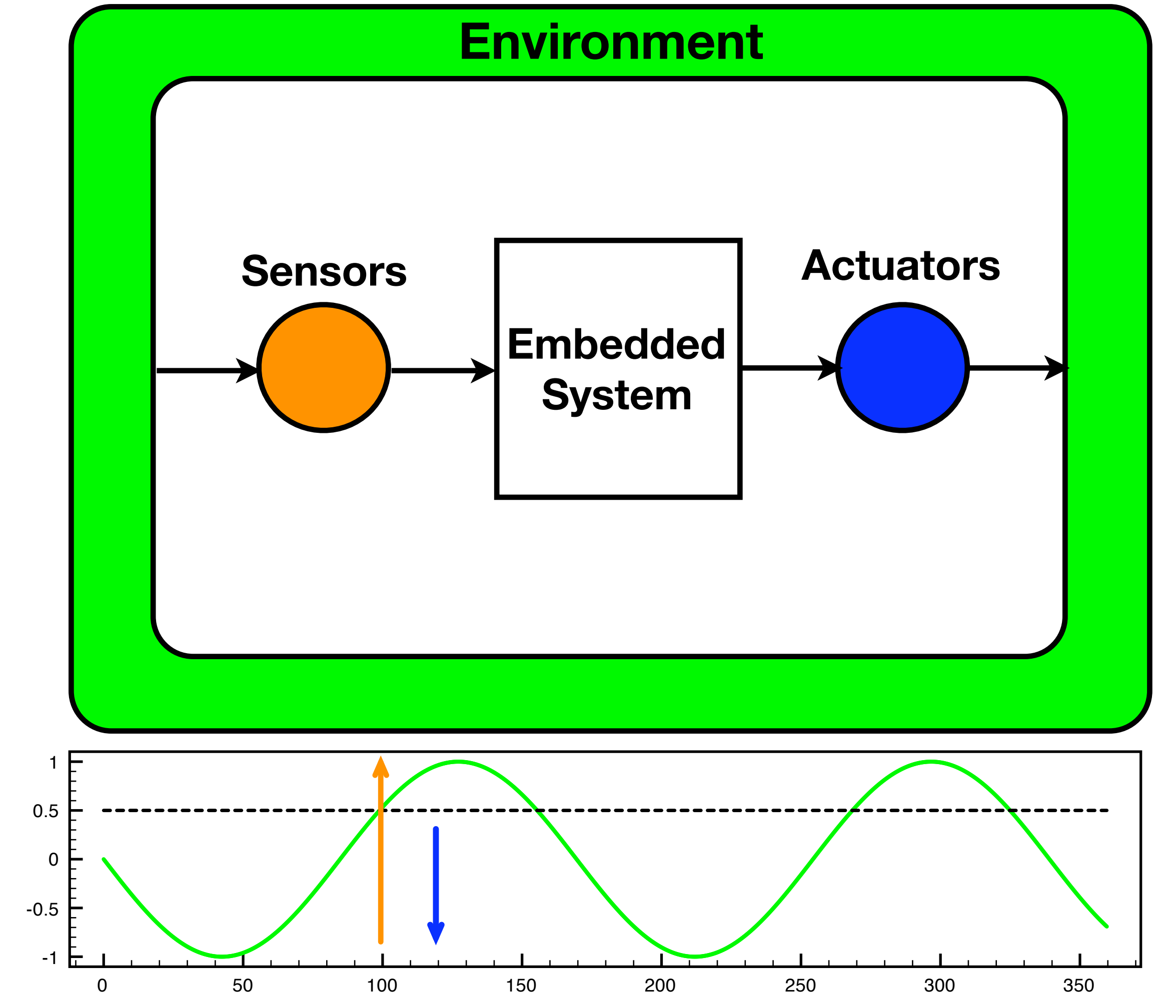
Communication-Based, Embedded System Design

Alessandro Pinto
University of California, Berkeley

apinto@eecs.berkeley.edu

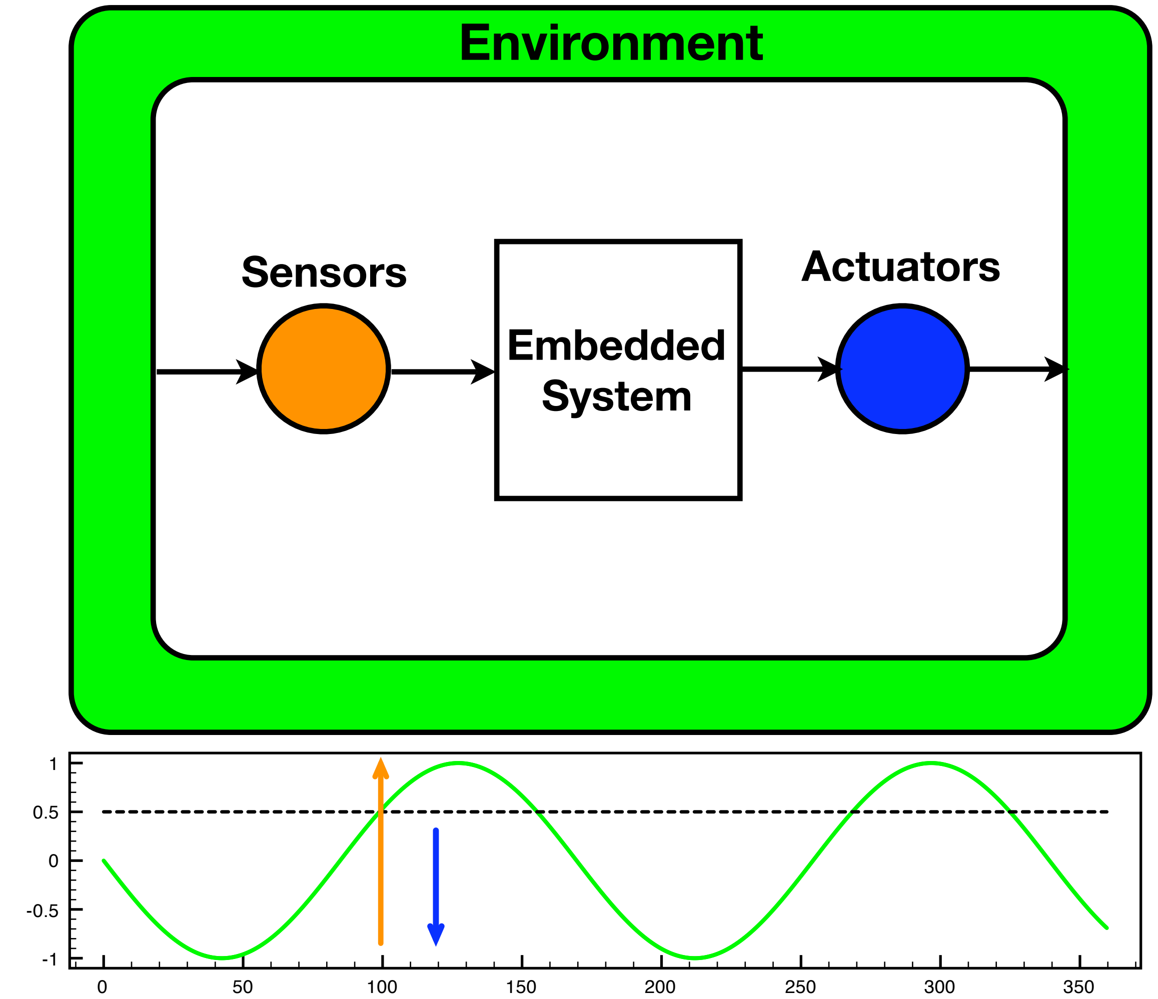
Dissertation Talk, Berkeley, 11/27/2007

Embedded Systems



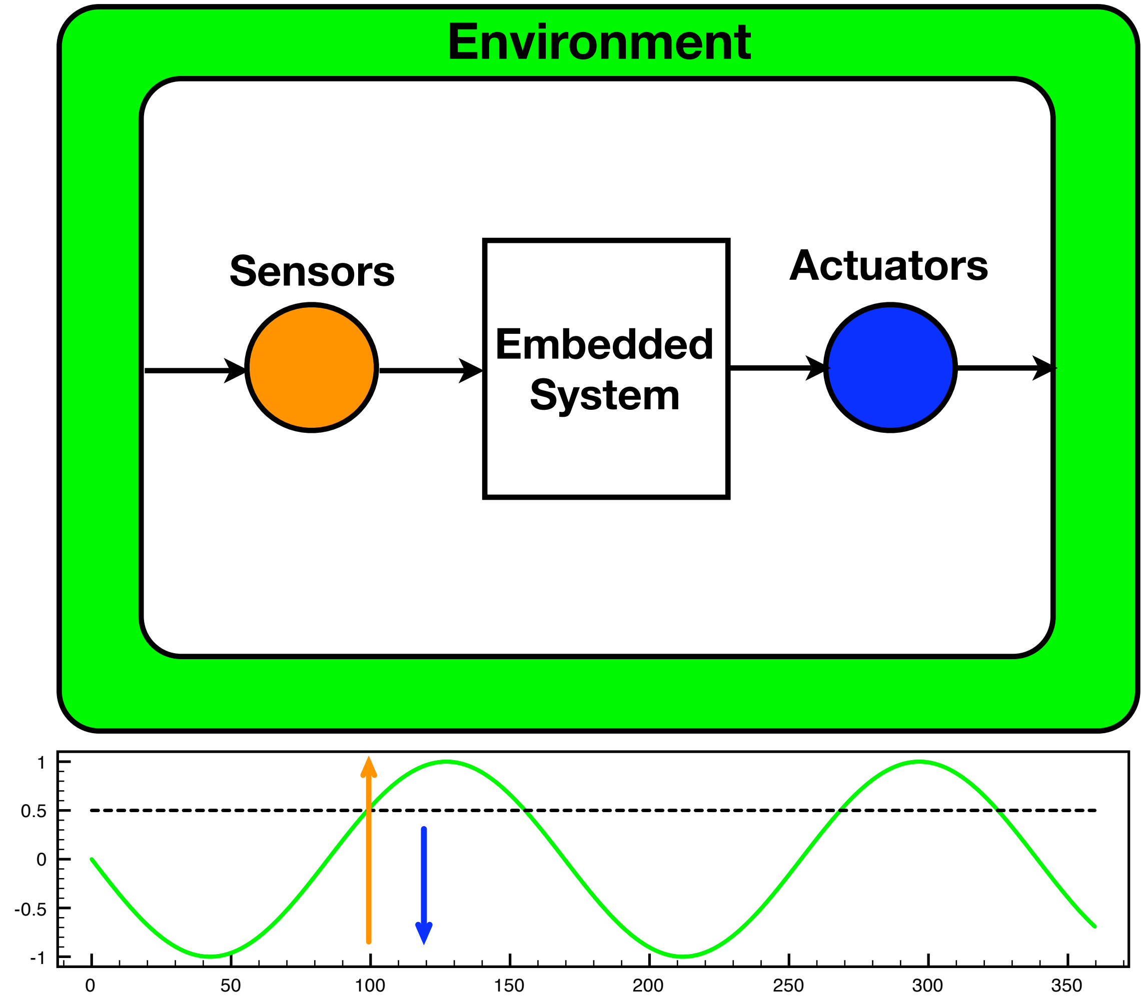
Embedded Systems

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



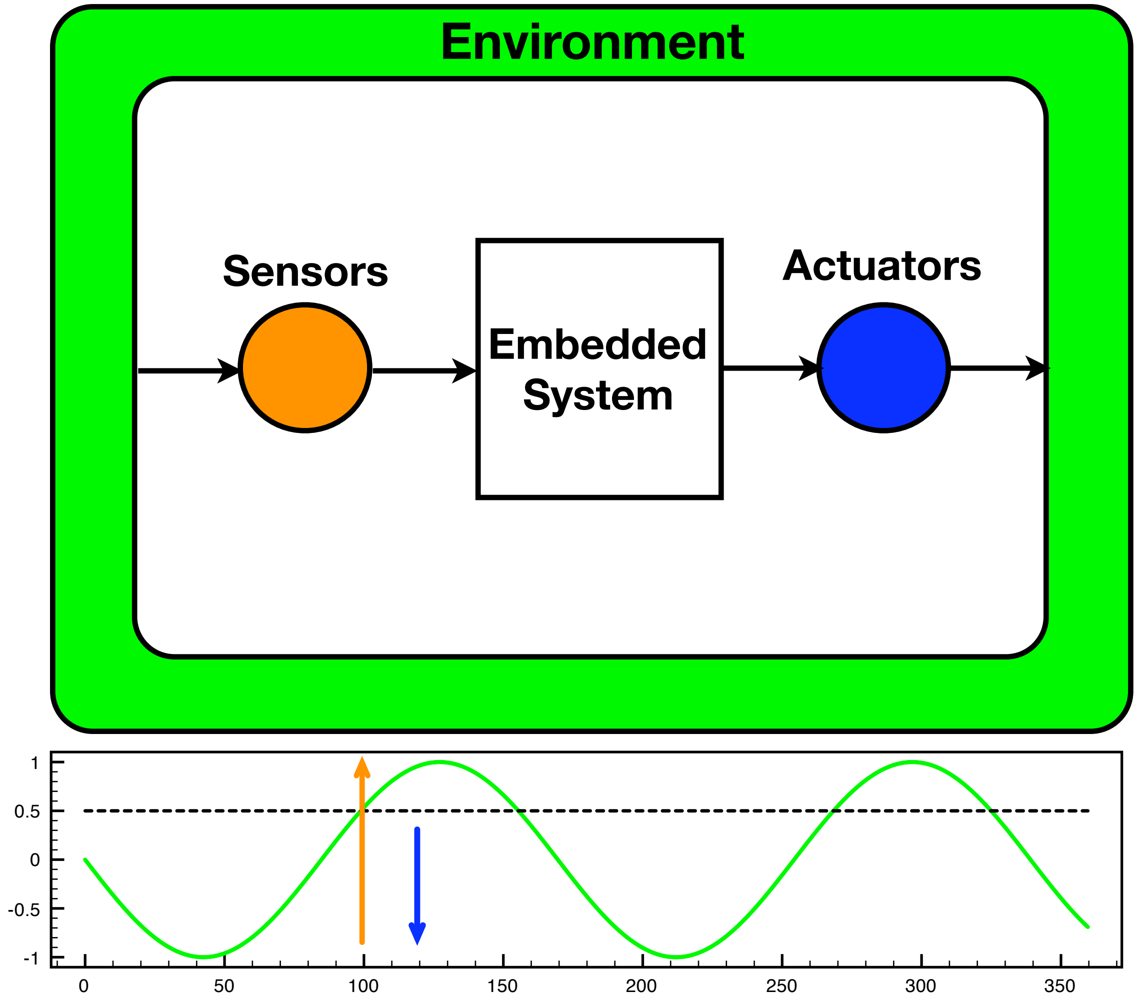
Embedded Systems

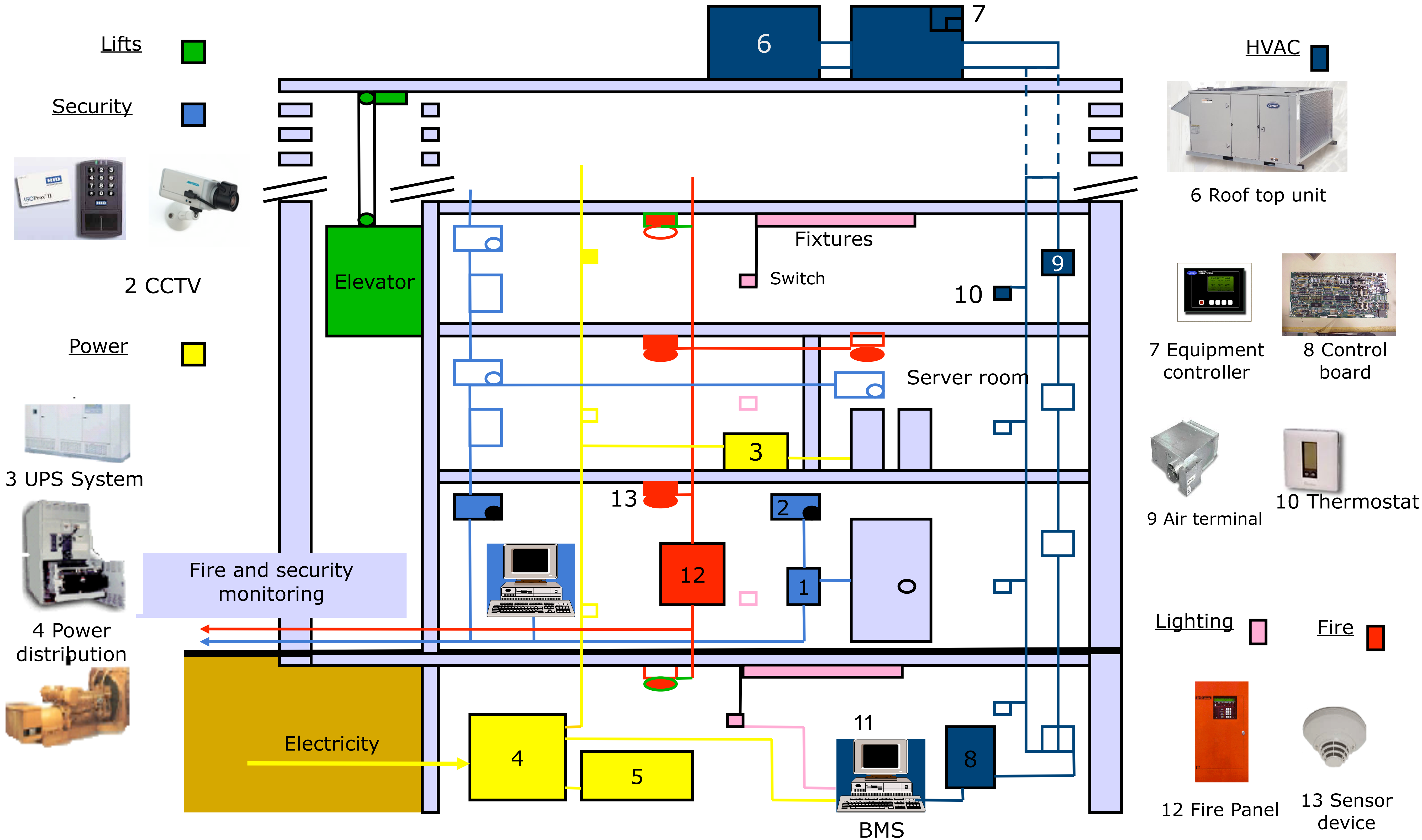
- ***Embedded*** in a physical environment
- ***Reacting*** at the speed of the environment



Embedded Systems

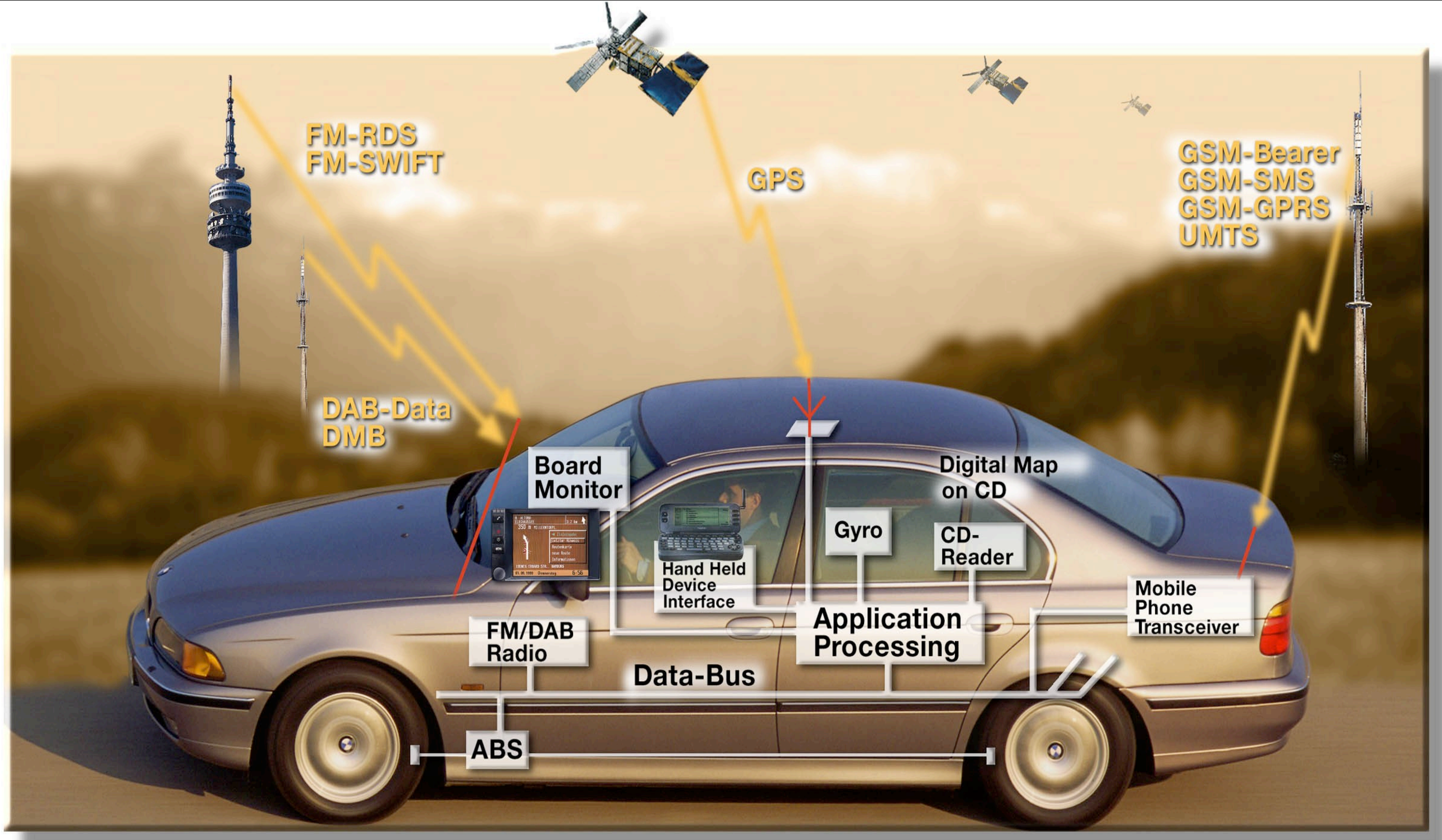
- ***Embedded*** in a physical environment
- ***Reacting*** at the speed of the environment
- ***Heterogeneous*** composition of subsystems
- ***Networked***
  - Spatially distributed
  - Cooperative





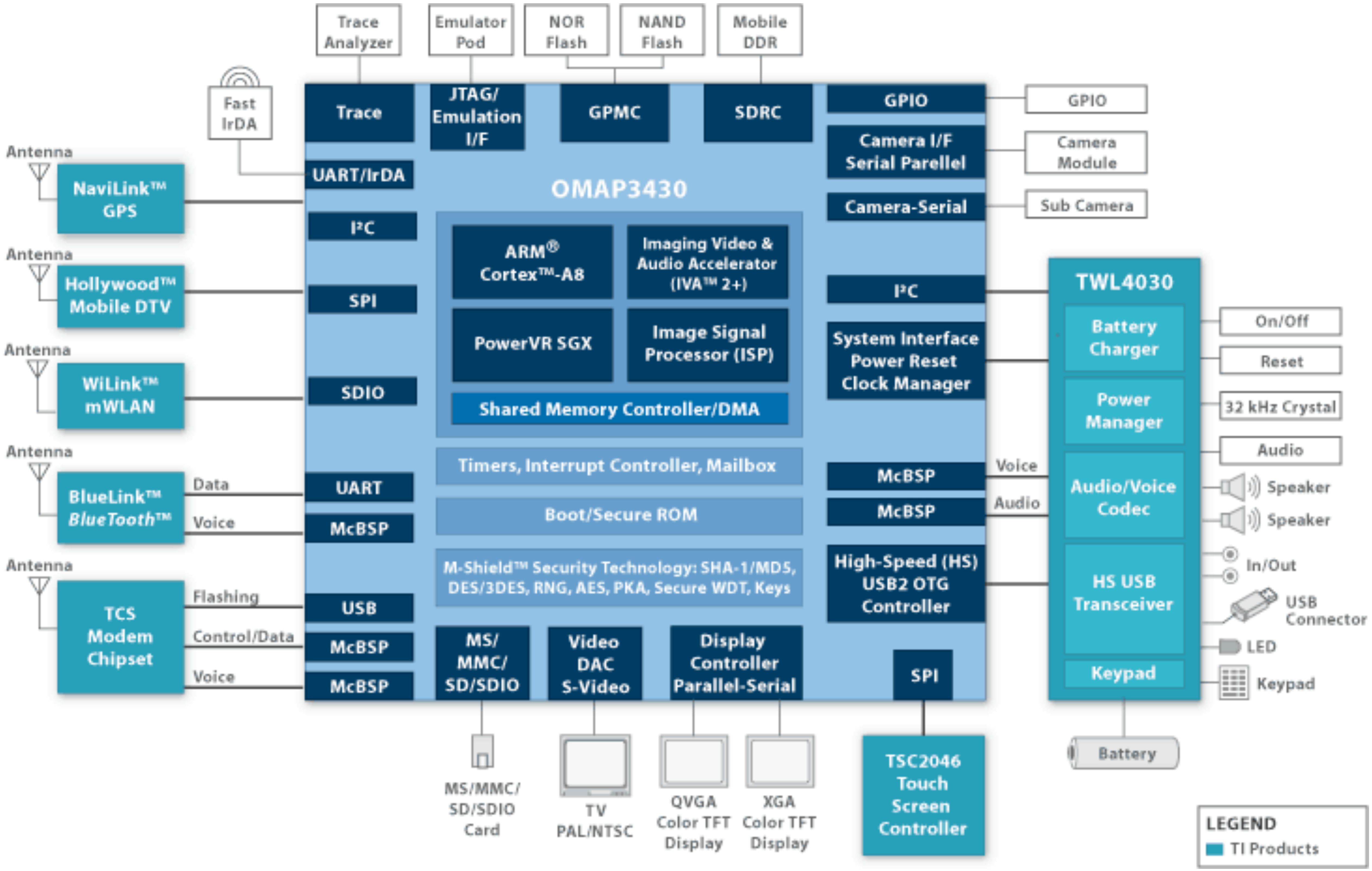
Examples of Embedded Systems

Building Automation (Source: Clas Jacobson, UTRC)



