorithms

- The idea is to cover each sensor and each actuator with exactly one chain

- Suppose that there is an oracle that gives us the set of all possible valid daisy chain busses containing exactly one router

j-th chain

chain j contains router i

$$\min \quad \sum_{j=1}^{n} f_j z_j$$

$$s.t. \quad \sum_{j=1}^{n} x_{ij} z_j = 1, \forall i \quad \sum_{j=1}^{n} y_{jk} z_j = 1, \forall k$$

$$z_j, x_{ij}, y_{jk} \in \{0,1\}$$

Algorithms

- The idea is to cover each sensor and each actuator with exactly one chain

- Suppose that there is an oracle that gives us the set of all possible valid daisy chain busses containing exactly one router

j-th chain

chain j contains router i

$$\min \quad \sum_{j=1}^{n} f_j z_j$$

$$s.t. \quad \sum_{j=1}^{n} x_{ij} z_j = 1, \forall i \quad \sum_{j=1}^{n} y_{jk} z_j = 1, \forall k$$

$$z_j, x_{ij}, y_{jk} \in \{0, 1\}$$

node k belongs to chain j

Algorithms: elcarO ehT

$$v_4 \quad v_3 \quad i_1 \quad v_2 \quad i_2 \quad v_1$$

v_1

i_2

v_2

i_1

v_3

v_4

Algorithms: elcarO ehT

$$v_4 \quad v_3 \quad i_1 \quad v_2 \quad i_2 \quad v_1$$

v_1

c_1

i_2

v_2

i_1

v_3

v_4

Algorithms: elcarO ehT

Algorithms: The Oracle

Algorithms: The Oracle

Algorithms: The Oracle

Algorithms: The Oracle

Algorithms: The Oracle

Algorithms: The Oracle

Algorithms: The Oracle

Algorithms: The Oracle

Modeling the ZigBee Protocol

Modeling the ZigBee Protocol

Physical Layer

$$P_b = f(SNR)$$

$$L = L_1 + 20\log_{10} r + n_f a_f + n_w a_w$$

Modeling the ZigBee Protocol

MAC layer

$aUnitBackoffPeriod$

$2^{SO} \cdot aBaseSlotDuration$

CAP

CFP

GTS_1 GTS_2

ACTIVE

INACTIVE

$SD = aBaseSuperframeDuration \cdot 2^{SO}$

$BI = aBaseSuperframeOrder \cdot 2^{BO}$

Physical Layer

$P_b = f(SNR)$

$L = L_1 + 20\log_{10} r + n_f a_f + n_w a_w$

Modeling the ZigBee Protocol

Modeling the ZigBee Protocol

i-th router

q-th end-to-end flow
uses link i to j

$$\min_{\mathbf{x},\mathbf{y}_q} \ F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

$s.t.$

i-th router

q-th end-to-end flow
uses link i to j

j is the parent
of i

Rules

$$\min_{\mathbf{x},\mathbf{y}_q} \; F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

$s.t.$

1. $x_i + x_j - 2e_{i,j} \geq 0 \qquad \forall i,j \in N \cup M$

2. $e_{ij} + e_{ji} \leq 1 \qquad \forall i,j \in N \cup M$

3. $e_{ij} = 0 \qquad \forall j \in N$

4. $\sum_{ij} e_{ij} - \sum_i x_i = -1 \qquad \forall i \in N \cup M$

5. $\sum_i e_{ij} \leq in_{max} \qquad \forall j \in M$

i-th router

q-th end-to-end flow
uses link i to j

$$\min_{\mathbf{x}, \mathbf{y}_q} \ F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

j is the parent
of i

$s.t.$

1. $x_i + x_j - 2e_{i,j} \geq 0$ $\forall i, j \in N \cup M$

2. $e_{ij} + e_{ji} \leq 1$ $\forall i, j \in N \cup M$

3. $e_{ij} = 0$ $\forall j \in N$

4. $\sum_{ij} e_{ij} - \sum_i x_i = -1$ $\forall i \in N \cup M$

5. $\sum_i e_{ij} \leq in_{max}$ $\forall j \in M$

Rules

Flow conditions

6. $e_{ij} + e_{ji} - y_{ijq} \geq 0$ $\forall i, j \in N \cup M, \forall q \in Q$

7. $e_{ij} + e_{ji} - y_{jiq} \geq 0$ $\forall i, j \in N \cup M, \forall q \in Q$

8. $A_q \mathbf{y}_q = \mathbf{b}_q$ $\forall q \in Q$

i-th router

q-th end-to-end flow
uses link i to j

$$\min_{\mathbf{x},\mathbf{y}_q} F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

j is the parent
of i

$s.t.$

1. $x_i + x_j - 2e_{i,j} \geq 0 \qquad \forall i, j \in N \cup M$
2. $e_{ij} + e_{ji} \leq 1 \qquad \forall i, j \in N \cup M$
3. $e_{ij} = 0 \qquad \forall j \in N$
4. $\sum_{ij} e_{ij} - \sum_i x_i = -1 \qquad \forall i \in N \cup M$
5. $\sum_i e_{ij} \leq in_{max} \qquad \forall j \in M$

Rules

Flow conditions

6. $e_{ij} + e_{ji} - y_{ijq} \geq 0 \qquad \forall i, j \in N \cup M, \forall q \in Q$
7. $e_{ij} + e_{ji} - y_{jiq} \geq 0 \qquad \forall i, j \in N \cup M, \forall q \in Q$
8. $A_q \mathbf{y}_q = \mathbf{b}_q \qquad \forall q \in Q$

Bw constraints

9. $\sum_q y_{ijq}(b_q + O) \leq b_{max} \qquad \forall i, j \in N \cup M$

i-th router

q-th end-to-end flow
uses link i to j

$$\min_{\mathbf{x},\mathbf{y}_q} F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

j is the parent
of i

$s.t.$

1. $x_i + x_j - 2e_{i,j} \geq 0 \qquad \forall i,j \in N \cup M$
2. $e_{ij} + e_{ji} \leq 1 \qquad \forall i,j \in N \cup M$
3. $e_{ij} = 0 \qquad \forall j \in N$
4. $\sum_{ij} e_{ij} - \sum_i x_i = -1 \quad \forall i \in N \cup M$
5. $\sum_i e_{ij} \leq in_{max} \qquad \forall j \in M$

Rules

Flow conditions

6. $e_{ij} + e_{ji} - y_{ijq} \geq 0 \quad \forall i,j \in N \cup M, \forall q \in Q$
7. $e_{ij} + e_{ji} - y_{jiq} \geq 0 \quad \forall i,j \in N \cup M, \forall q \in Q$
8. $A_q \mathbf{y}_q = \mathbf{b}_q \qquad \forall q \in Q$

Bw constraints

9. $\sum_q y_{ijq}(b_q + O) \leq b_{max} \quad \forall i,j \in N \cup M$
10. $\sum_{i \in M} x_i \leq n_{max}$

i-th router

q-th end-to-end flow
uses link i to j

$$\min_{\mathbf{x},\mathbf{y}_q} \; F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

j is the parent
of i

$s.t.$

1. $x_i + x_j - 2e_{i,j} \geq 0$ $\qquad \forall i,j \in N \cup M$

2. $e_{ij} + e_{ji} \leq 1$ $\qquad \forall i,j \in N \cup M$

3. $e_{ij} = 0$ $\qquad \forall j \in N$

Rules

4. $\sum_{ij} e_{ij} - \sum_i x_i = -1$ $\quad \forall i \in N \cup M$

5. $\sum_i e_{ij} \leq in_{max}$ $\qquad \forall j \in M$

Flow conditions

6. $e_{ij} + e_{ji} - y_{ijq} \geq 0$ $\quad \forall i,j \in N \cup M, \forall q \in Q$

7. $e_{ij} + e_{ji} - y_{jiq} \geq 0$ $\quad \forall i,j \in N \cup M, \forall q \in Q$

8. $A_q \mathbf{y}_q = \mathbf{b}_q$ $\qquad \forall q \in Q$

Bw constraints

9. $\sum_q y_{ijq}(b_q + O) \leq b_{max}$ $\quad \forall i,j \in N \cup M$

10. $\sum_{i \in M} x_i \leq n_{max}$

Latency constraints

11. $\sum_{ij} y_{ijq} l(i,j) \leq l_q$ $\quad \forall q \in Q$

i-th router

q-th end-to-end flow
uses link i to j

$$\min_{\mathbf{x},\mathbf{y}_q} F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

j is the parent
of i

$s.t.$

1. $x_i + x_j - 2e_{i,j} \geq 0 \qquad \forall i,j \in N \cup M$

2. $e_{ij} + e_{ji} \leq 1 \qquad \forall i,j \in N \cup M$

3. $e_{ij} = 0 \qquad \forall j \in N$

4. $\sum_{ij} e_{ij} - \sum_i x_i = -1 \quad \forall i \in N \cup M$

5. $\sum_i e_{ij} \leq in_{max} \qquad \forall j \in M$

Rules

Flow conditions

6. $e_{ij} + e_{ji} - y_{ijq} \geq 0 \quad \forall i,j \in N \cup M, \forall q \in Q$

7. $e_{ij} + e_{ji} - y_{jiq} \geq 0 \quad \forall i,j \in N \cup M, \forall q \in Q$

8. $A_q \mathbf{y}_q = \mathbf{b}_q \qquad \forall q \in Q$

Bw constraints

9. $\sum_q y_{ijq}(b_q + O) \leq b_{max} \quad \forall i,j \in N \cup M$

10. $\sum_{i \in M} x_i \leq n_{max}$

Latency constraints

11. $\sum_{ij} y_{ijq} l(i,j) \leq l_q \quad \forall q \in Q$

PER constraints

12. $\sum_{ij} y_{ijq} \log p'(i,j) \leq \log(1 - p_q) \quad \forall q \in Q$

i-th router

q-th end-to-end flow
uses link i to j

j is the parent
of i

$$\min_{\mathbf{x},\mathbf{y}_q} F = \sum_i c_i x_i + \sum_{ij} c_{ij} \sum_q y_{ijq}$$

$s.t.$

1. $x_i + x_j - 2e_{i,j} \geq 0$ $\quad \forall i,j \in N \cup M$
2. $e_{ij} + e_{ji} \leq 1$ $\quad \forall i,j \in N \cup M$
3. $e_{ij} = 0$ $\quad \forall j \in N$
4. $\sum_{ij} e_{ij} - \sum_i x_i = -1$ $\quad \forall i \in N \cup M$
5. $\sum_i e_{ij} \leq in_{max}$ $\quad \forall j \in M$

Rules

Flow conditions

6. $e_{ij} + e_{ji} - y_{ijq} \geq 0$ $\quad \forall i,j \in N \cup M, \forall q \in Q$
7. $e_{ij} + e_{ji} - y_{jiq} \geq 0$ $\quad \forall i,j \in N \cup M, \forall q \in Q$
8. $A_q \mathbf{y}_q = \mathbf{b}_q$ $\quad \forall q \in Q$

Bw constraints

9. $\sum_q y_{ijq}(b_q + O) \leq b_{max}$ $\quad \forall i,j \in N \cup M$

10. $\sum_{i \in M} x_i \leq n_{max}$

Latency constraints

11. $\sum_{ij} y_{ijq} l(i,j) \leq l_q$ $\quad \forall q \in Q$

PER constraints

12. $\sum_{ij} y_{ijq} \log p'(i,j) \leq \log(1 - p_q)$ $\quad \forall q \in Q$

13. $x_i,\ e_{ij},\ y_{ijq} \in \{0,1\}$ $\quad \forall i,j \in N \cup M, \forall q \in Q$

NETWORKS

PBD Communication Synthesis Flow

PLATFORMS

File · Data Structure · Algorithm

IdGraph
AdjList
...
AddVertex(V : int)
AddEdge(U : int , V : int)
InV(V : int) : bool
InE(U : int , V : int) : bool
...

P_B_Network
mP : map< int , Point >
mB : map< pair< int, int > CosiDouble >
GetP(V : int) : Point
GetB(U : int , V : int) : CosiDouble
operator +(G : P_B_Network&) : P_B_Network
...

Variable
mAny : bool
...
IsAny( ) : bool
...

Point
mX : CosiDouble
mY : CosiDouble
mZ : CosiDouble
...
GetX( ) : Point
...

CosiDouble
mValue : double
...
SetValue(D : double)
GetValue( ) : double
...

Synthesis
...
Initialize(Spec : XC_WC_Network* , Plt : Platform*)
Run( )
GetResult( ) : XI_WI_Network

LonWorksModel
...
double NodePrice(string)
...

PerformanceCostModel
mNodes : vector< Node >
mLinks : vector< Links >
...
virtual NodePrice(string) : double
...

1...*
1

Library
mCmp : map< int , Component >
Cost(Cmp : int , X : XI , W : WI) : double
...

1...*
1

Platform
mW : vector< Wall >
mS : vector< Surface >
...
GetPath(P1 : Point , P2 : Point) : vector< Point >
Distance(P1 : Point , P2 : Point) : double
Cost(Cmp : XI_WI_Network) : double
...

Project · Comm. Constrains · Opt · Algorithm 1 · Algorithm 2 · Algorithm 3 · Building (Walls, Surfaces) · $\mathcal{L}$ · $\langle\mathcal{L}\rangle$ · Components · Quantities

G_C · G_I · Verification · Y/N

Code Generation · Dot · Svg · Report

Next level of abstraction · Synthesis · $\langle\mathcal{L}'\rangle$ · Comp · Quantities · G_I'

COSI-BAS

- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
- .Capture network components
- .Automatic synthesis

NETWORKS

PBD Communication Synthesis Flow

PLATFORMS

COSI-BAS

- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
- .Capture network components
- .Automatic synthesis

File Data Structure Algorithm

Synthesis

...
Initialize(Spec : XC_WC_Network* , Plt : Platform*)
Run( )
GetResult( ) : XI_WI_Network

LonW
...
double Nod...

Performa
mNodes : vecto
mLinks : vecto
...
virtual NodePri...

L
mCmp : map< int ,
Cost(Cmp : int , ...
...

P
mW : vector< Wall >
mS : vector< Surface >
...
GetPath(P1 : Point ,
Distance(P1 : Point ,
Cost(Cmp : XI_WI_Ne...
...

PBD Communication Synthesis Flow

Project
Opt
Comm. Constrains
Building (Walls, Surfaces)
G_C
Algorithm 1
Algorithm 2
Algorithm 3
$\langle \mathcal{L} \rangle$
$\mathcal{L}$
Components
Quantities
Verification
G_I
Y/N
Next level of abstraction
Code Generation
Dot Svg Report
Synthesis
$\langle \mathcal{L}' \rangle$
Comp
Quantities
G_I'

: int)
: bool

Double >
...
uble
) : P_B_Network

Variable
mAny : bool
...
IsAny( ) : bool
...

CosiDouble
mValue : double
...
SetValue(D : double)
GetValue( ) : double
...

NETWORKS

PLAT

COSI-BAS

- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
- .Capture network components
- .Automatic synthesis

File Data Structure Algorithm

IdGraph

AdjList

...

AddVertex(V : int)
AddEdge(U : int , V : int)
InV(V : int) : bool
InE(U : int , V : int) : bool

...

P_B_Network

mP : map< int , Point >
mB : map< pair< int, int > CosiDouble >

GetP(V : int) : Point
GetB(U : int , V : int) : CosiDouble
operator +(G : P_B_Network&) : P_B_Network

...

Variable

mAny : bool

...

IsAny( ) : bool

...

Point

mX : CosiDouble
mY : CosiDouble
mZ : CosiDouble

...

GetX( ) : Point

...

CosiDouble

mValue : double

...

SetValue(D : double)
GetValue( ) : double

...