Examples of Embedded Systems

Automotive

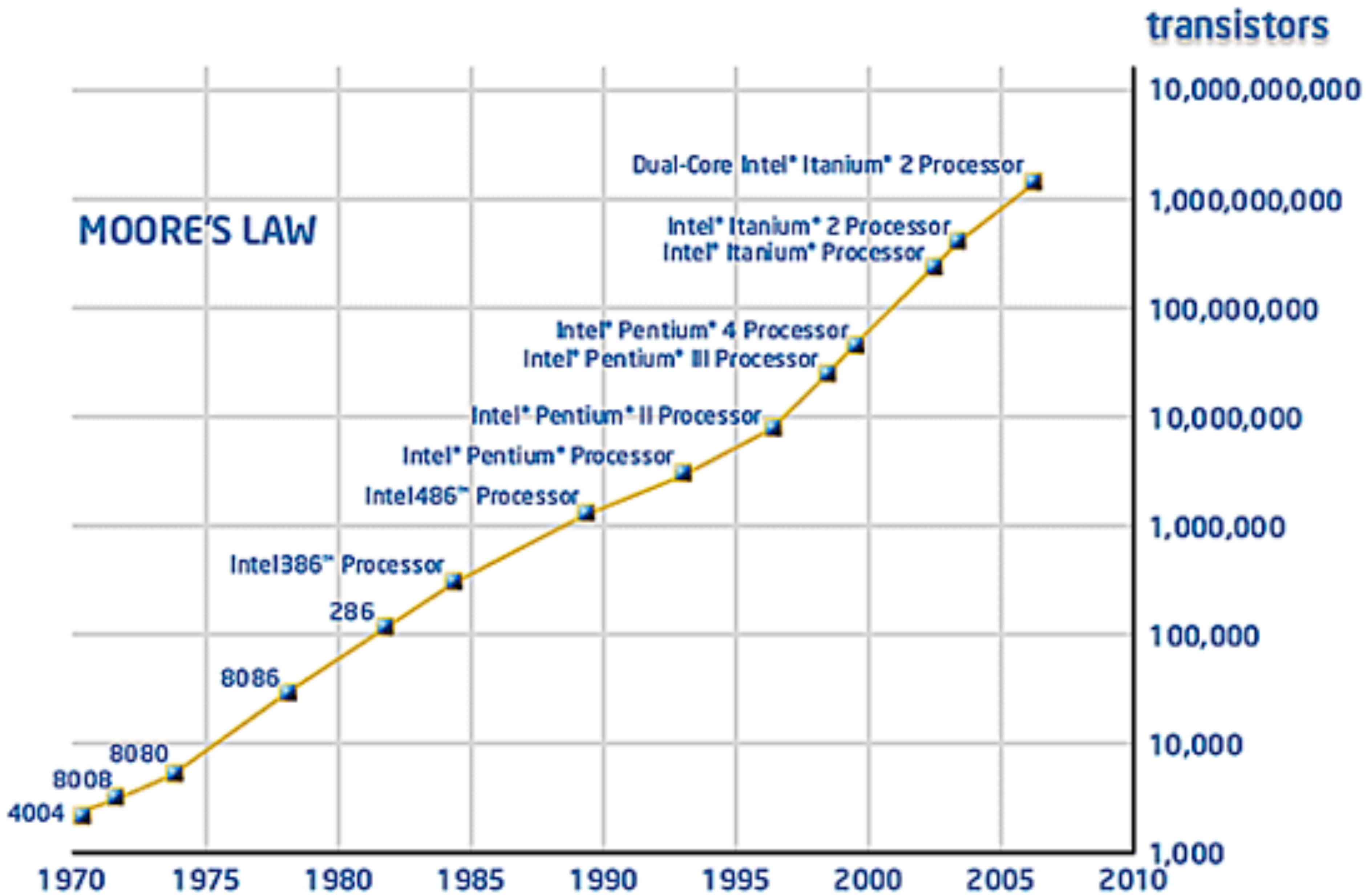


Examples of Embedded Systems

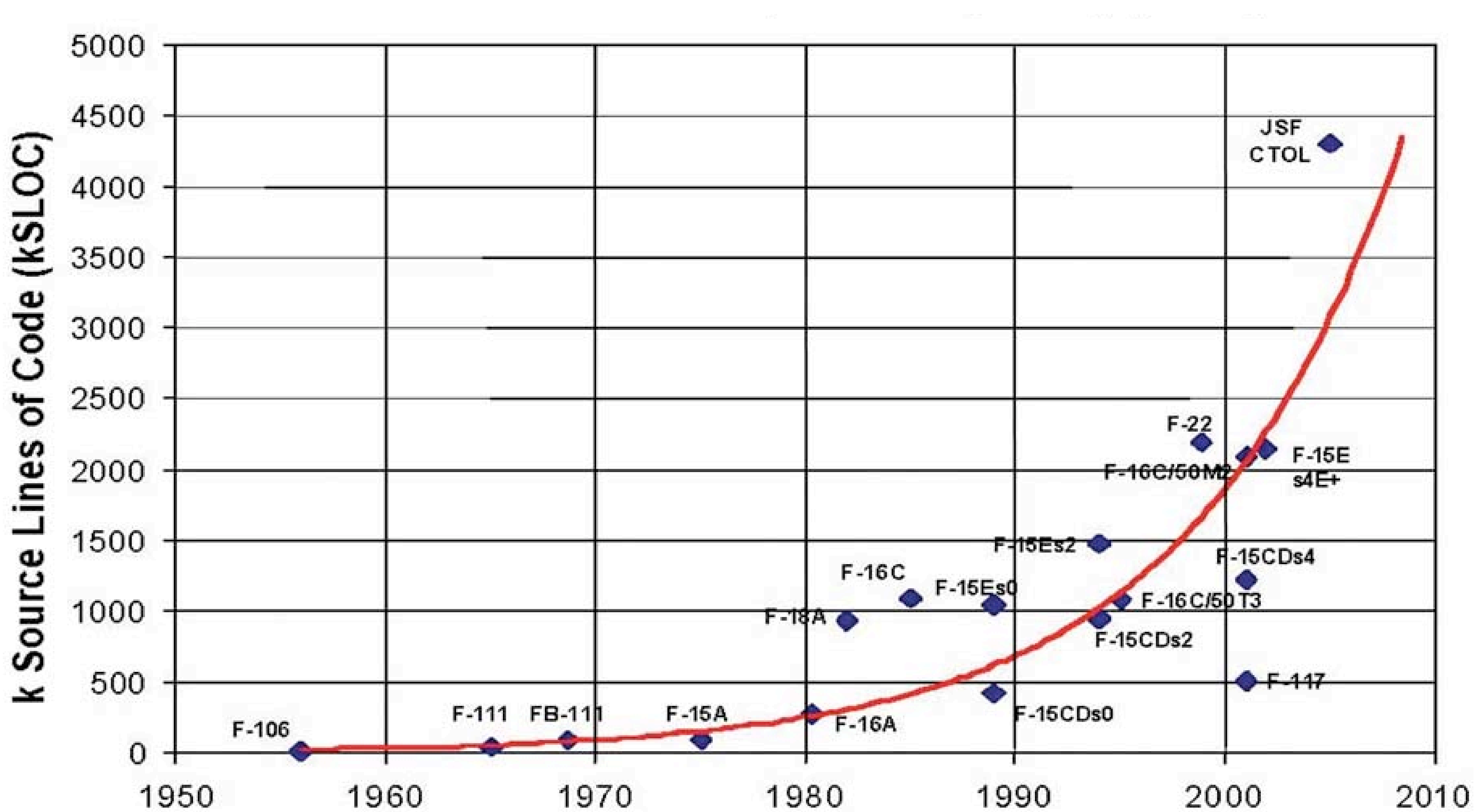
System-On-Chip

Design Complexity Trends

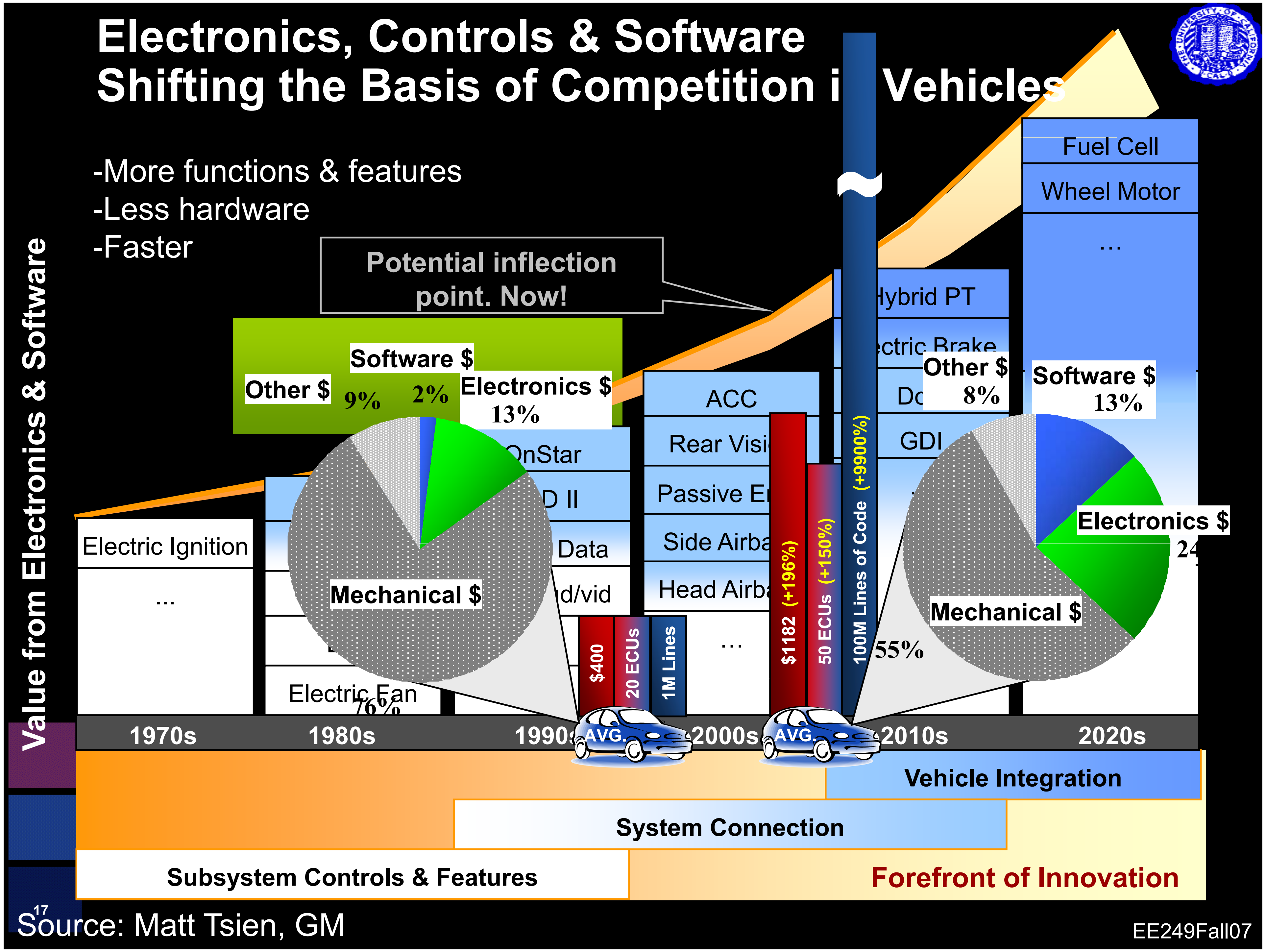
Design Complexity Trends

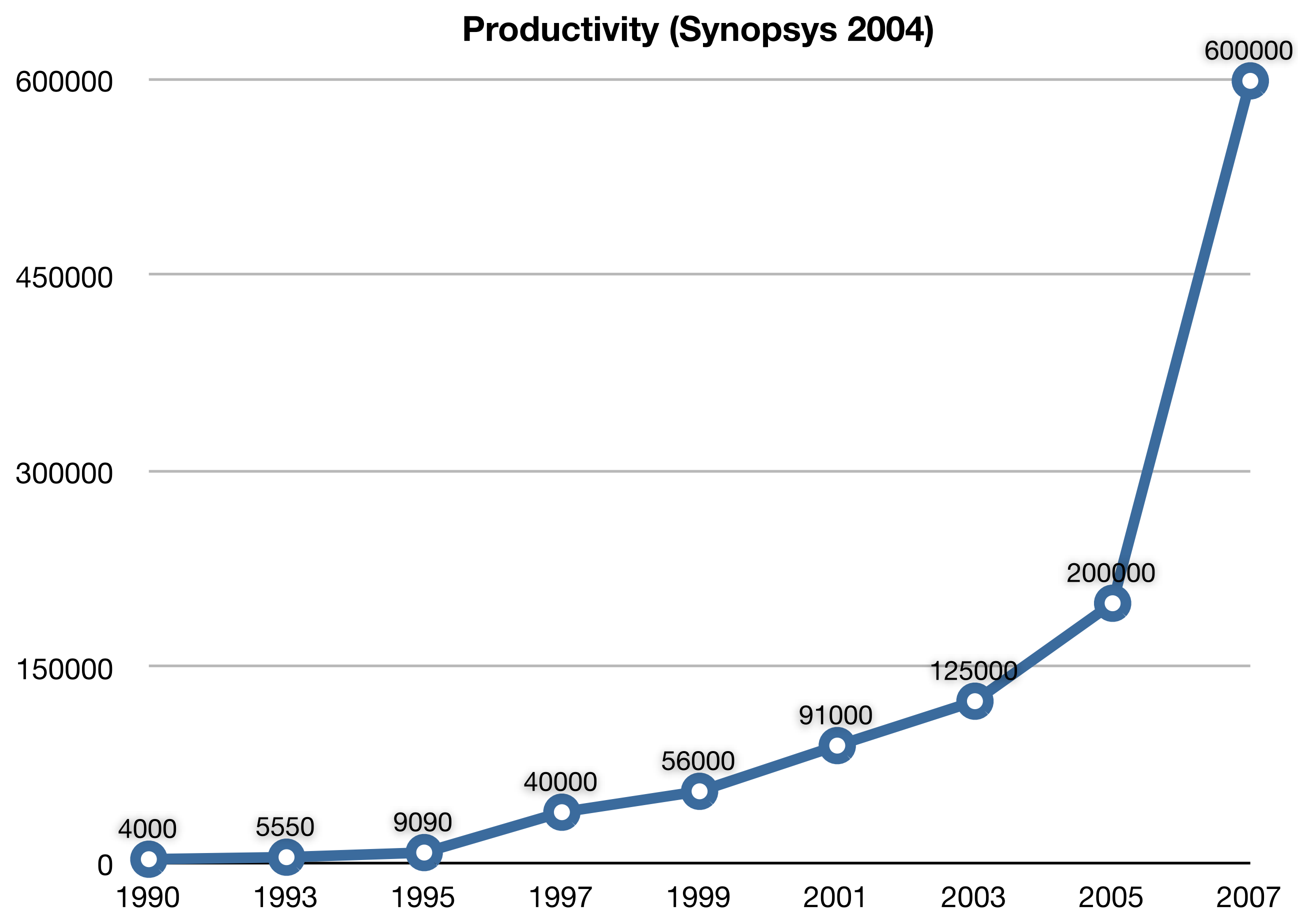


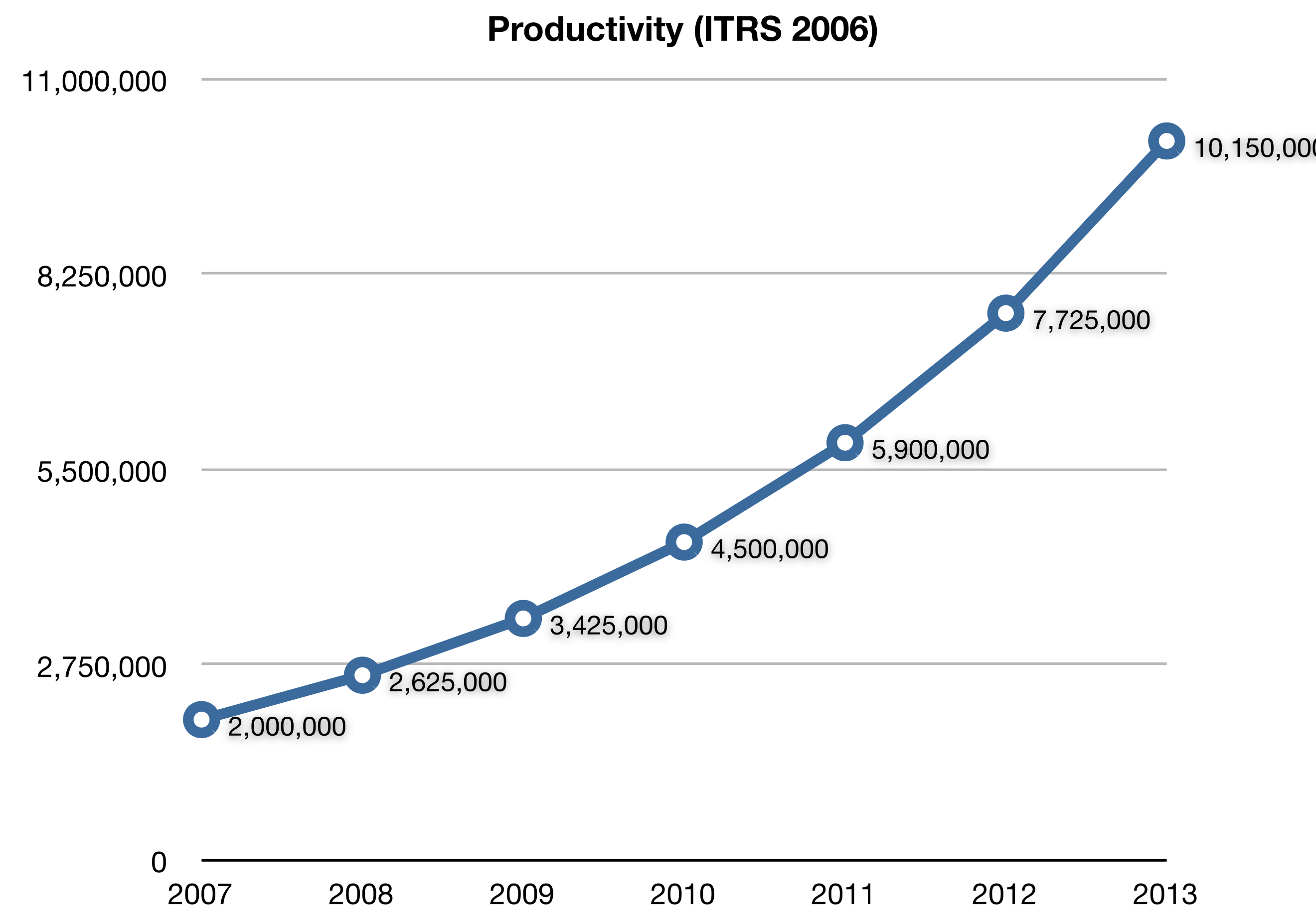
Design Complexity Trends



Design Complexity Trends

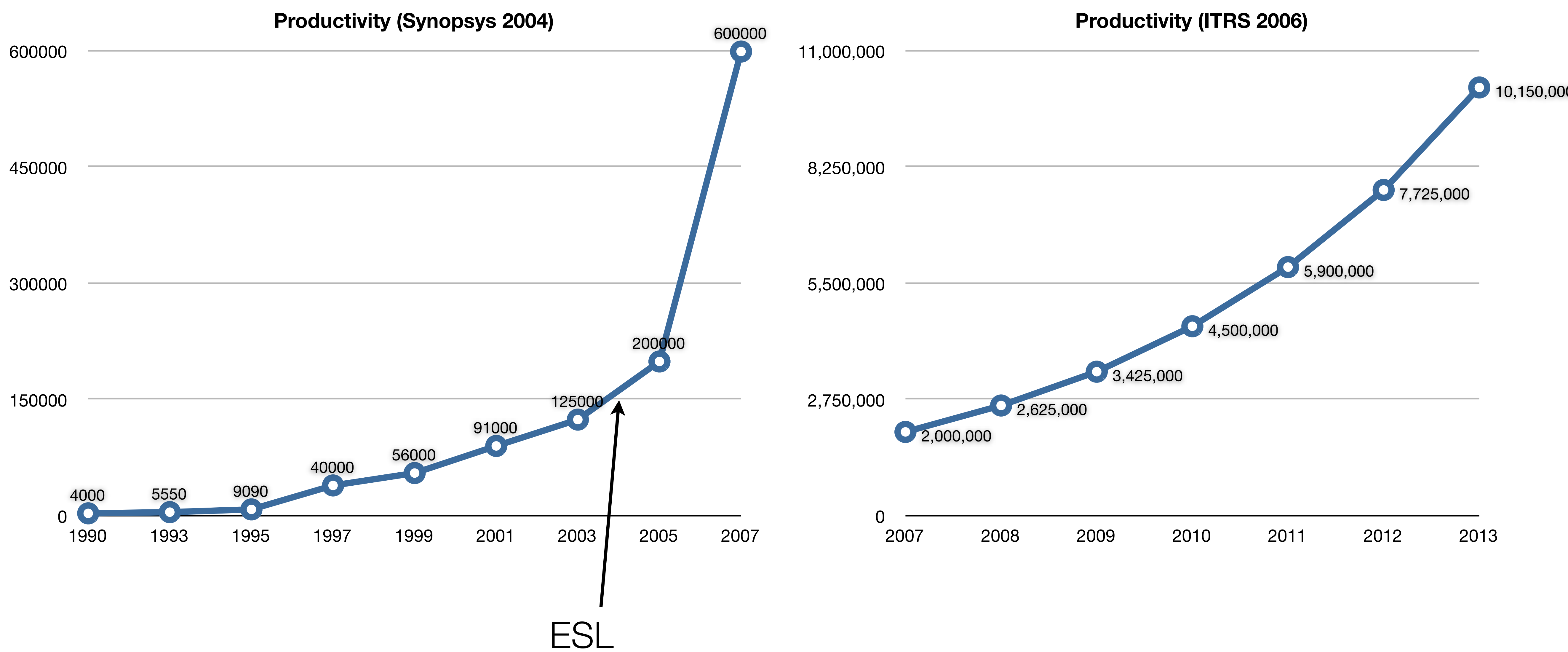






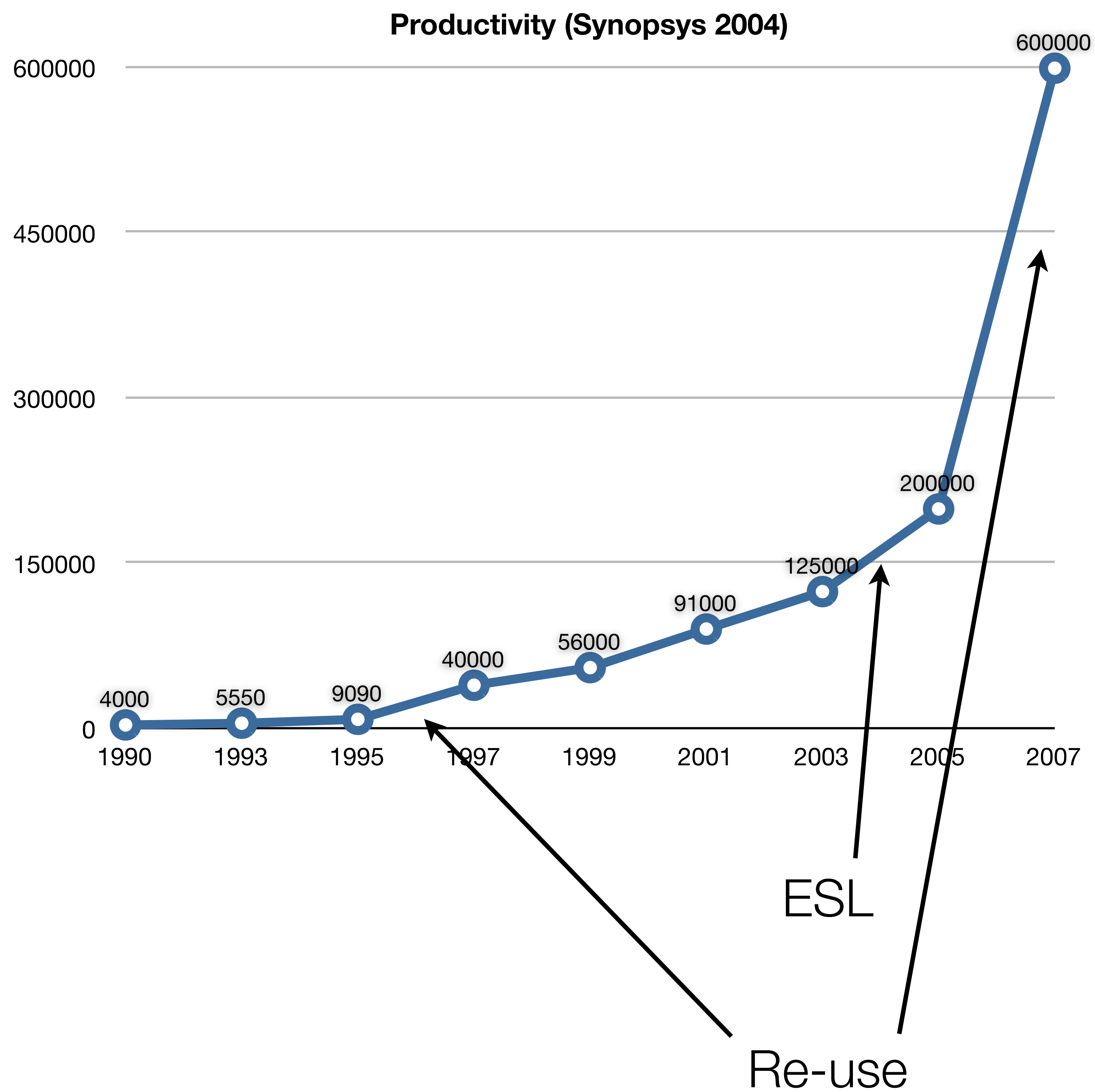
Productivity

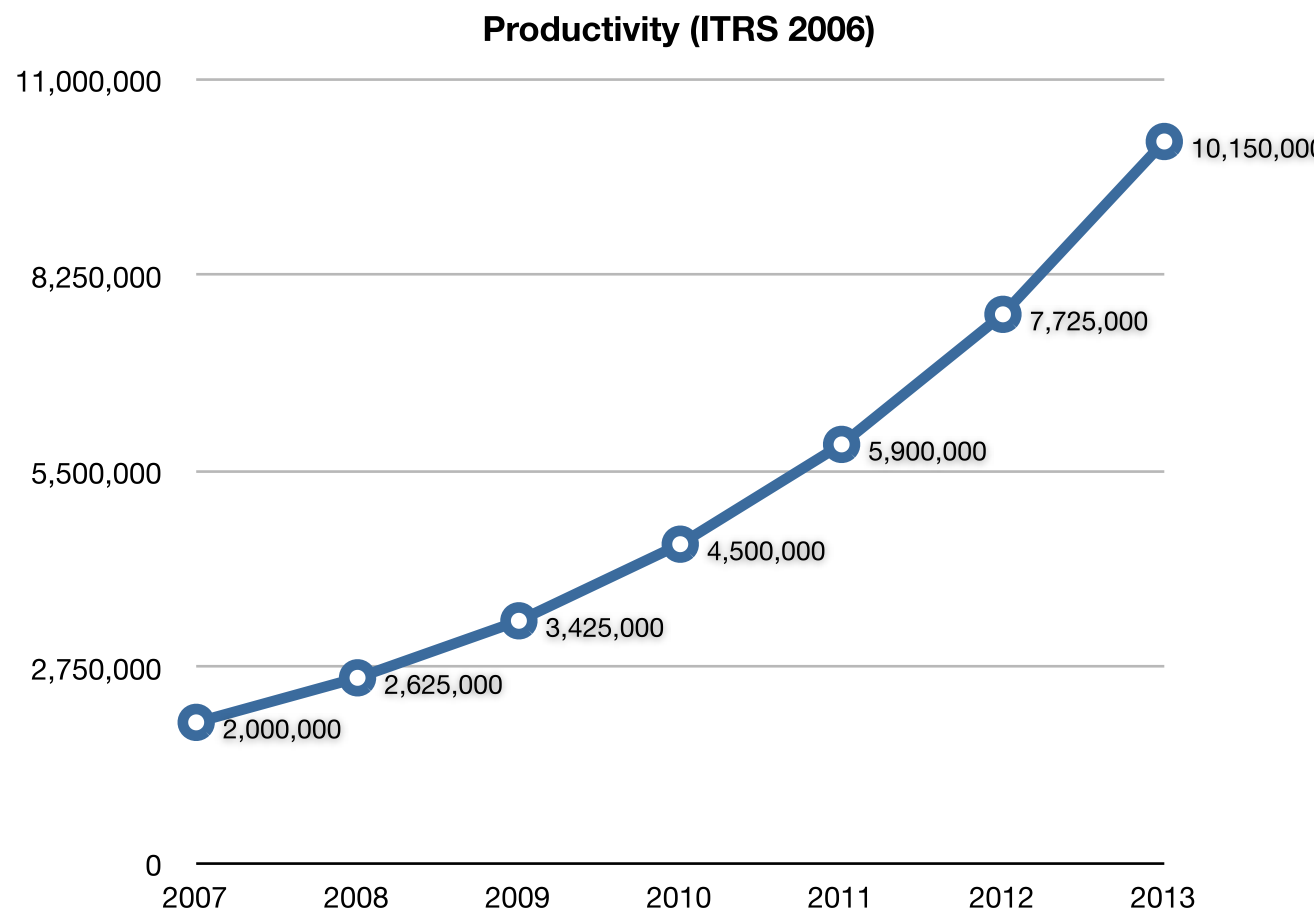
Productivity in number of gates per designer per year



Productivity

Productivity in number of gates per designer per year





Productivity

Productivity in number of gates per designer per year

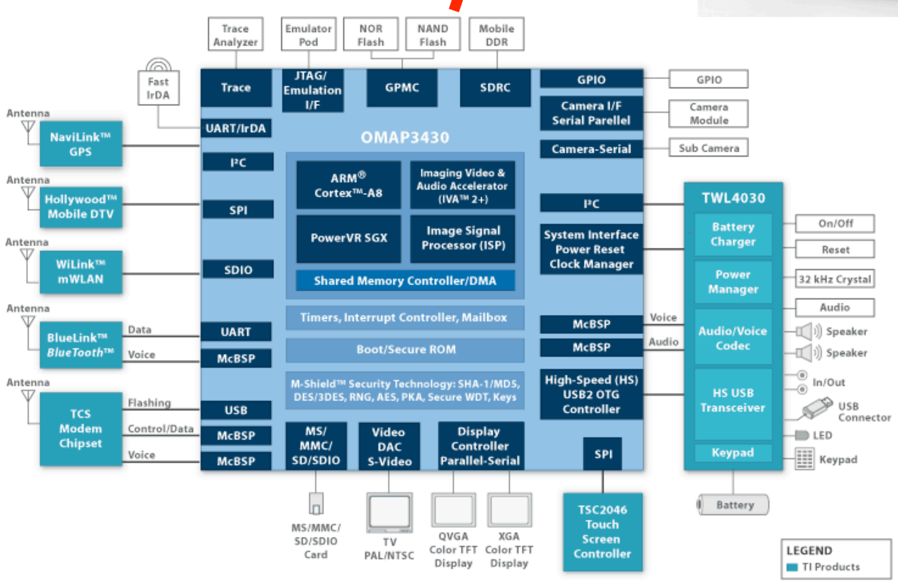
Reuse

Reuse


Email
SMS/MMS
Internet Explorer
Tasks
Comm Manager
MAY
15
Calendar
htc

Reuse

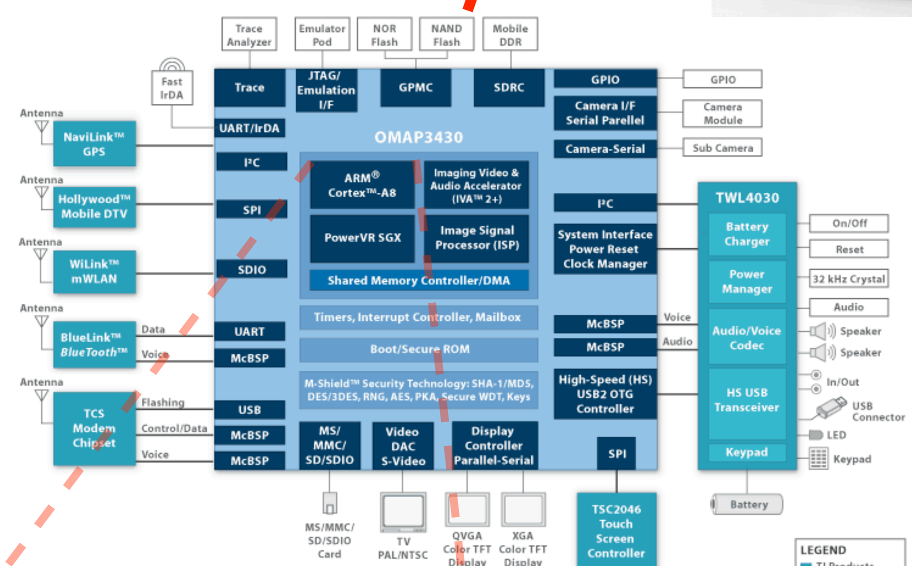
TI OMAP 3



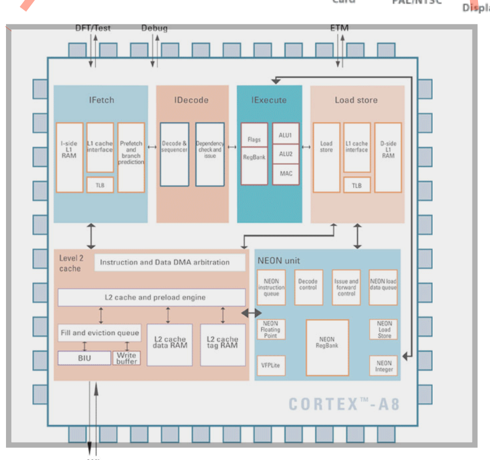
Designed for Windows Mobile™

Reuse

TI OMAP 3



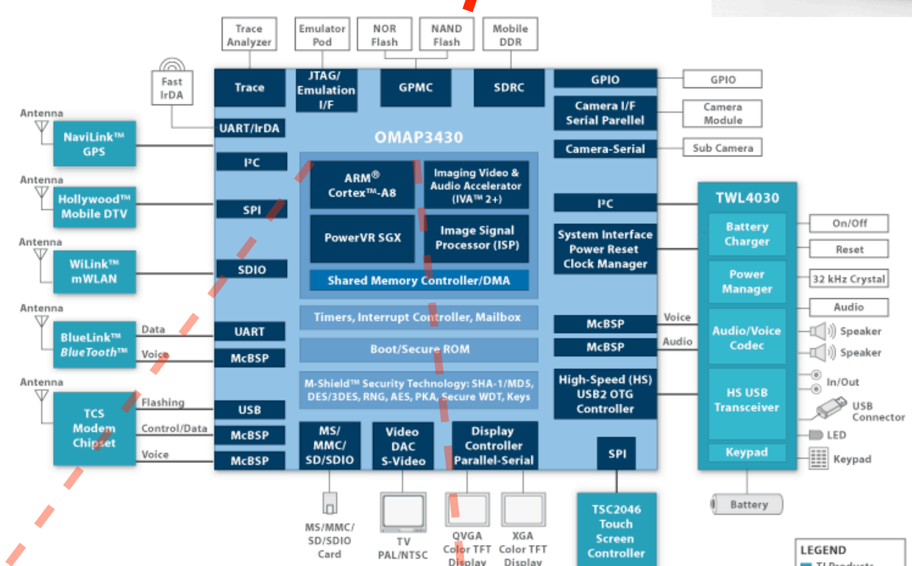
Designed for Windows Mobile™

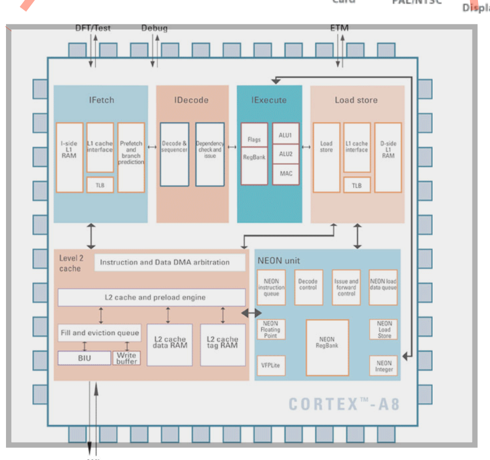


ARM Cortex-A8

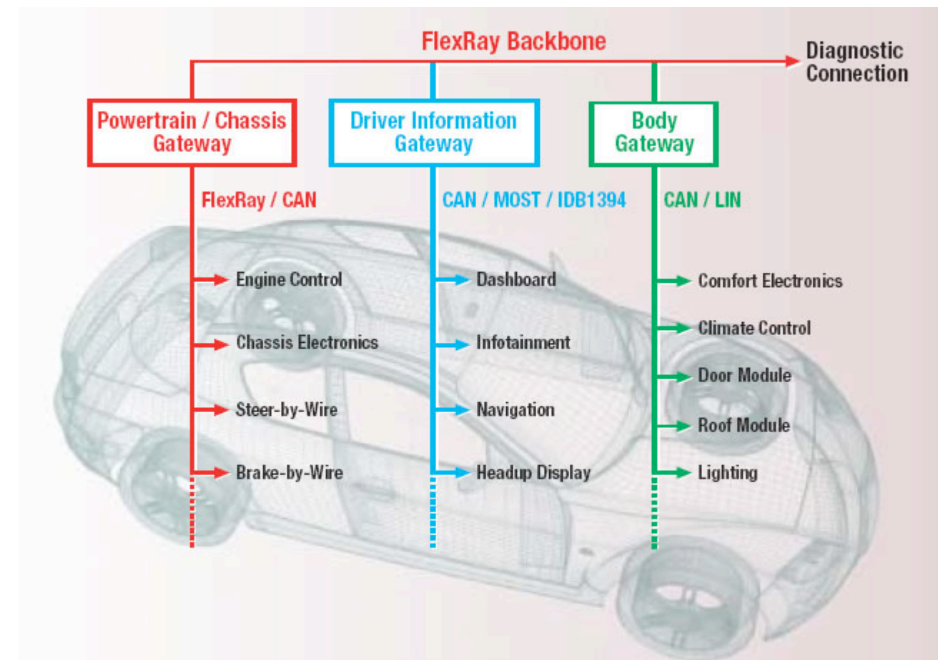
Reuse

TI OMAP 3



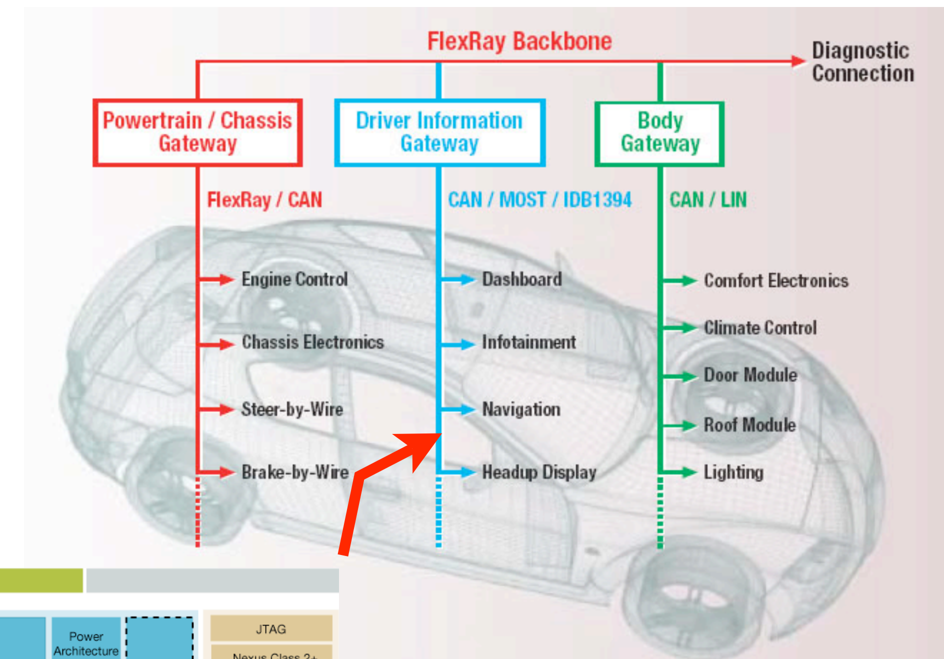


ARM Cortex-A8

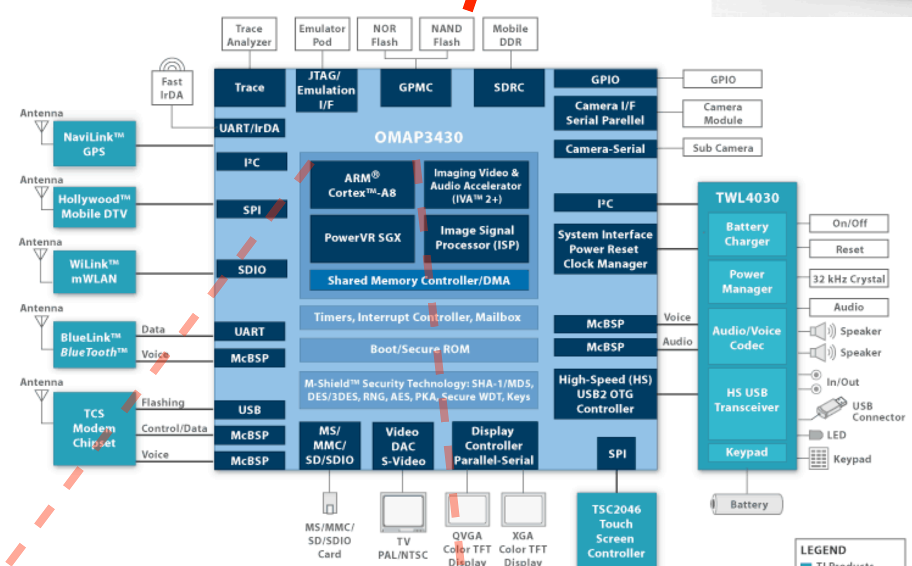


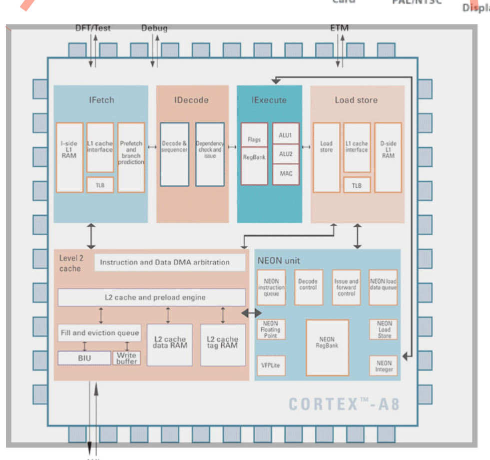
Reuse

Designed for Windows Mobile

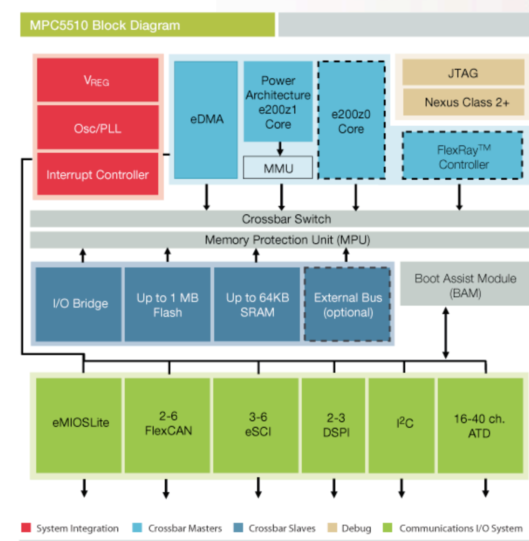


TI OMAP 3



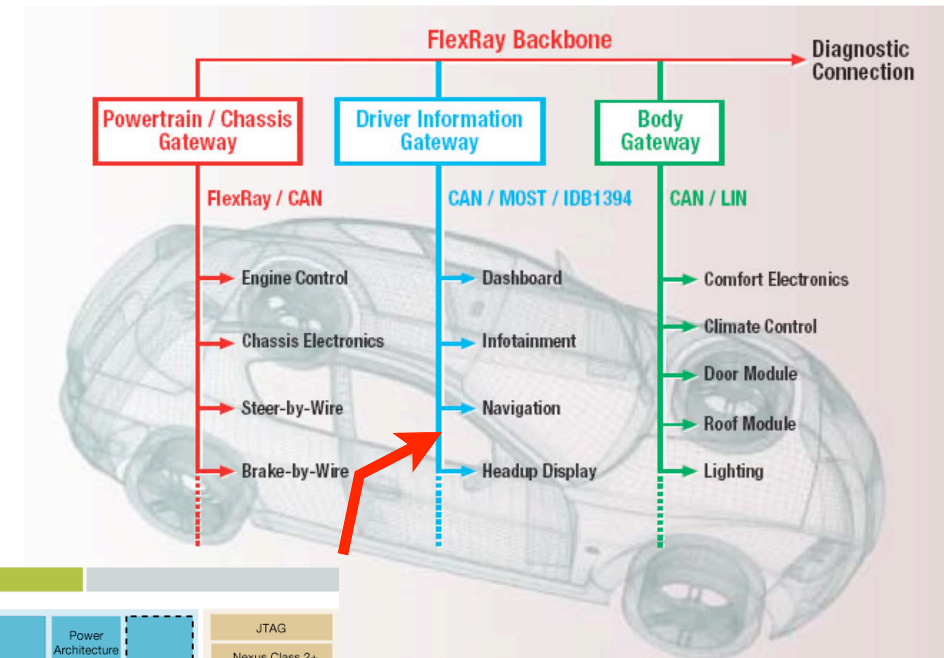


ARM Cortex-A8

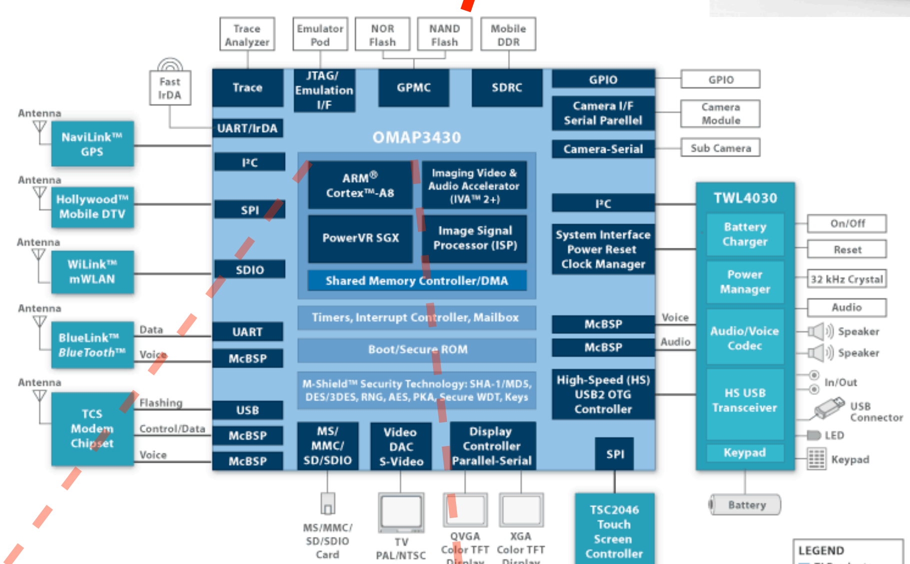


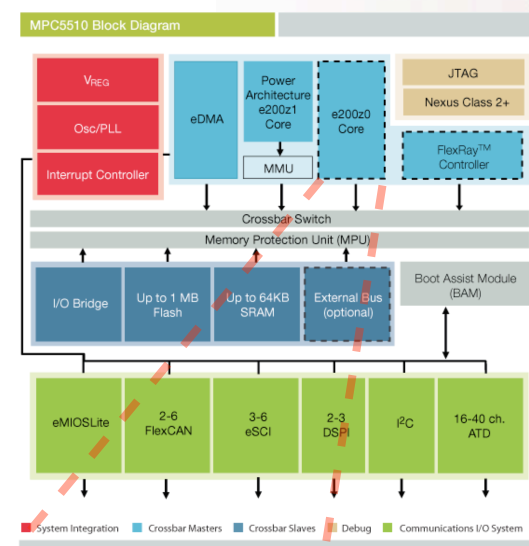
Freescale MPC5510

Reuse



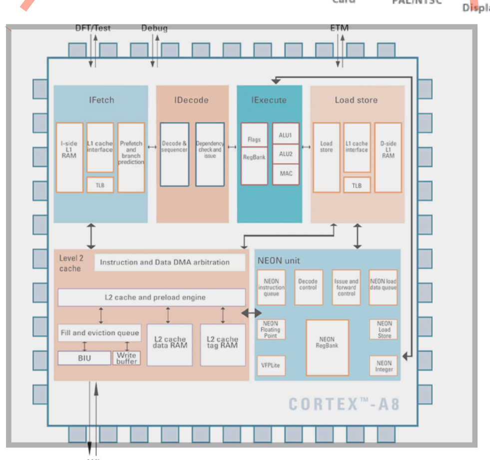
TI OMAP 3



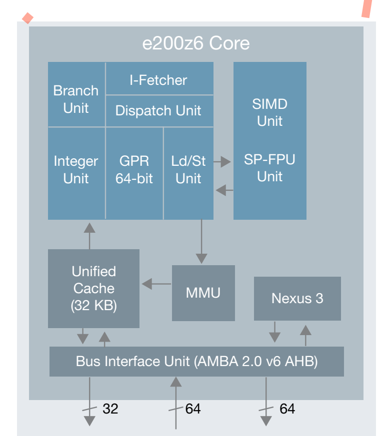


Designed for Windows Mobile™

Freescale MPC5510



ARM Cortex-A8



e200

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

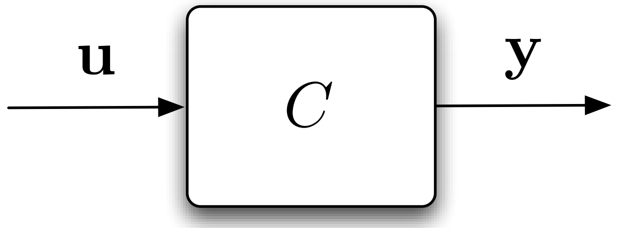
Outline

- Defining the Problem
  - Building Automation and On-Chip Communication
- A Formal Framework to Enable Synthesis
- Applications
  - Abstraction (Modeling)
  - Algorithms
  - Results

Defining the Problem

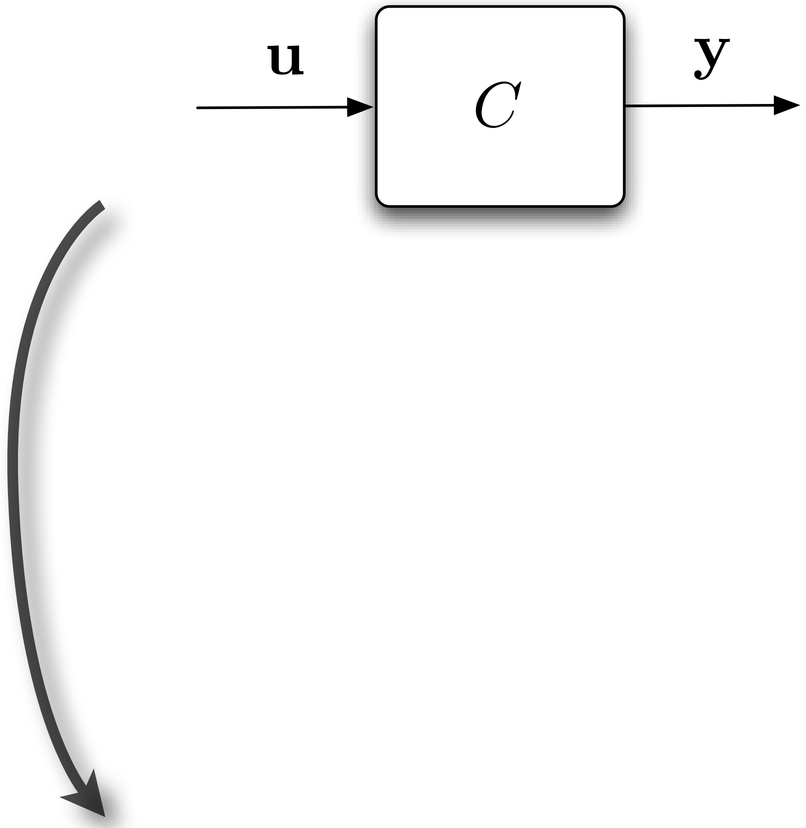
Distributed Control

Centralized Control



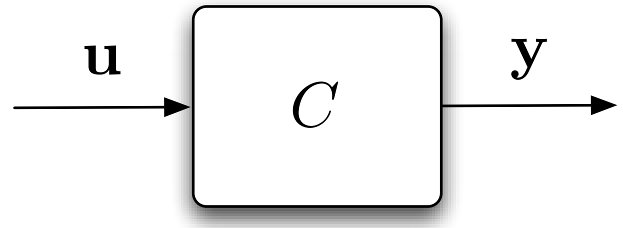
Distributed Control

Centralized Control

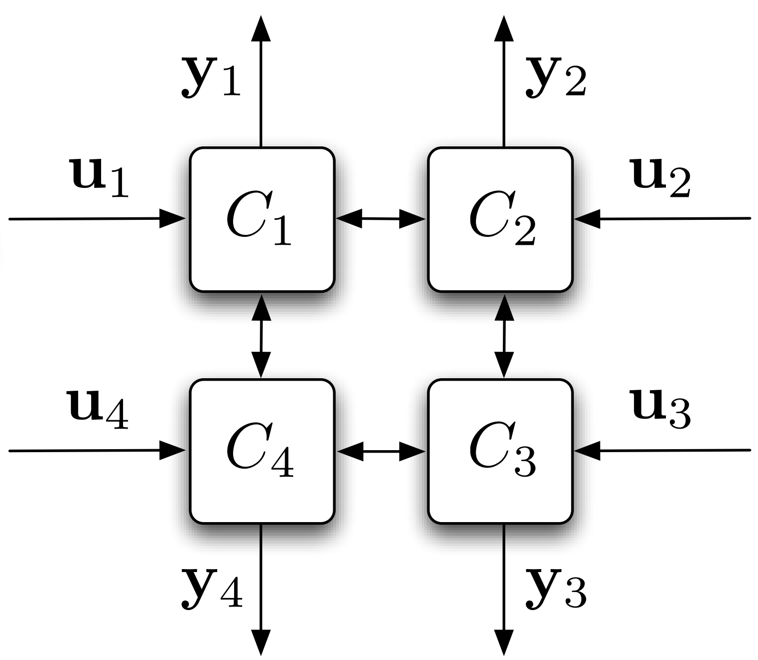


Distributed Control

Centralized Control

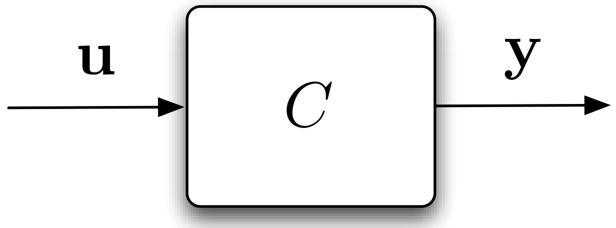


Distributed Control

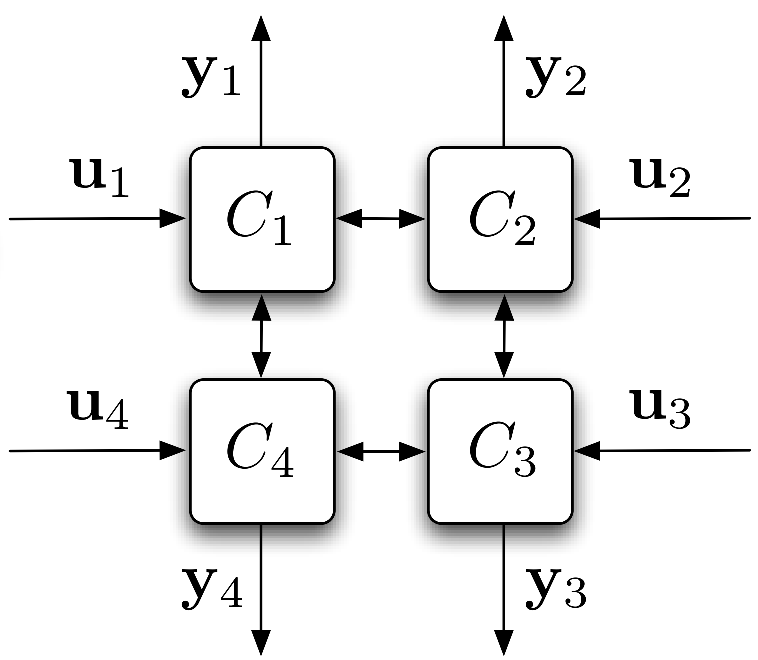


Distributed Control

Centralized Control

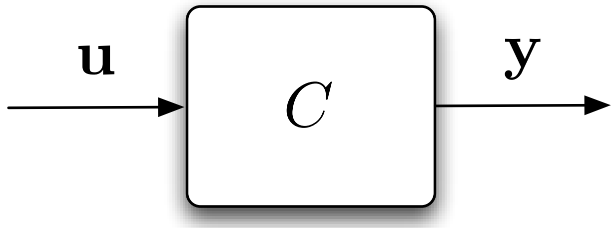


Distributed Control

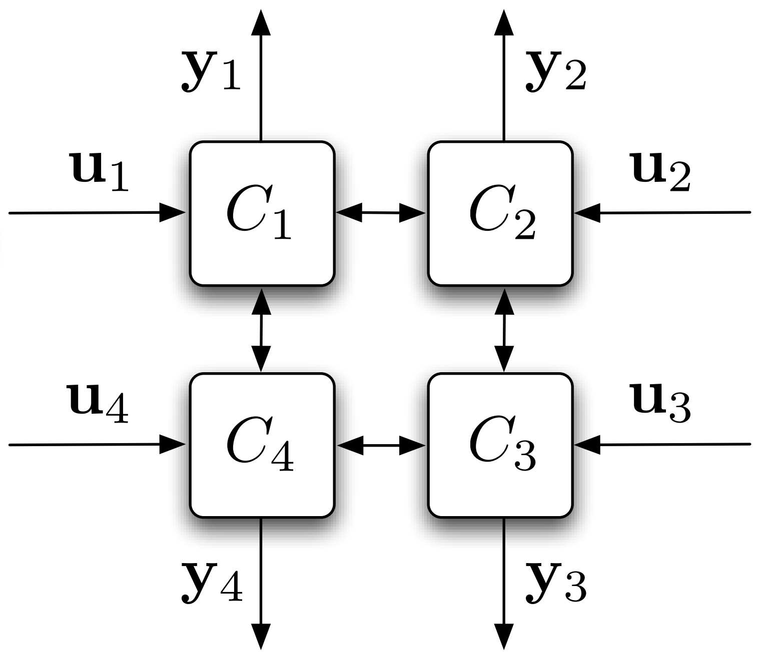


Distributed Control

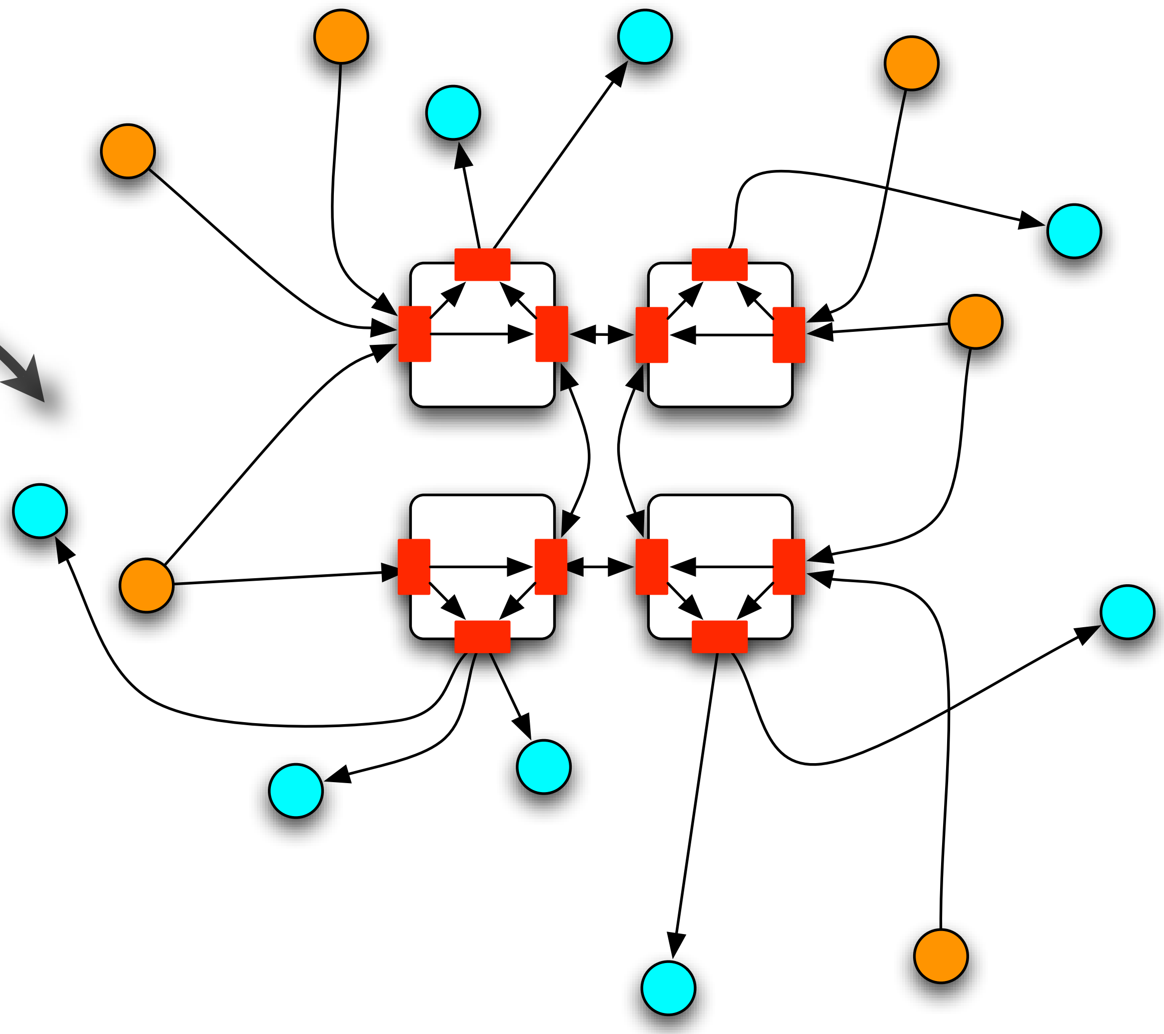
Centralized Control



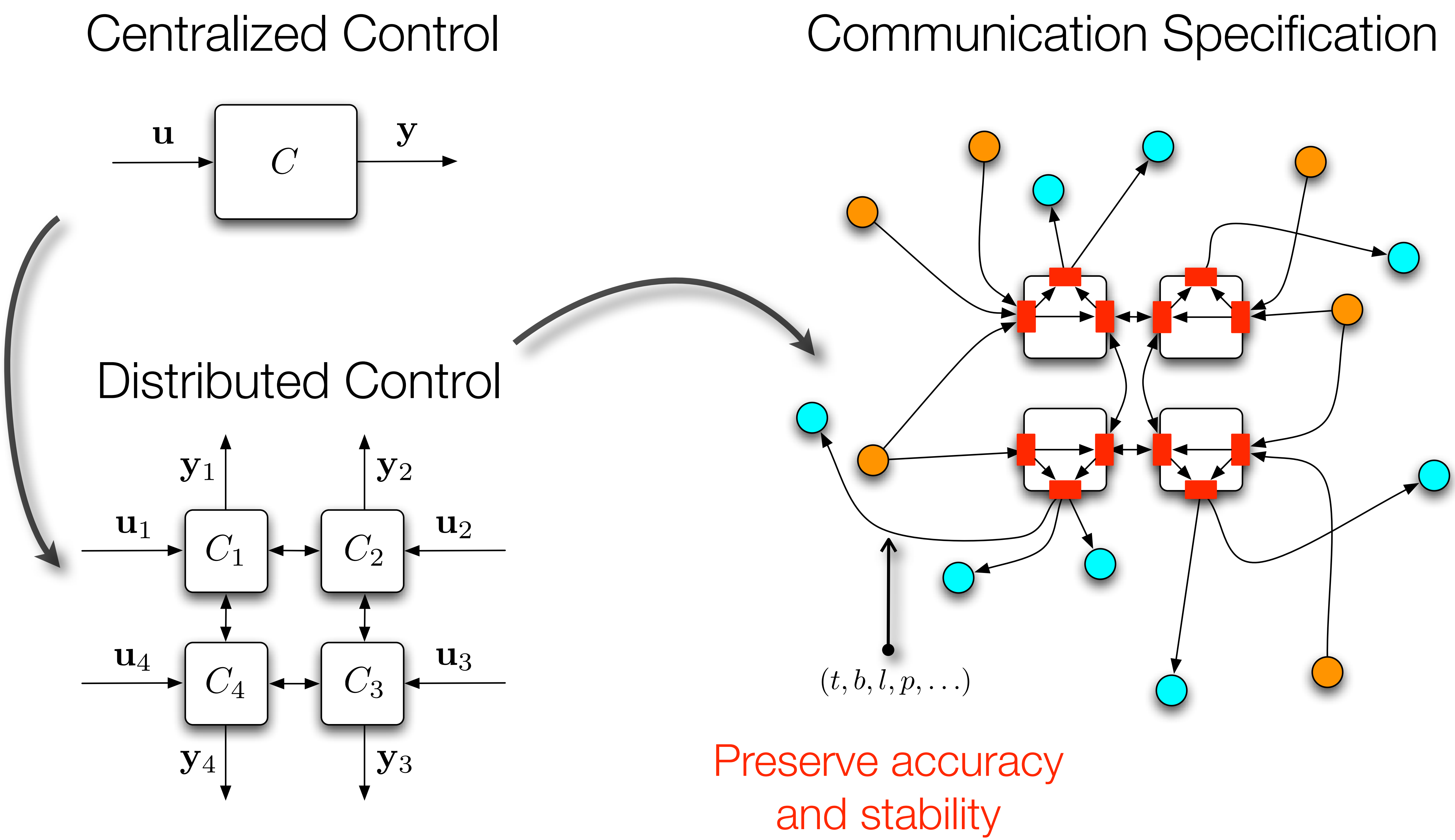
Distributed Control



Communication Specification



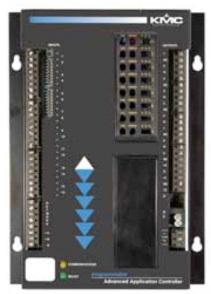
Distributed Control



Available Network Technologies

BacNet

Router



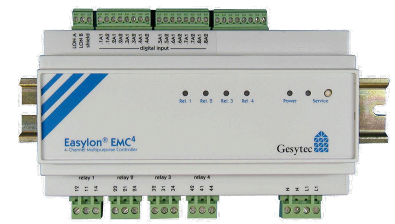
DDC

Sensor

etc.

LonWorks

Router



DDC

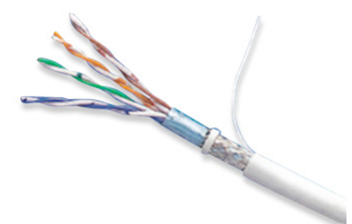
Sensor

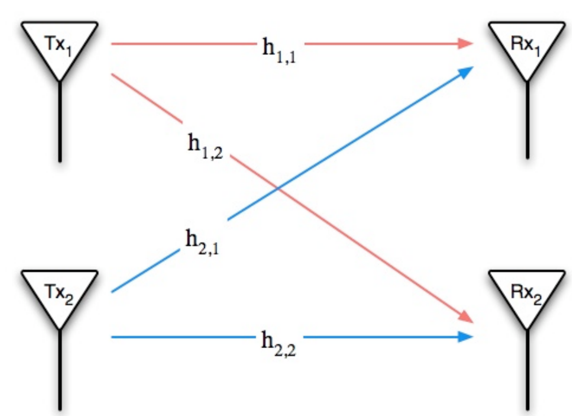
etc.