NETWORKS

Comm.
Constrains
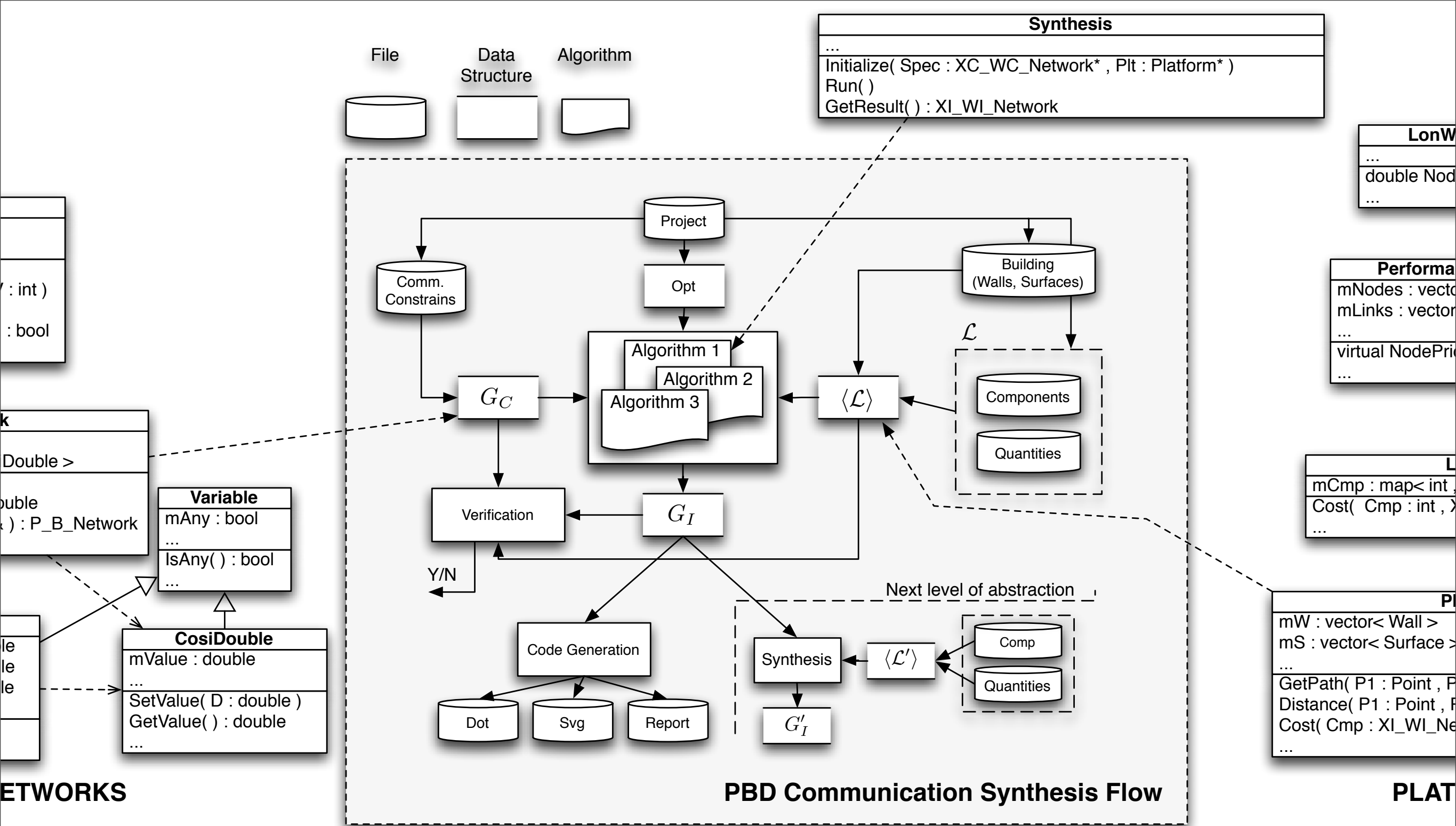
G_C

A

Verification

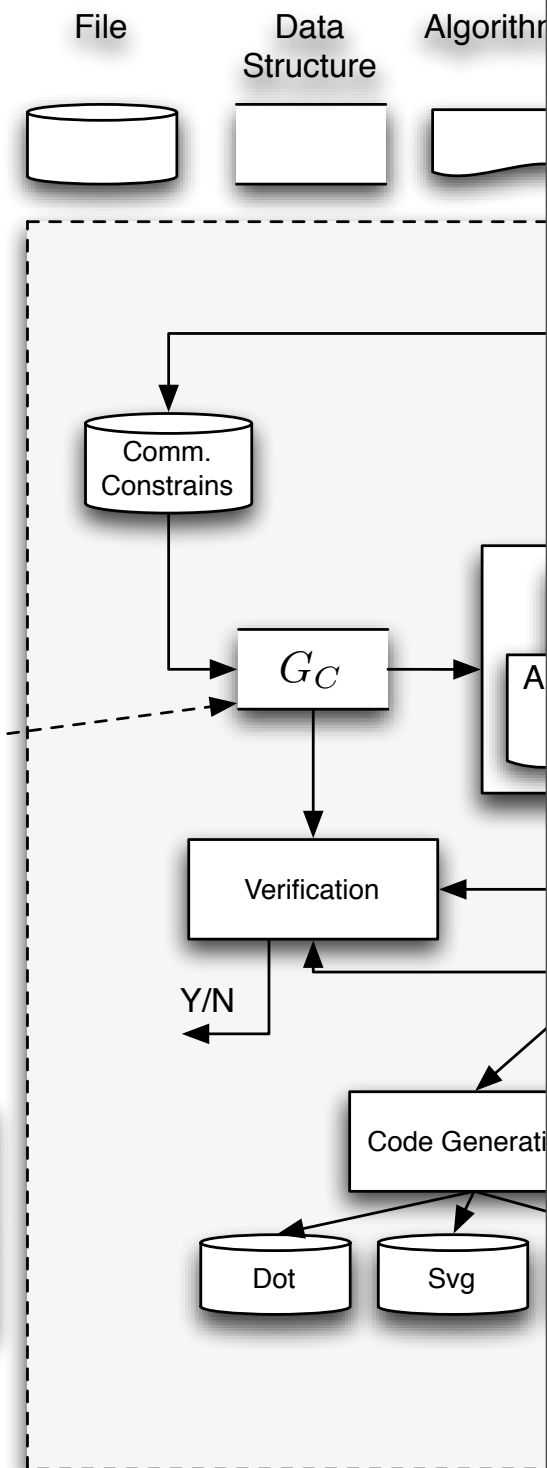Y/N

Code Generati

Dot   Svg

COSI-BAS

- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
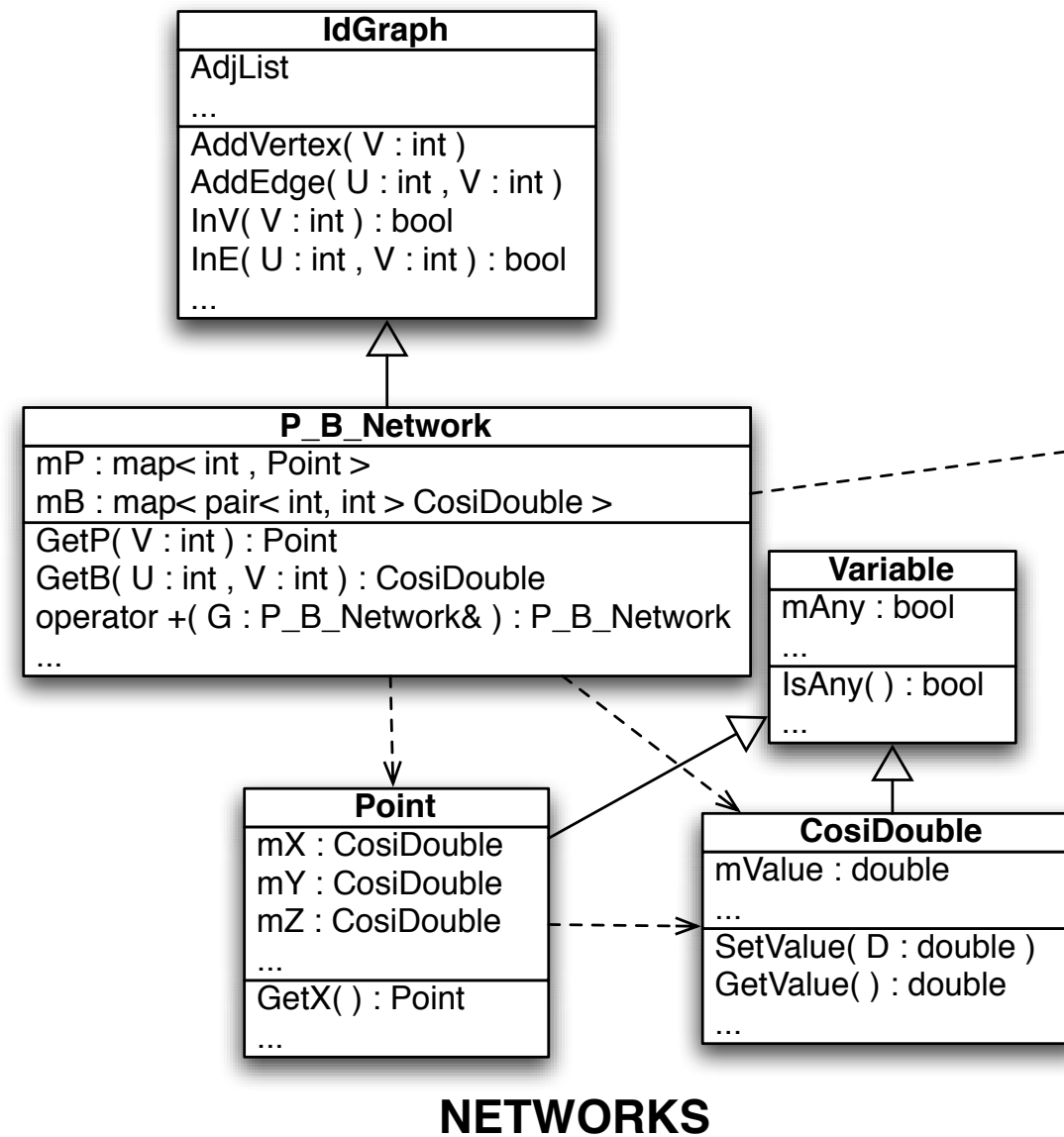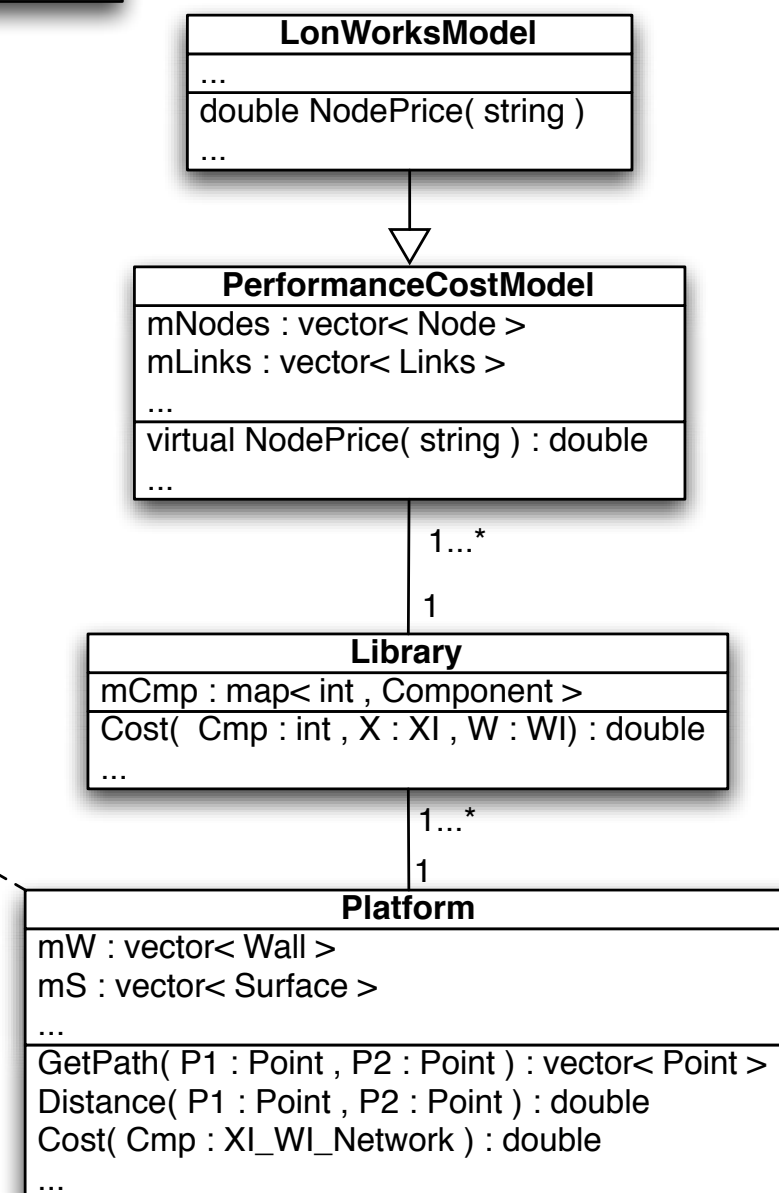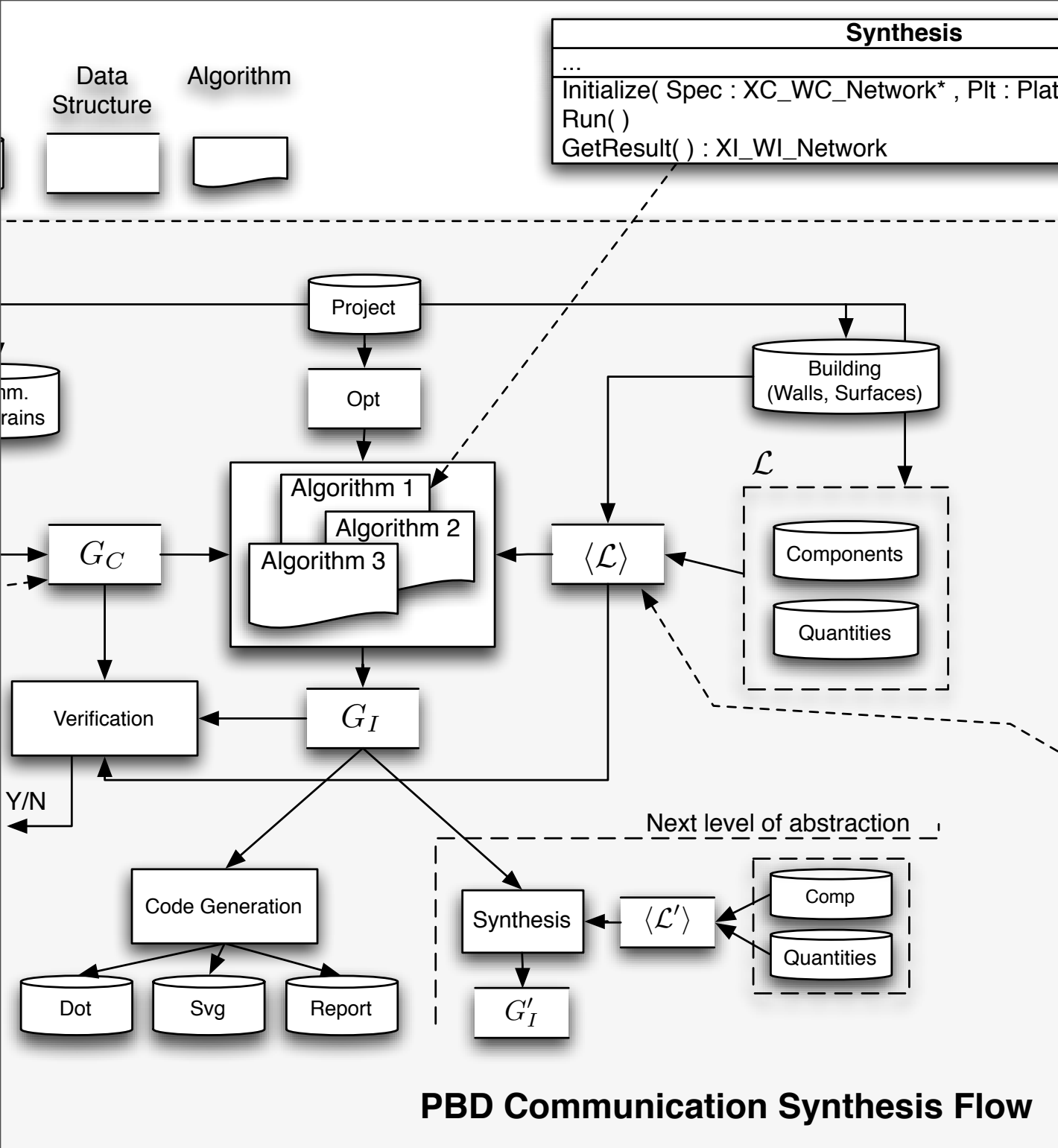- .Capture network components
- .Automatic synthesis

Data Structure

Algorithm

Synthesis

...