ZigBee

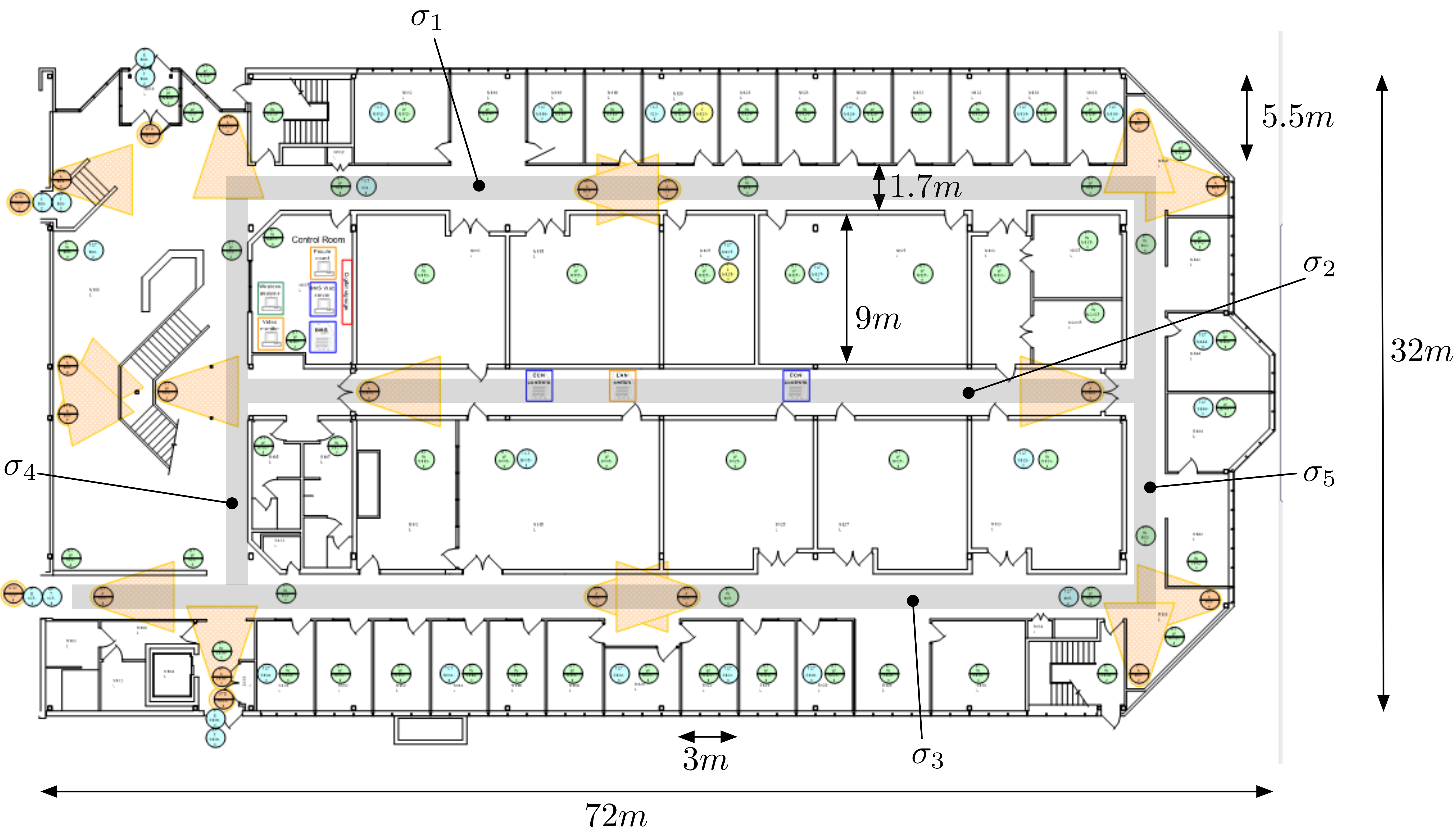
APOGEE

Channels

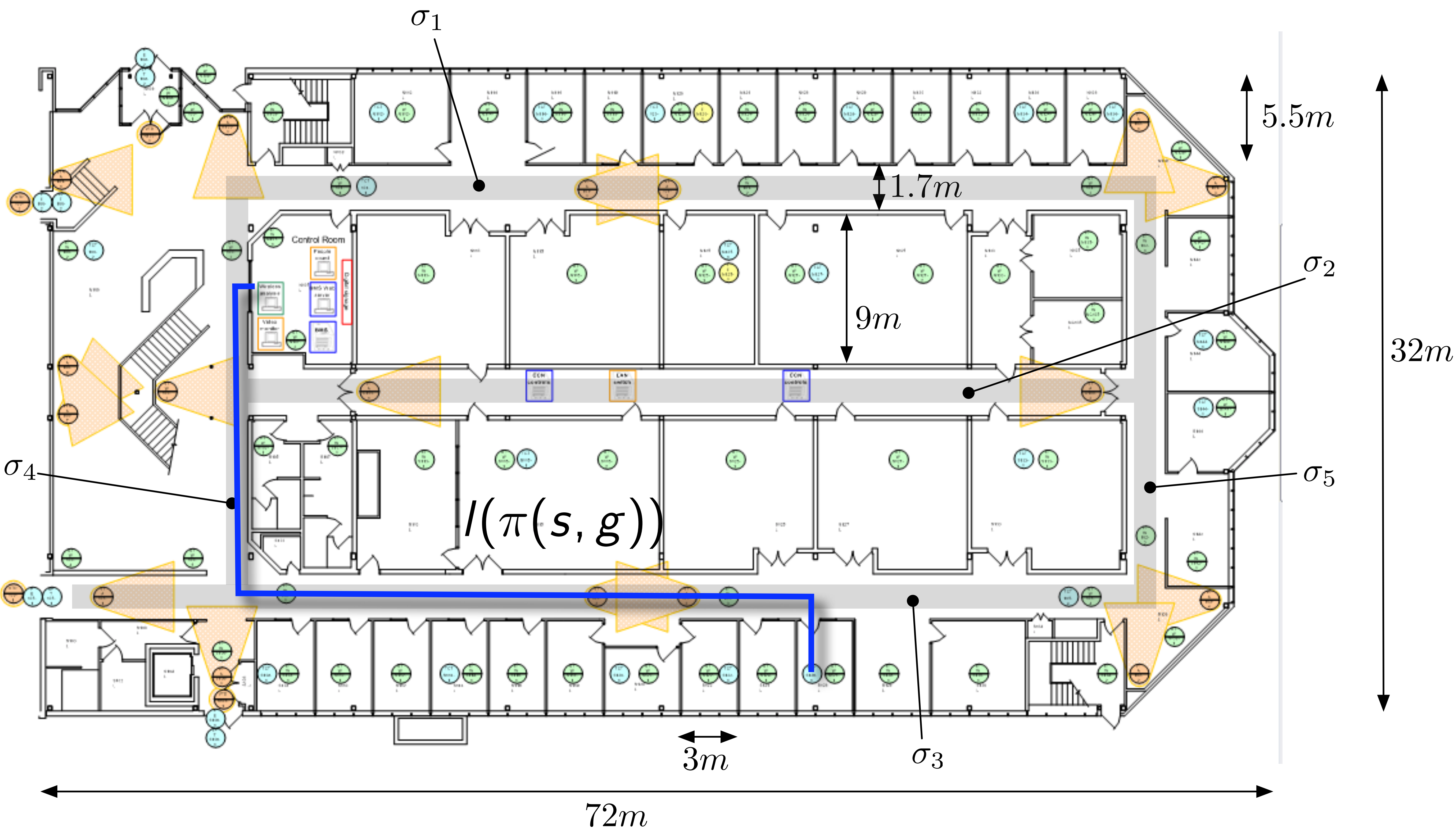




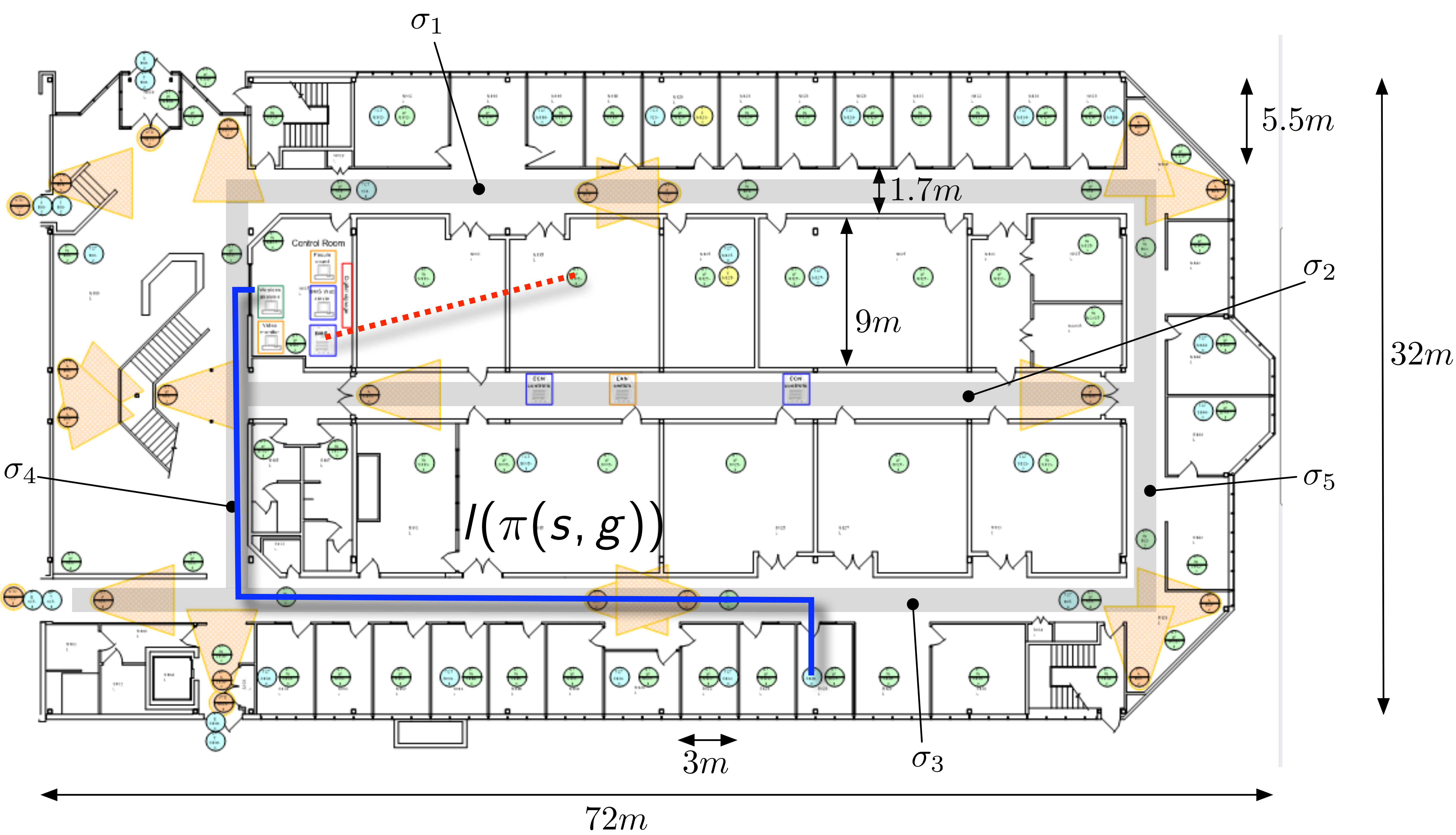
The Physical Aspect of the Problem



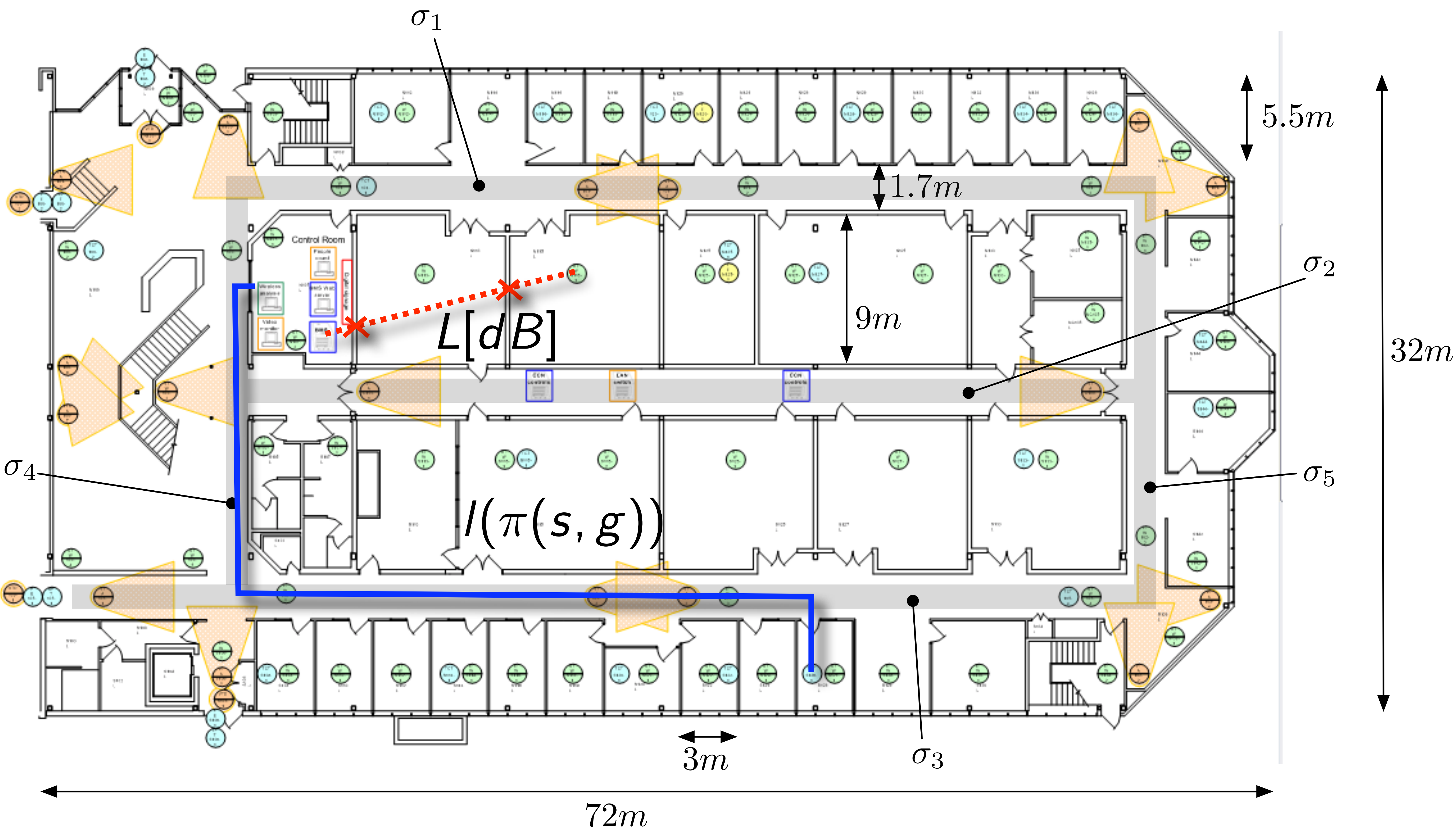
The Physical Aspect of the Problem



The Physical Aspect of the Problem

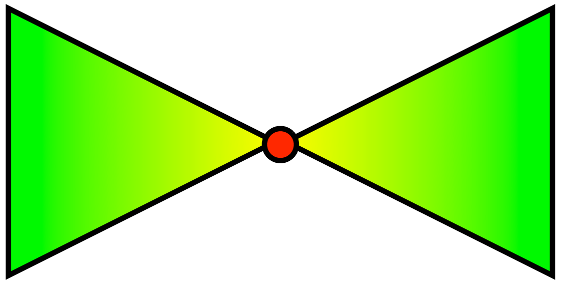


The Physical Aspect of the Problem

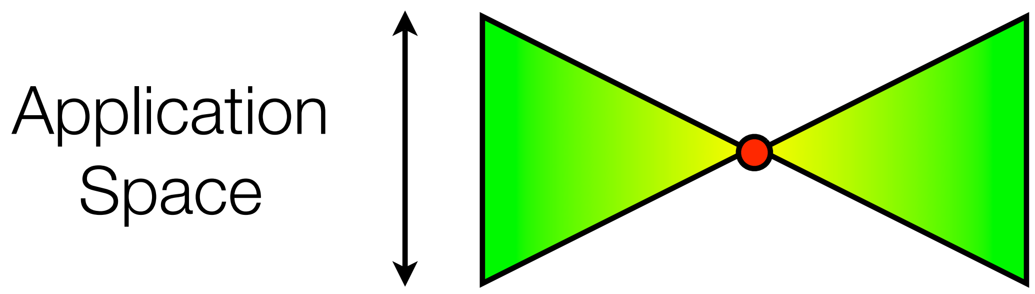


Platform-Based Design

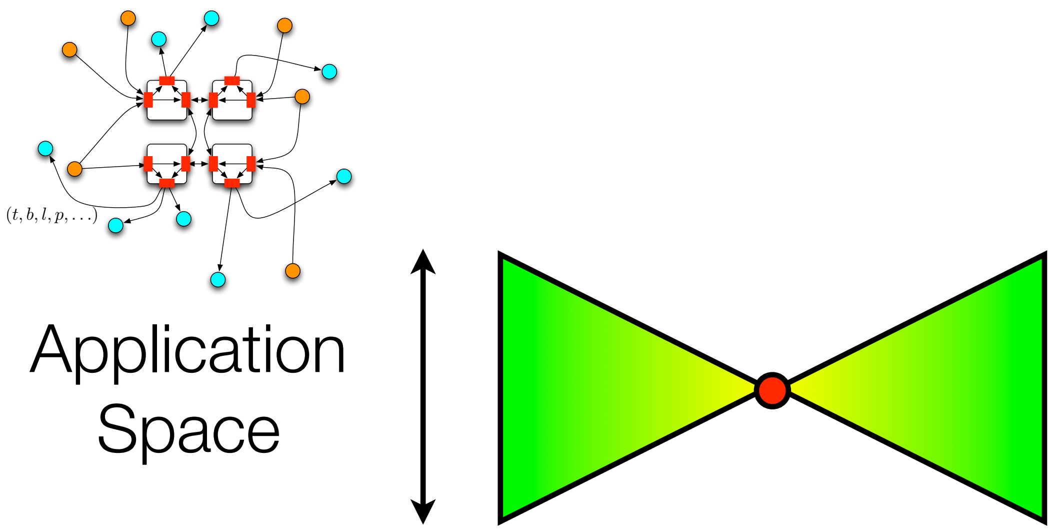
Platform-Based Design



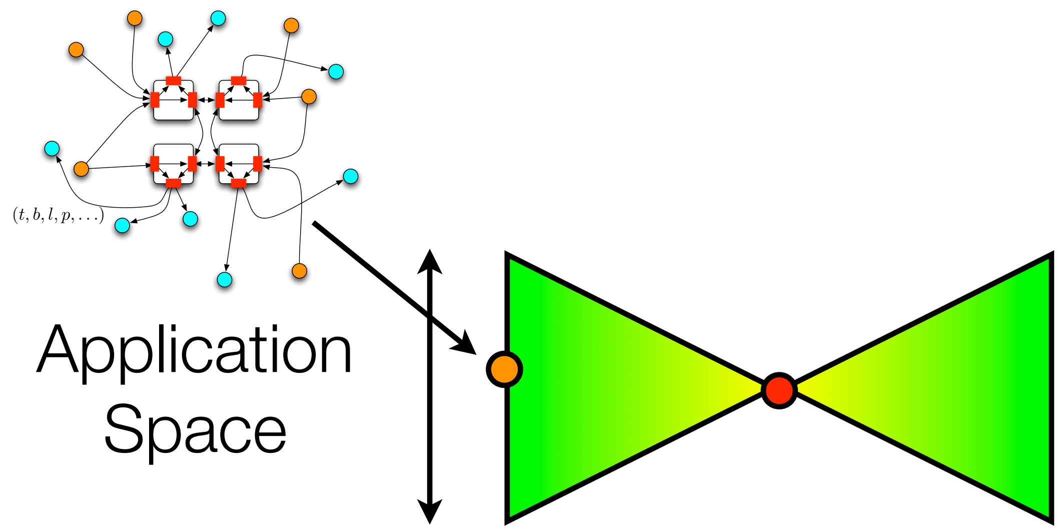
Platform-Based Design



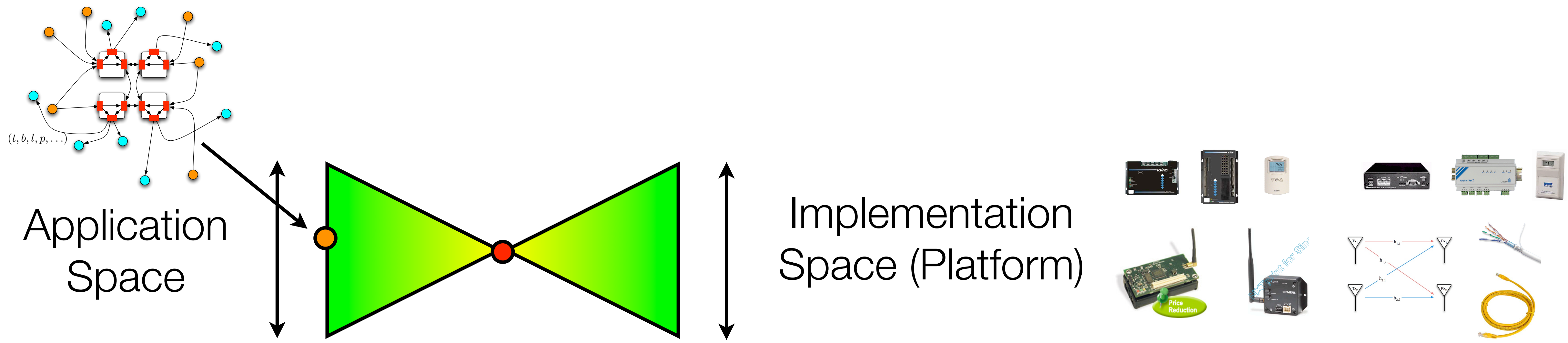
Platform-Based Design



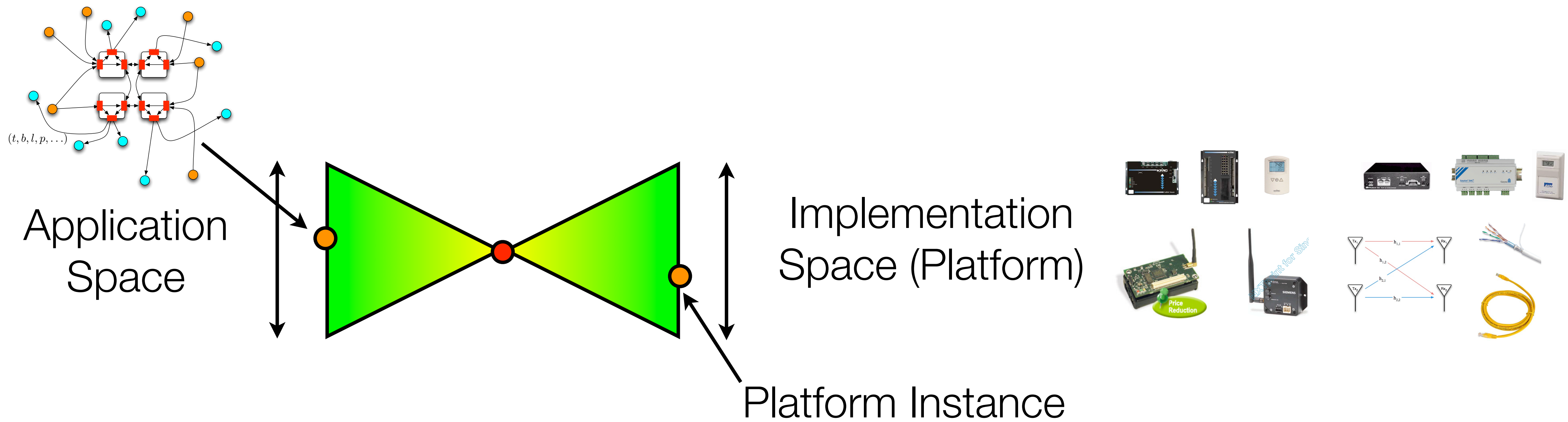
Platform-Based Design



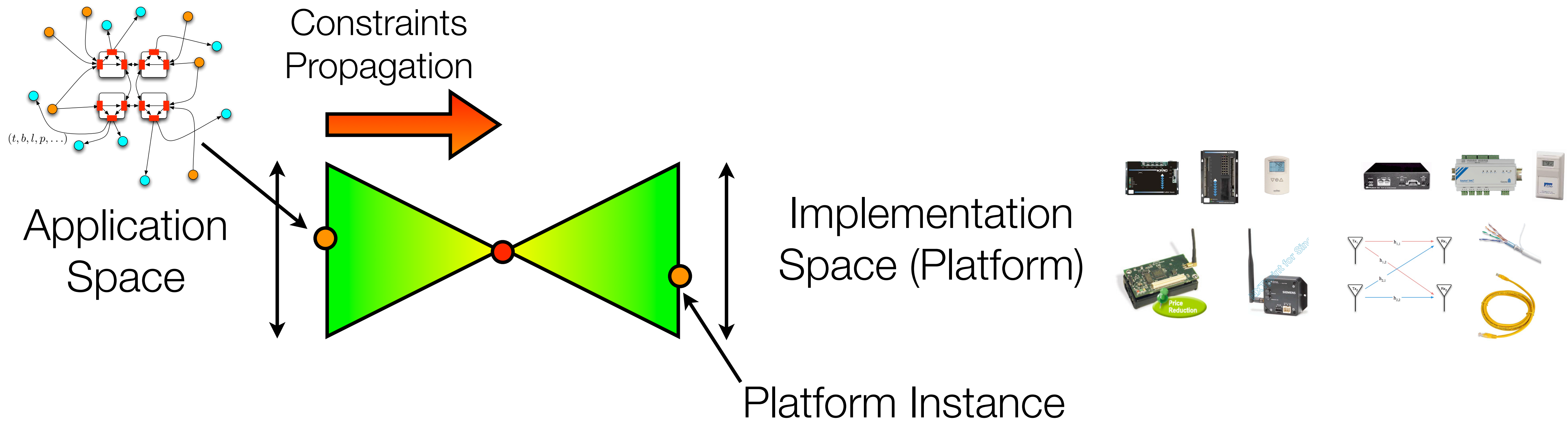
Platform-Based Design



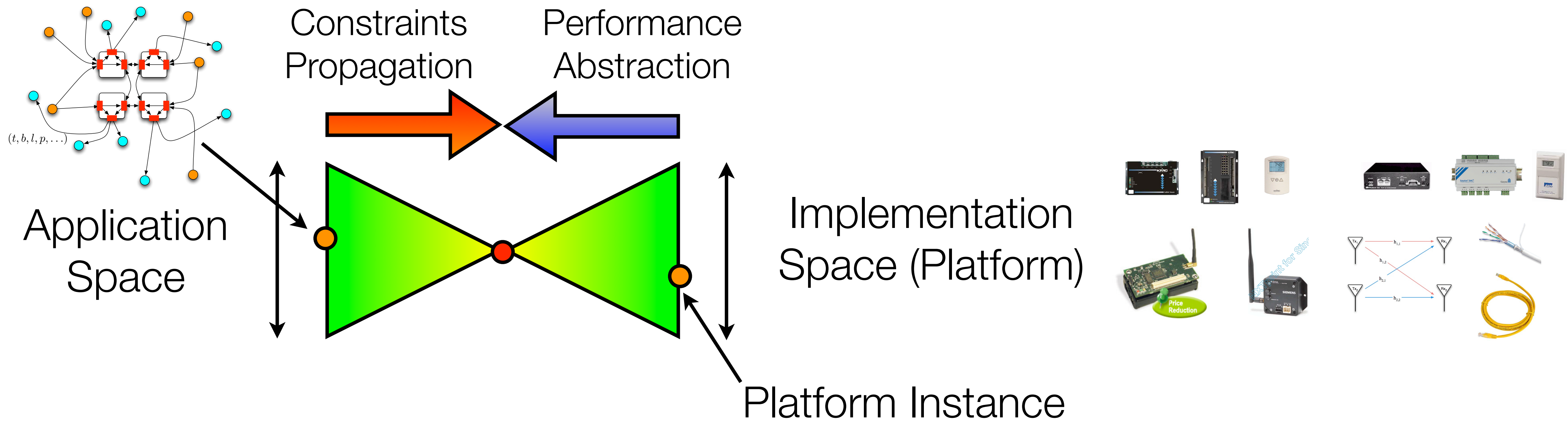
Platform-Based Design



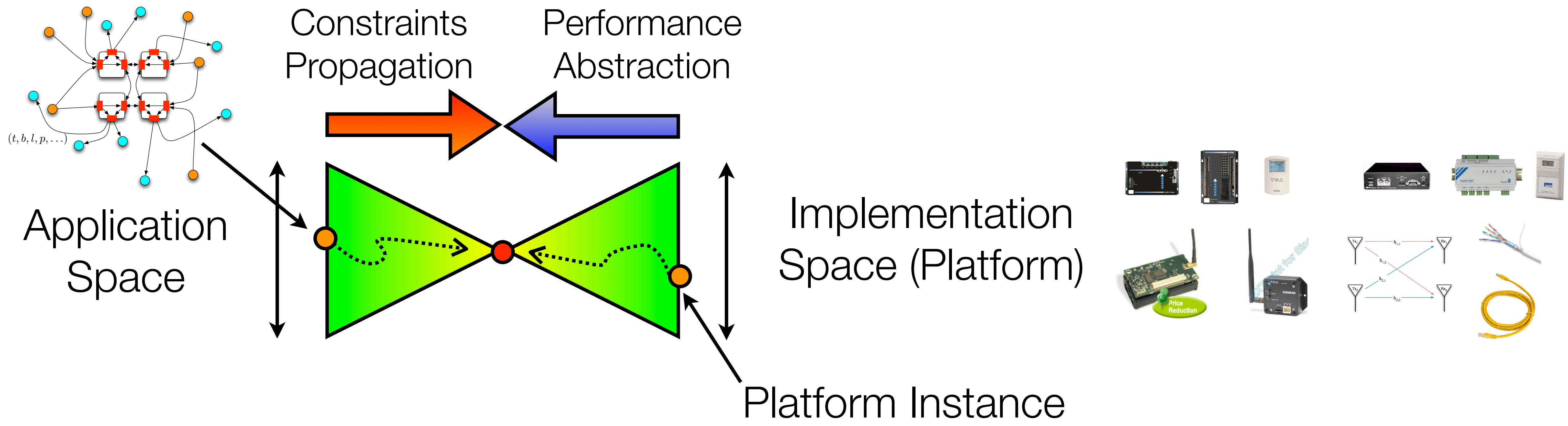
Platform-Based Design



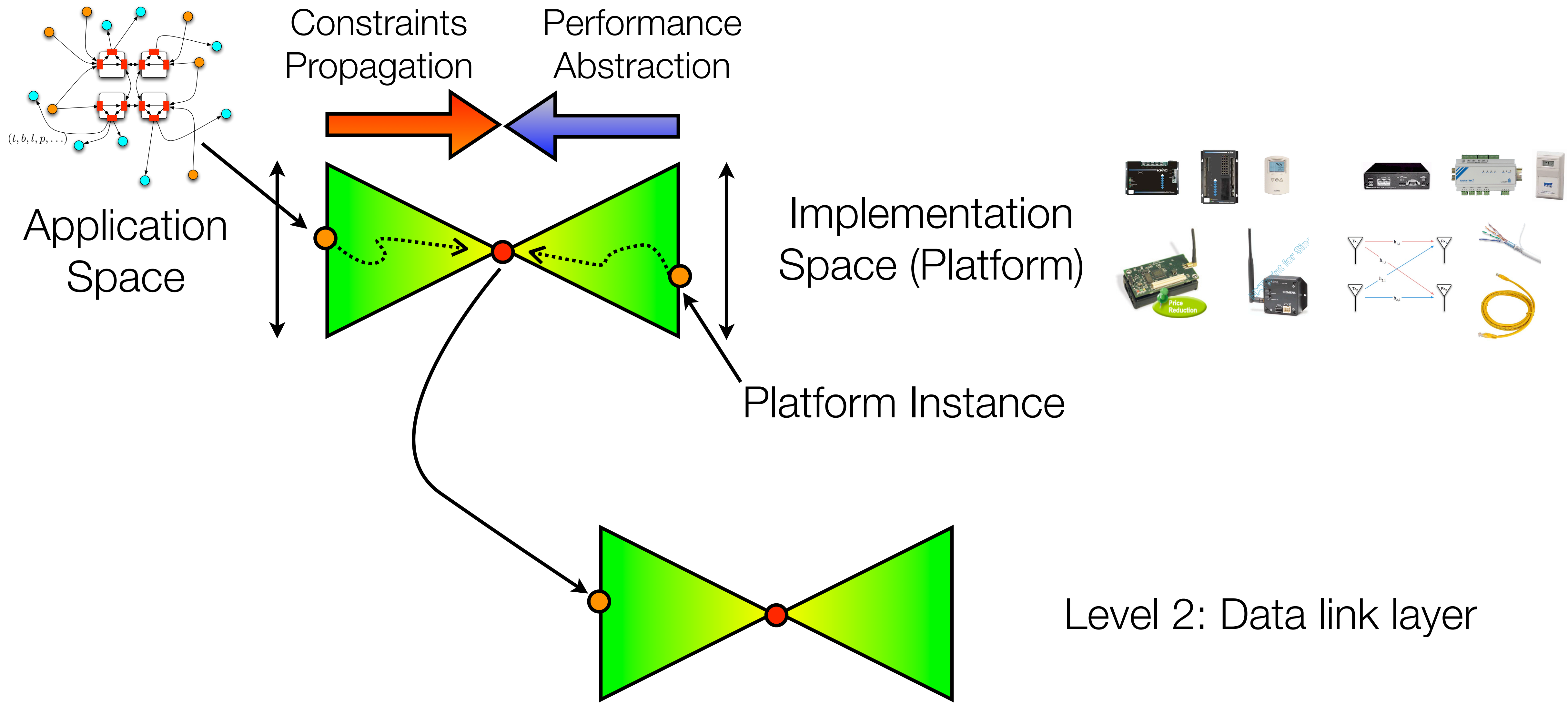
Platform-Based Design



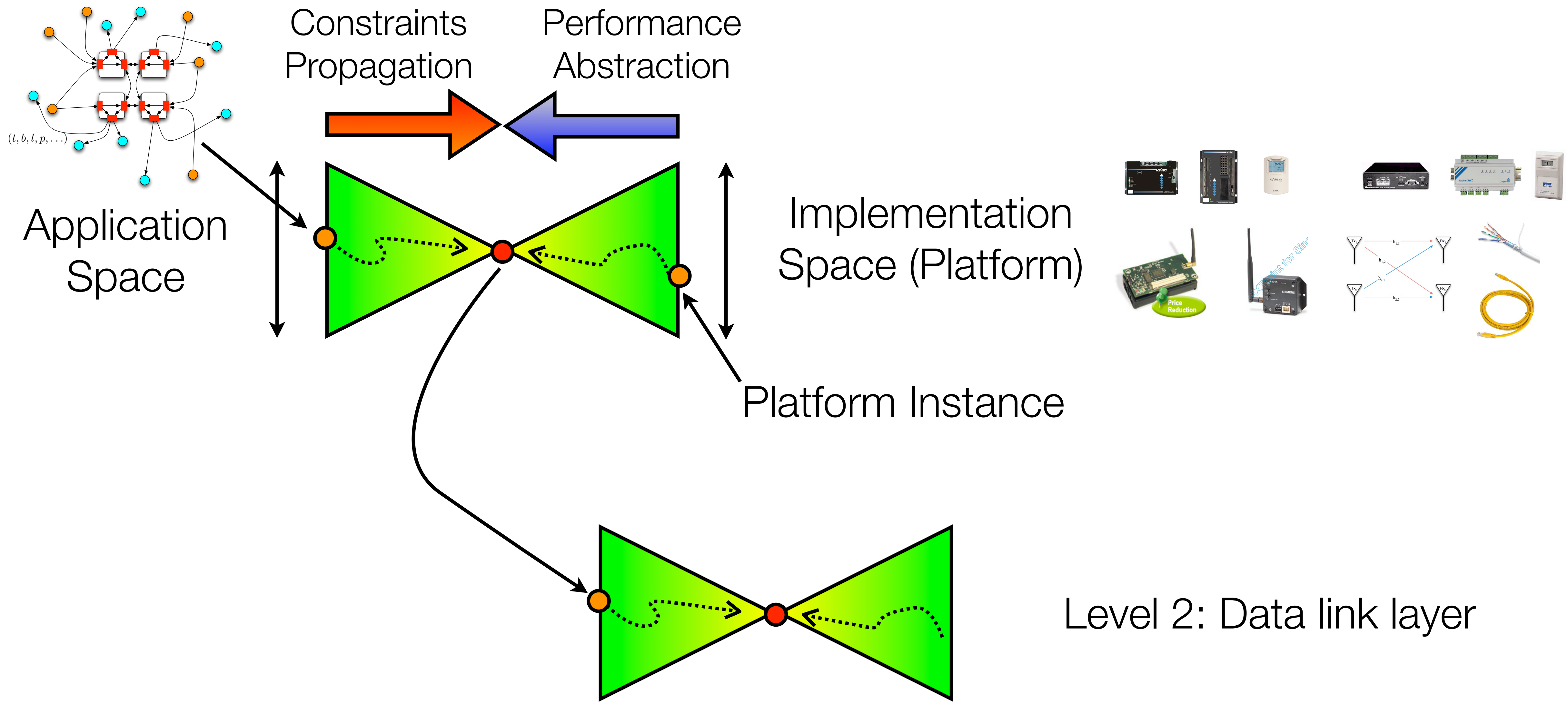
Platform-Based Design



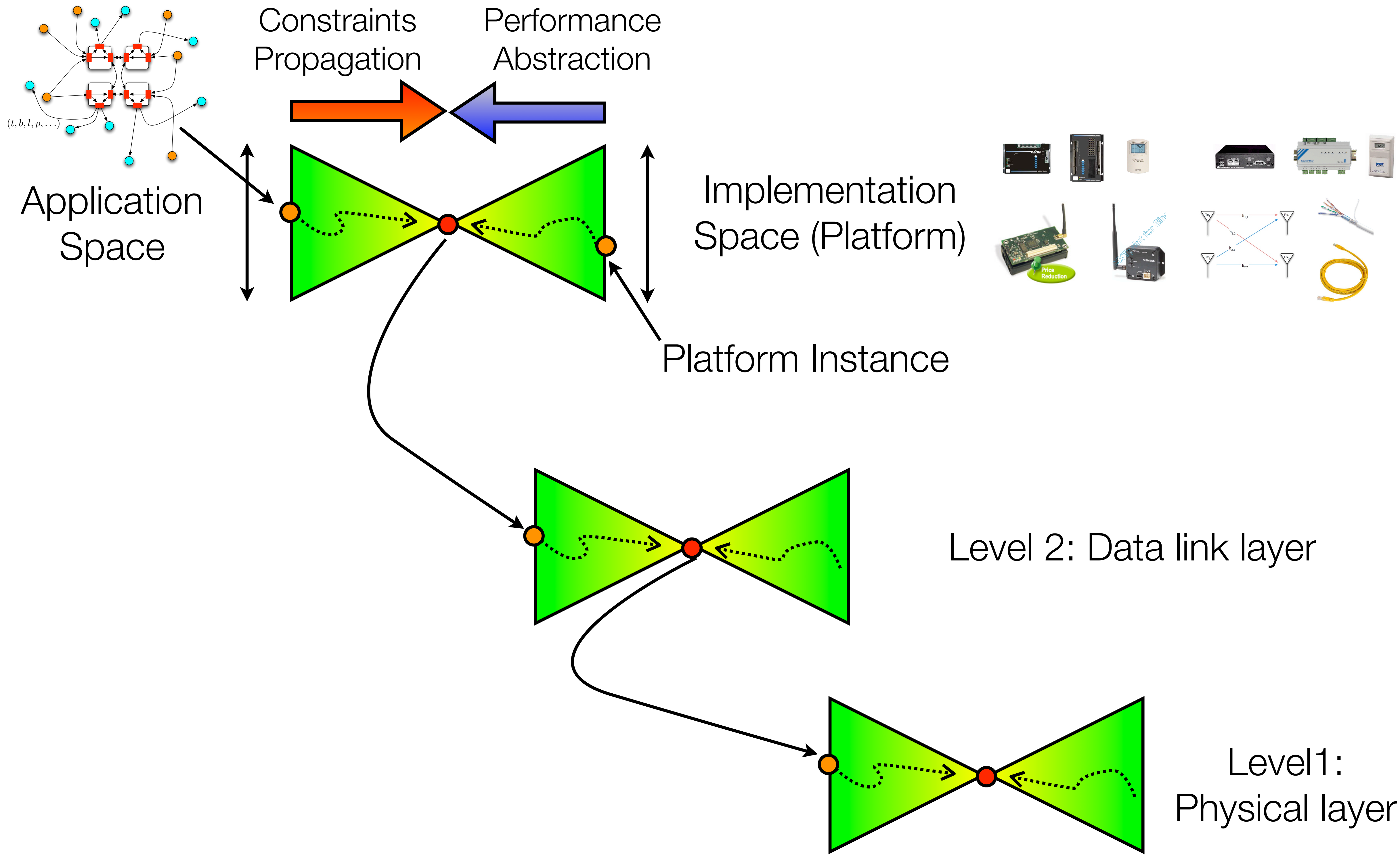
Platform-Based Design



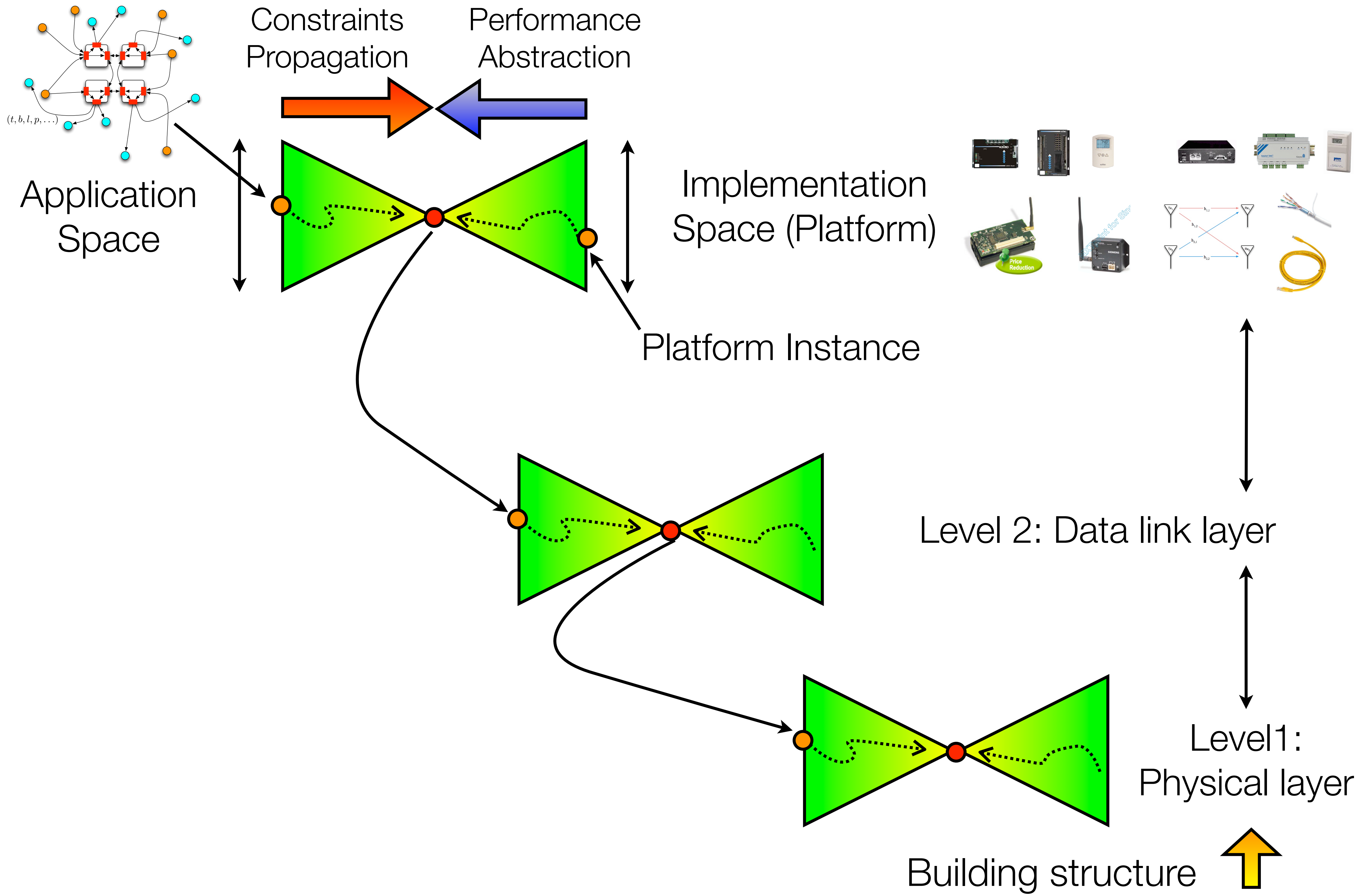
Platform-Based Design



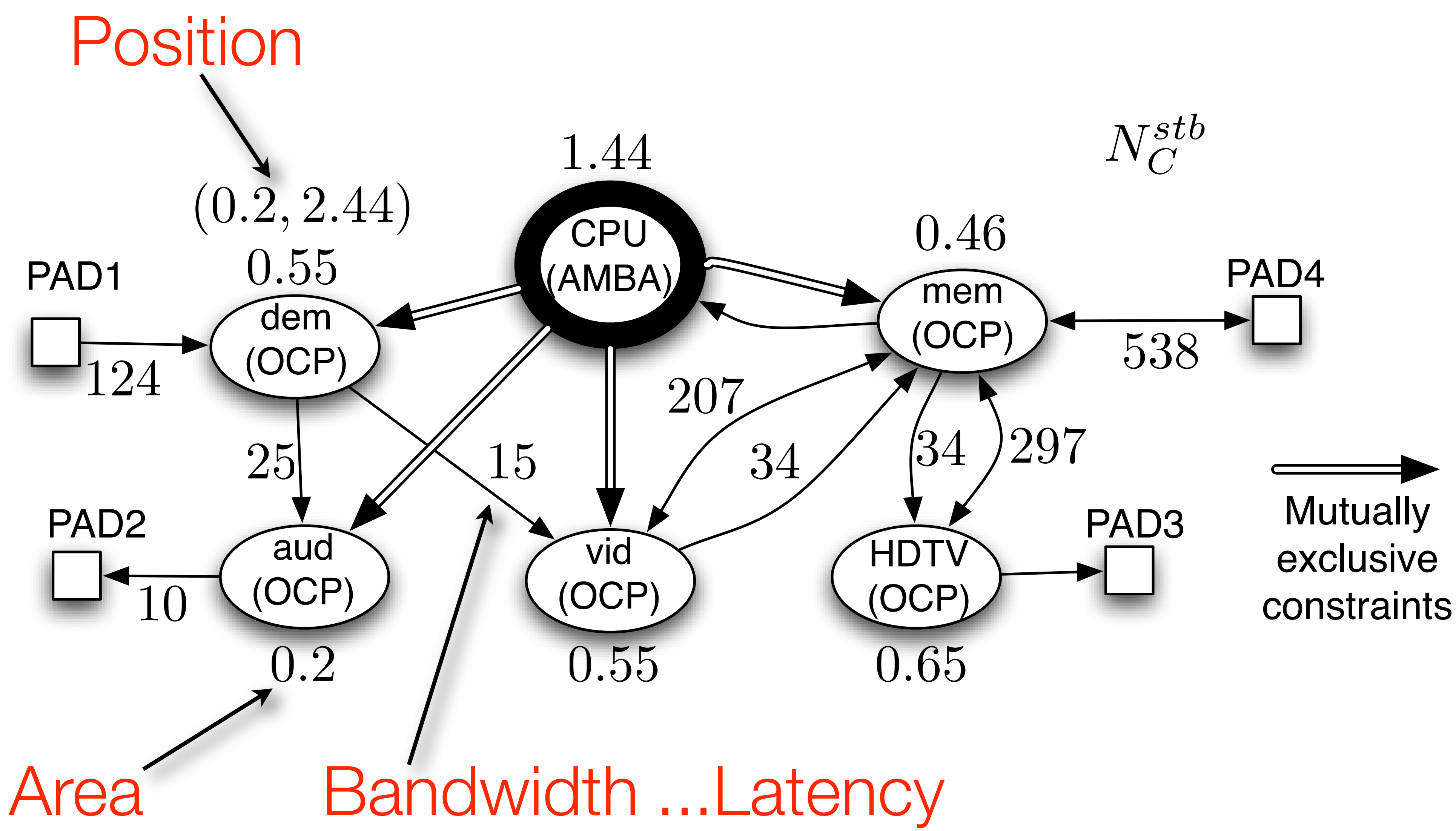
Platform-Based Design



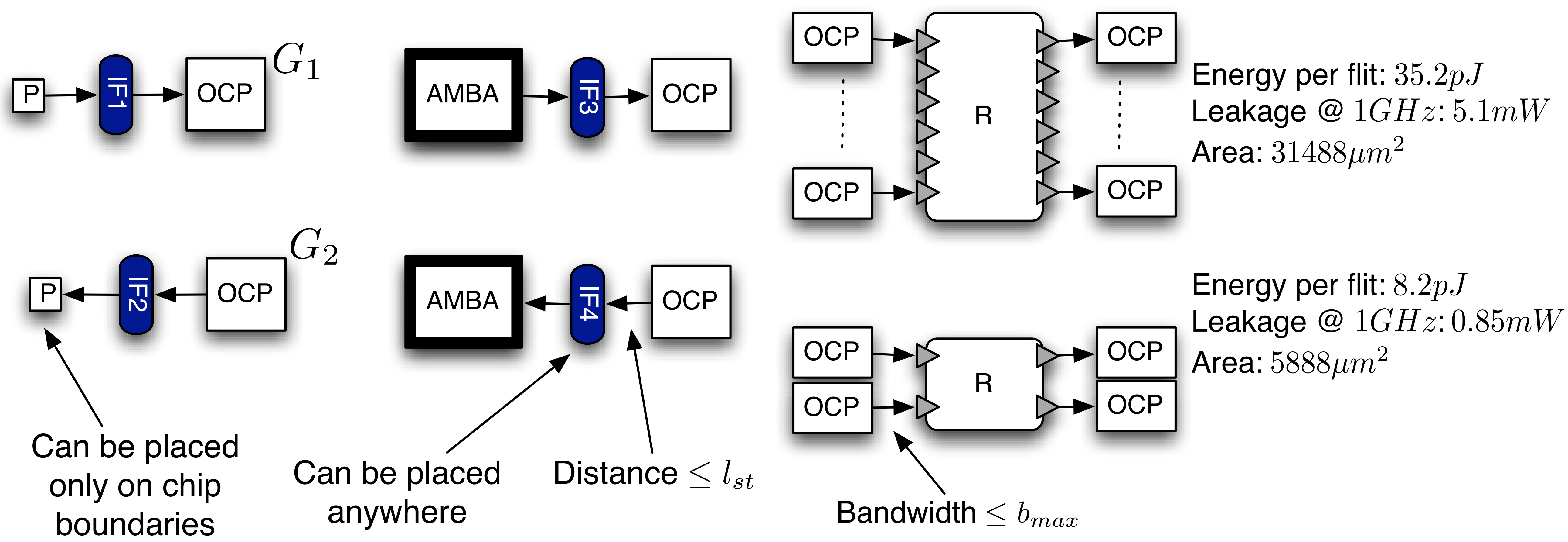
Platform-Based Design



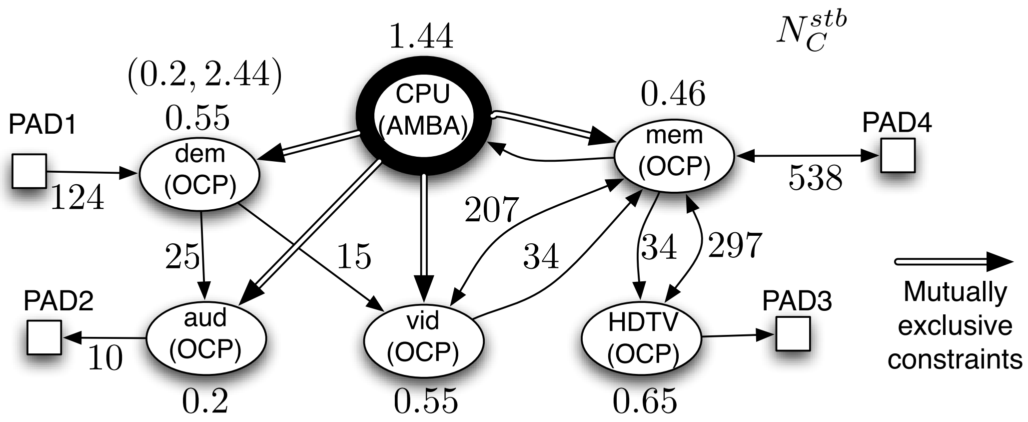
The On-Chip Communication Specification

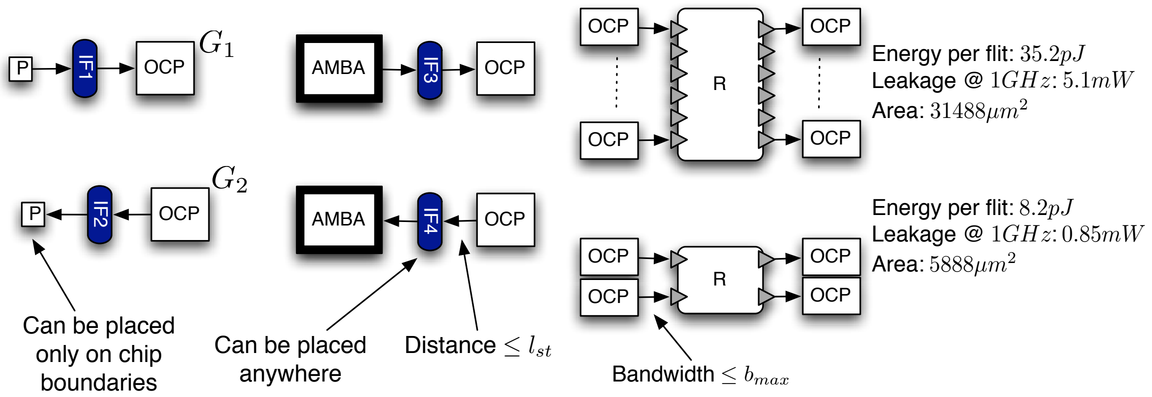


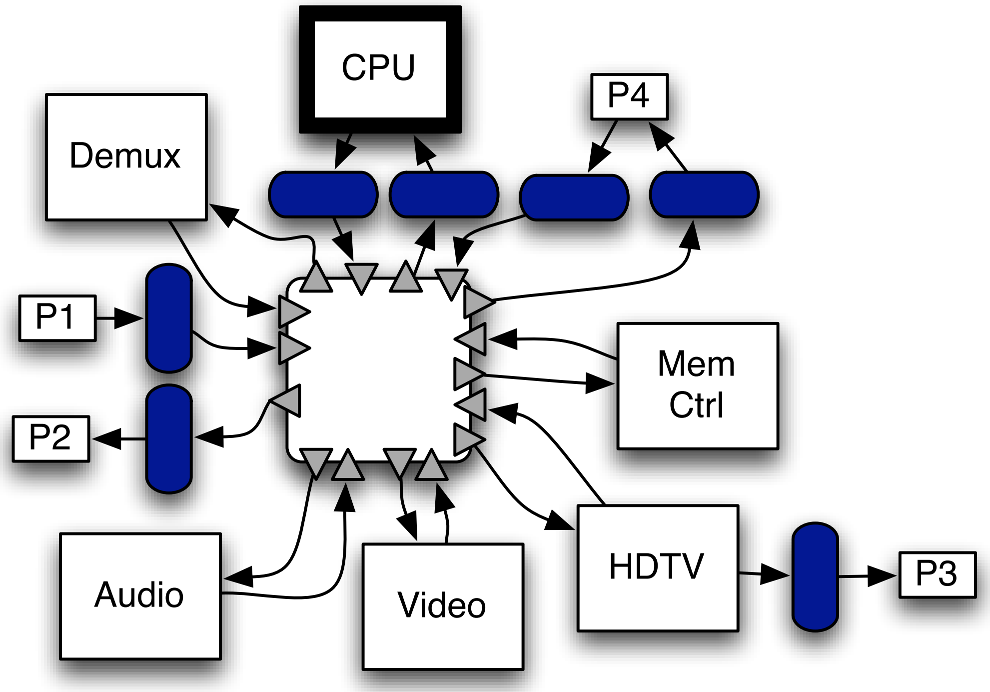
The On-Chip Communication Platform



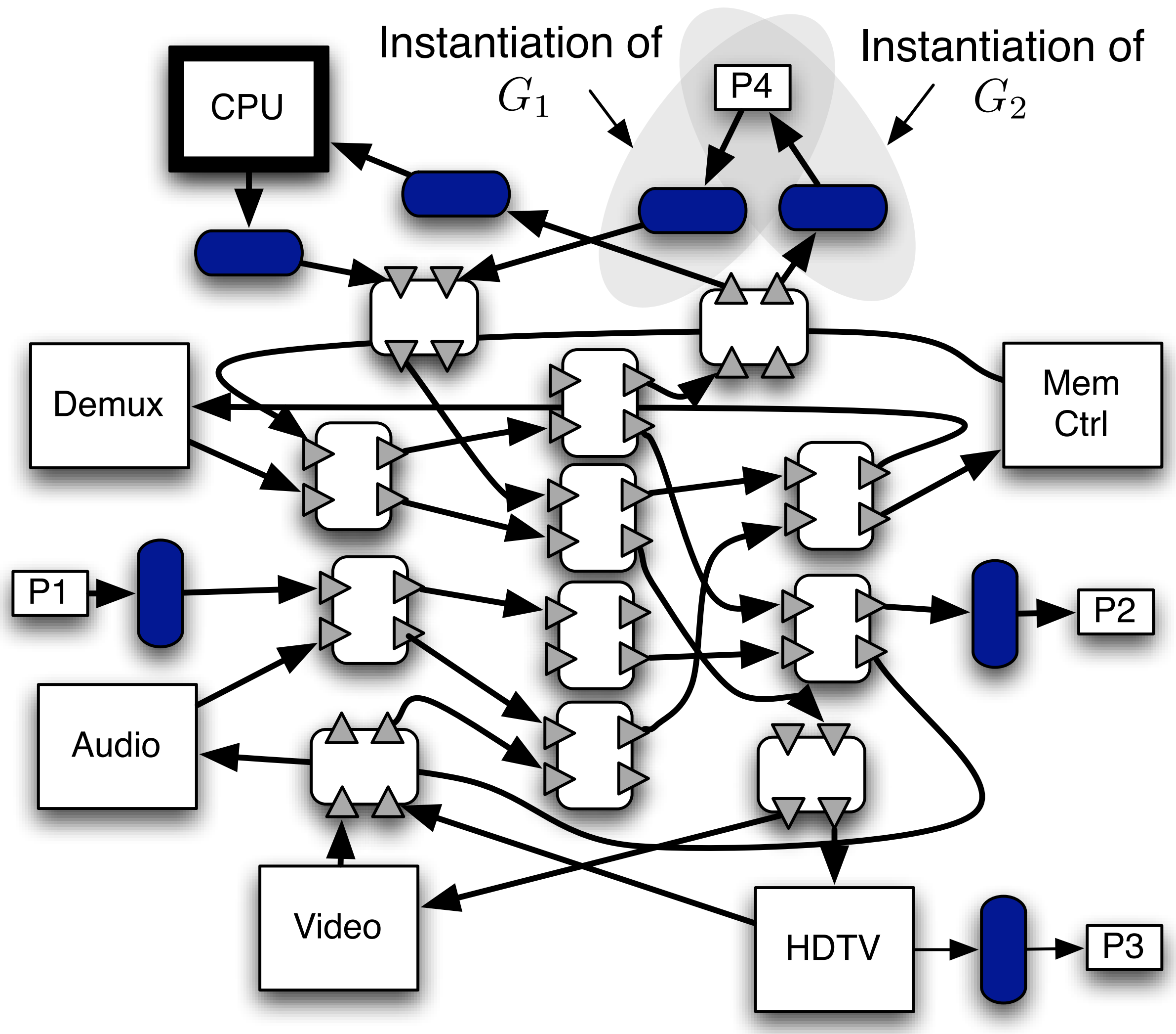
Example of Platform Instances







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

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

$$V \cup E$$

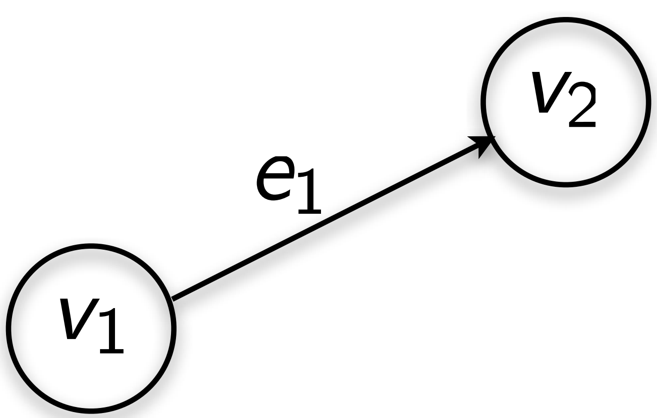
Components

Communication Structures

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

$V \cup E$

Components

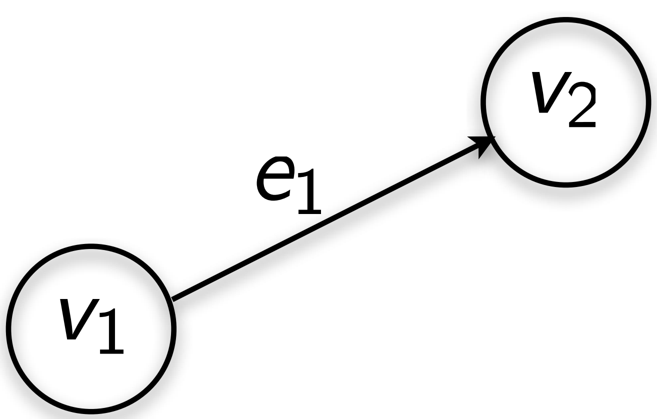


Communication Structures

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

$V \cup E$
Components

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



Communication Structures

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

$$V \cup E$$

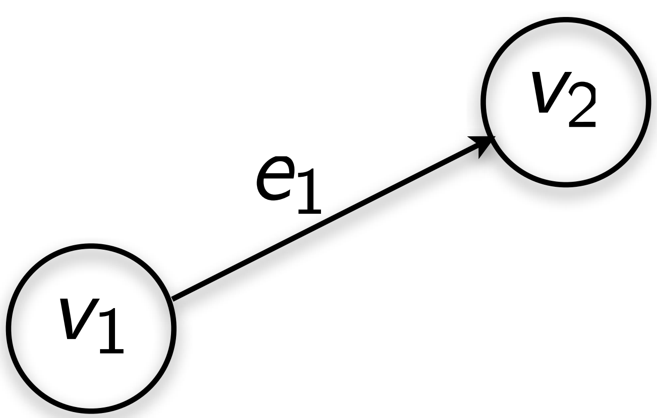
Components

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

Quantities

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

Domain



Communication Structures

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

$V \cup E$

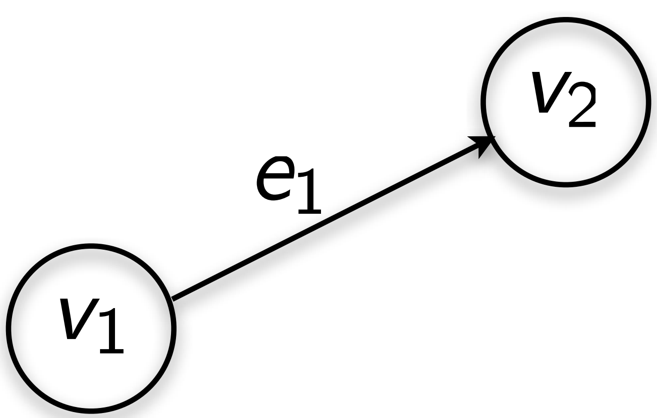
Components

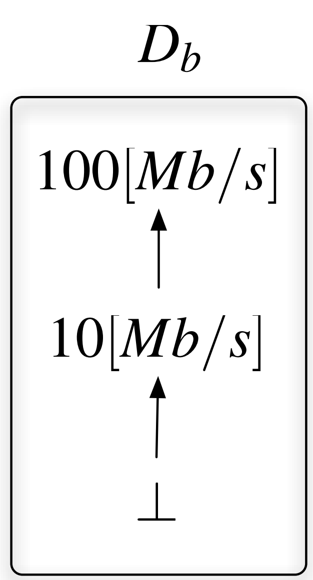
$\{x, y, b\}$

Quantities

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

Domain





Communication Structures

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

$V \cup E$

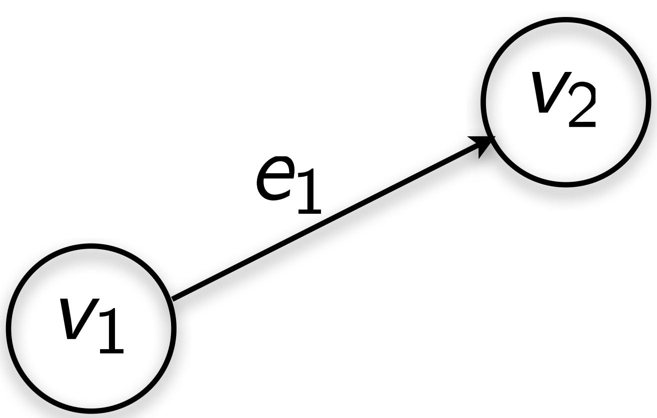
Components

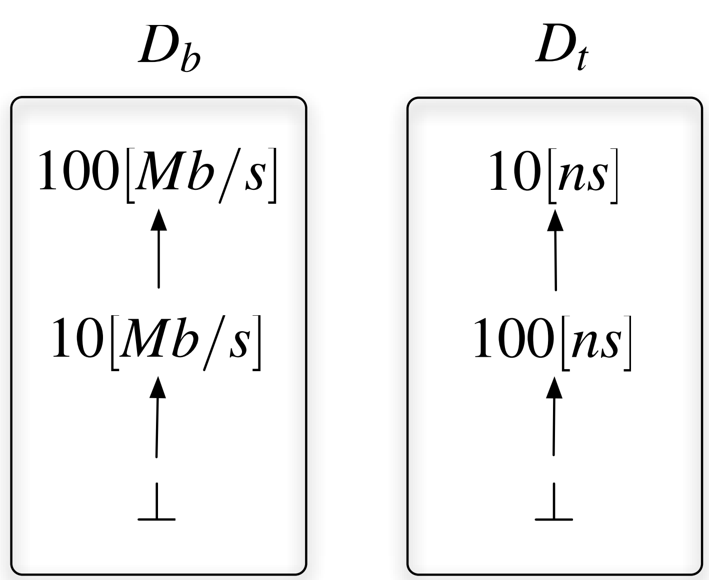
$\{x, y, b\}$

Quantities

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

Domain





Communication Structures

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