Initialize(Spec : XC_WC_Network* , Plt : Platform*)
Run( )
GetResult( ) : XI_WI_Network

LonWorksModel

...

double NodePrice(string)

...

PerformanceCostModel

mNodes : vector< Node >
mLinks : vector< Links >
...

virtual NodePrice(string) : double
...

1...*

1

Library

mCmp : map< int , Component >

Cost(Cmp : int , X : XI , W : WI) : double
...

1...*

1

Platform

mW : vector< Wall >
mS : vector< Surface >
...

GetPath(P1 : Point , P2 : Point) : vector< Point >
Distance(P1 : Point , P2 : Point) : double
Cost(Cmp : XI_WI_Network) : double
...

Project

Opt

Building
(Walls, Surfaces)

$\mathcal{L}$

Algorithm 1

Algorithm 2

Algorithm 3

G_C

$\langle \mathcal{L} \rangle$

Components

Quantities

Verification

G_I

Y/N

Next level of abstraction

Code Generation

Synthesis

$\langle \mathcal{L}' \rangle$

Comp

Quantities

Dot

Svg

Report

G_I'

PBD Communication Synthesis Flow

PLATFORMS

COSI-BAS

- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
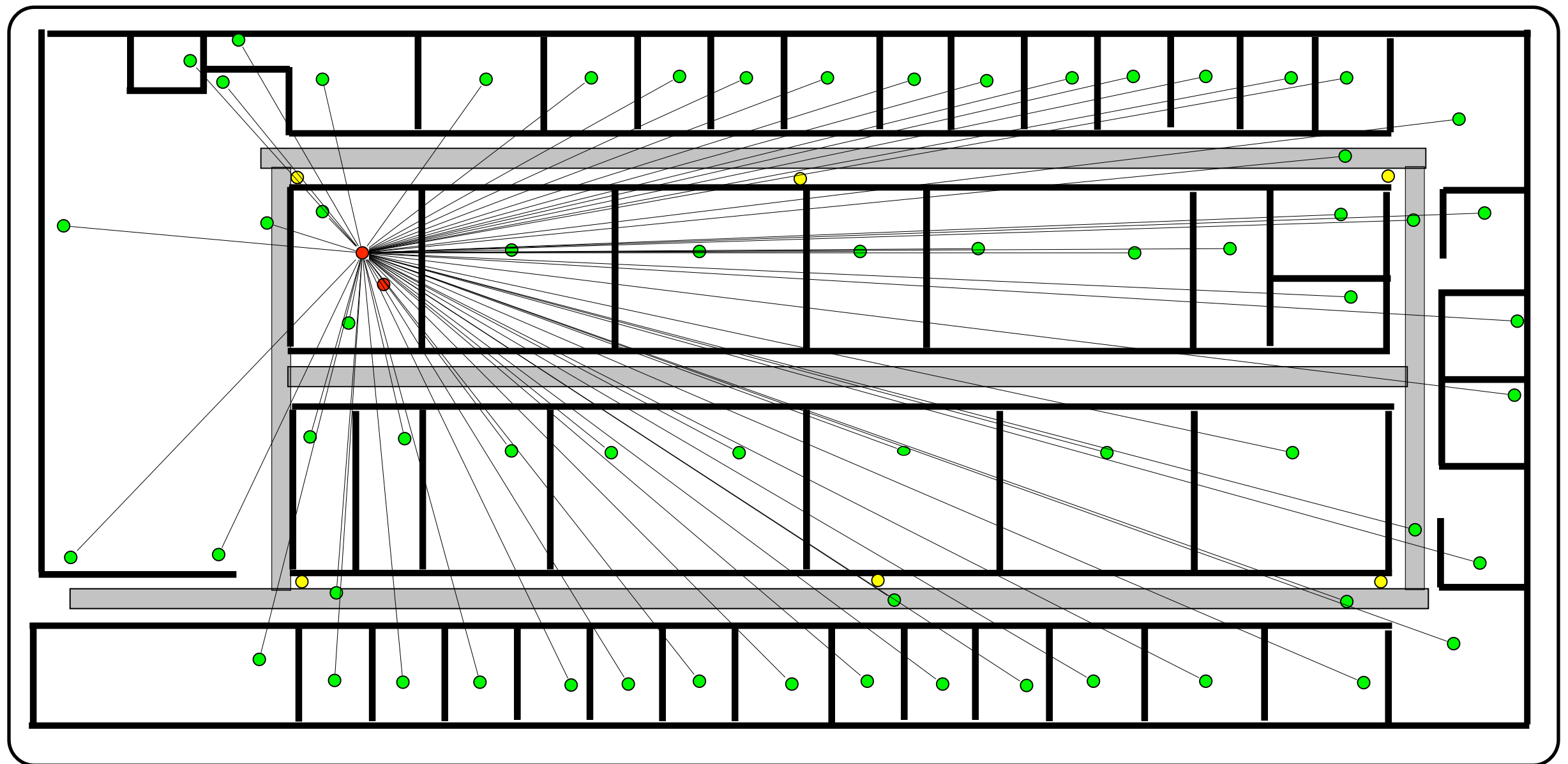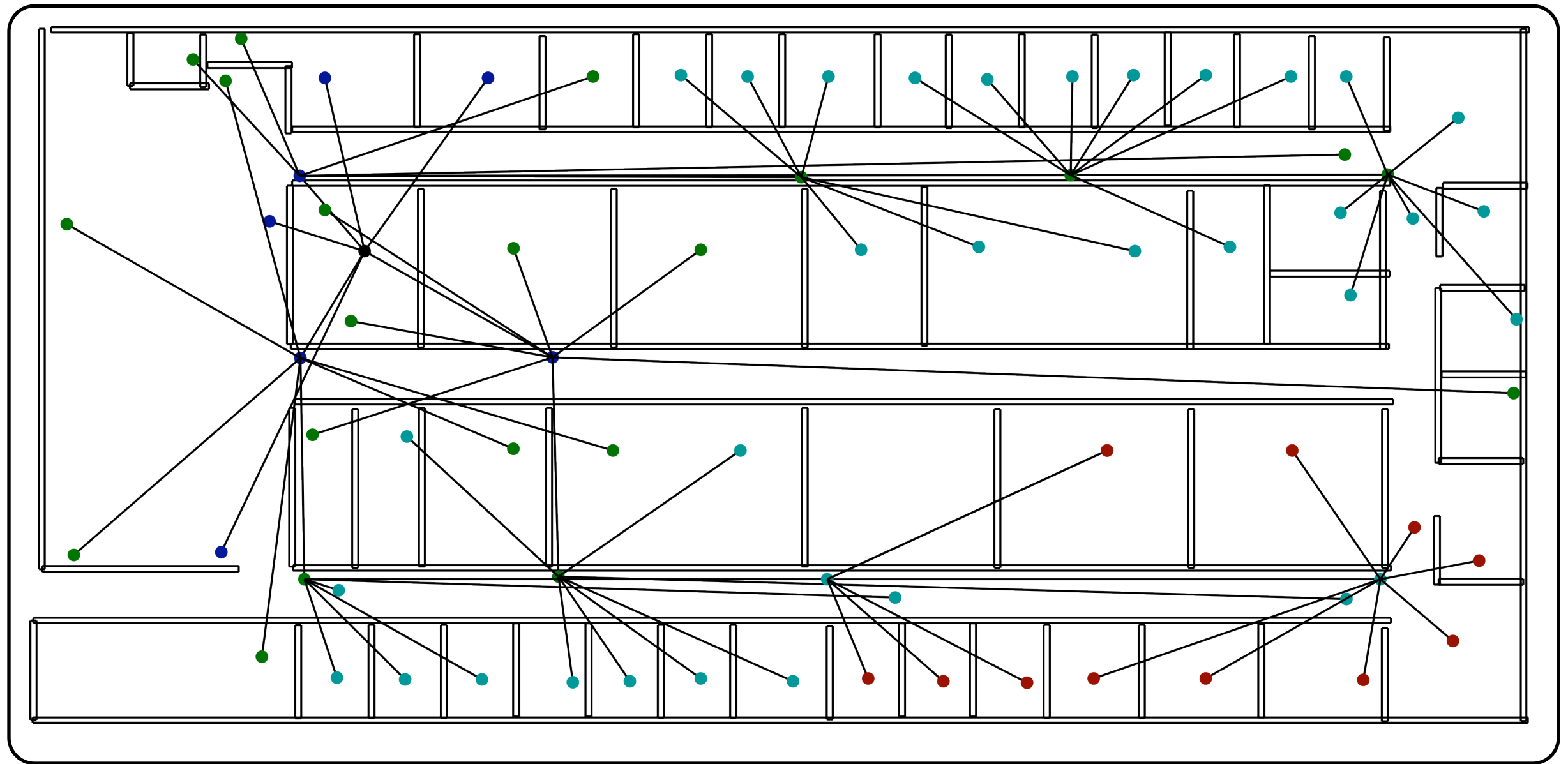- .Capture network components
- .Automatic synthesis

Examples

Examples

Examples

Results

- L-Buildings: 70 x 30 m^2, 64 nodes, period=0.1s, b=16 bits

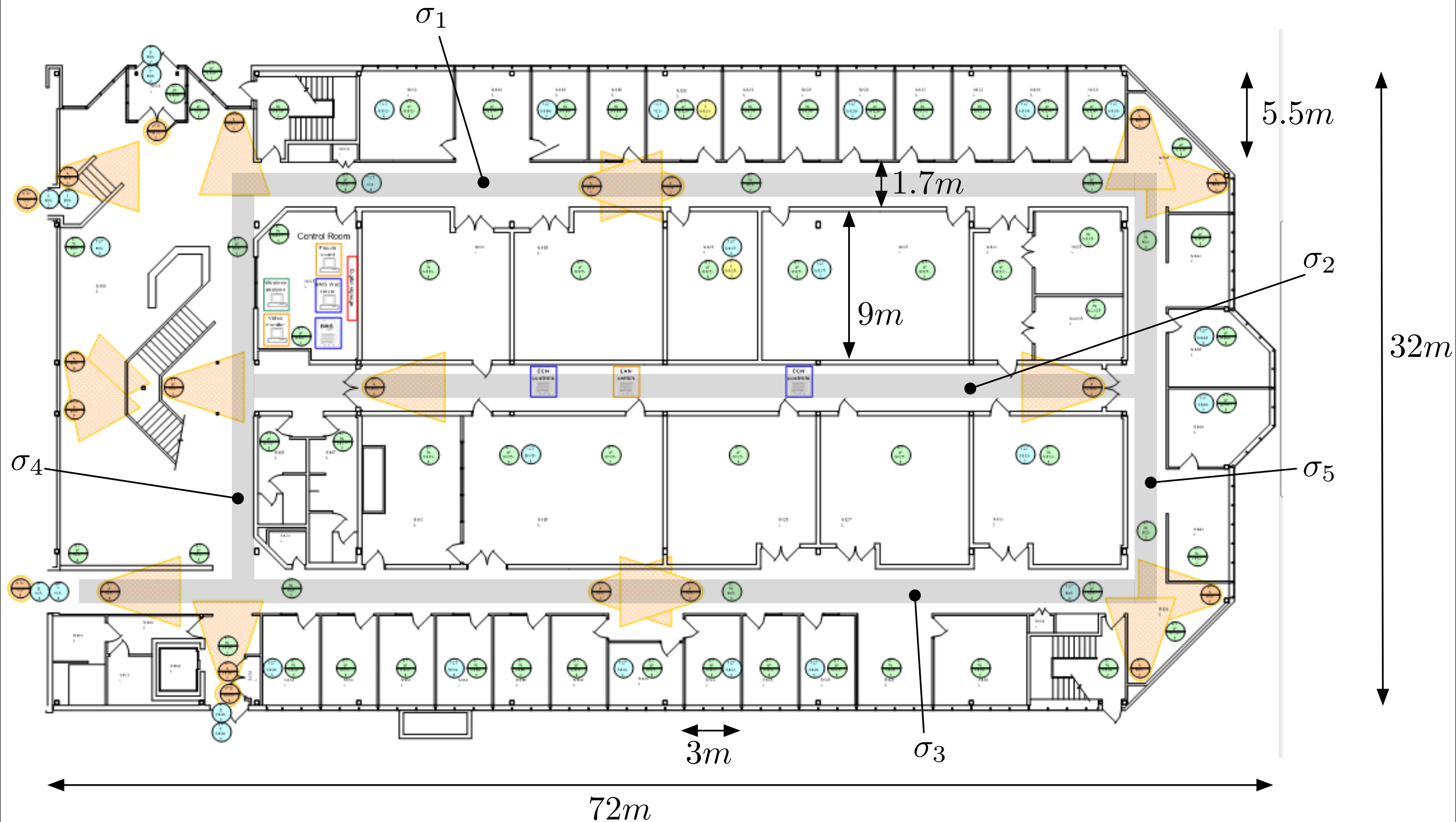- Big-box office: 60 x 56 m^2, 64 nodes, period=0.1s, b=16 bits

New
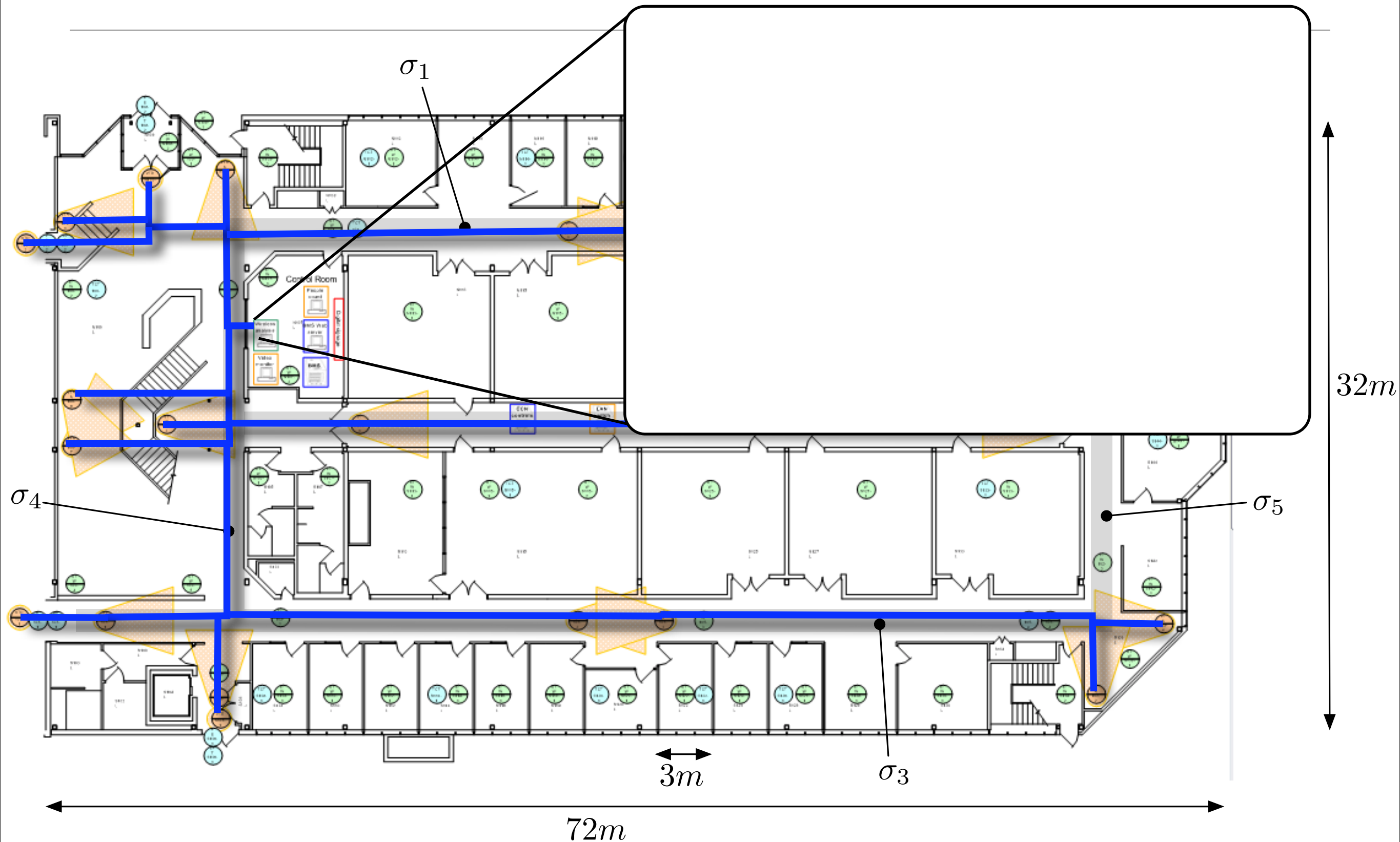
Building	Bw (Kb/s)	Max Length (m)	Max #devices	Max delay (ms)	Max Utilization (%)	Router ($)	Nodes ($)	Wires ($)	Total ($)
L-Building	78	1000	32	91	89%	3700	10240	5020	18960
	250	400	20	22	20%	5180	10240	4939	20359
Big-Box Office	78	1000	32	91	89	2220	10240	4317	16777
	250	400	20	19	20%	4440	10240	4131	18811