$V \cup E$

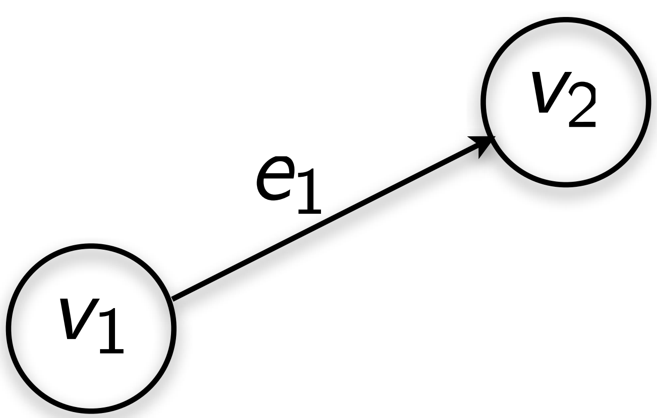
Components

$\{x, y, b\}$

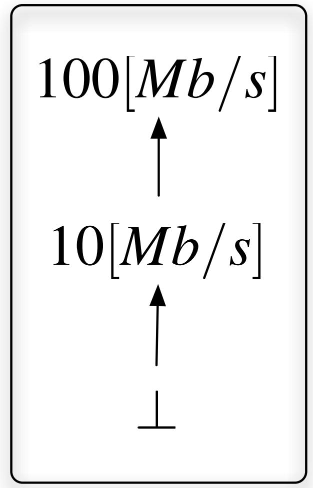
Quantities

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

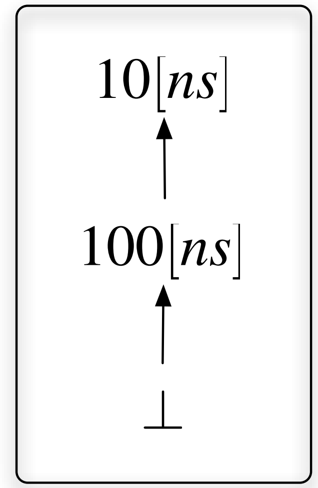
Domain



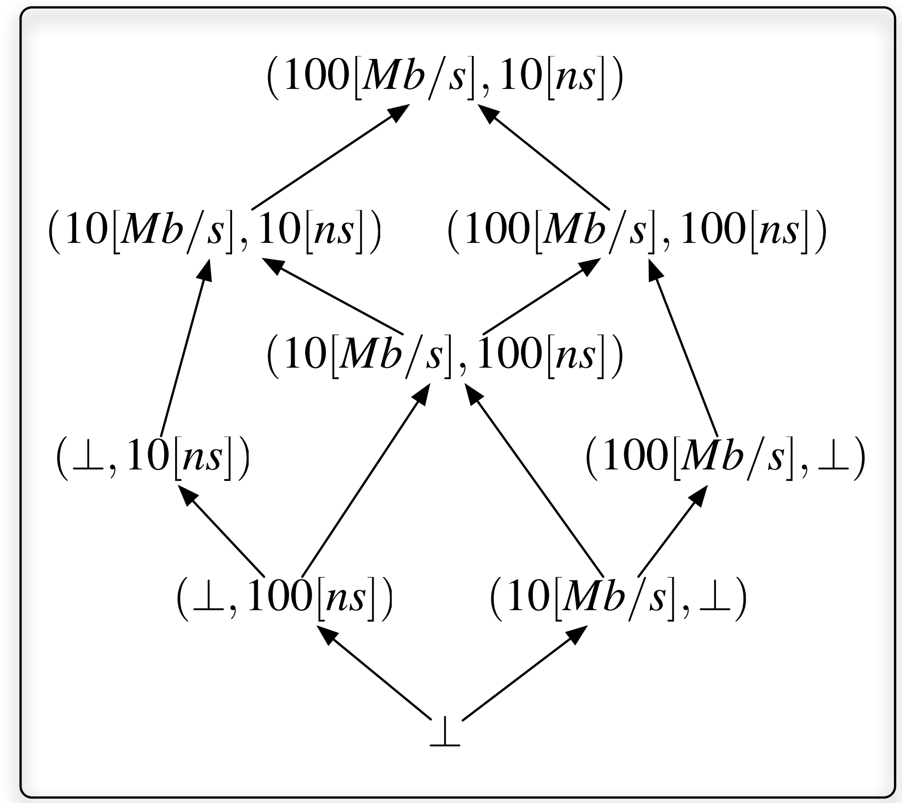
D_b



D_t



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



Communication Structures

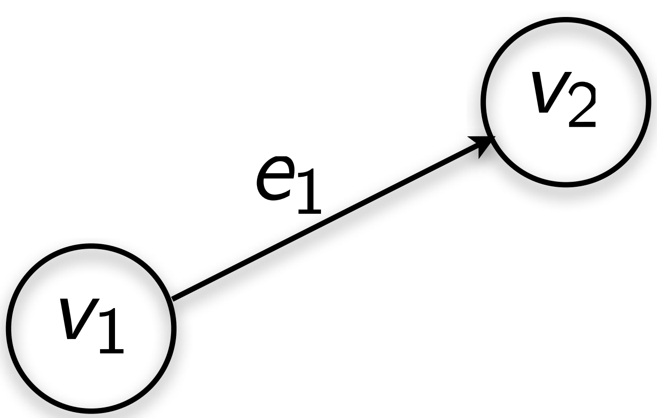
$$N(\mathcal{C}, Q, L)$$

$V \cup E$
Components

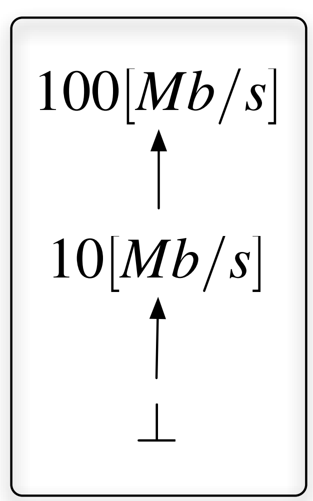
$\{x, y, b\}$
Quantities

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

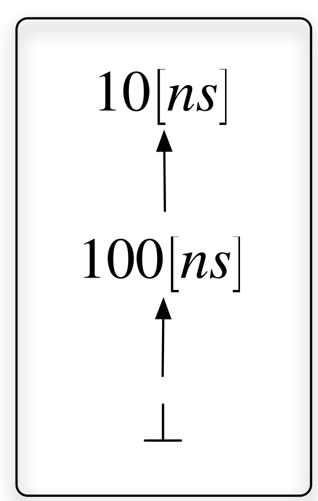
$D_Q = D_x \times D_y \times D_b$
Domain



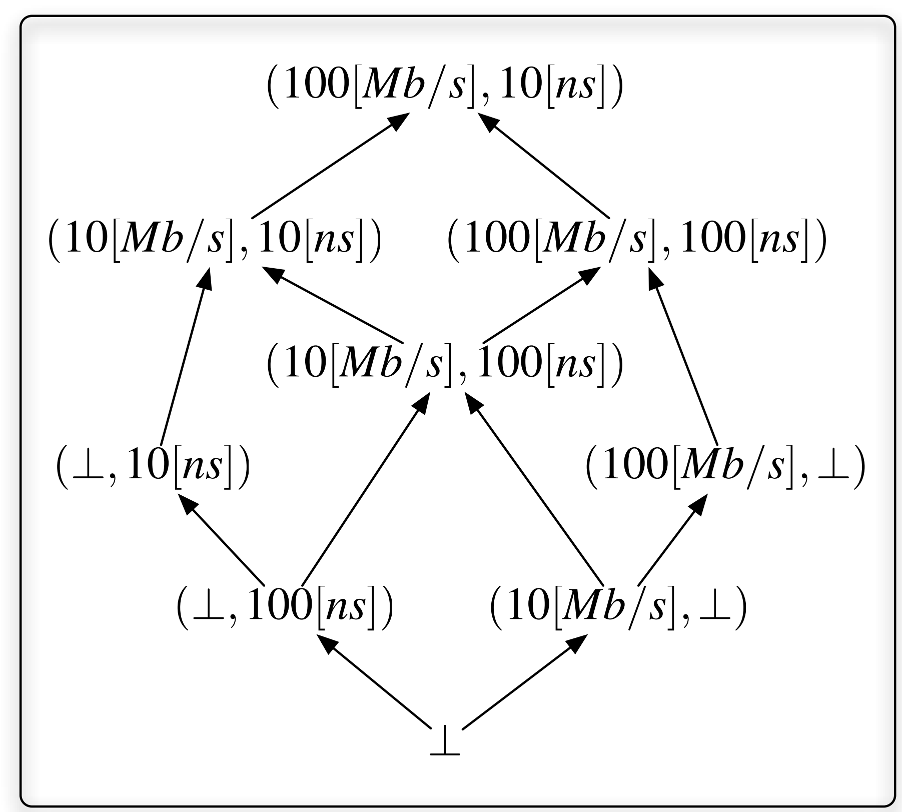
D_b



D_t



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



Communication Structures

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

$V \cup E$

Components

$\{x, y, b\}$

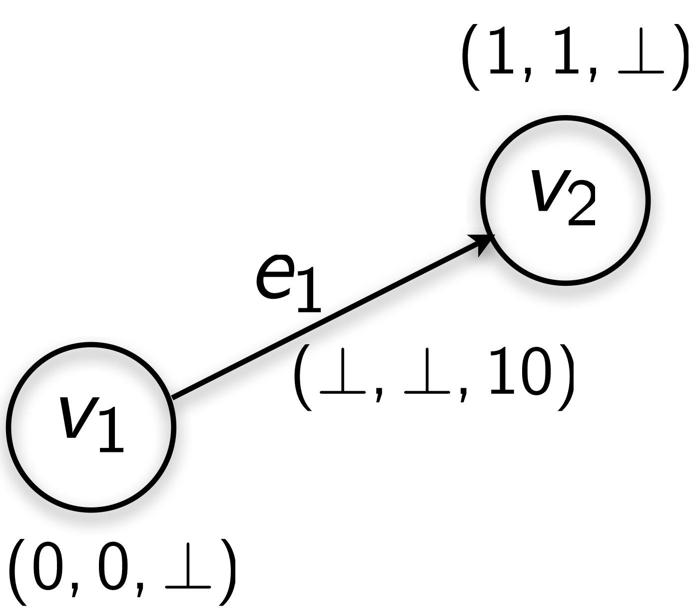
Quantities

$D_Q = D_x \times D_y \times D_b$

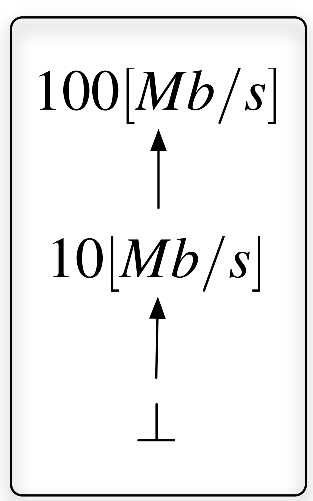
Domain

$\{l | l : \mathcal{C} \rightarrow D_Q\}$

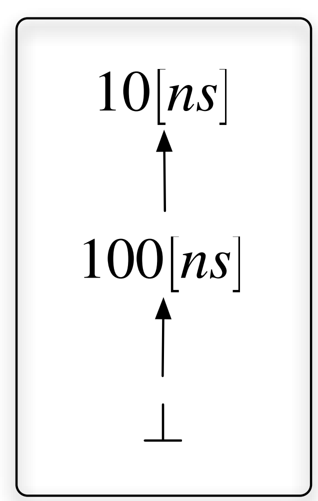
Configurations



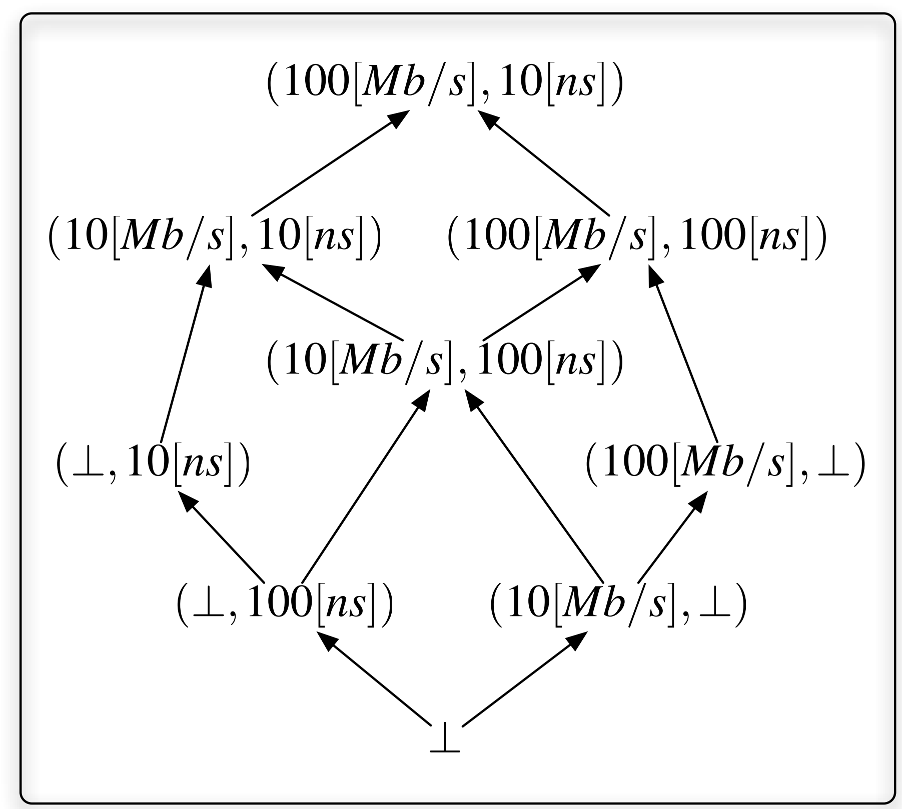
D_b



D_t



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



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

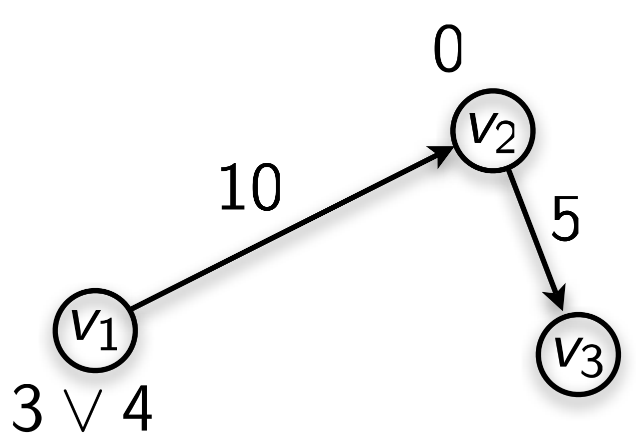
$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

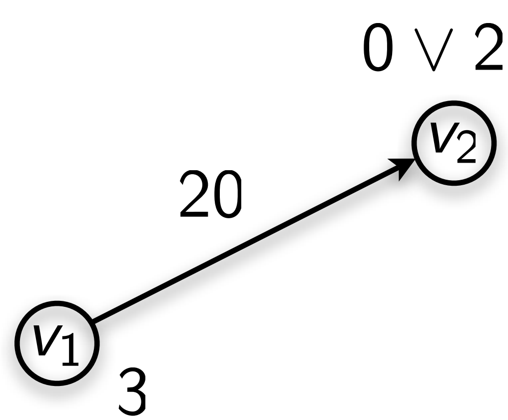
$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad\qquad L_1 \oplus_Q L_2$$

Composition

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

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

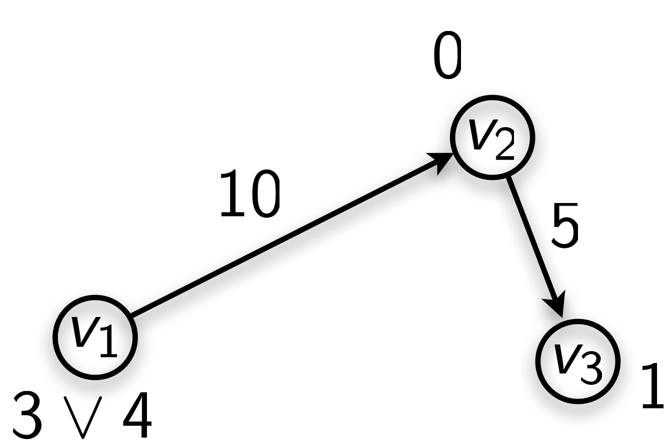


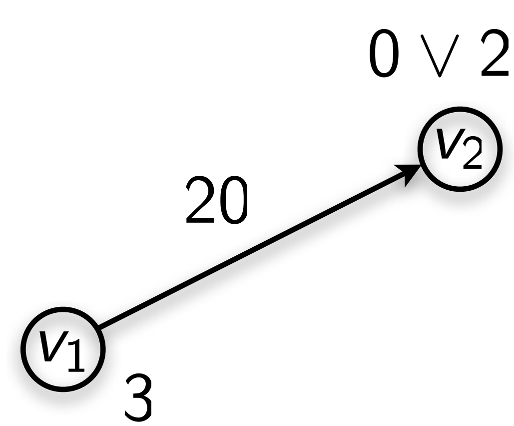


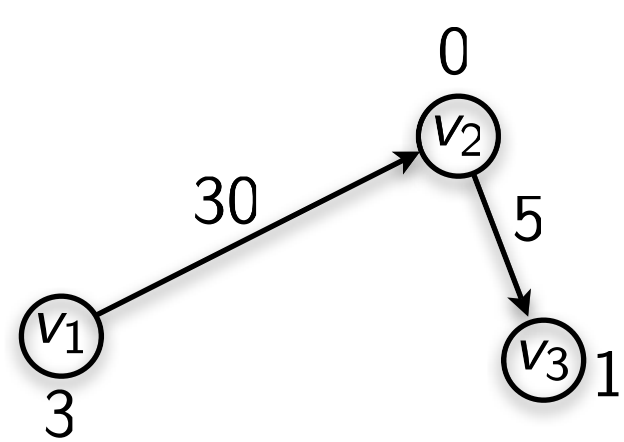
Composition

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

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





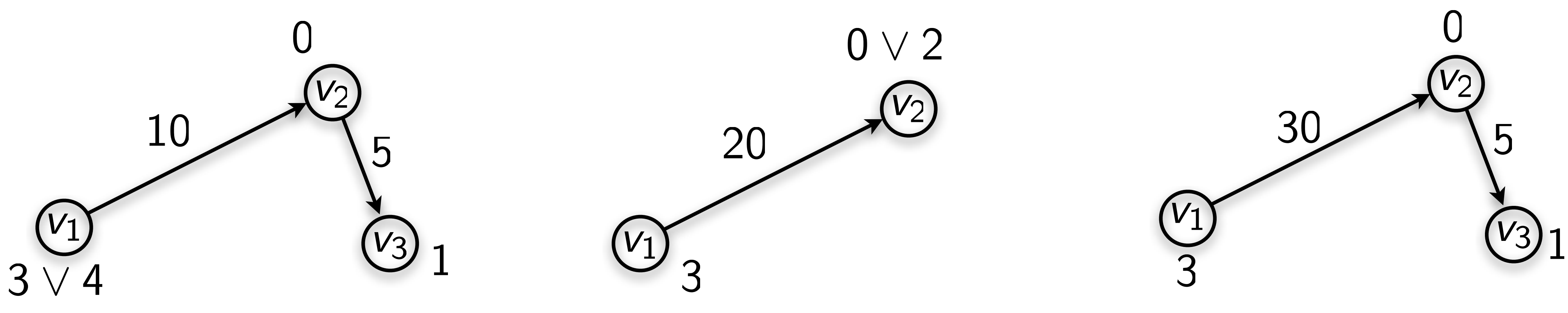


Composition

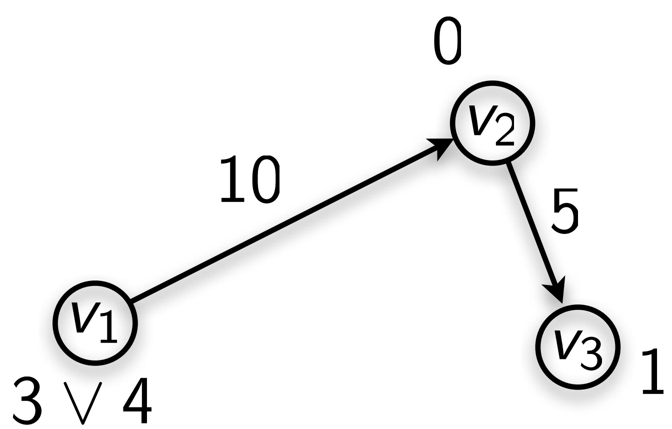
$$|E| = |V| - 1$$

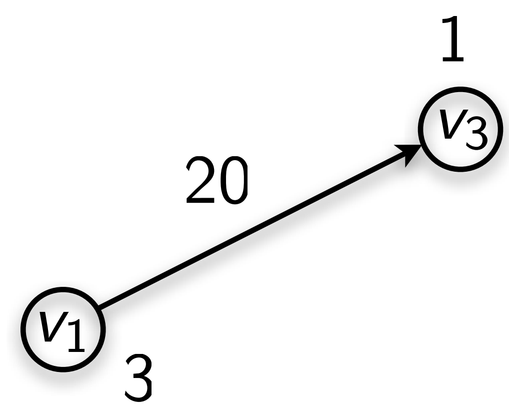
$$N(\mathcal{C}, Q, L) = N_1(\mathcal{C}_1, Q, L_1) \|_Q^{\mathcal{R}} N_2(\mathcal{C}_2, Q, L_2)$$

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



Composition

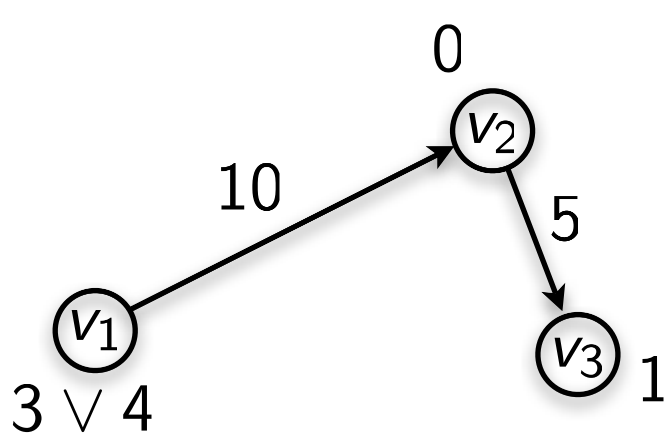


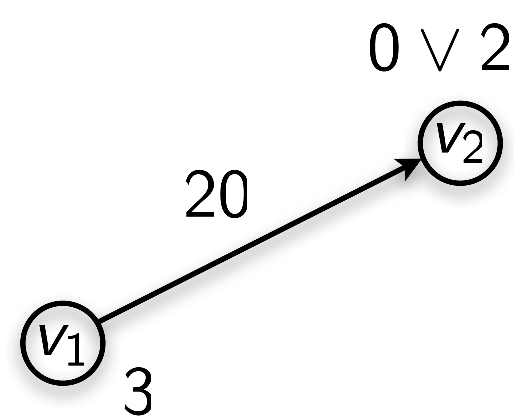


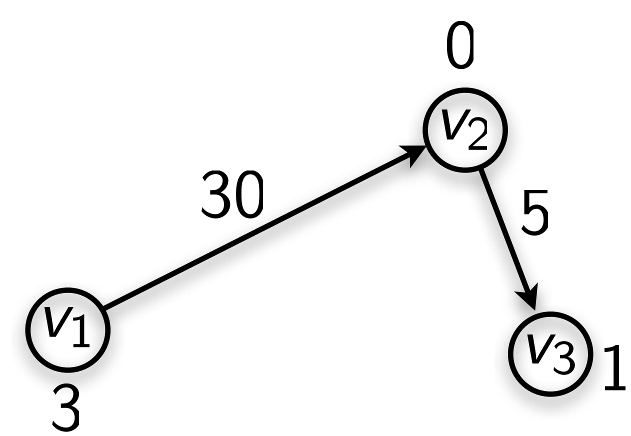
$$|E| = |V| - 1$$

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

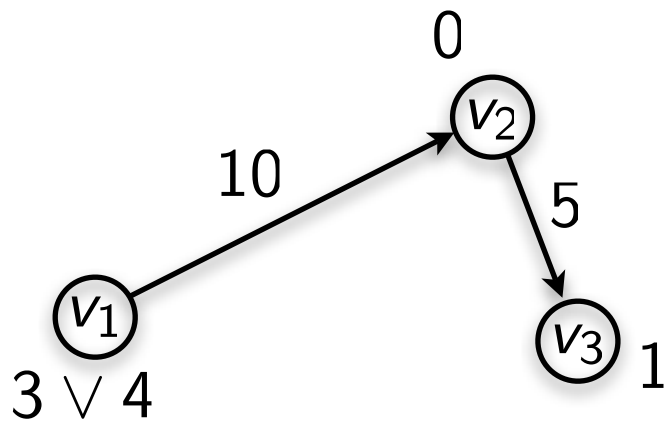
$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad L_1 \oplus_Q L_2$$

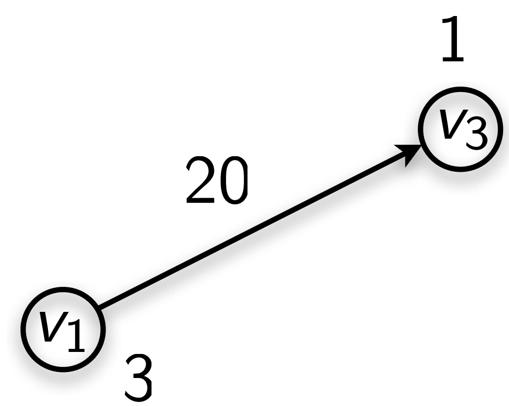


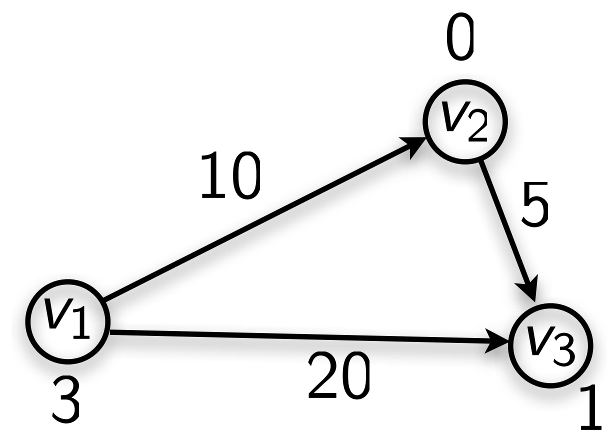




Composition



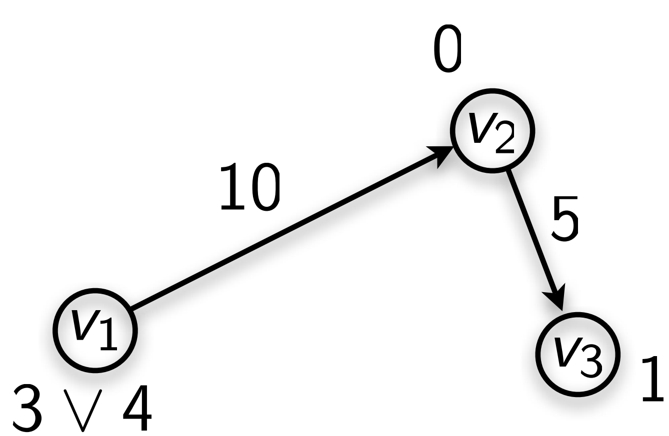


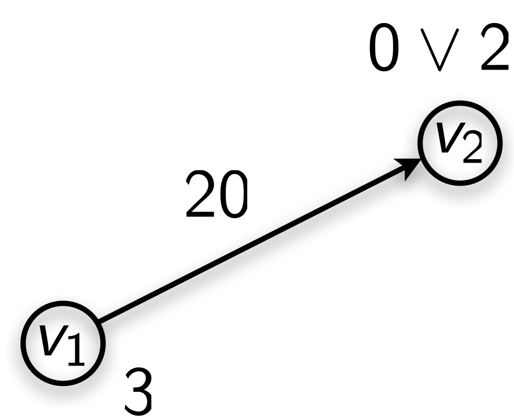


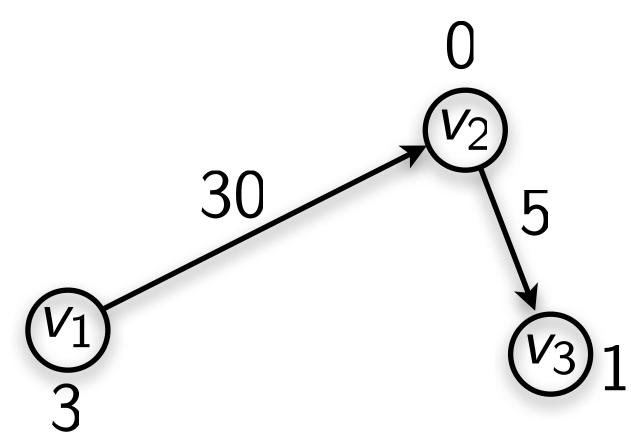
$$|E| = |V| - 1$$

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

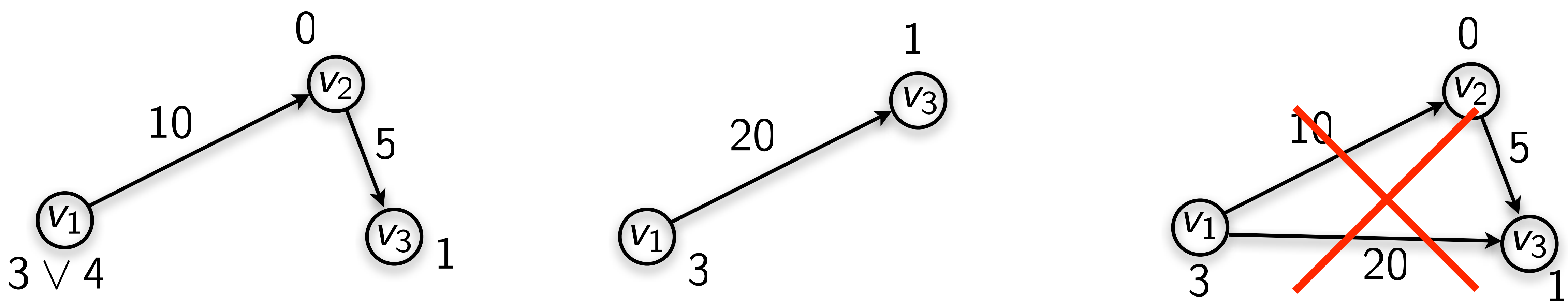
$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad L_1 \oplus_Q L_2$$