Retrofit

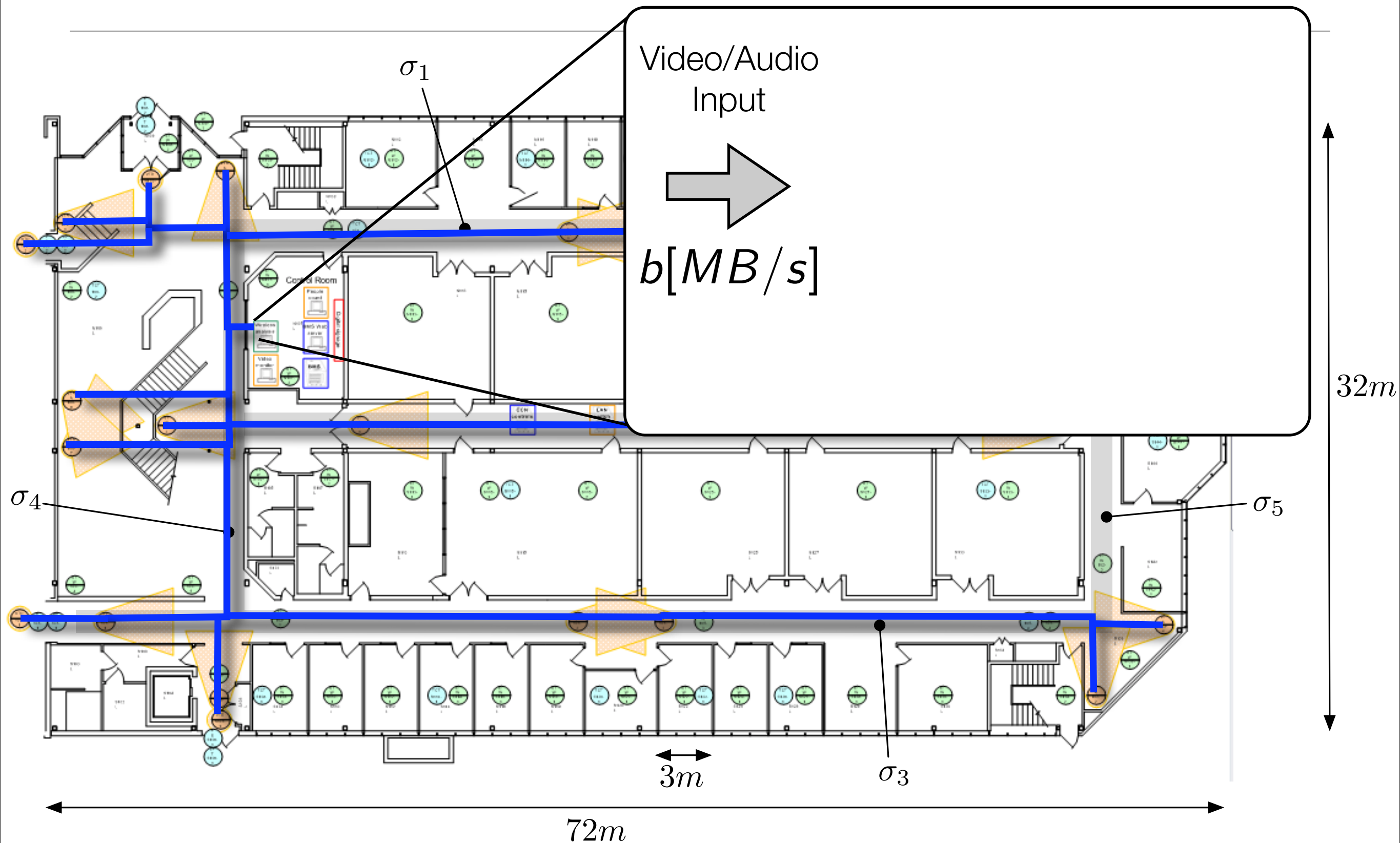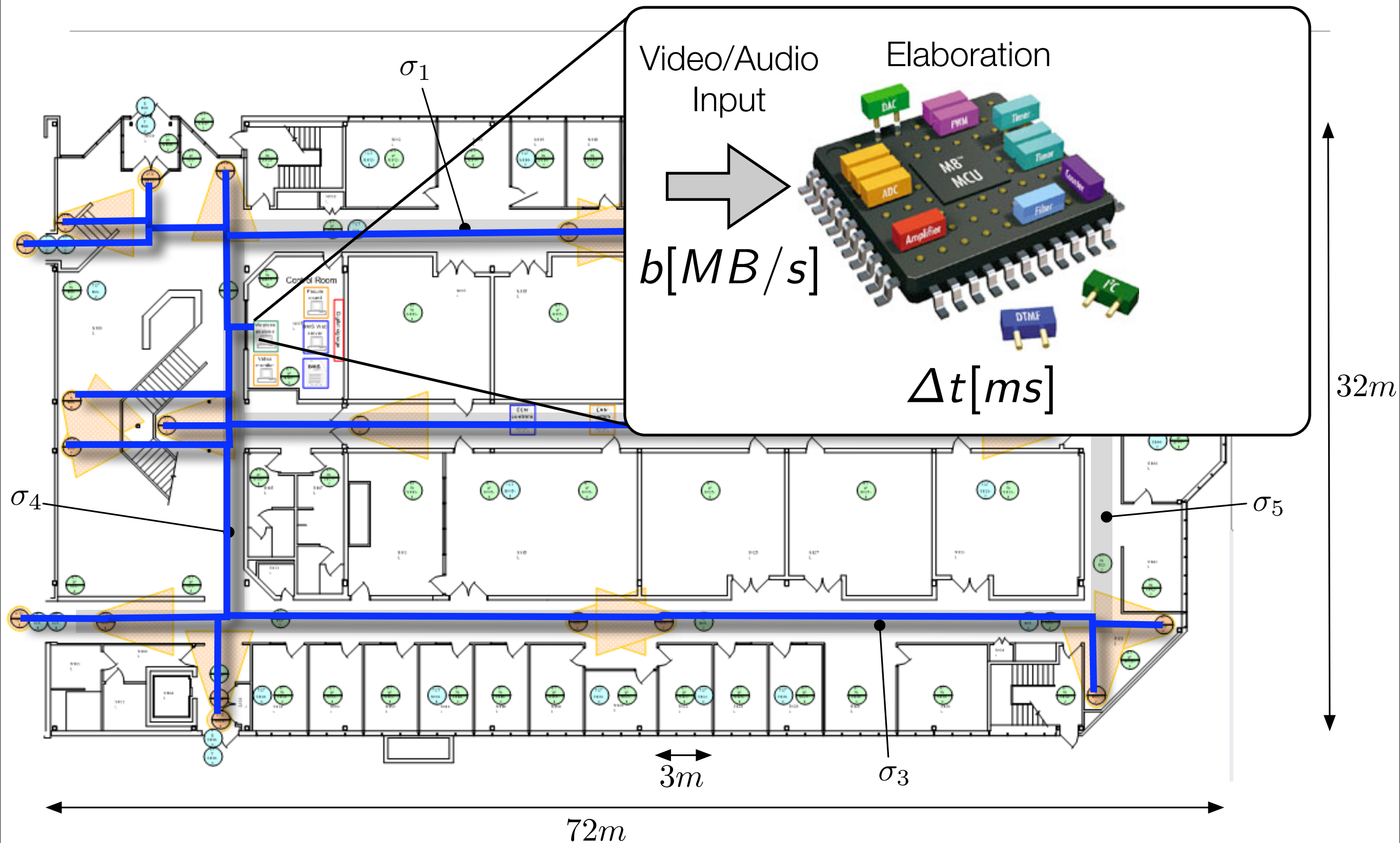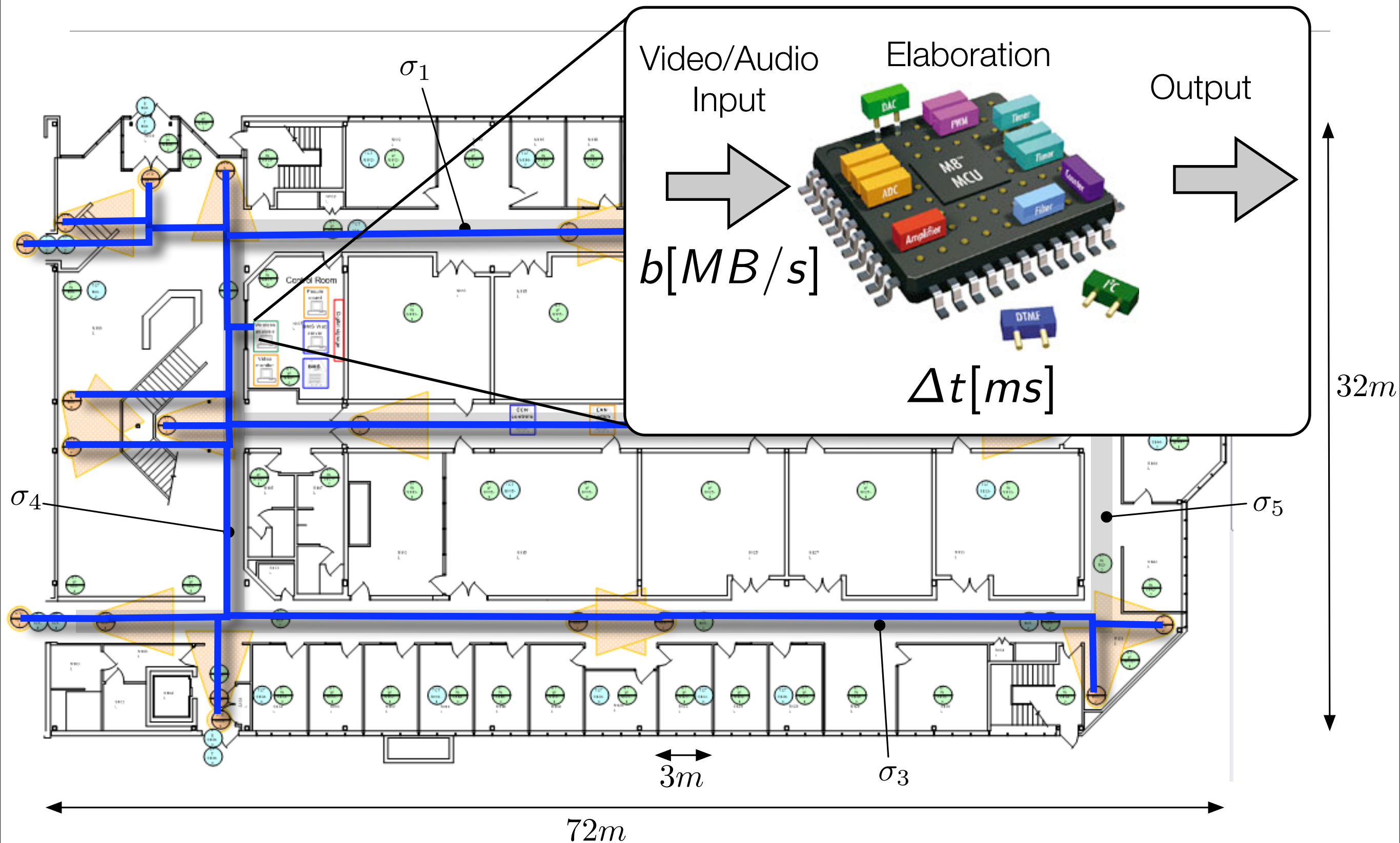| Building	Bw (Kb/s)	Max Length (m)	Max #devices	Max delay (ms)	Max Utilization (%)	Router ($)	Nodes ($)	Wires ($)	Total ($)
L-Building	78	1000	32	91	89%	5920	10240	12680	28844
	250	400	20	22	20%	5180	10240	13744	29198
Big-Box Office	78	1000	32	91	89	3700	10240	12044	25984
	250	400	20	19	20%	4440	10240	11855	26535

Keep Guaranteeing Properties: Constraint Propagation

Keep Guaranteeing Properties: Constraint Propagation

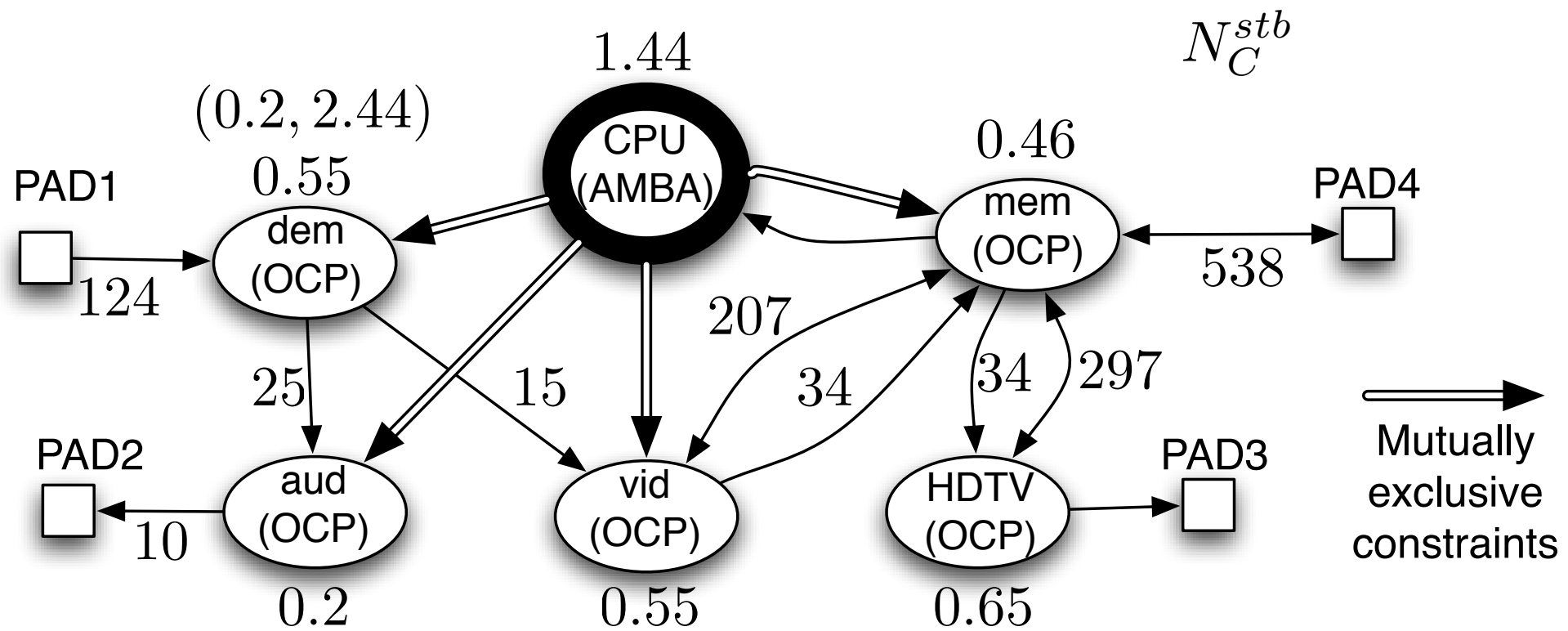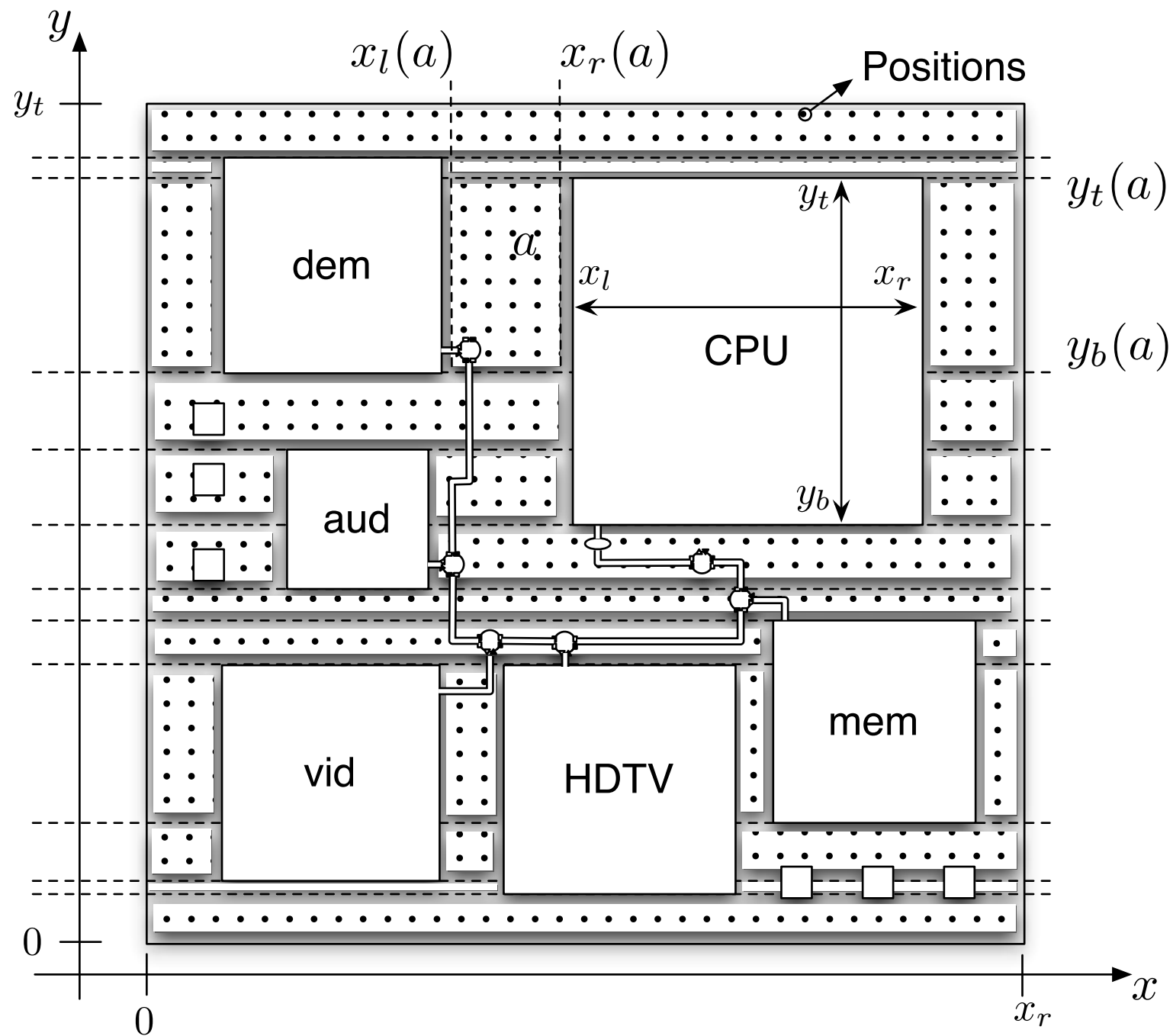# Keep Guaranteeing Properties: Constraint Propagation