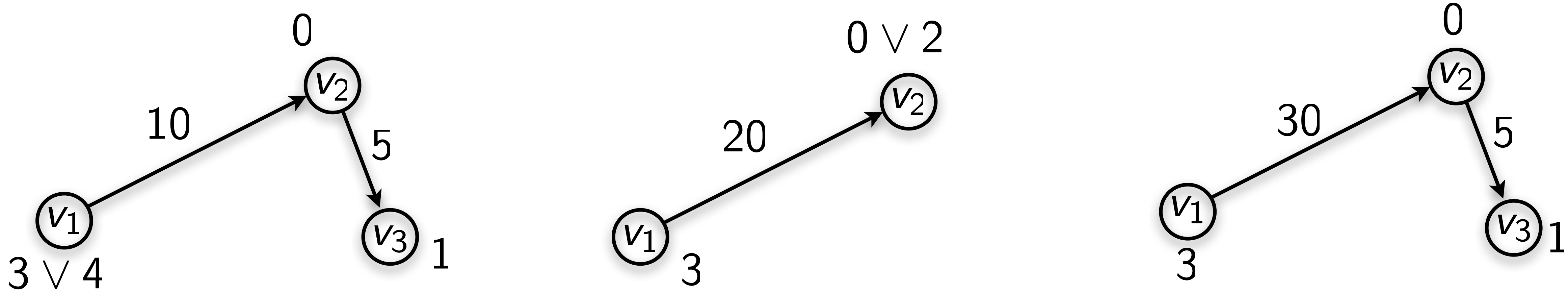
Composition



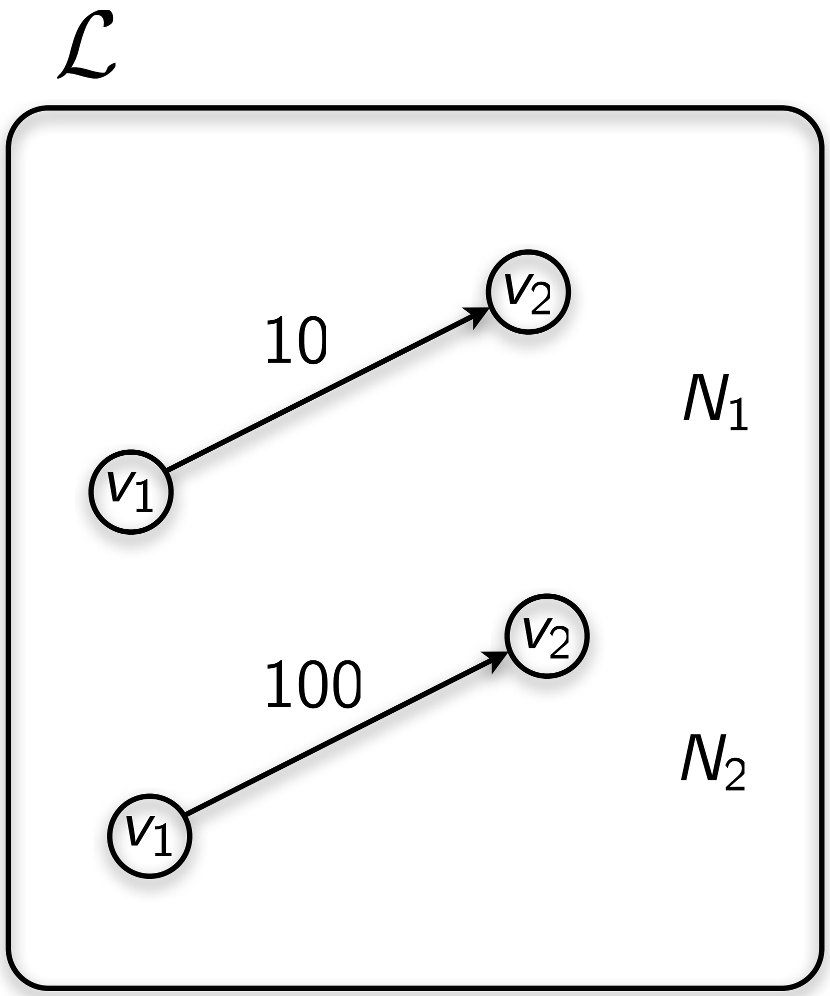
$$|E| = |V| - 1$$

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

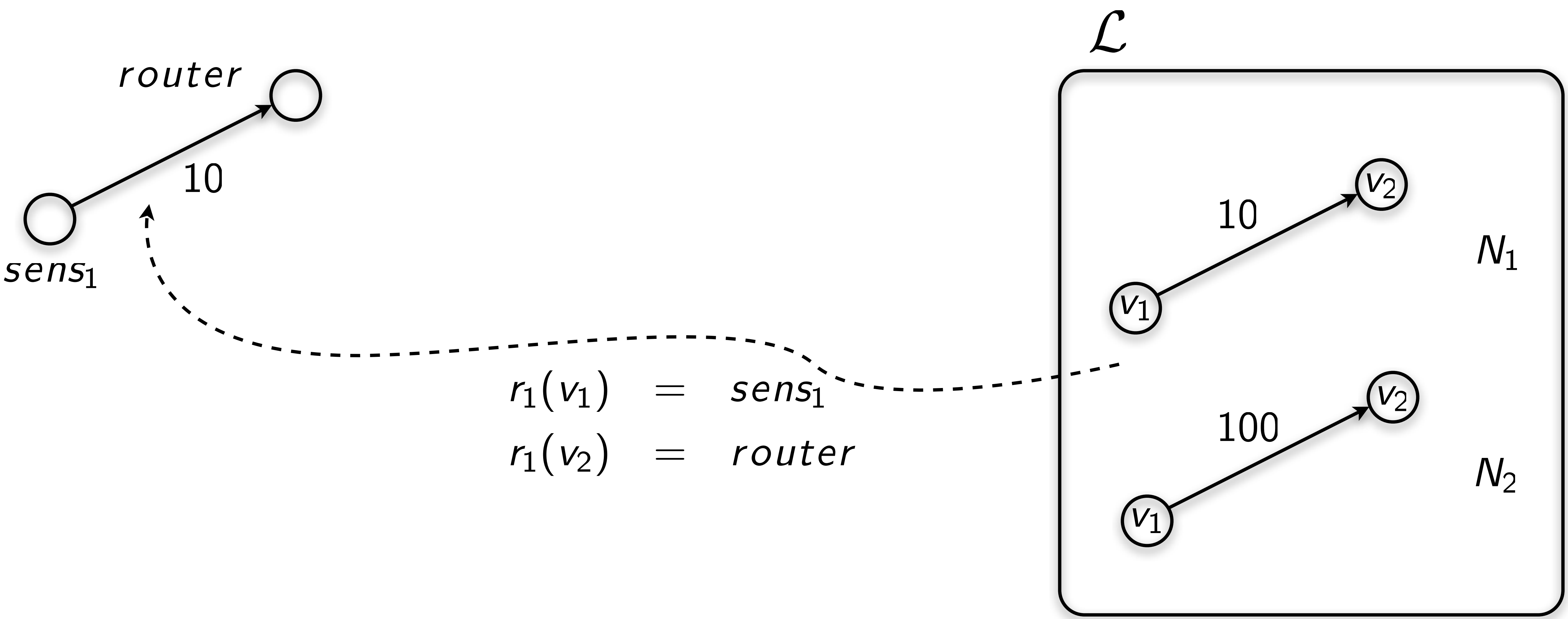
$$\mathcal{C}_1 \cup \mathcal{C}_2 \qquad L_1 \oplus_Q L_2$$



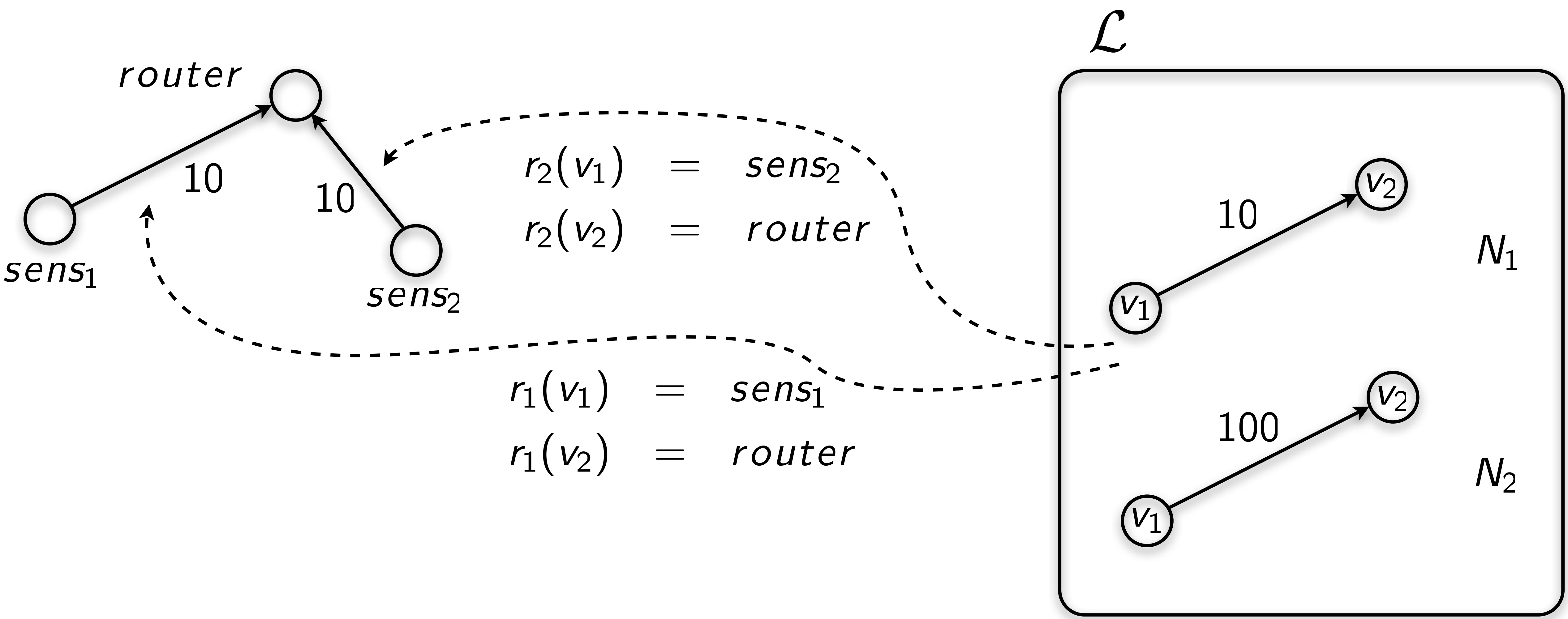
Definition of a Platform



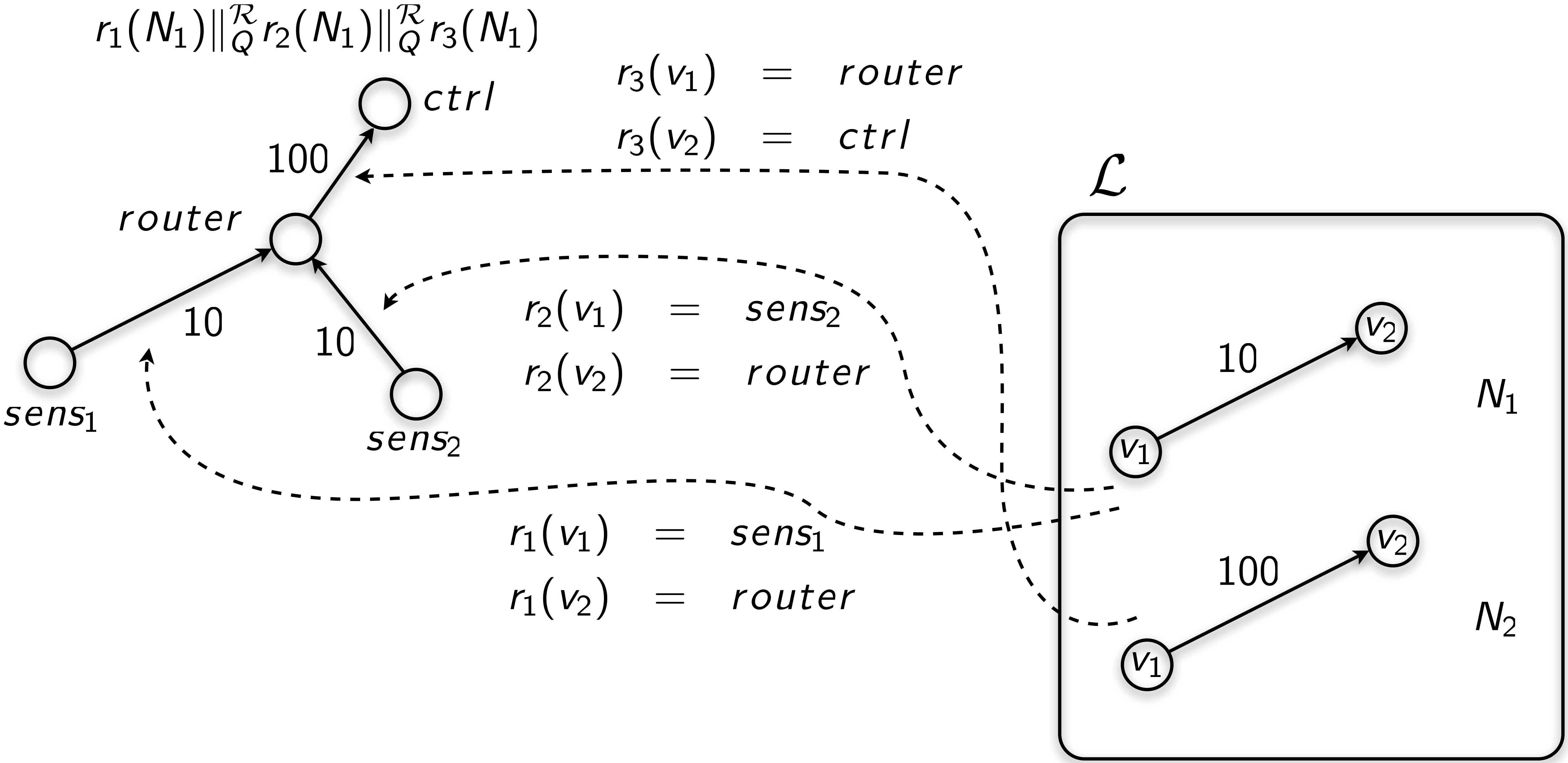
Definition of a Platform



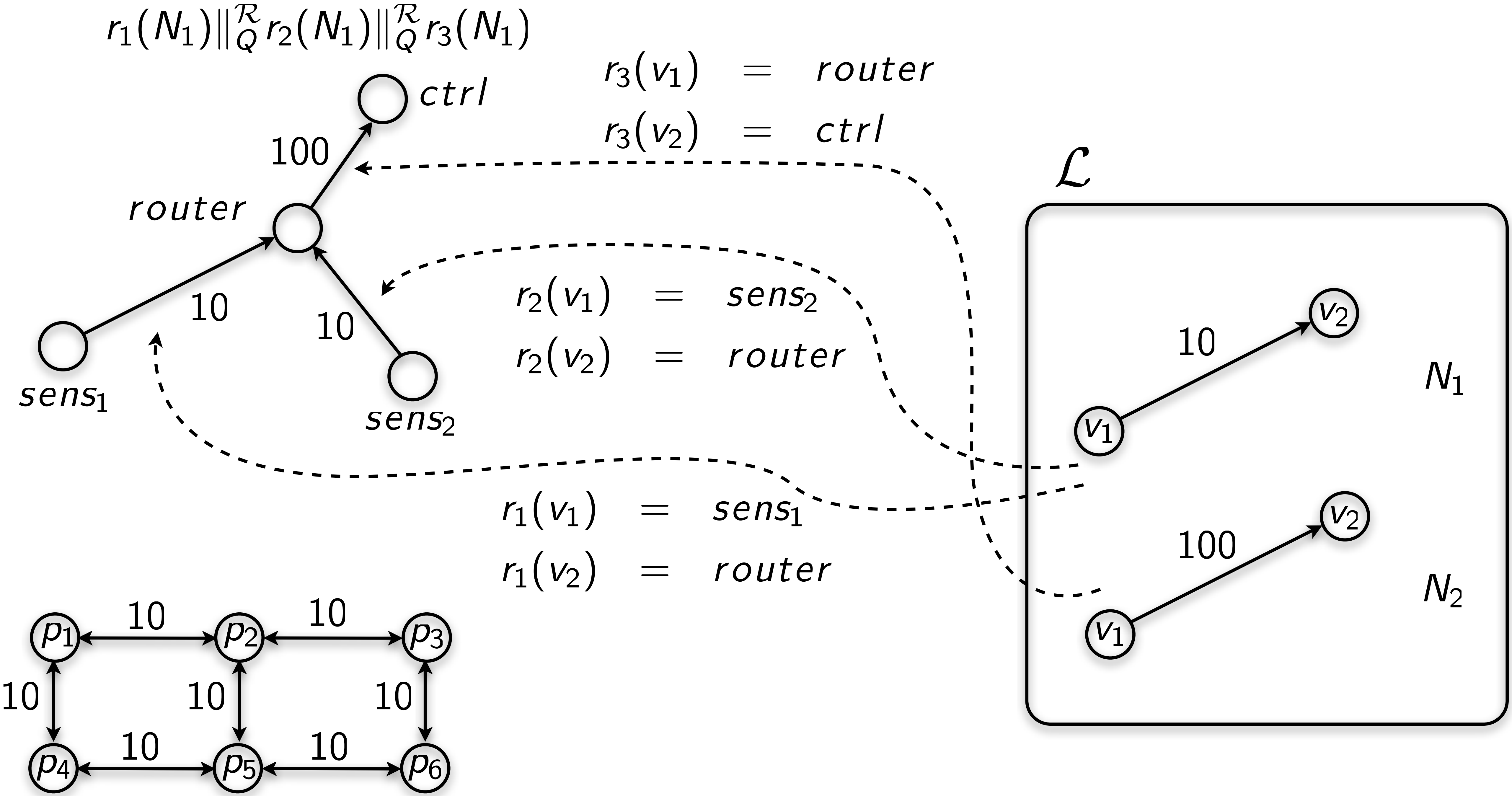
Definition of a Platform



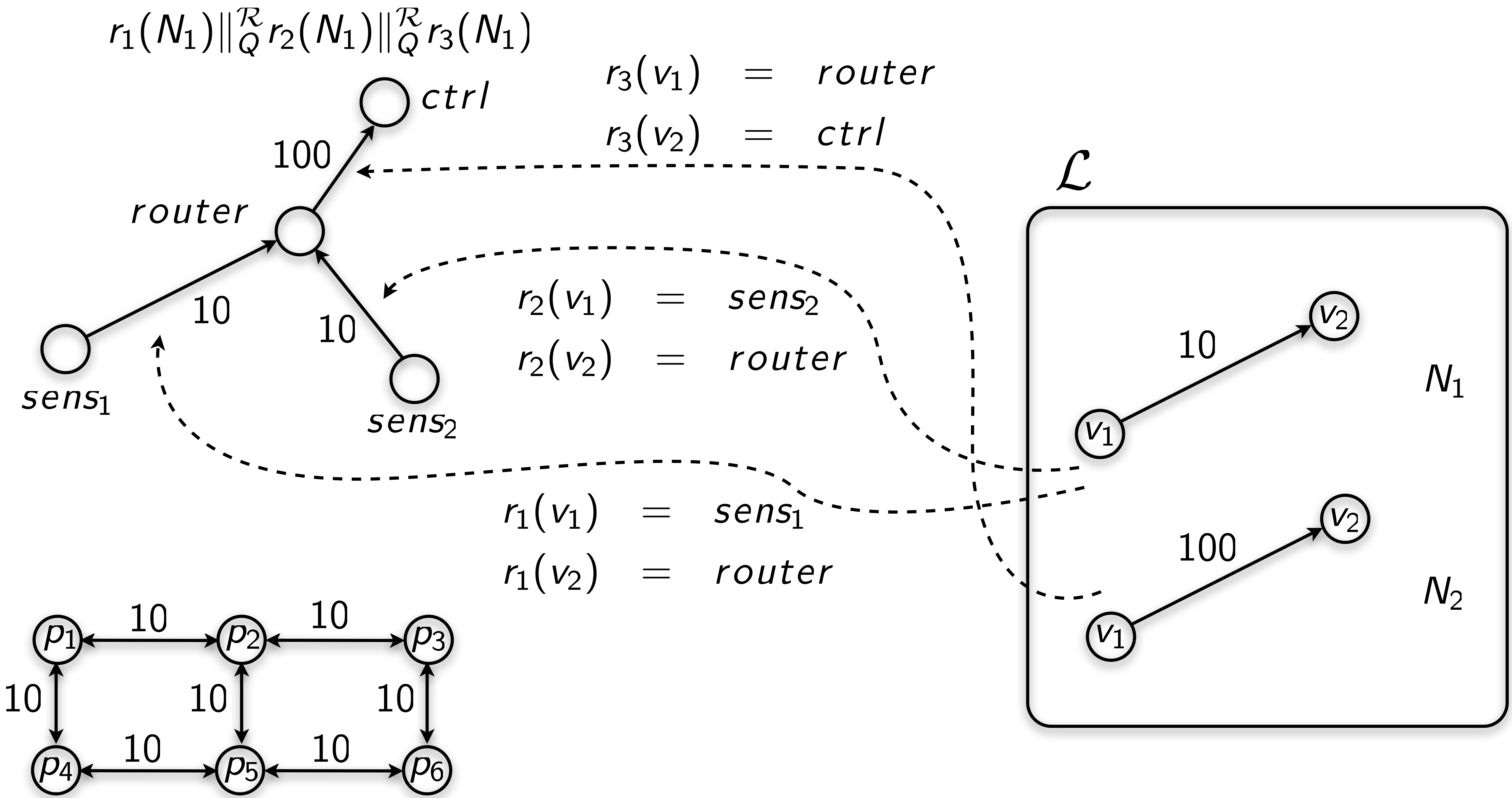
Definition of a Platform



Definition of a Platform



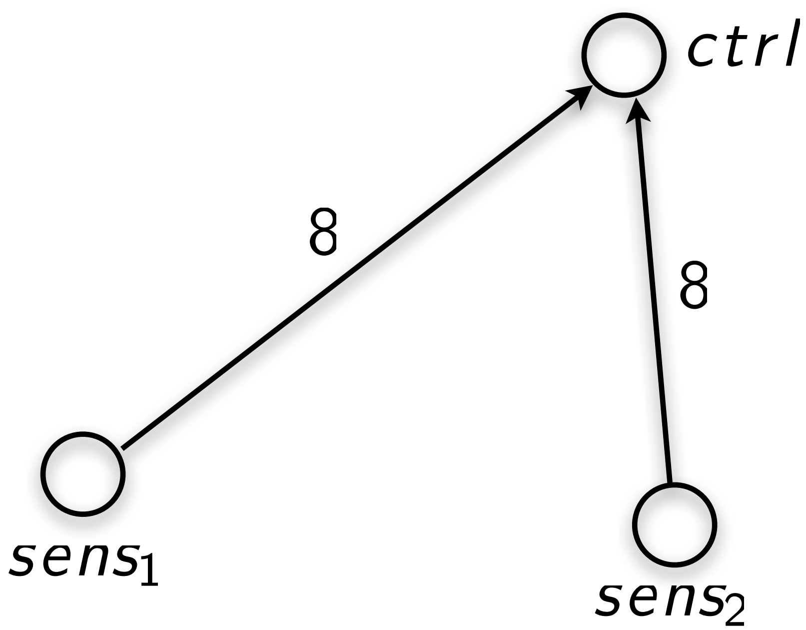
Definition of a Platform



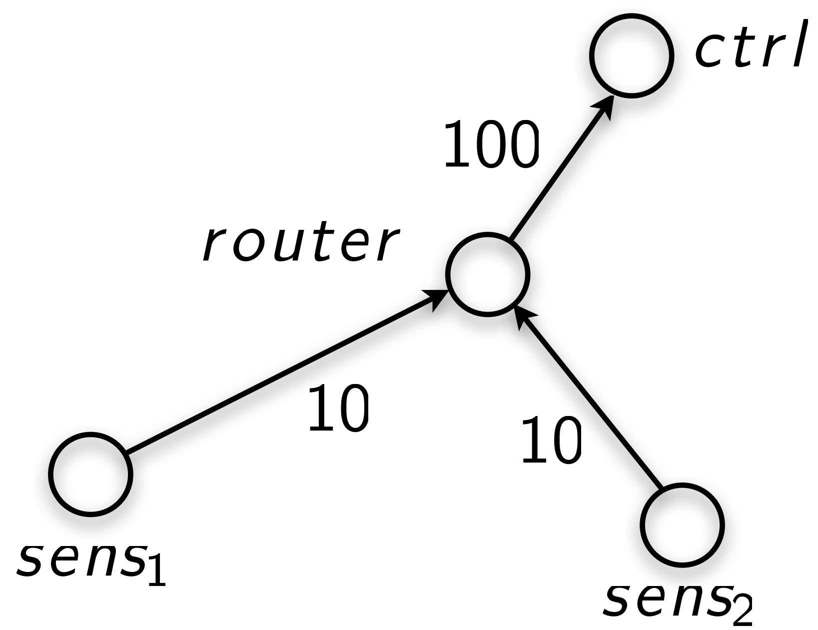
$$\langle \mathcal{L} \rangle = \mathcal{L} \ \cup \ \{ N = N' \|^{\mathcal{R}}_{Q} N'_L : N'_L \subseteq r(N_L), r \in R', \ N_L \in \mathcal{L}, \ N' \in \langle \mathcal{L} \rangle \}$$

Specification, Platform, Mapping

Specification $N_C(C_C, Q_C, L_C)$

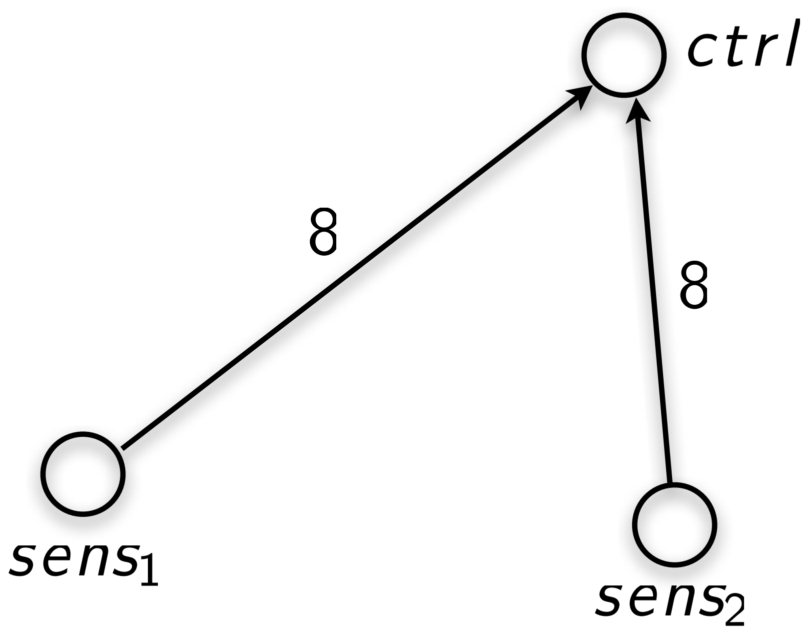


Platform Instance $N_P(C_P, Q_P, L_P)$

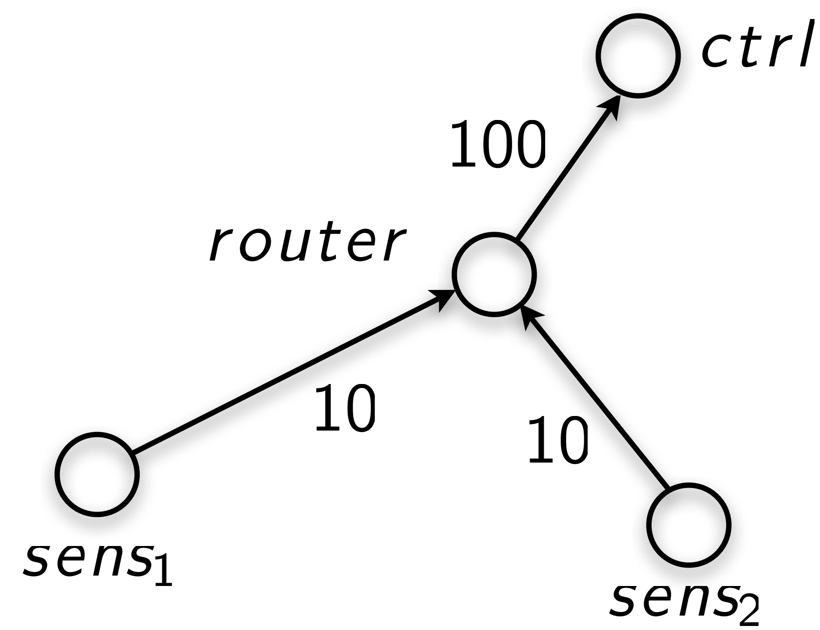


Specification, Platform, Mapping

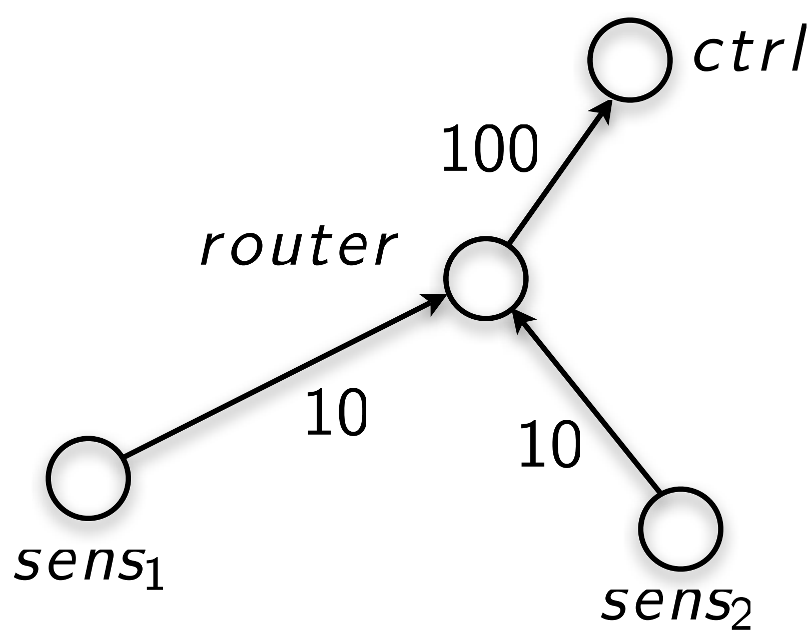
Specification $N_C(C_C, Q_C, L_C)$



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

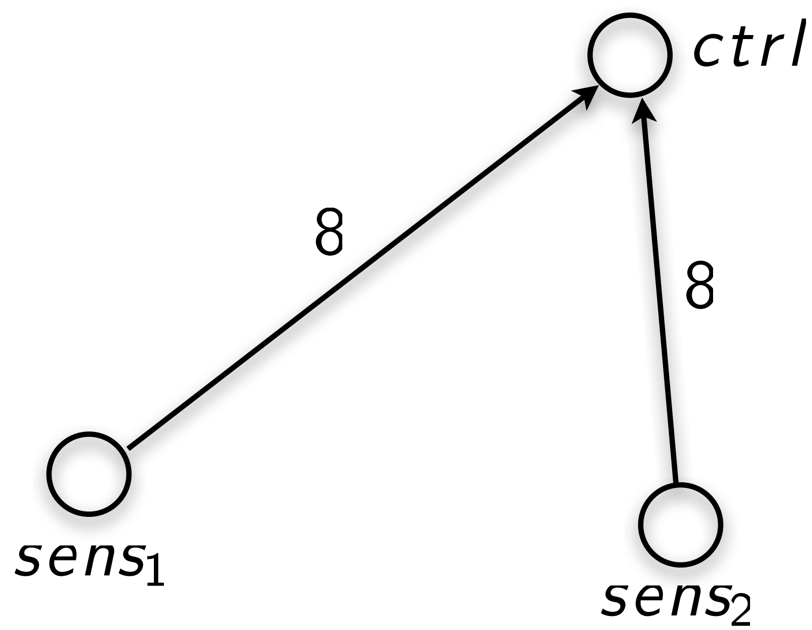


Implementation $N_I(C_I, Q_I, L_I)$

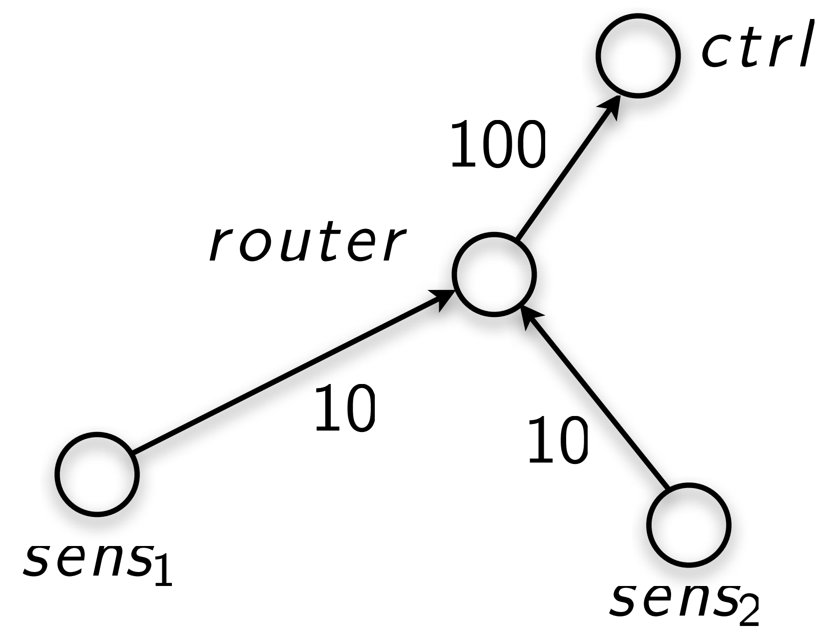


Specification, Platform, Mapping

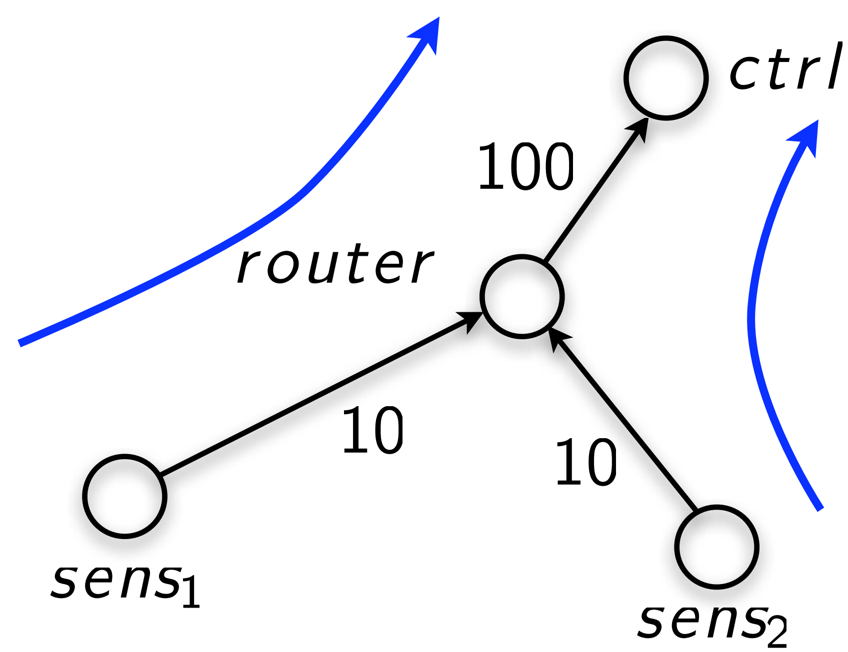
Specification $N_C(C_C, Q_C, L_C)$



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

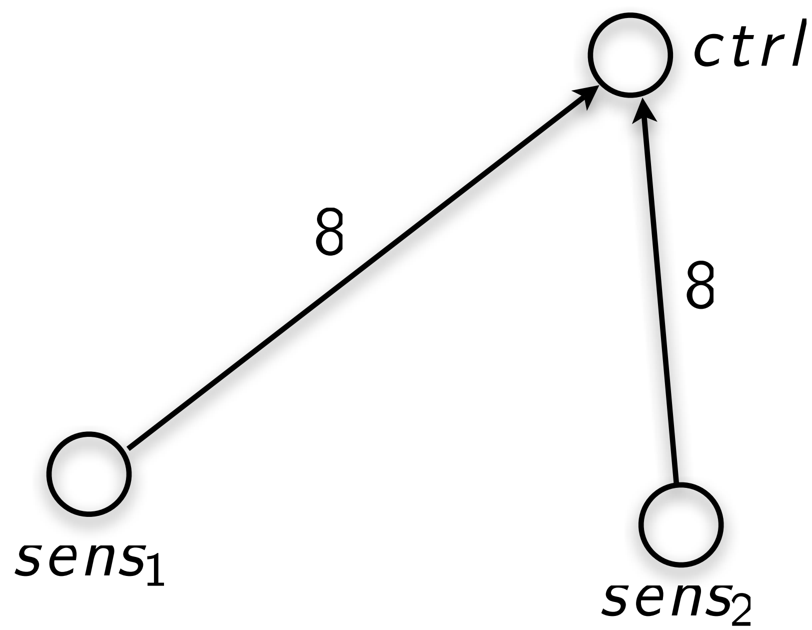


Implementation $N_I(C_I, Q_I, L_I)$

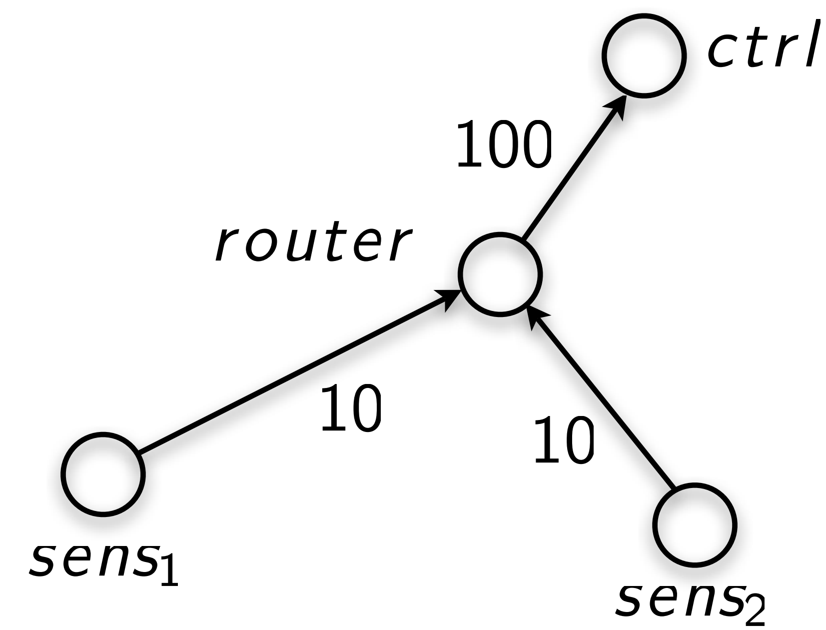


Specification, Platform, Mapping

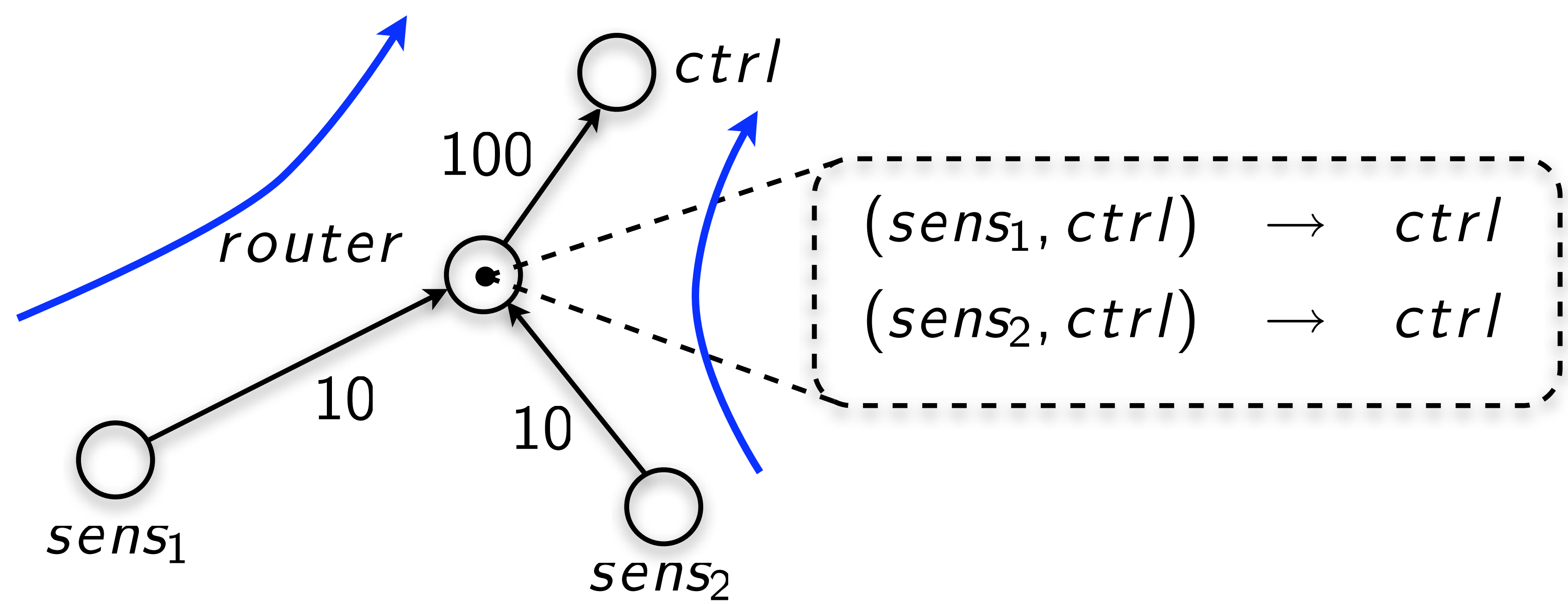
Specification $N_C(C_C, Q_C, L_C)$



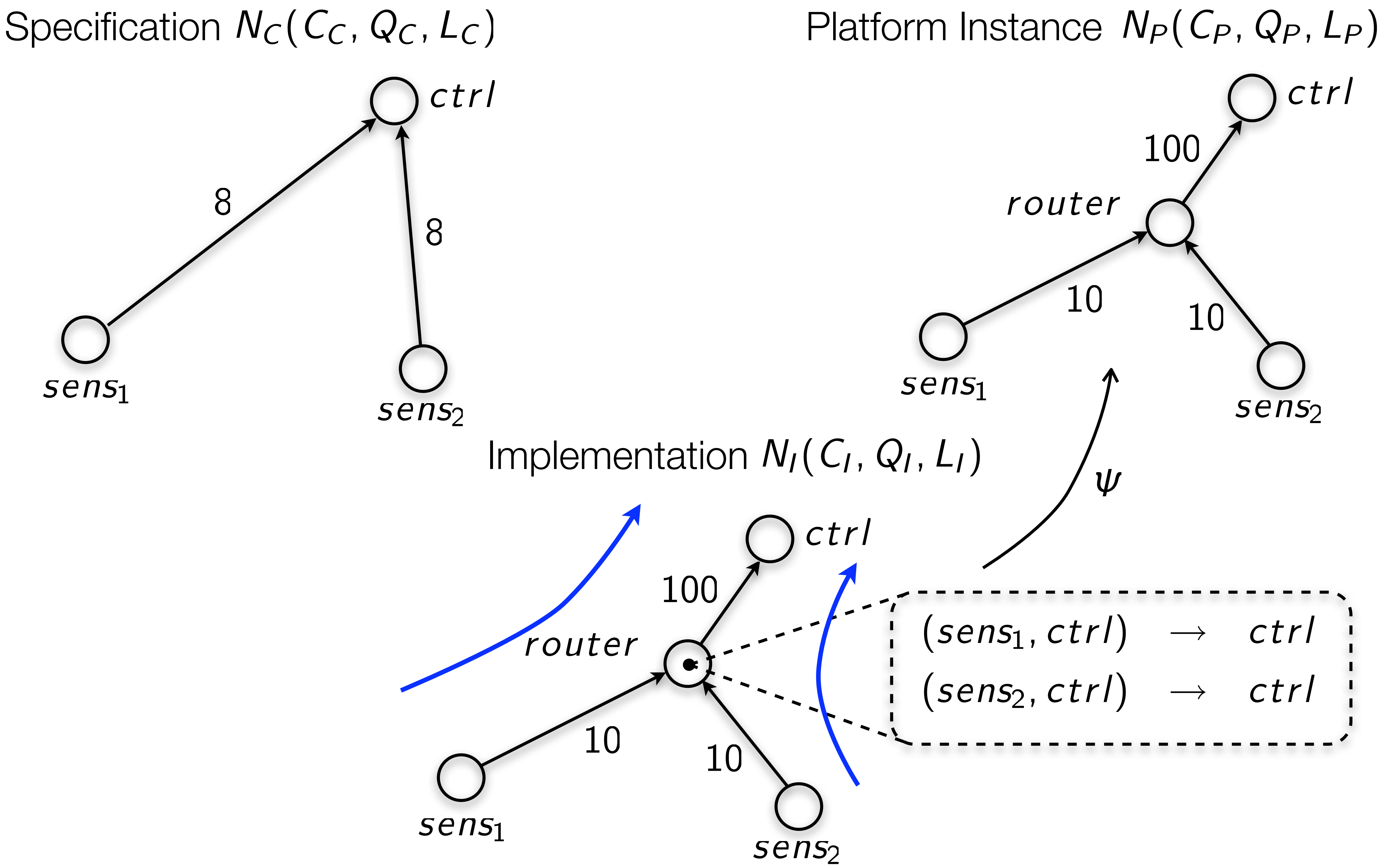
Platform Instance $N_P(C_P, Q_P, L_P)$



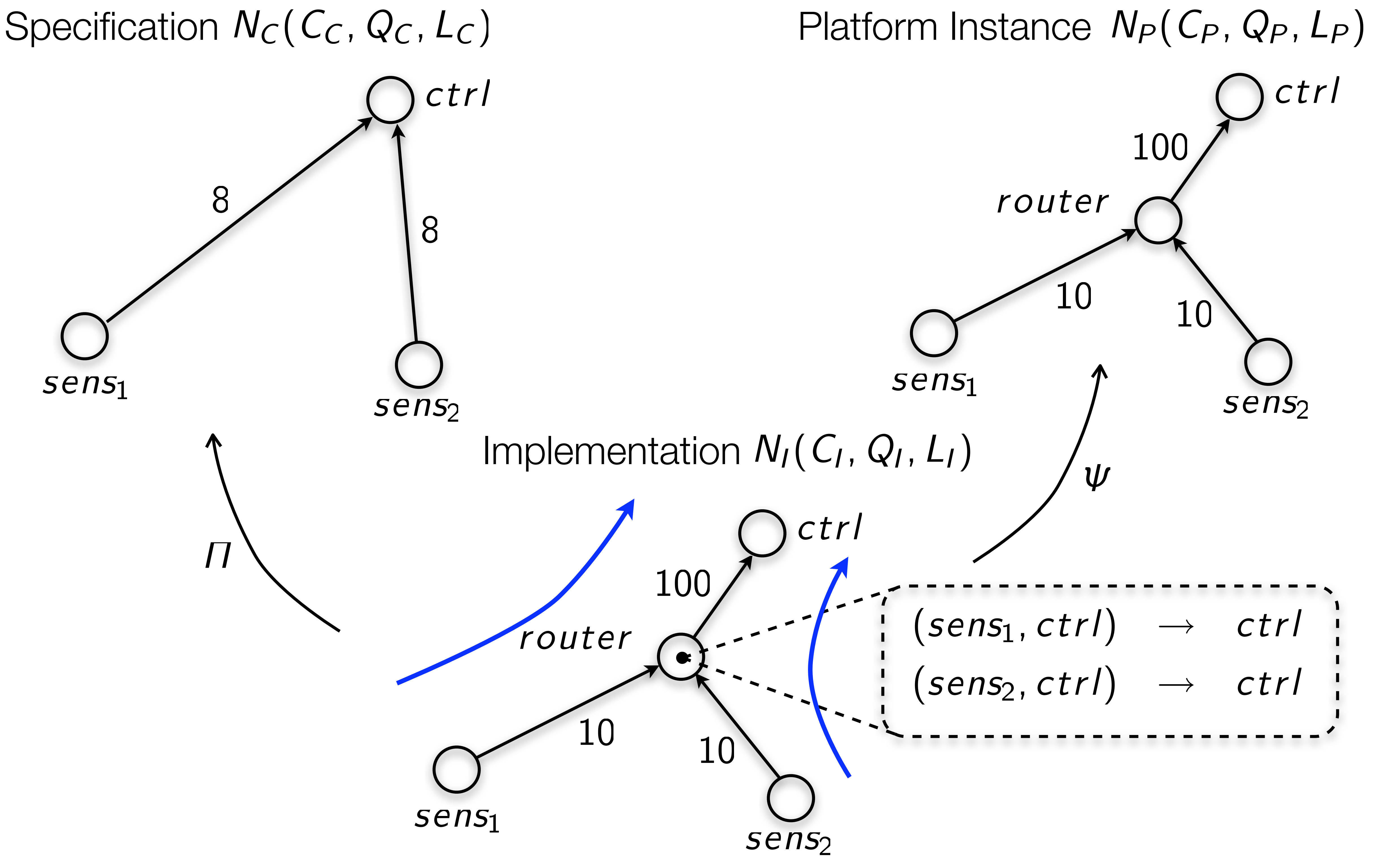
Implementation $N_I(C_I, Q_I, L_I)$



Specification, Platform, Mapping

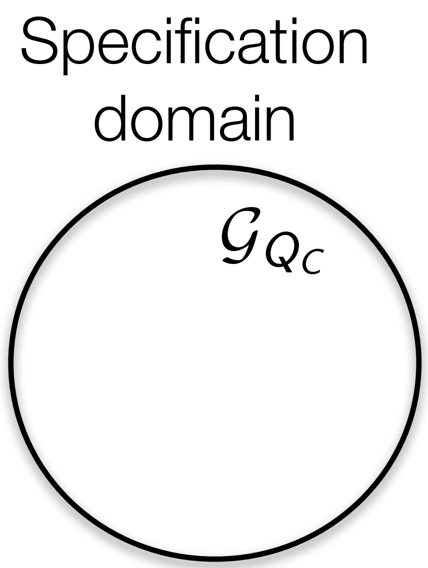


Specification, Platform, Mapping

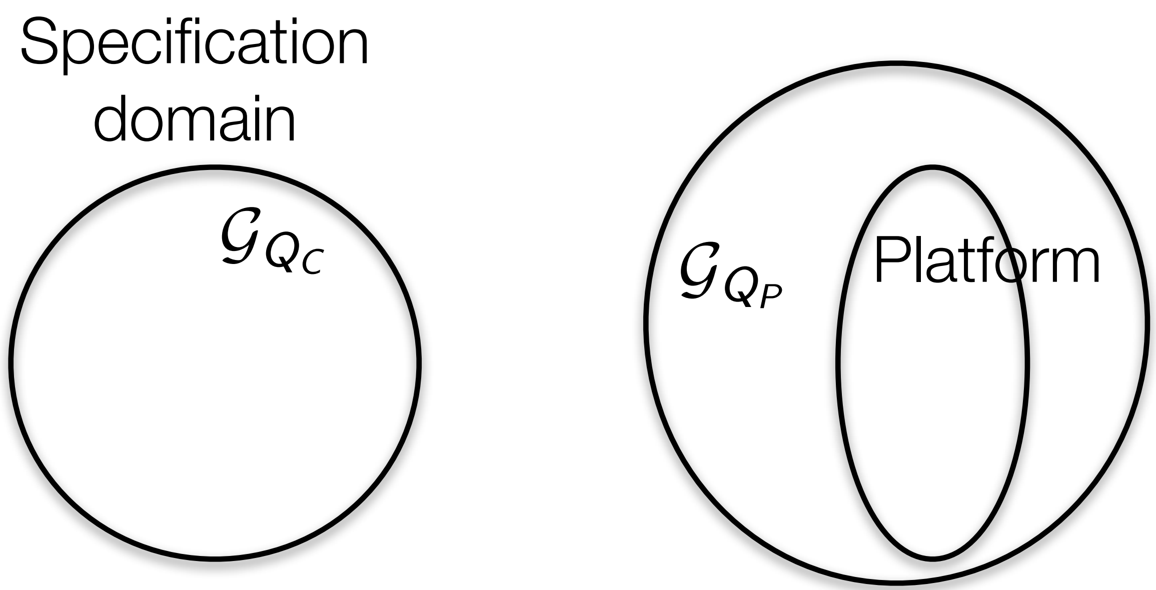


The General Synthesis Problem

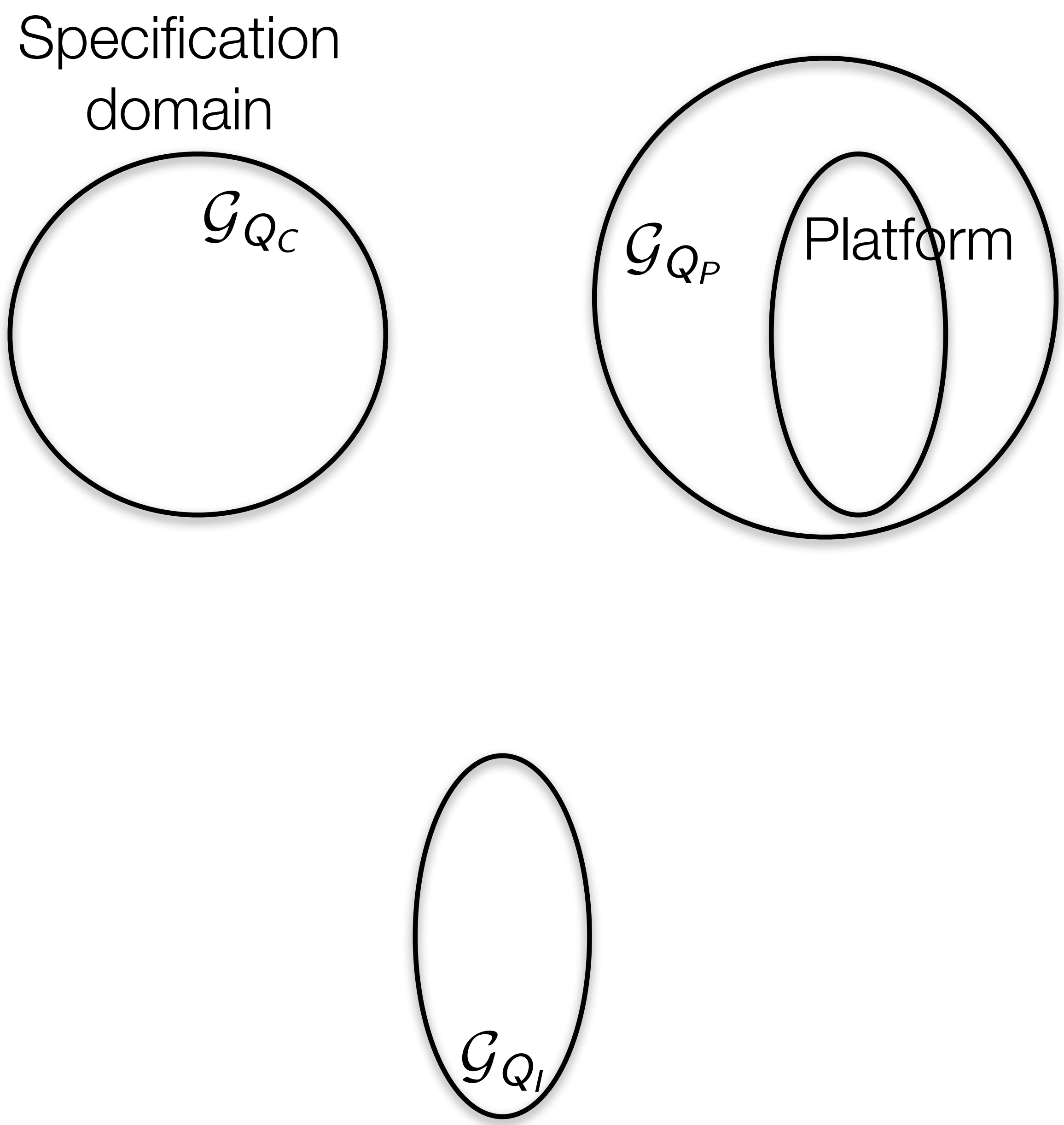
The General Synthesis Problem



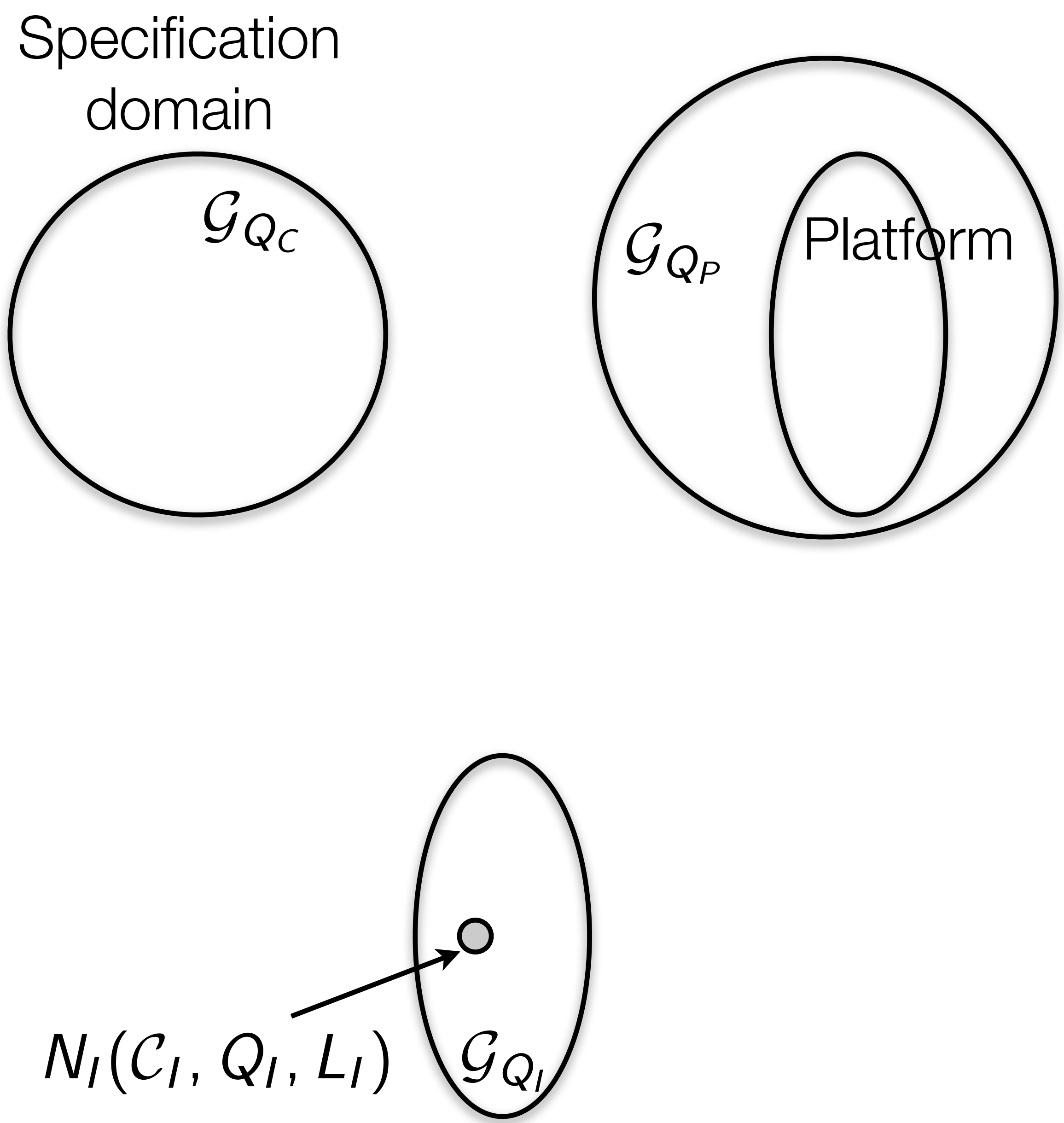
The General Synthesis Problem



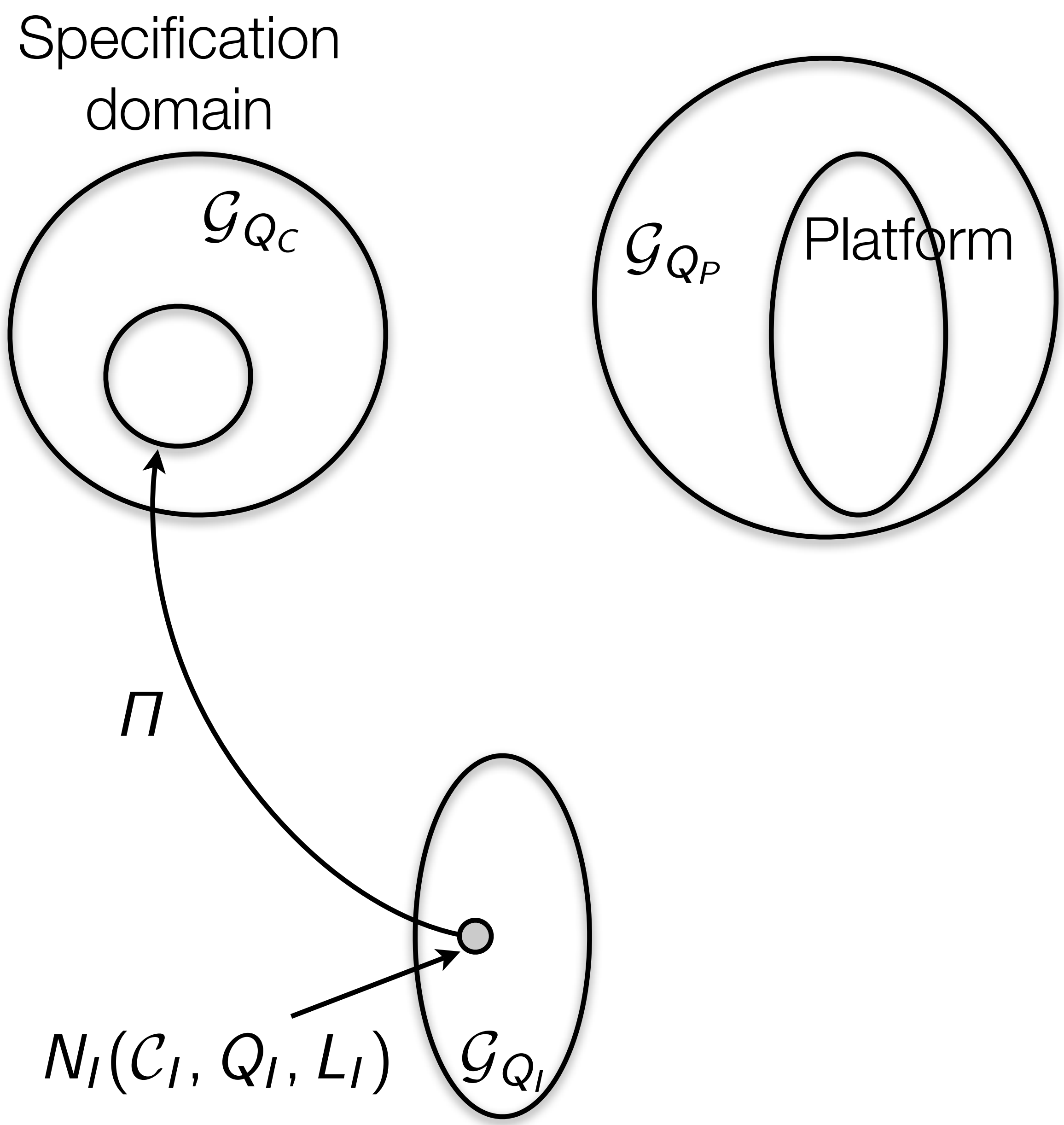
The General Synthesis Problem



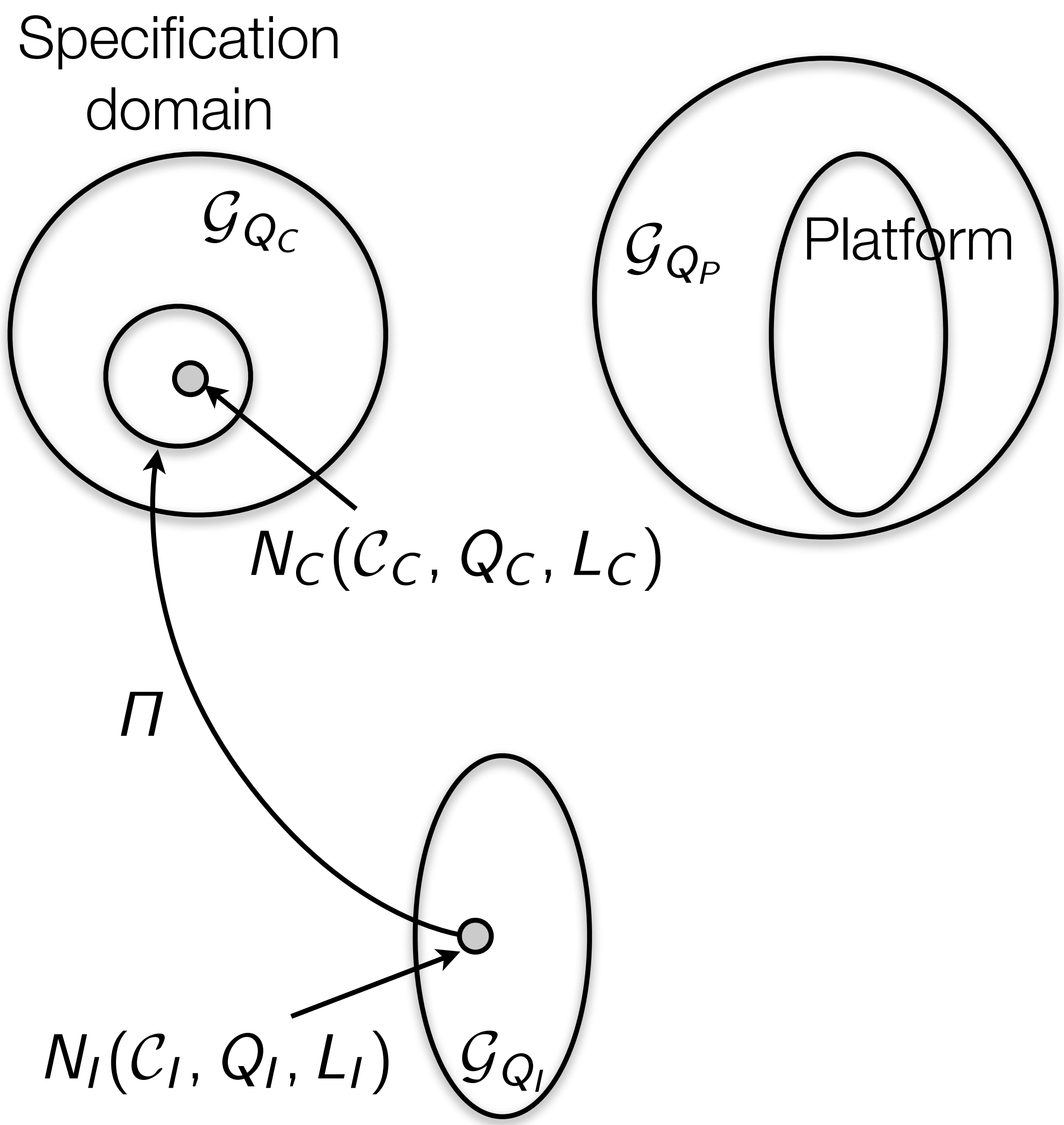
The General Synthesis Problem



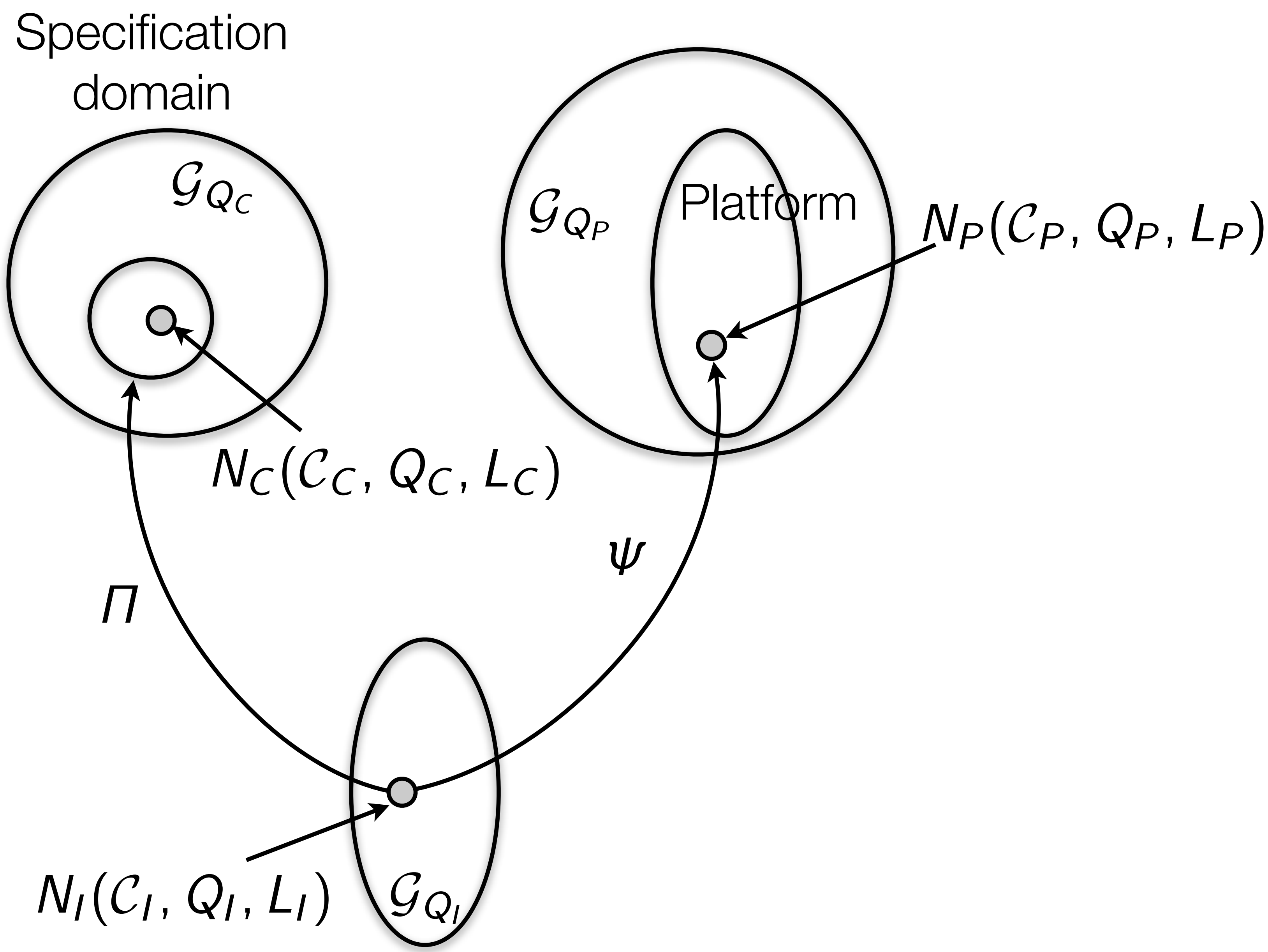
The General Synthesis Problem



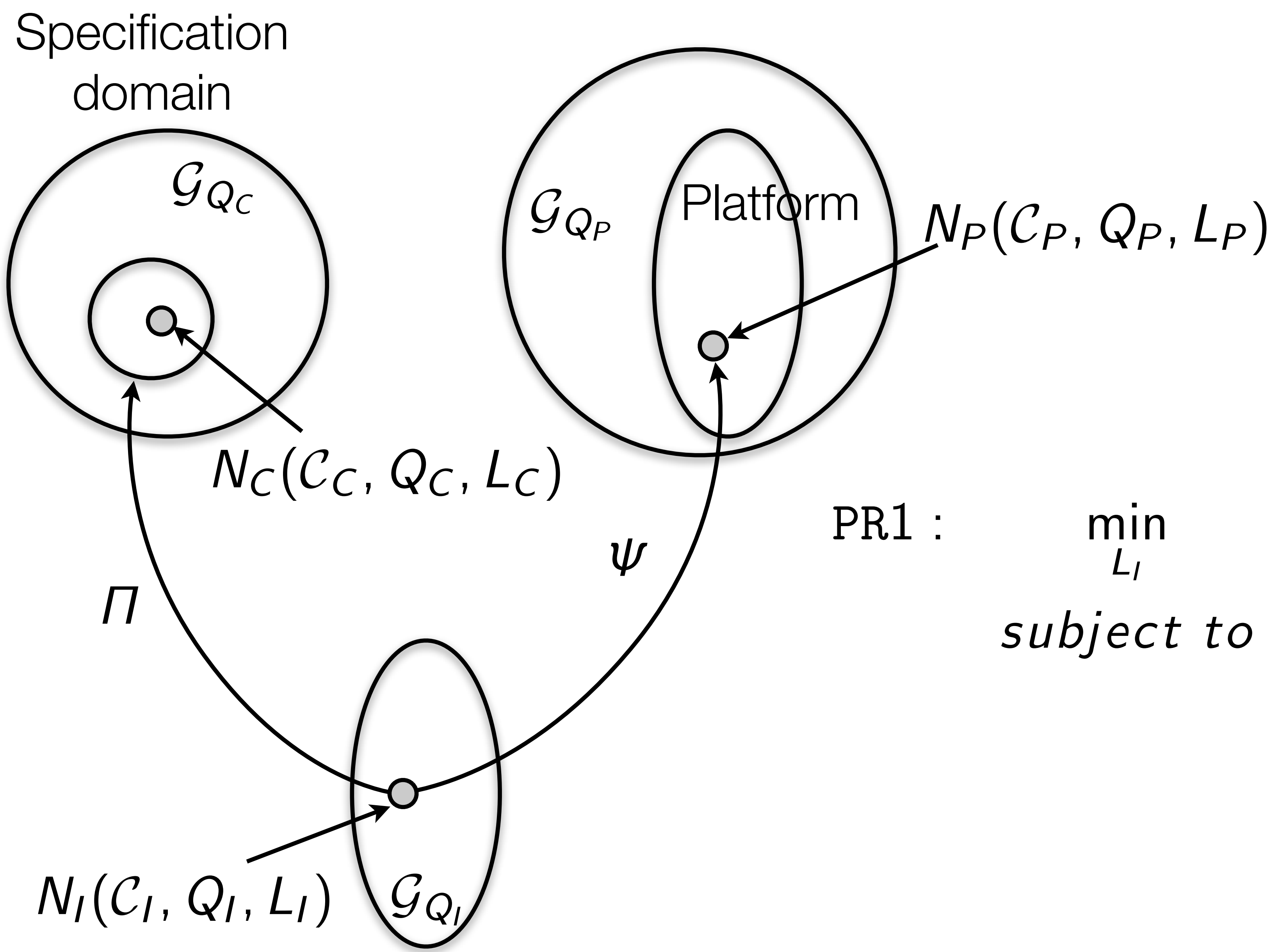
The General Synthesis Problem



The General Synthesis Problem

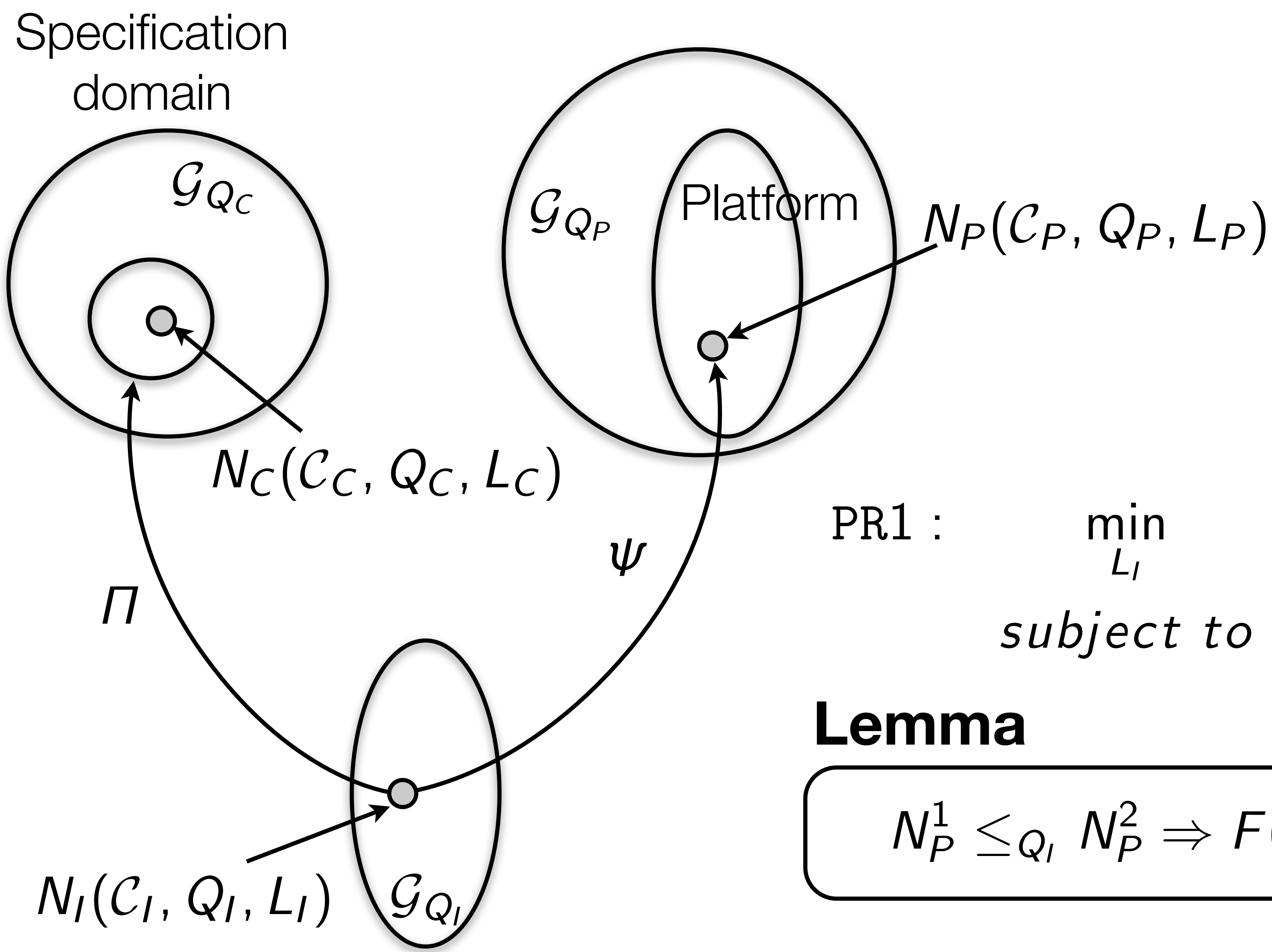


The General Synthesis Problem



$$\mathtt{PR1}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq_{Q_c} \Pi(N_I), \;\; \Psi(N_I) \subseteq N_P$$