Keep Guaranteeing Properties: Constraint Propagation

Video/Audio Input

$b[MB/s]$

σ_1

σ_4

σ_5

σ_3

$32m$

$72m$

$3m$

Keep Guaranteeing Properties: Constraint Propagation

σ_1

Video/Audio Input

Elaboration

$b[MB/s]$

$\Delta t[ms]$

σ_4

σ_5

σ_3

$32m$

$3m$

$72m$

Keep Guaranteeing Properties: Constraint Propagation

σ_1

Video/Audio Input

Elaboration

Output

$b[MB/s]$

$\Delta t[ms]$

$32m$

σ_4

σ_5

$3m$

σ_3

$72m$

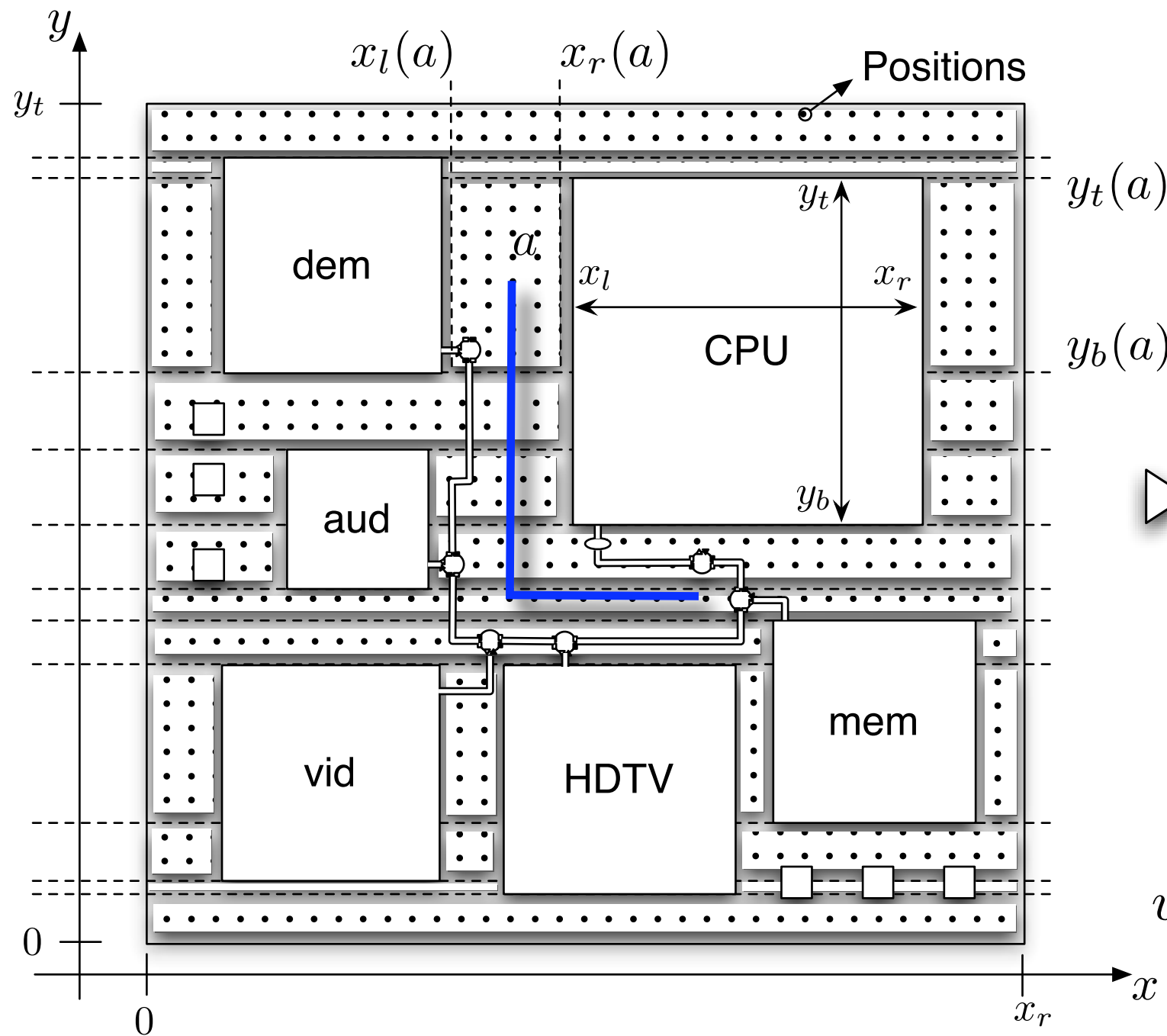# Communication Synthesis for On-Chip Communication
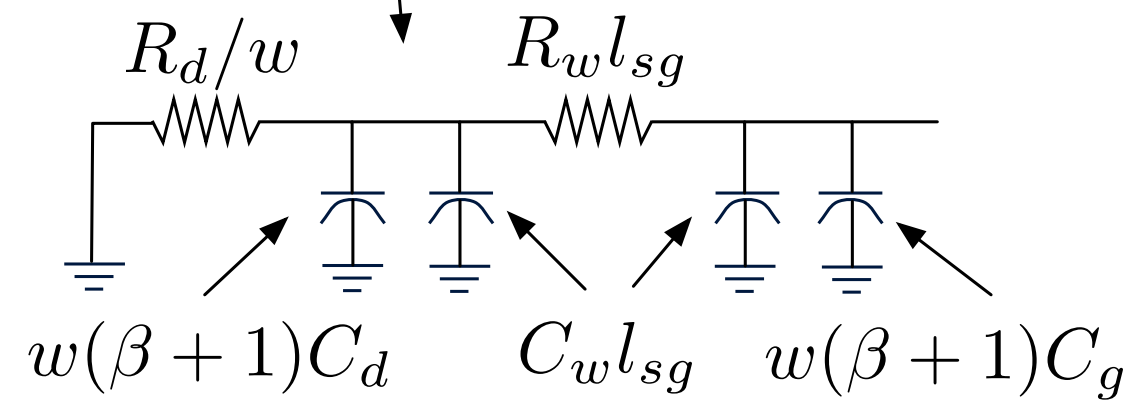
Physical Aspects
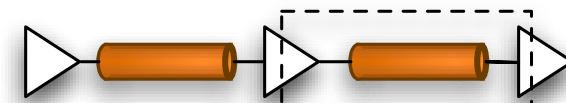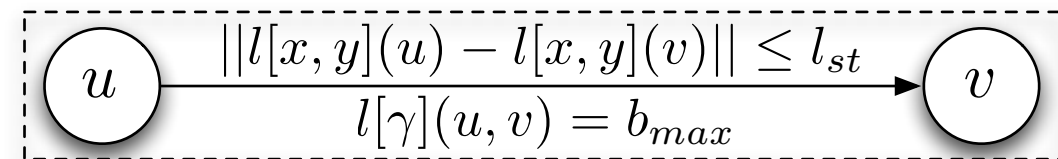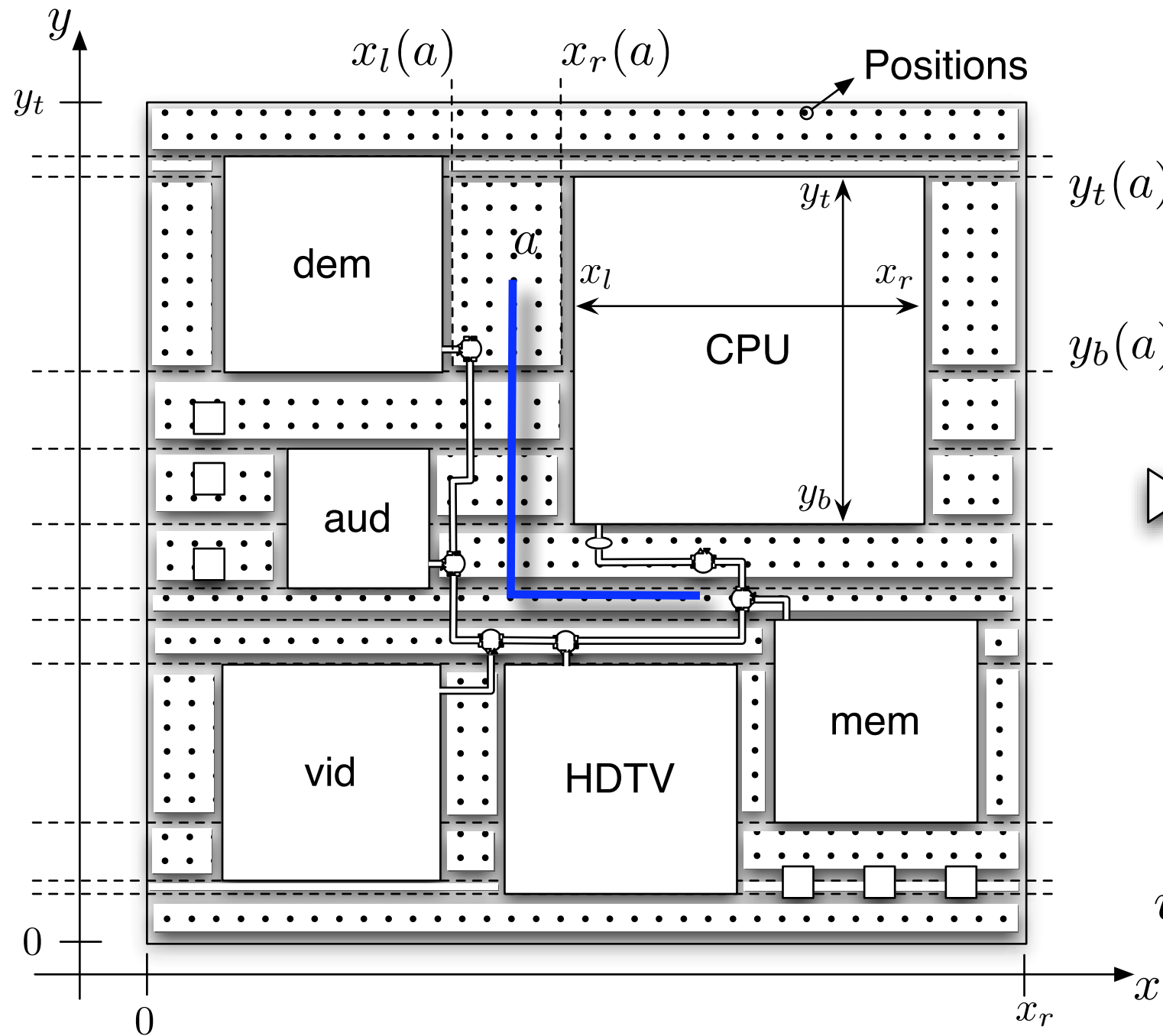
Physical Aspects

Physical Aspects

Physical Aspects

Physical Aspects
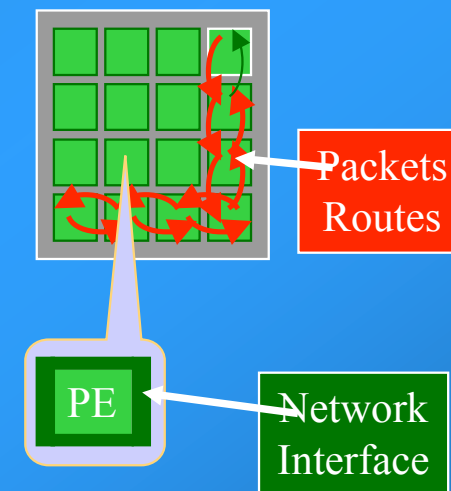
Controversial Views

- Networks-On-Chip vs. Bus-based Communication

Controversial Views

- Networks-On-Chip vs. Bus-based Communication

Why on-chip networking ?

- Provide a structured methodology for realizing on-chip communication
  - Modularity
  - Flexibility
- Cope with inherent limitations of busses
  - Performance and power of busses do not scale up
- Support reliable operation
  - Layered approach to error detection and correction

Packets Routes

PE

Network Interface

De Micheli

39

Giovanni De Micheli [ISCAS2006]

Controversial Views

- Networks-On-Chip vs. Bus-based Communication

Summary

Point to point busses are not necessary for multi-core chip

Rings and meshes were devised for point to point busses over long distances—overkill for on chip network?

Router power could be prohibitive

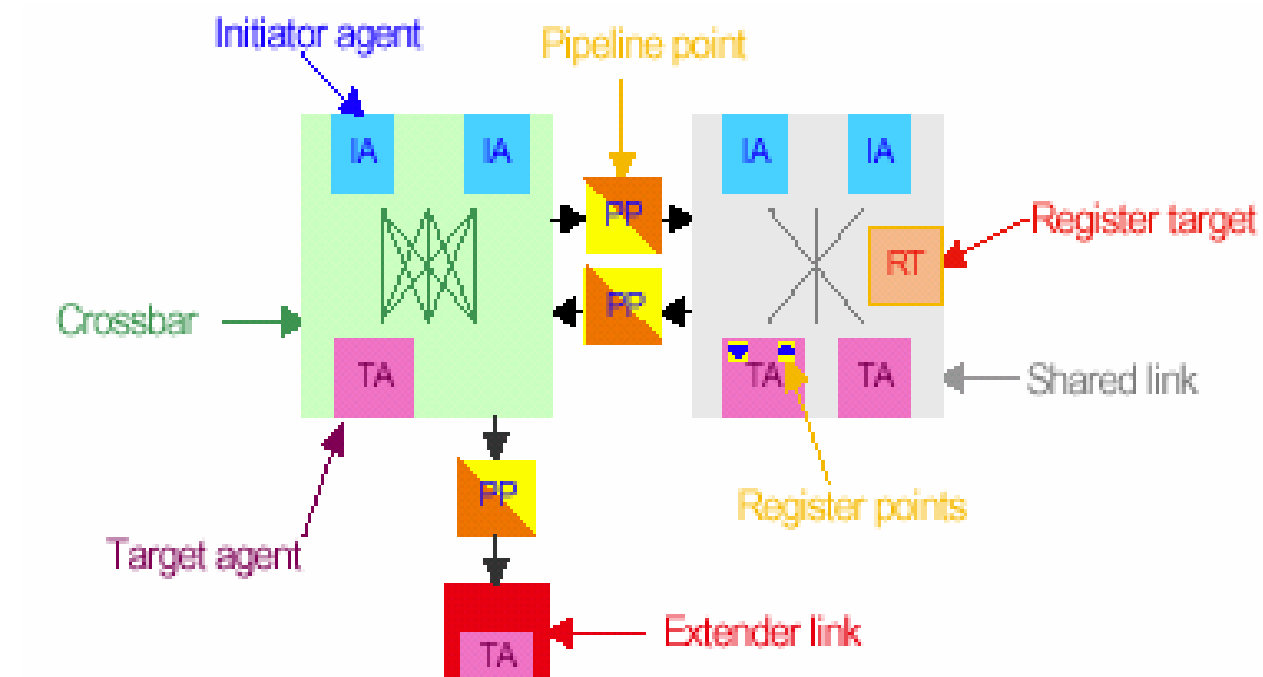Wide bus or busses, may be adequate

- Simple to implement
- Simpler coherency
- Lower power
- Maybe lower latency

Go slower, wider, and simpler

13

Shekhar Borkar [OCIN06]

Controversial Views

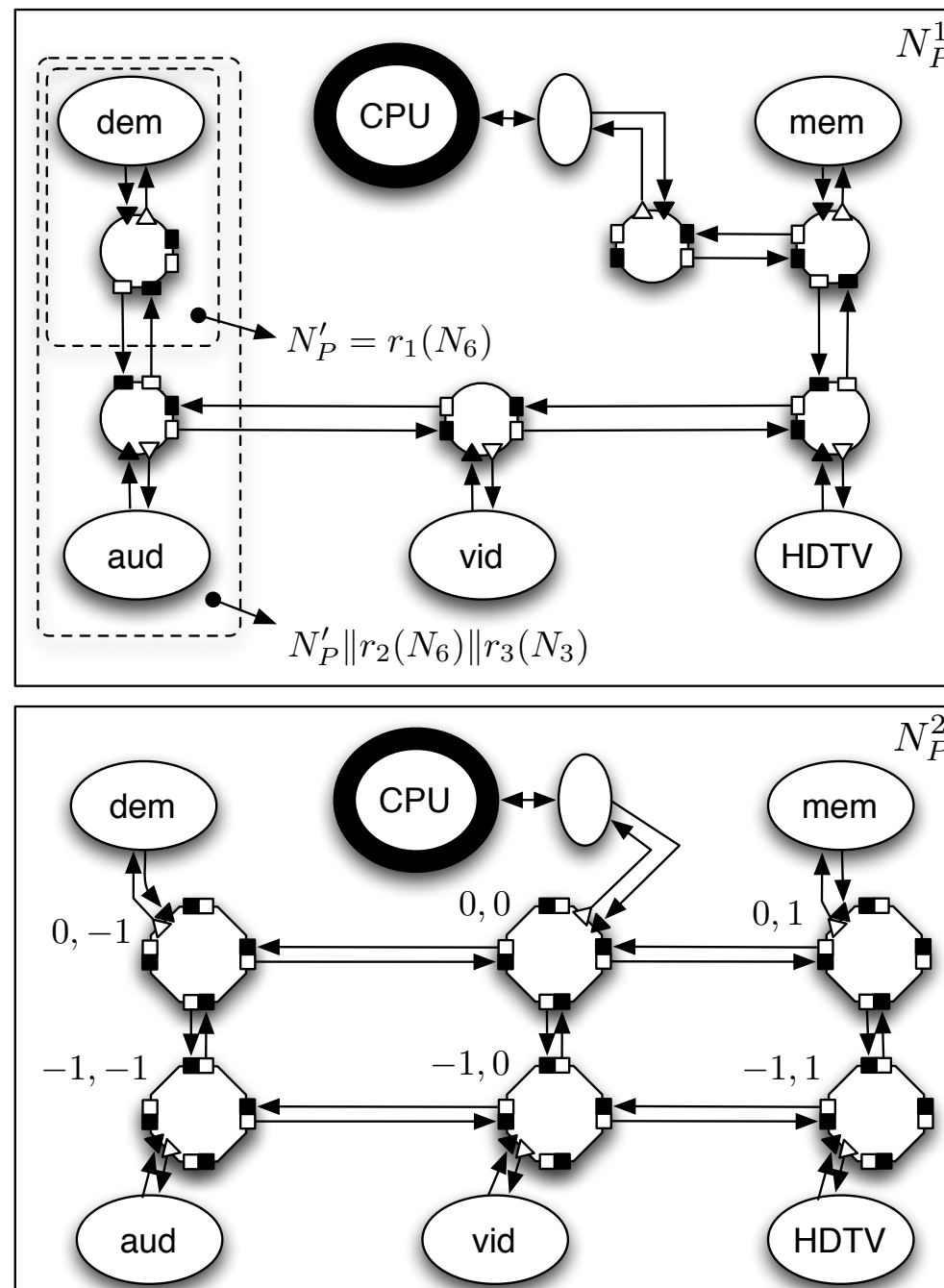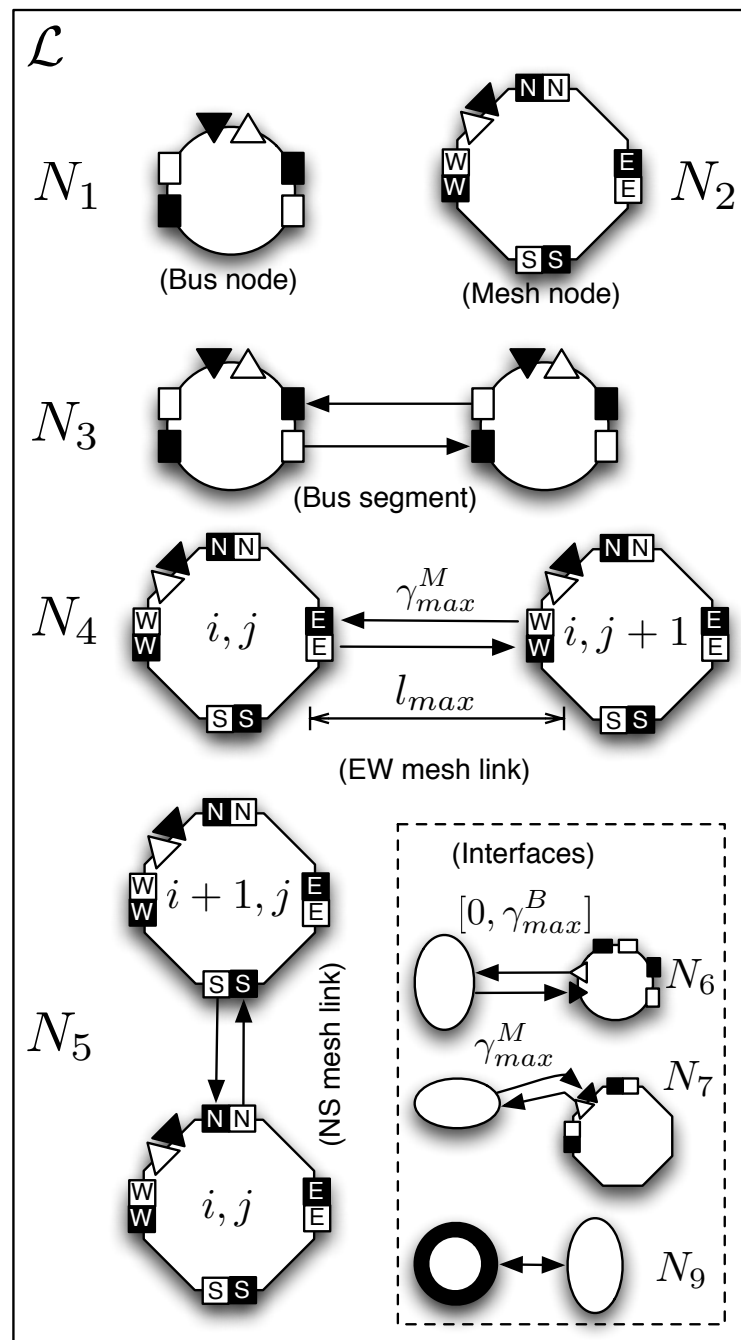- Networks-On-Chip vs. Bus-based Communication

SonicsMX

Controversial Views

- Networks-On-Chip vs. Bus-based Communication

Our Approach

- Define the communication library, define composition rules, find the best communication implementation

Rule 1: Number of bus nodes at most number of bus segment minus one
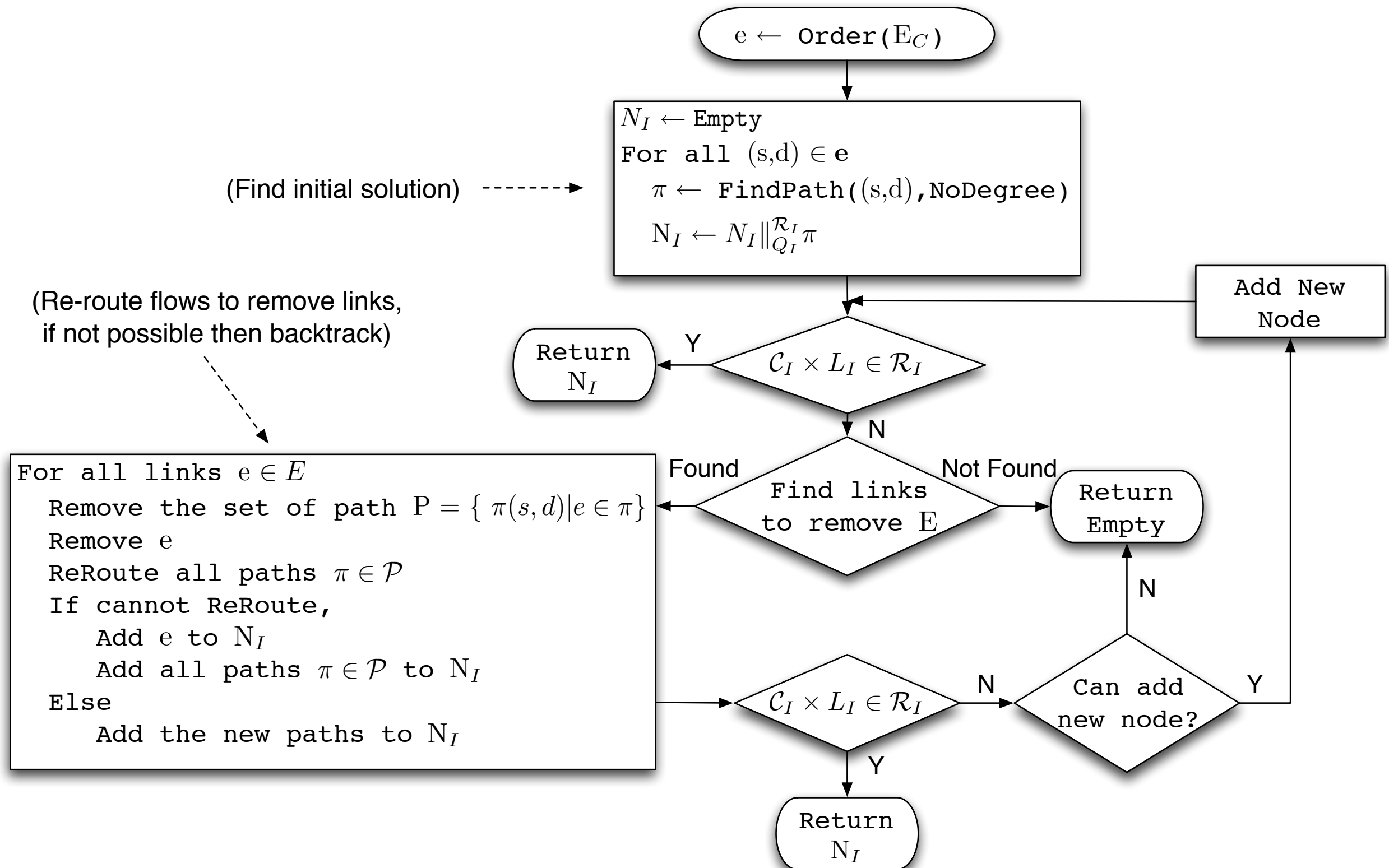
Rule 2: Constraint on the total bus capacity

Rule 3: Mesh segment only between (i,j) and (i,j+1) or (i,j) and (i+1,j)
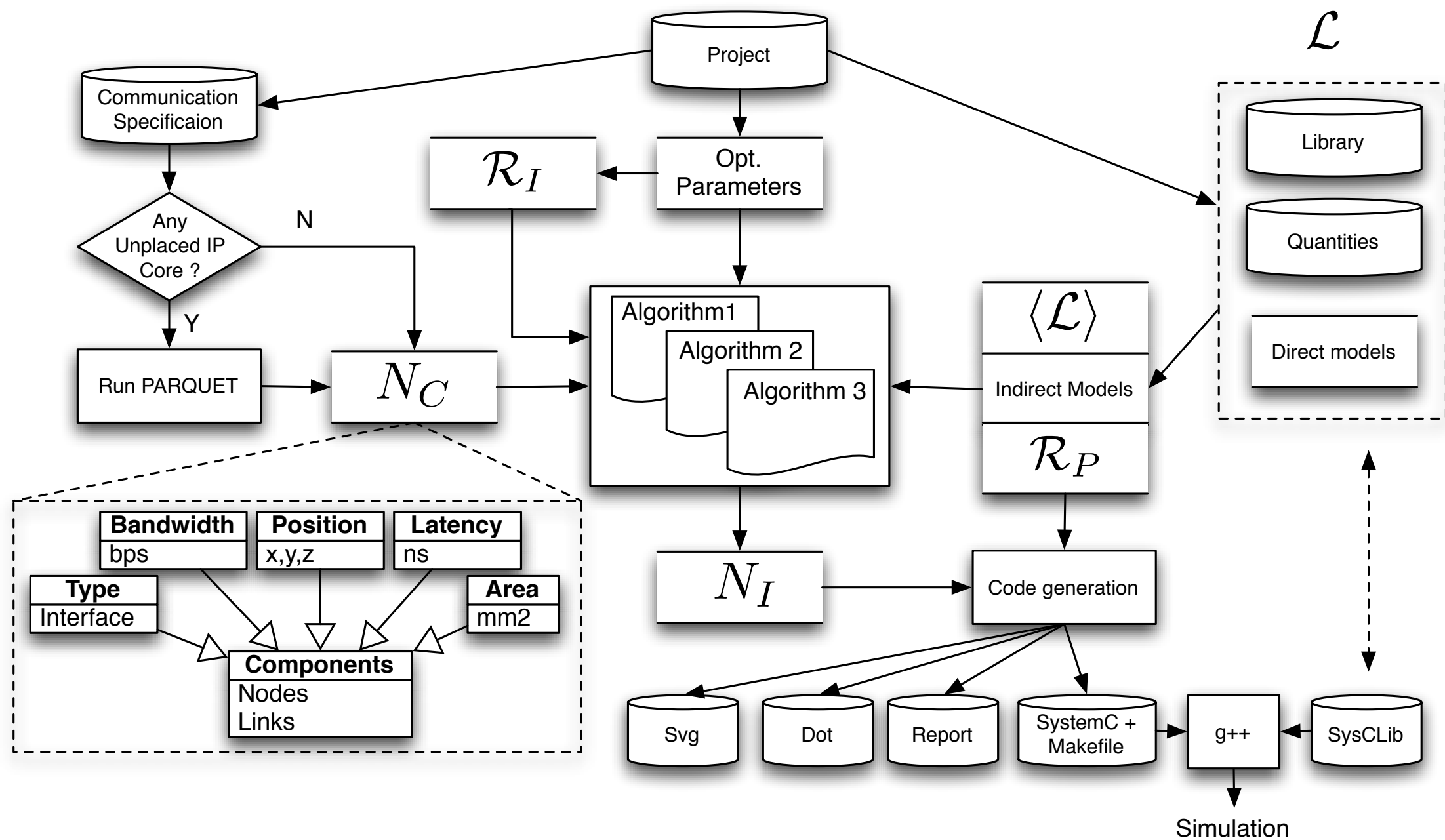
Hardness Results for Network Design

- Consider the bicriteria network design problem $(\mathbf{A}, \mathbf{B}, \mathcal{S})$

    - $\mathbf{A}$ is a budget requirement and $\mathbf{B}$ a minimization objective

    - $\mathcal{S}$ is a membership requirement in a class of subgraphs

- Consider $\mathbf{A}$ a fixed node degree, and $\mathbf{B}$ cost

- Unless P = NP, there is not polynomial time $(1, \rho)$-approximation algorithm

- Unless P = NP, there is not polynomial time $(\rho, 1)$-approximation algorithm

- Unless P = NP, there is not polynomial time $(2 - \epsilon, \rho)$-approximation algorithm

- Unless P = NP, there is not polynomial time $(\rho, \tau - \epsilon)$-approximation algorithm

- $\rho > 1, \ \epsilon > 0$

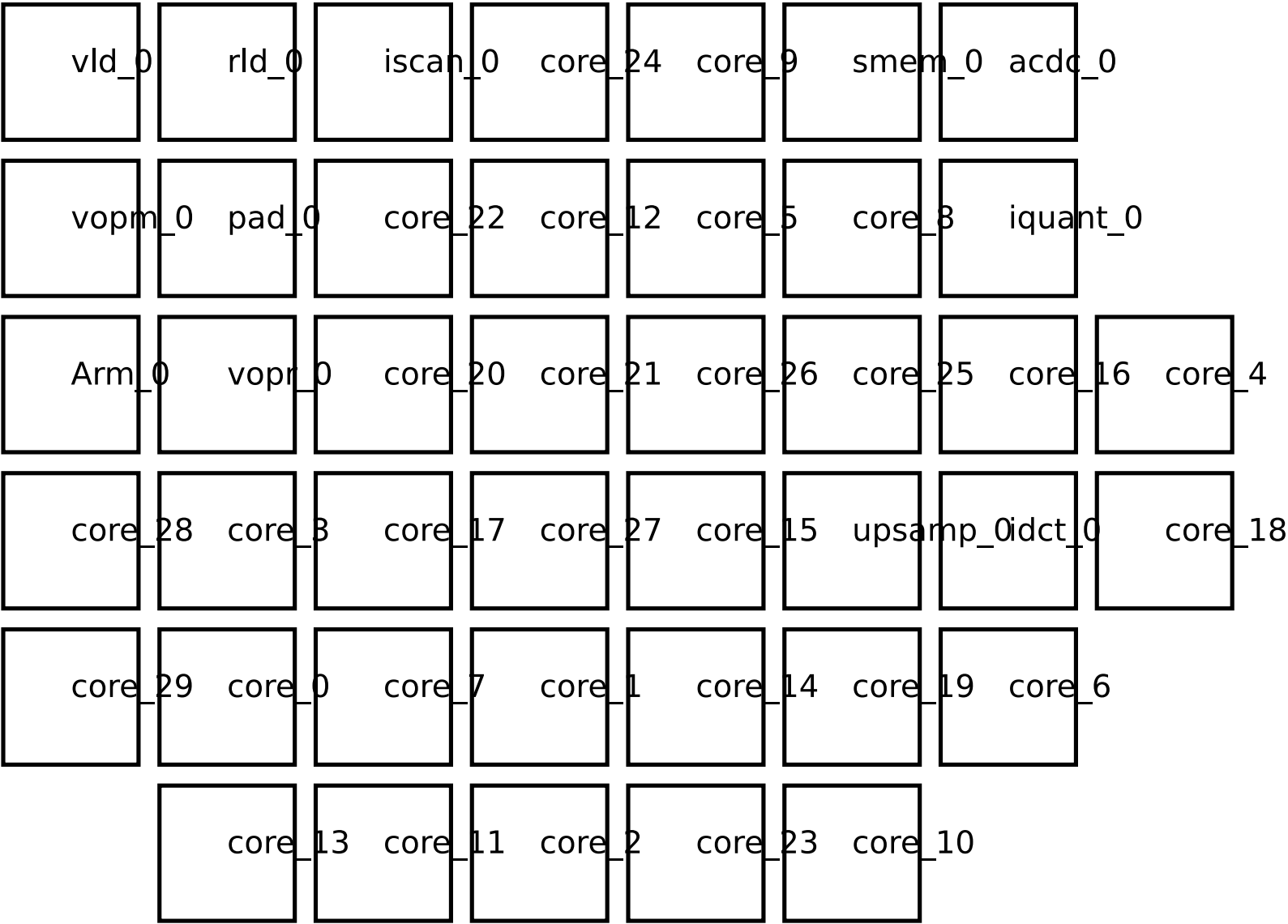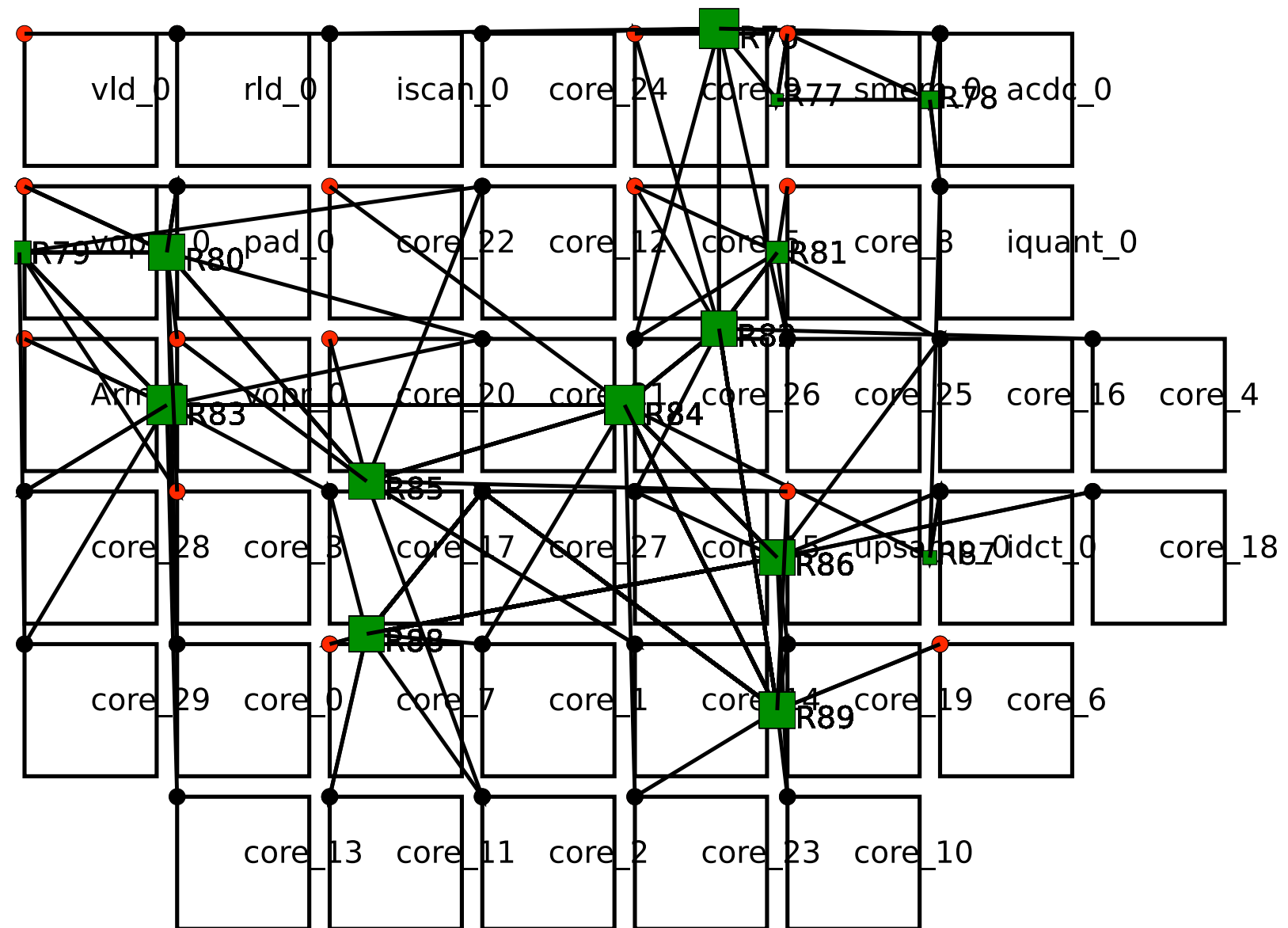- τ lower bound on the performance guarantee of Steiner Tree

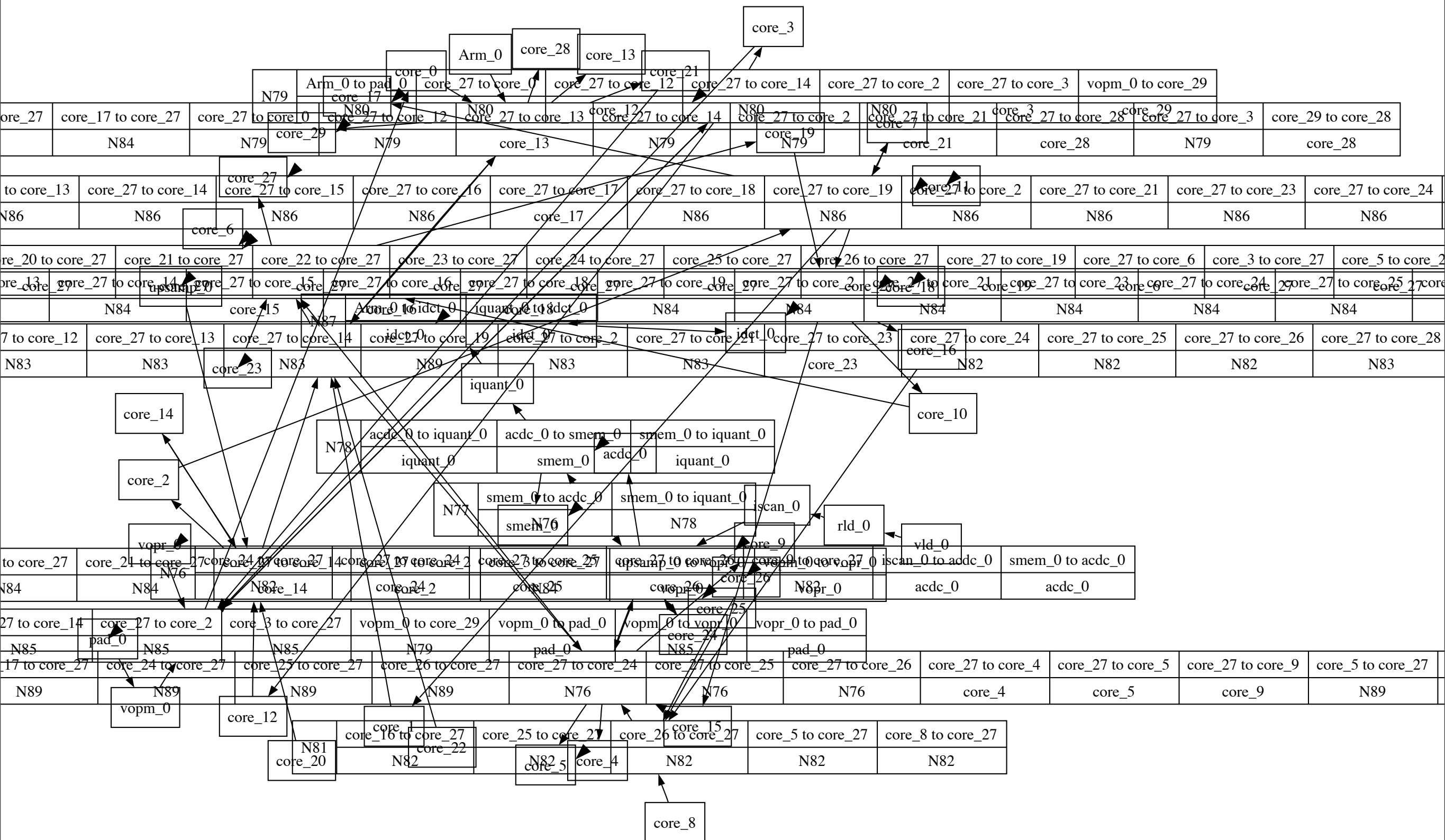R. Ravi et. al. "Approximation Algorithms for the Degree-Constrained Minimum-Cost Network-Design Problem", Algorithmica 2001
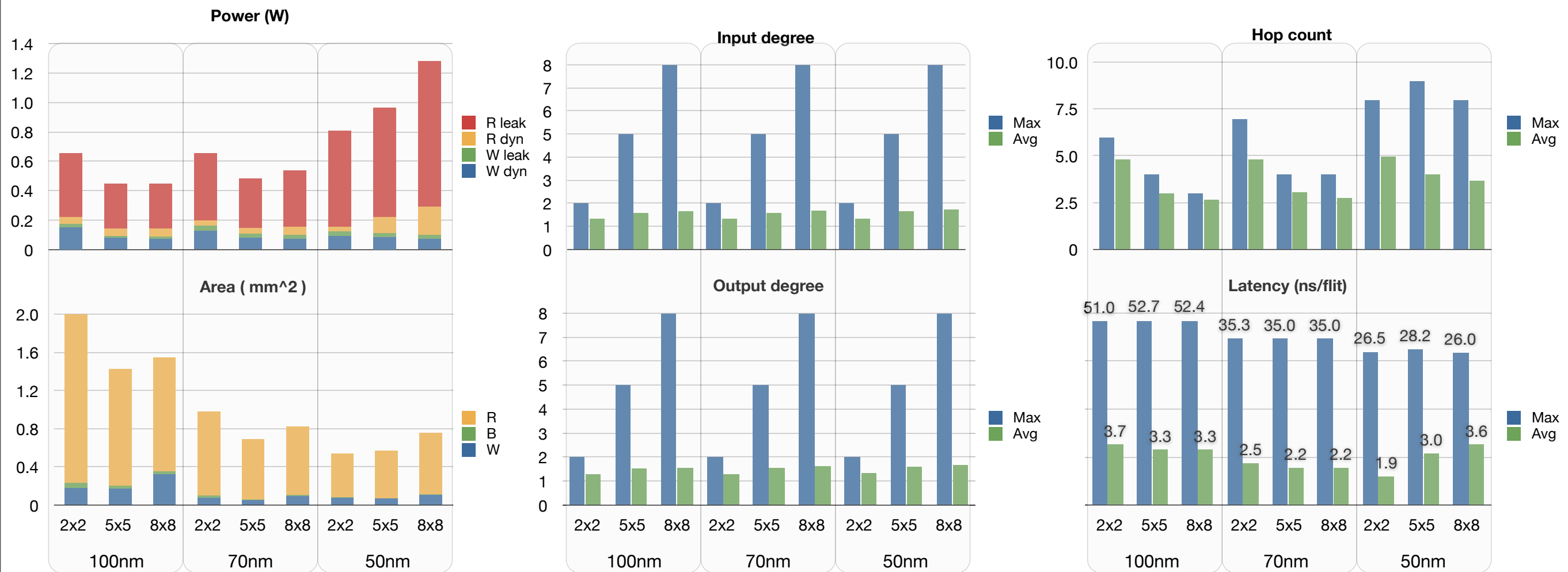
Heuristic Algorithm

COSI-OCC

COSI-OCC

COSI-OCC

COSI-OCC

Results

Results

Results

CONCLUSIONS RESEARCH DIRECTIONS

$o_s = F_s(i_s)$ F_c $o_r = F_r(i_r, o_r)$

i_s → (A) → o_s → (C) → i_r → (B) →

Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness

$o_s = F_s(i_s)$

F_c

$o_r = F_r(i_r)$

i_s → A → o_s → C → i_r → B
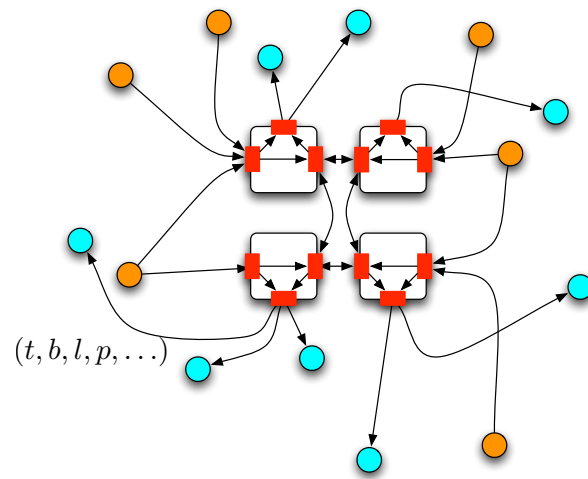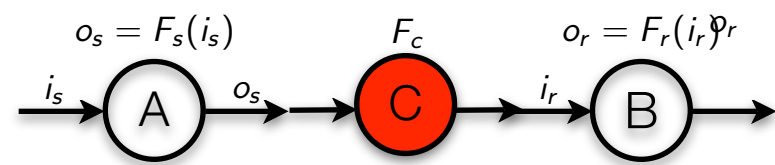
$(t, b, l, p, \ldots)$

Sensor, actuators, controllers, links...
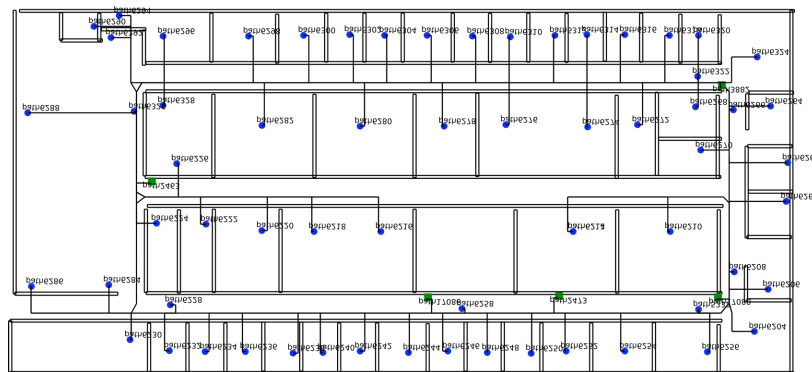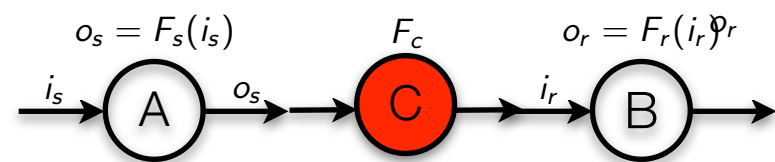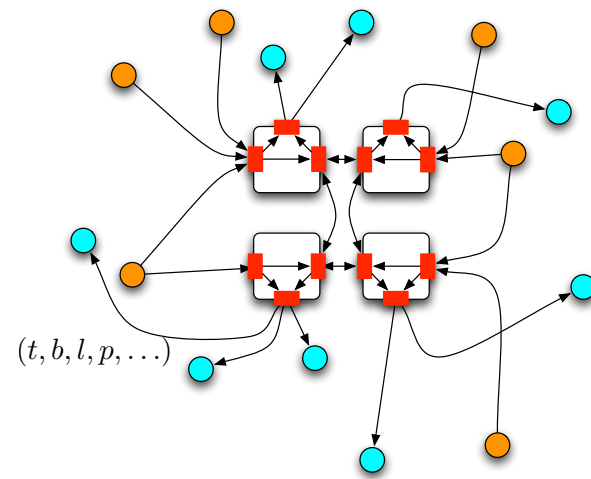
Buses, Routers, Gateways...

Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness

Compose sensors, actuators, controllers. Define accuracy and stability of the distributed control [preserve functionality, infer QoS]

Compose communication nodes and links to minimize installation and operation cost [preserve QoS, infer HW/SW performance]

$o_s = F_s(i_s)$ F_c $o_r = F_r(i_r)$

i_s A o_s C i_r B

$(t, b, l, p, \ldots)$

Sensor, actuators, controllers, links...

Buses, Routers, Gateways...

1.44 N_C^{stb}

$(0.2, 2.44)$ 0.55 CPU (AMBA) 0.46

PAD1 dem (OCP) mem (OCP) PAD4

124 538

25 15 207 34 34 297

PAD2 aud (OCP) vid (OCP) HDTV (OCP) PAD3

10 0.2 0.55 0.65

Mutually exclusive constraints

Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness

Compose sensors, actuators, controllers. Define accuracy and stability of the distributed control [preserve functionality, infer QoS]
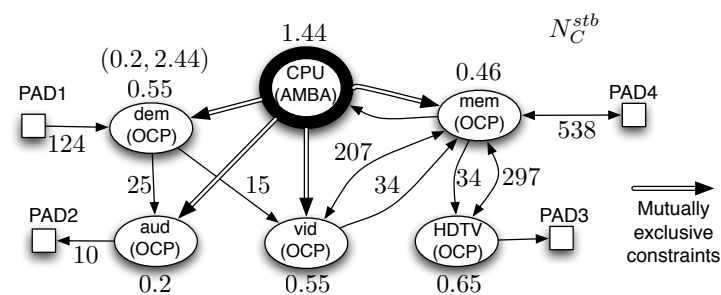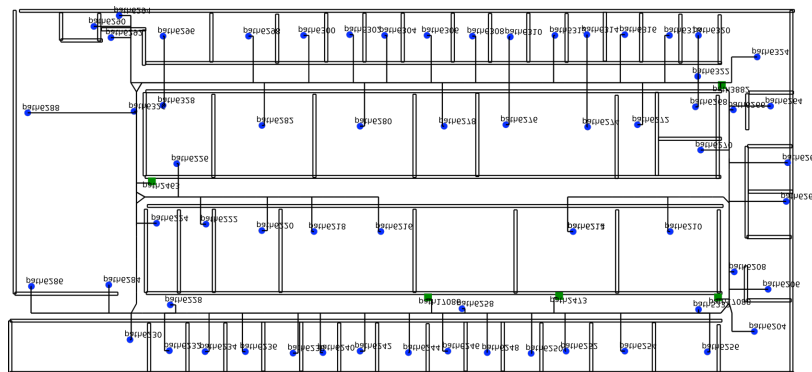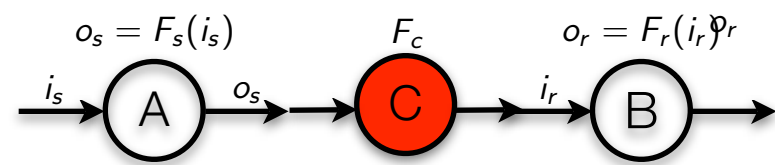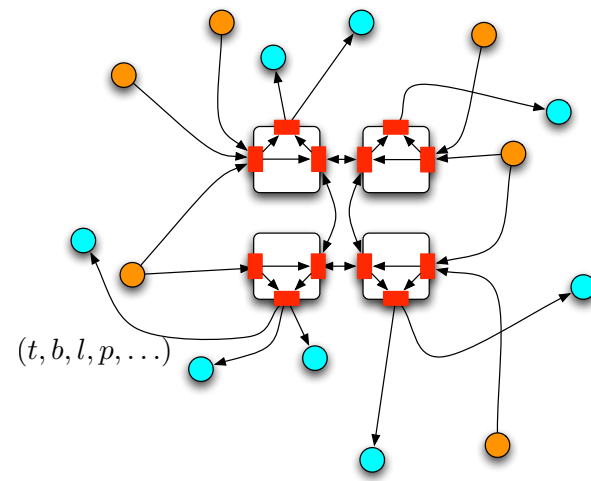
Compose communication nodes and links to minimize installation and operation cost [preserve QoS, infer HW/SW performance]
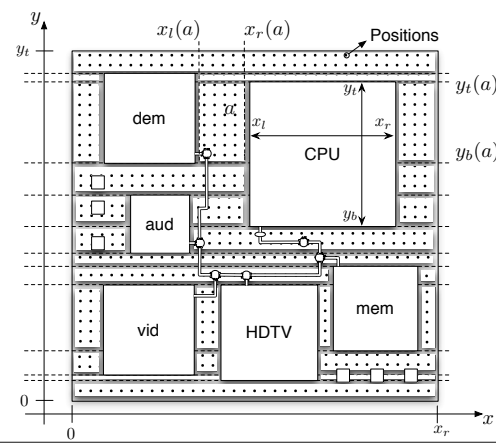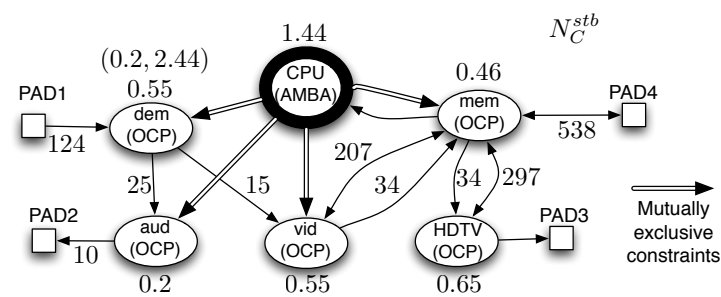
Compose platform [preserve performance, infer QoS]

$o_s = F_s(i_s)$ F_c $o_r = F_r(i_r)$

i_s A o_s C i_r B o_r

$(t, b, l, p, \ldots)$

Sensor, actuators, controllers, links...

Buses, Routers, Gateways...

N_C^{stb}

$(0.2, 2.44)$ 1.44 0.46

PAD1 0.55
dem (OCP) CPU (AMBA) mem (OCP) PAD4
124 538

25 15 207 34 34 297

PAD2 aud (OCP) vid (OCP) HDTV (OCP) PAD3
10 0.2 0.55 0.65

Mutually exclusive constraints

y $x_l(a)$ $x_r(a)$ Positions
y_t $y_t(a)$
dem μ x_l x_r
CPU $y_b(a)$
aud y_b
vid HDTV mem
0
0 x_r x

Buses, Rings, Mesh, Routers, Interfaces...

Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness

Compose sensors, actuators, controllers. Define accuracy and stability of the distributed control [preserve functionality, infer QoS]

Compose communication nodes and links to minimize installation and operation cost [preserve QoS, infer HW/SW performance]
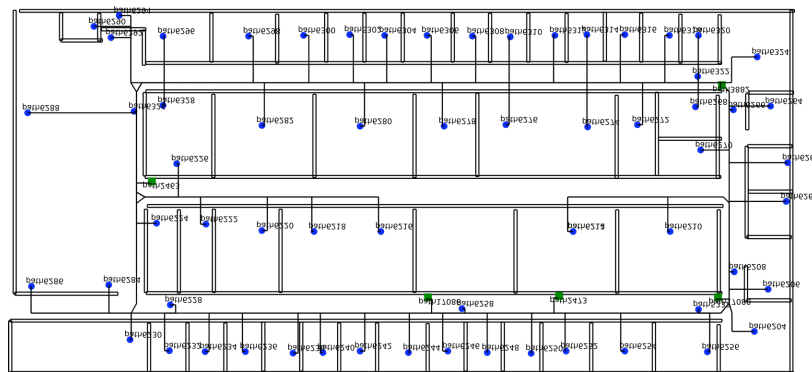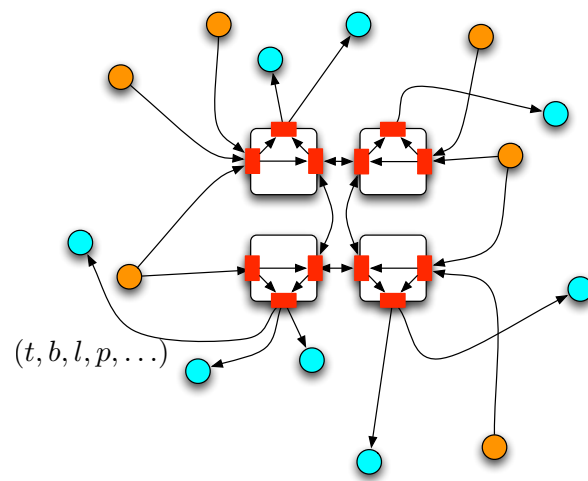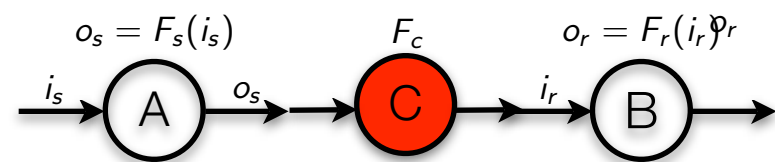
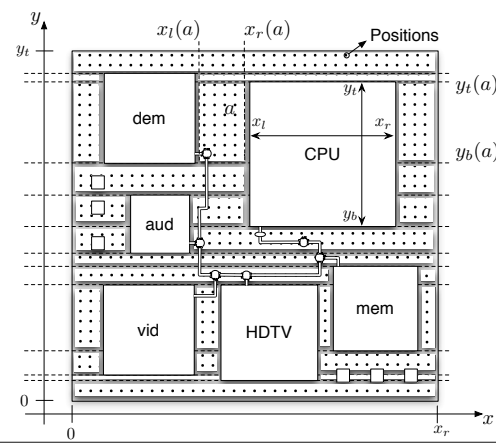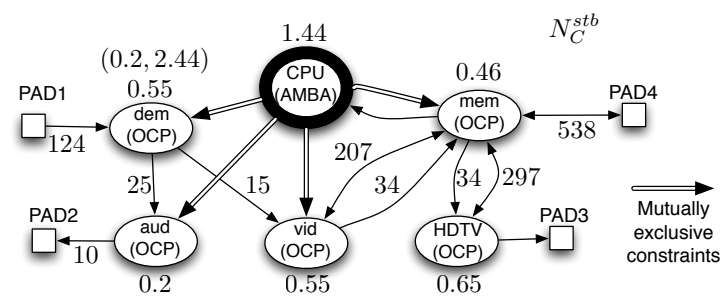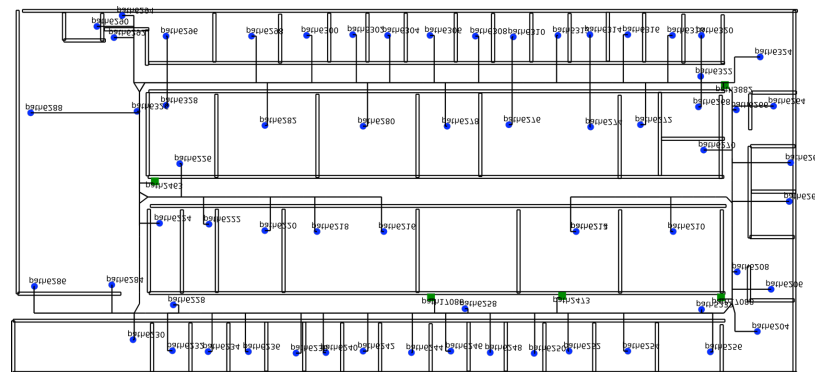Compose platform [preserve performance, infer QoS]

Compose communication nodes and links to minimize power and area [preserve QoS]

$o_s = F_s(i_s)$ F_c $o_r = F_r(i_r)$

Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness

Sensor, actuators, controllers, links...

Compose sensors, actuators, controllers. Define accuracy and stability of the distributed control [preserve functionality, infer QoS]

Buses, Routers, Gateways...

Compose communication nodes and links to minimize installation and operation cost [preserve QoS, infer HW/ SW performance]

Buses, Rings, Mesh, Routers, Interfaces...

Compose platform [preserve performance, infer QoS]

Compose communication nodes and links to minimize power and area [preserve QoS]

Abstraction

Thank you!

Alessandro Pinto, U.C. Berkeley,
"Communication-Based, Embedded System Design"
Dissertation Talk, Berkeley, 11/27/2007