The General Synthesis Problem



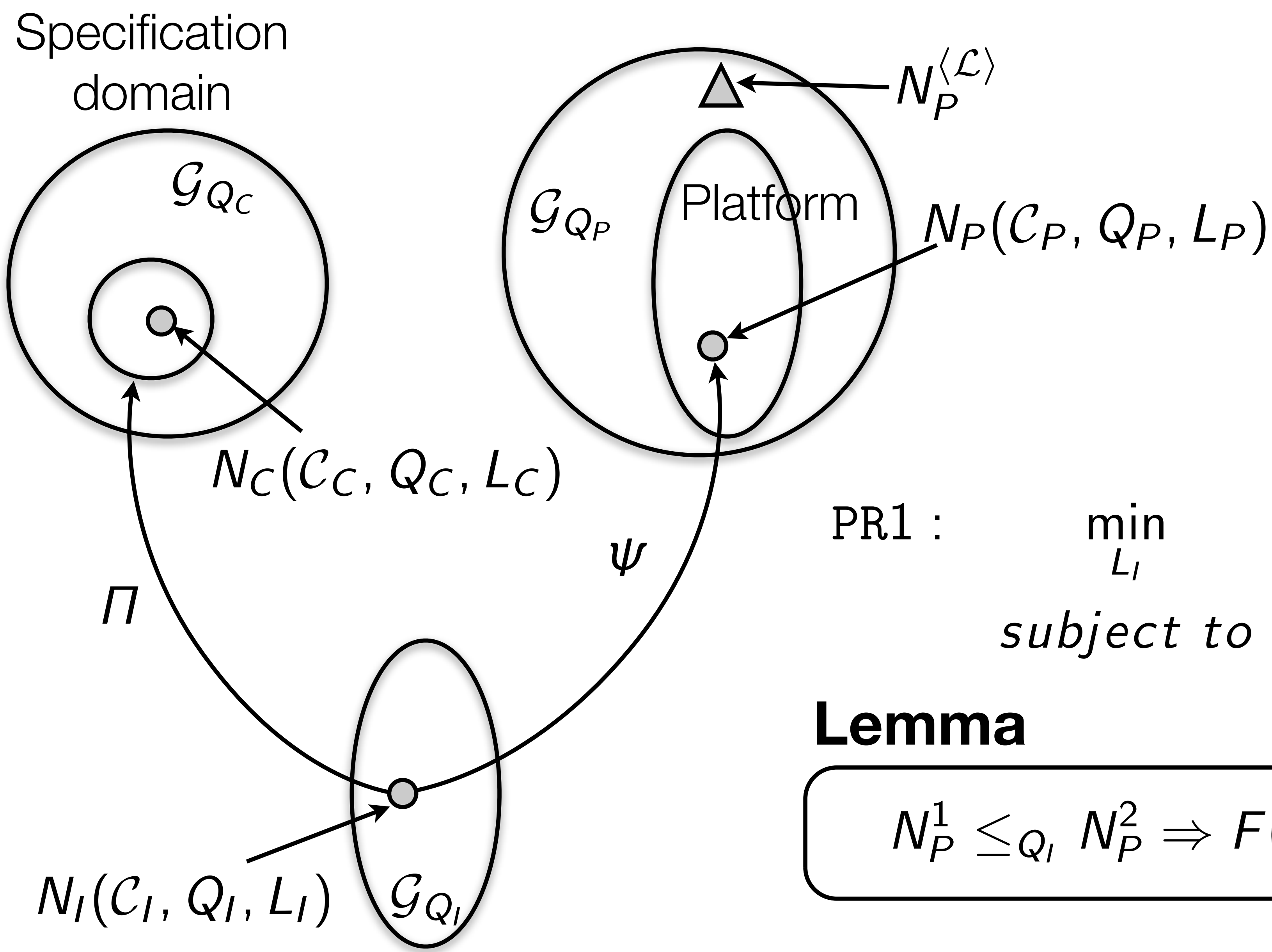
$$\texttt{PR1}: \quad \min_{L_I} \quad F(N_I)$$

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

Lemma

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

The General Synthesis Problem



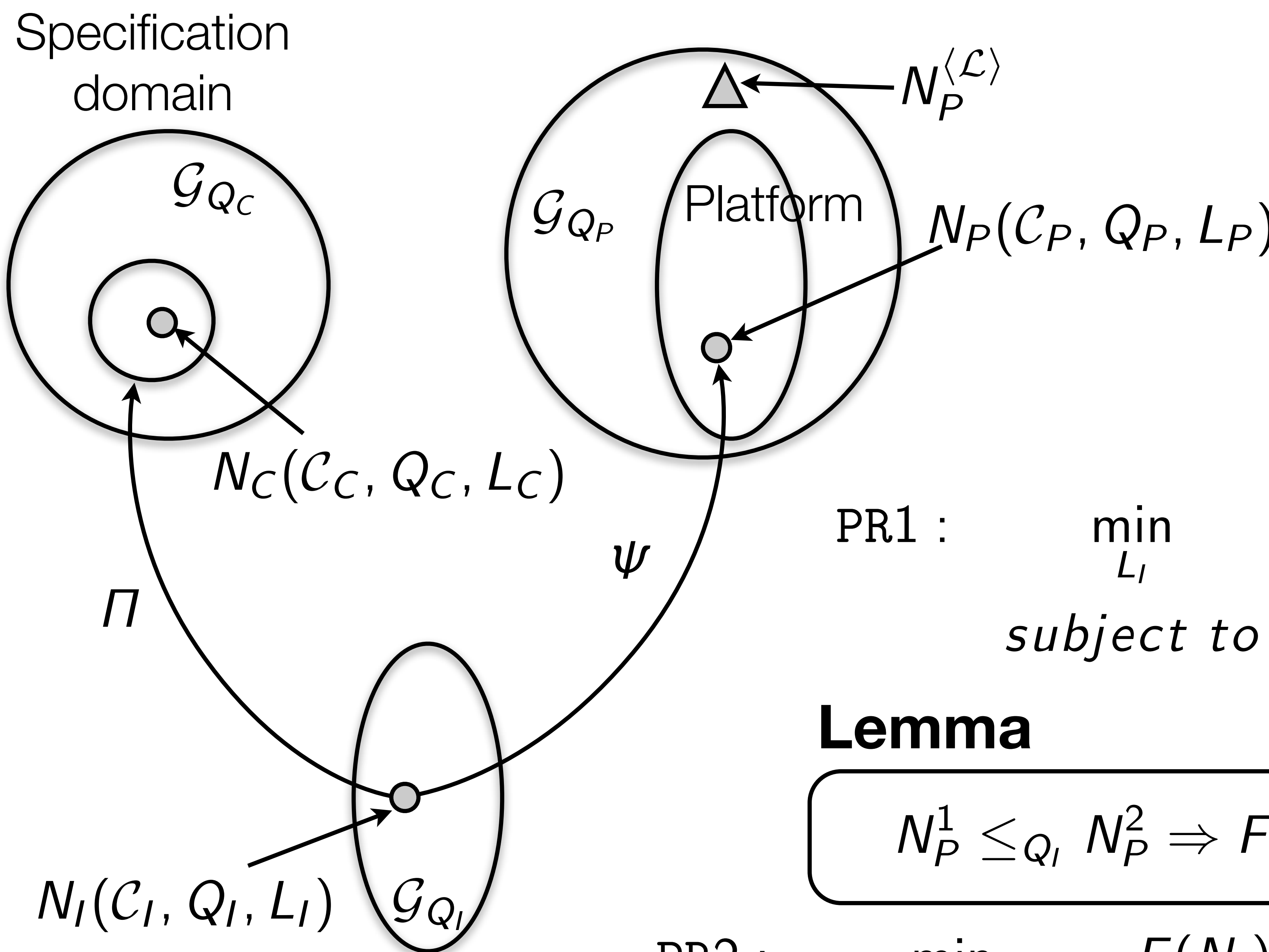
$$\mathtt{PR1}: \quad \min_{L_I} \quad F(N_I)$$

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

Lemma

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

The General Synthesis Problem



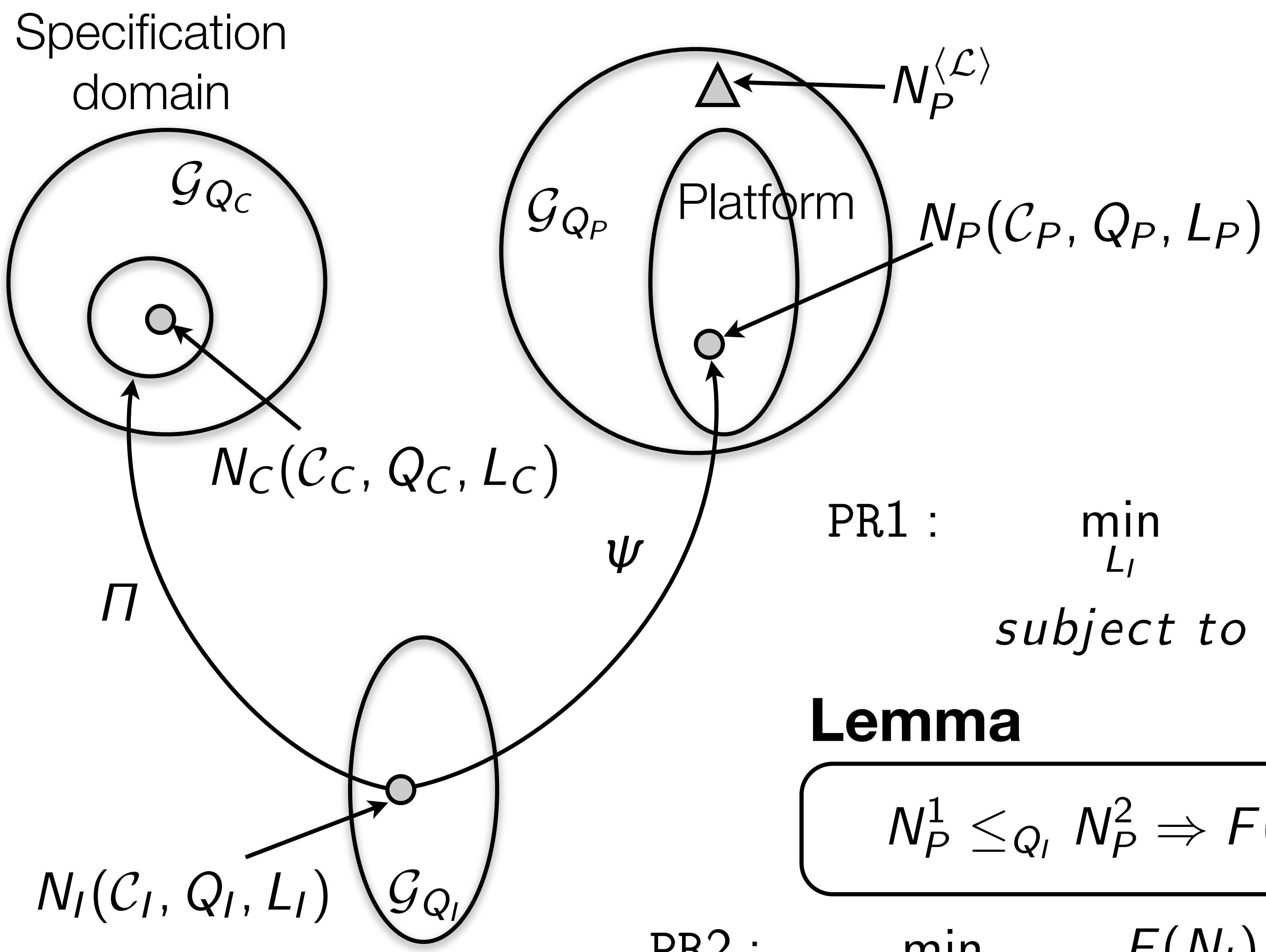
$$\text{PR1}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq_{Q_C} \Pi(N_I), \;\; \Psi(N_I) \subseteq N_P$$

Lemma

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

$$\text{PR2}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq \Pi(\Phi(N_I)), \;\; \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$$
$$\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$$

The General Synthesis Problem



$$\text{PR1}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq_{Q_C} \Pi(N_I), \;\; \Psi(N_I) \subseteq N_P$$

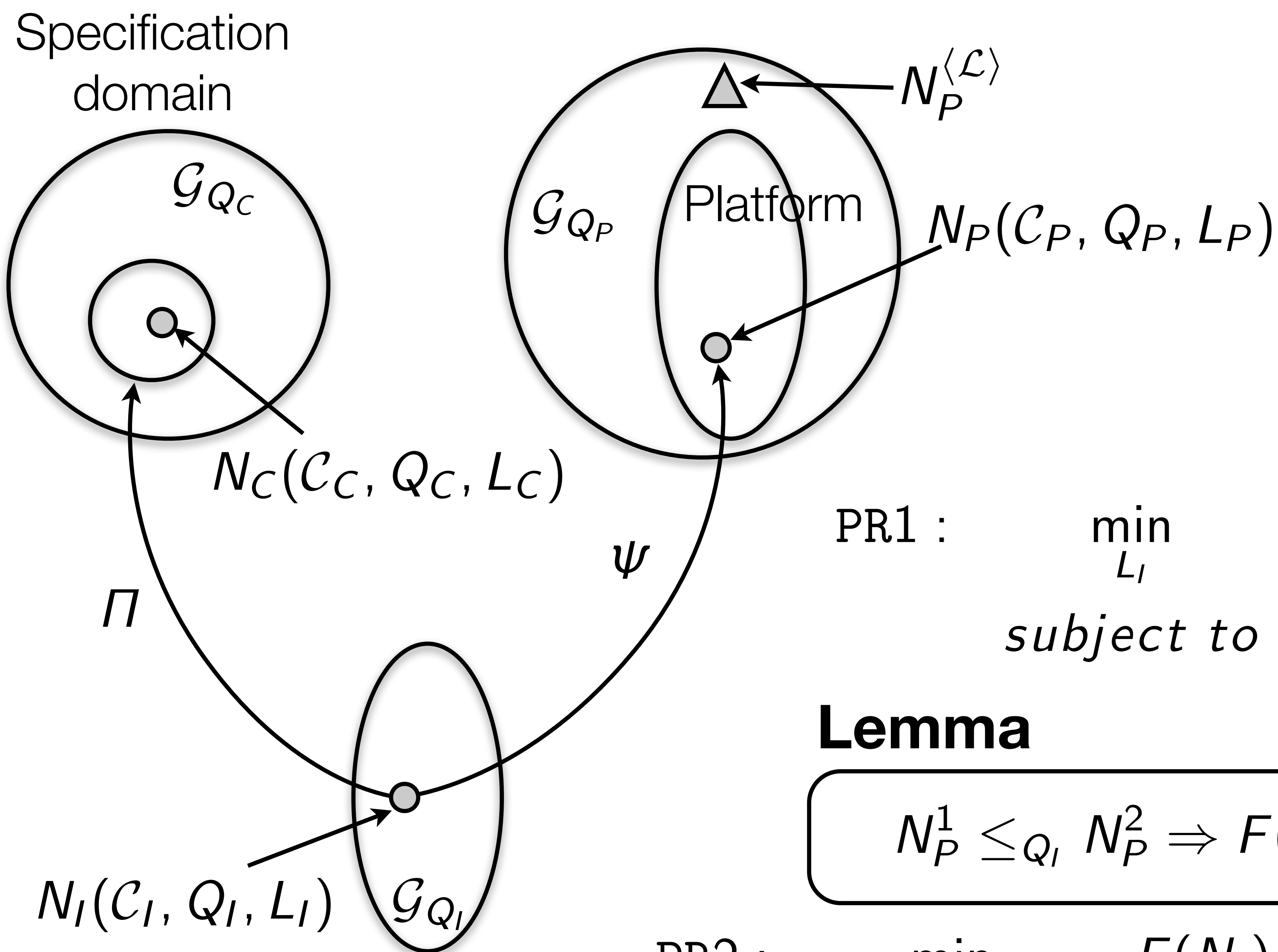
Lemma

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

$$\text{PR2}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq \Pi(\Phi(N_I)), \;\; \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$$
$$\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$$

Depends on the composition rules

The General Synthesis Problem



$$\text{PR1}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq_{Q_C} \Pi(N_I), \ \ \Psi(N_I) \subseteq N_P$$

Lemma

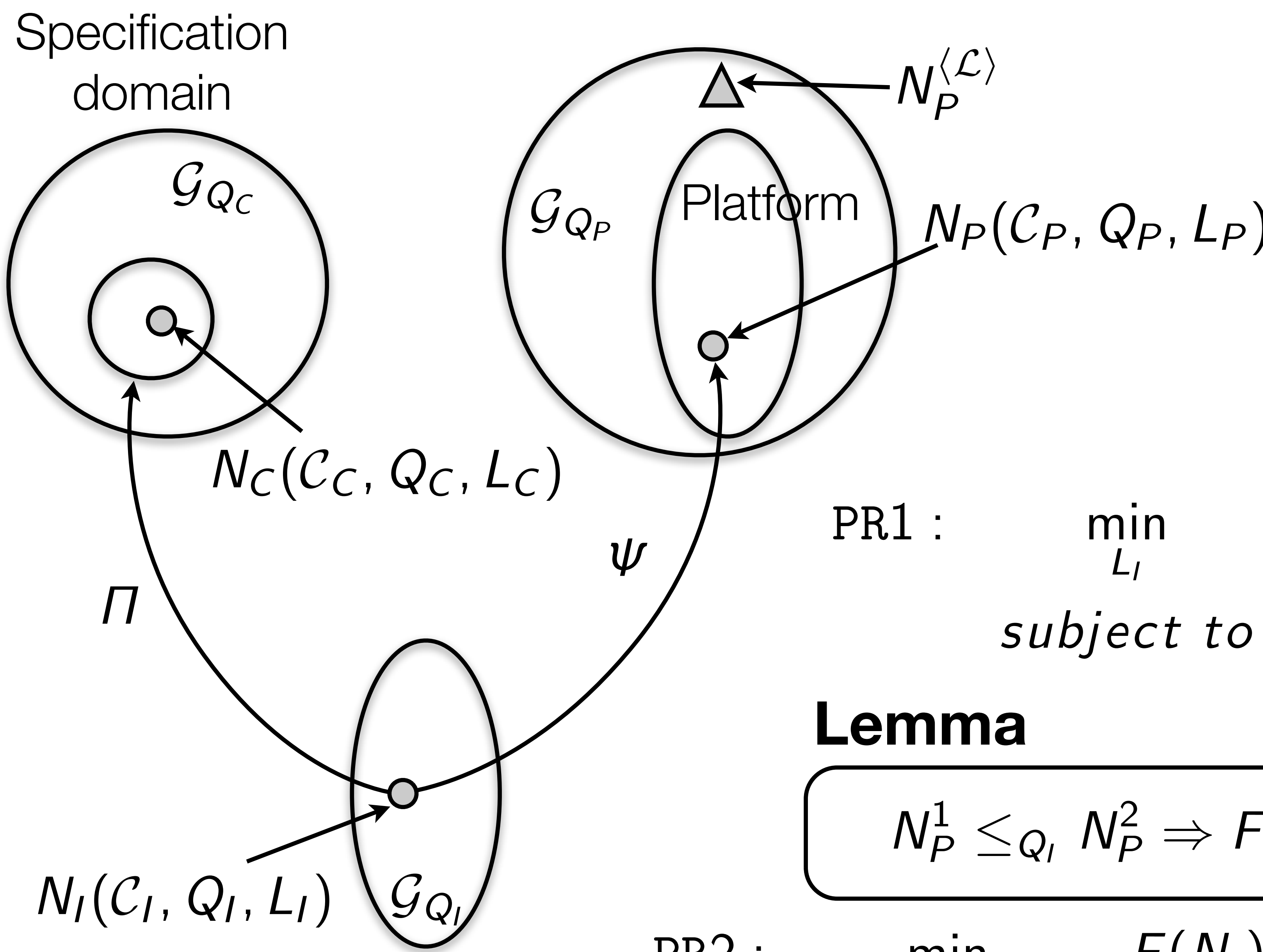
$$N_P^1 \leq_{Q_I} N_P^2 \Rightarrow F(N_I^1) \leq F(N_I^2)$$

Depends on quantities and library

$$\text{PR2}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq \Pi(\Phi(N_I)), \ \ \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$$
$$\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$$

Depends on the composition rules

The General Synthesis Problem



Theorem

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

$$\mathrm{PR1}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq_{Q_C} \Pi(N_I), \;\; \Psi(N_I) \subseteq N_P$$

Lemma

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

Depends on quantities and library

$$\mathrm{PR2}: \quad \min_{L_I} \quad F(N_I)$$
$$\textit{subject to} \quad N_C \leq \Pi(\Phi(N_I)), \;\; \Psi(N_I) = N_P^{\langle \mathcal{L} \rangle},$$
$$\Psi(\Phi(N_I)) \in \langle \mathcal{L} \rangle$$

Depends on the composition rules

A General Method for Communication Design

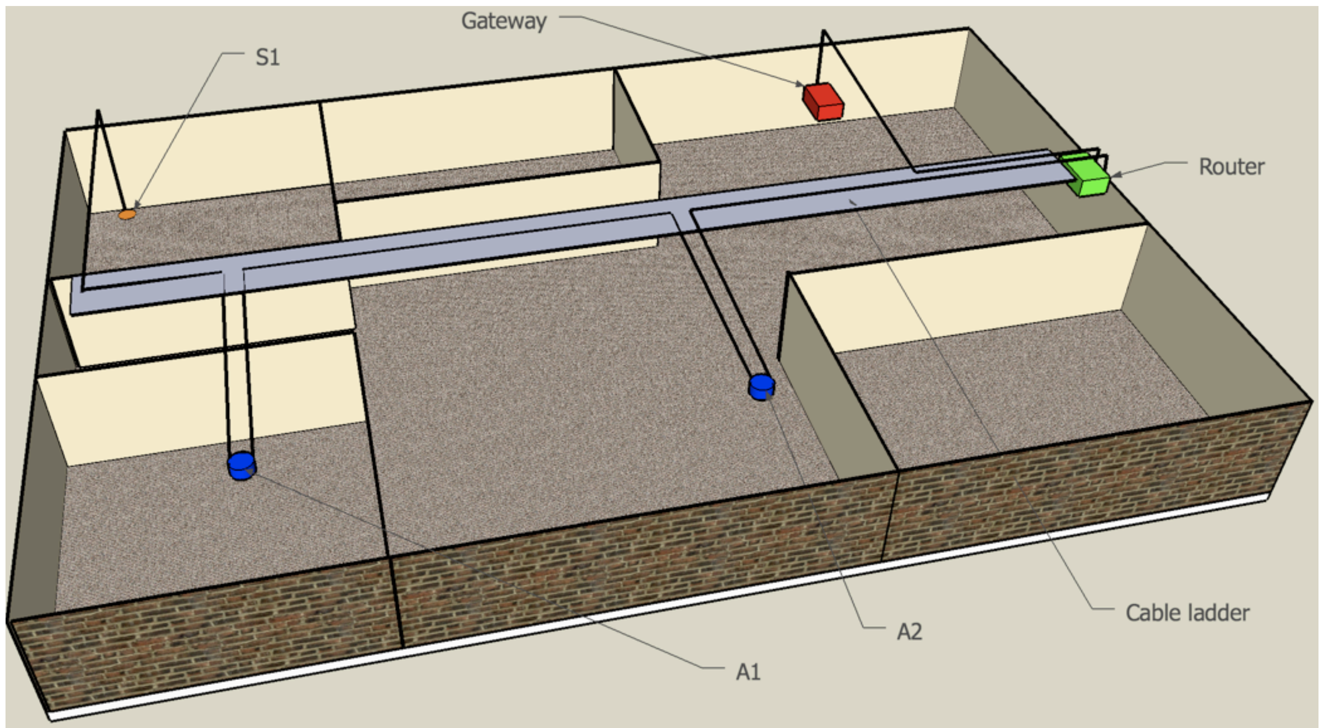
- Define the library L
- Define the rules R
- Define the models M
- Find the upper bound
- Formulate the specific problem: Given the properties to preserve and L, R, M
- Find an efficient algorithm

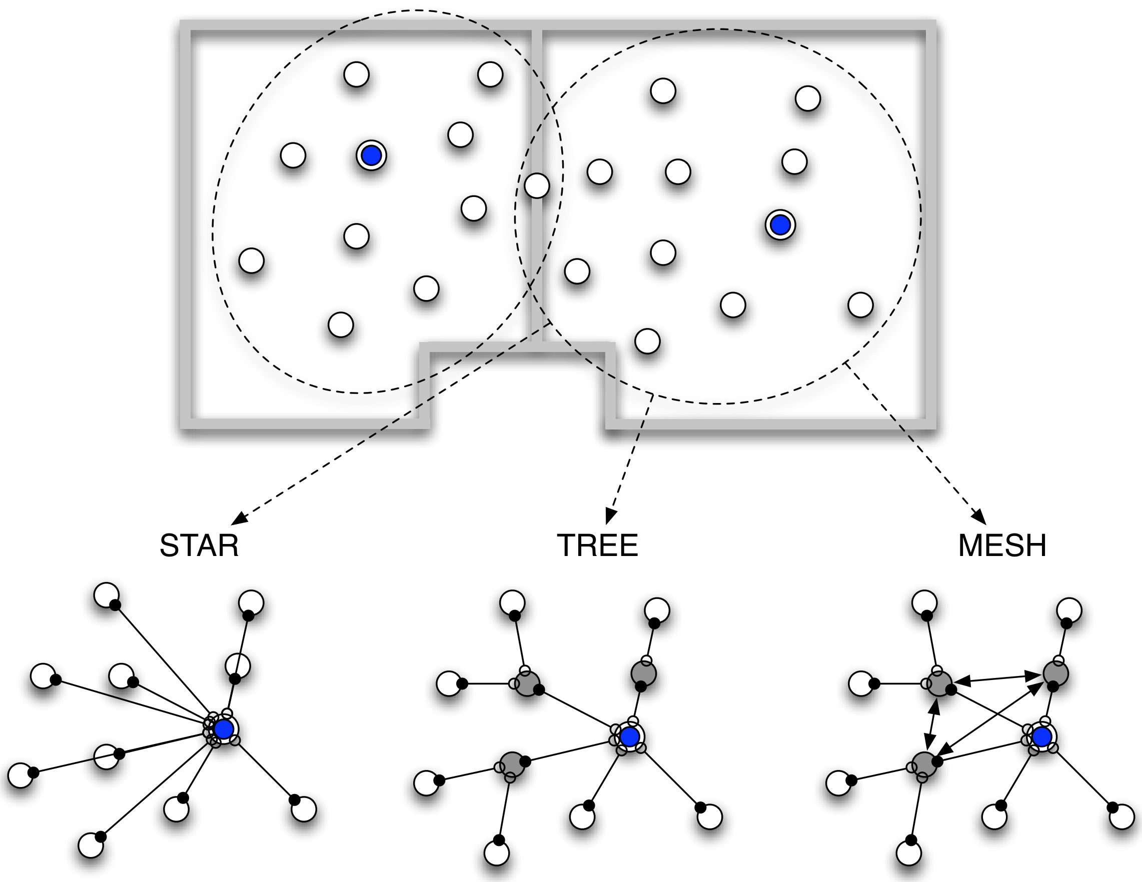
Communication Synthesis for Building Automation Systems

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



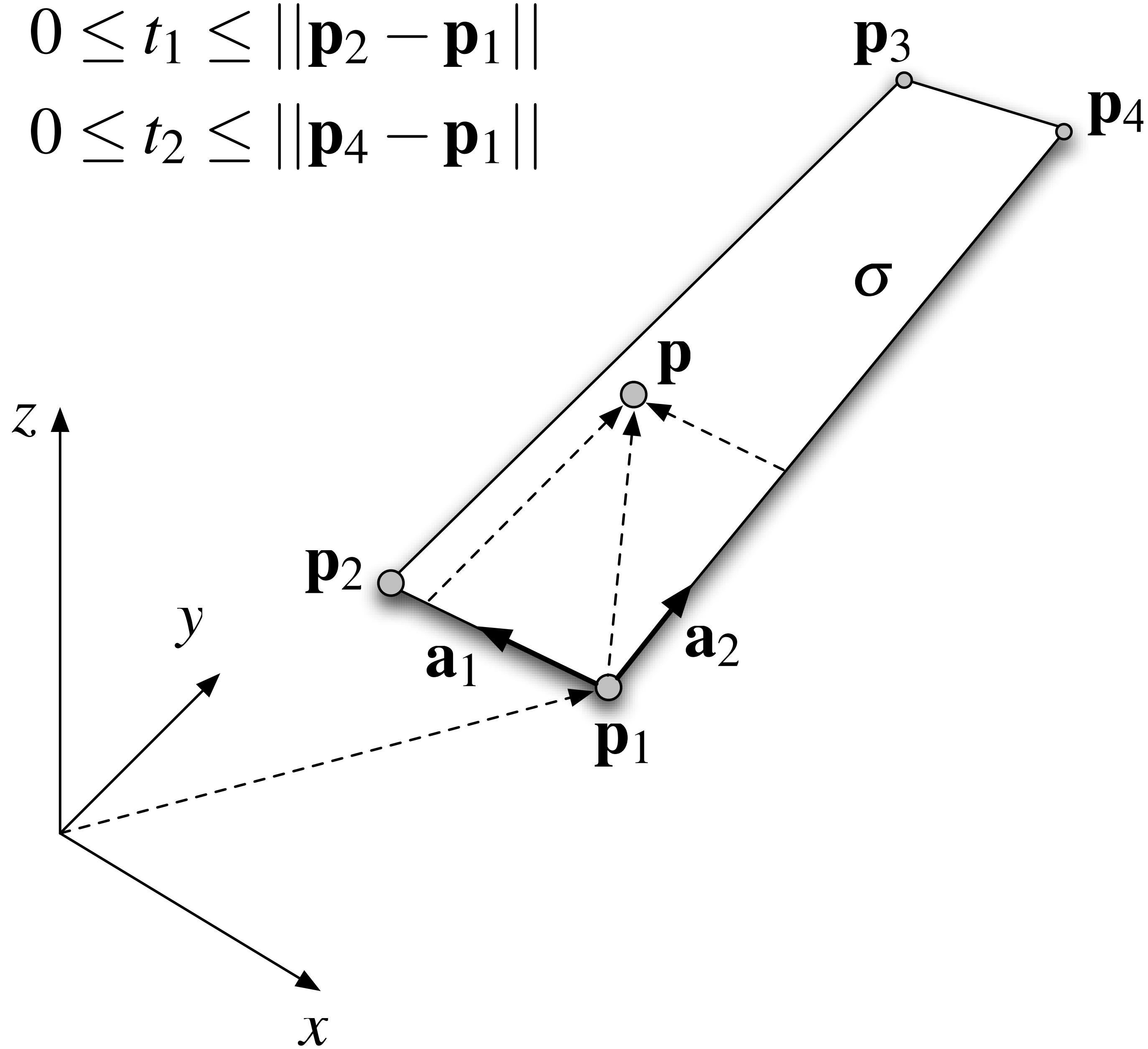


- Twisted-pair wires
- Daisy-chain connection
- ARCNET protocol (token ring bus)

- Wireless communication channels
- Tree topology
- ZigBee (802.15.4)

Capturing the Building Geometry

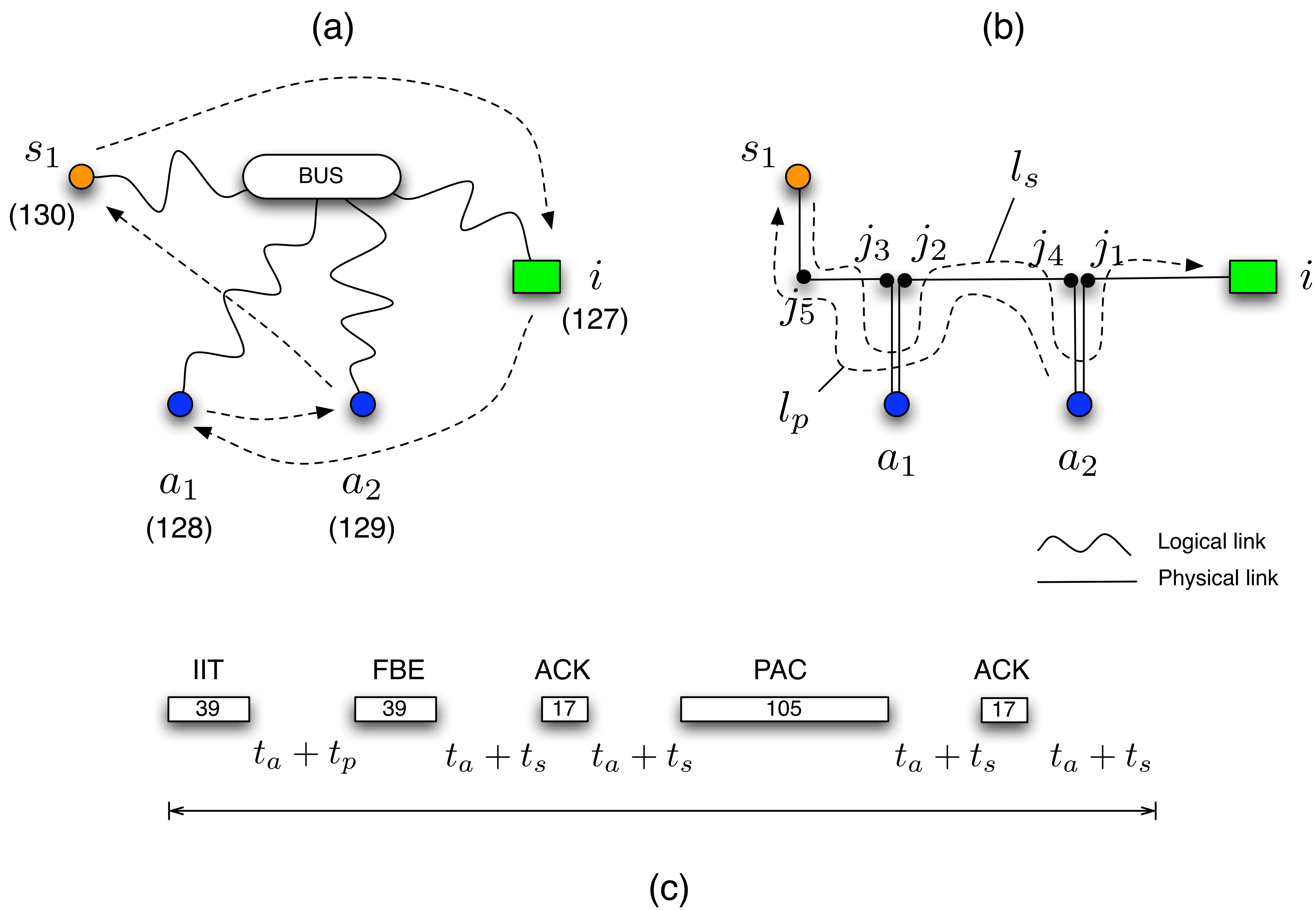
$$\mathbf{p} = \mathbf{p_1} + t_1\mathbf{a}_1 + t_2\mathbf{a}_2$$
$$0 \le t_1 \le ||\mathbf{p}_2 - \mathbf{p}_1||$$
$$0 \le t_2 \le ||\mathbf{p}_4 - \mathbf{p}_1||$$



- 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

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{n} f_j z_j \\
s.t. \quad & \sum_{j=1}^{n} x_{ij} z_j = 1, \forall i \quad \sum_{j=1}^{n} y_{jk} z_j = 1, \forall k \\
& z_j, x_{ij}, y_{jk} \in \{0, 1\}
\end{aligned}
$$

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

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

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

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

j-th chain

chain j contains router i

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

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

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

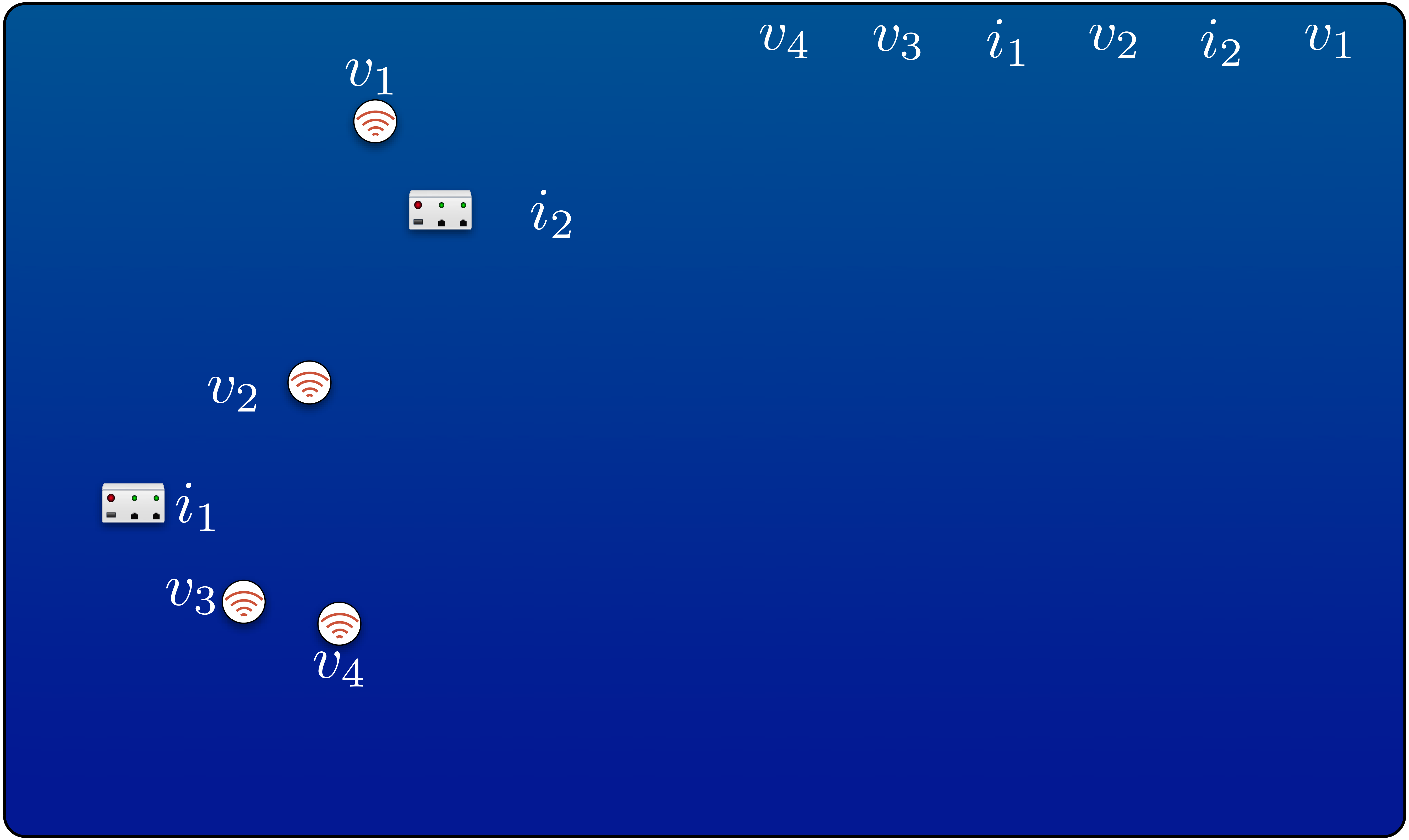
$$z_j, x_{ij}, y_{jk} \in \{0, 1\}$$

j-th chain (z_j)

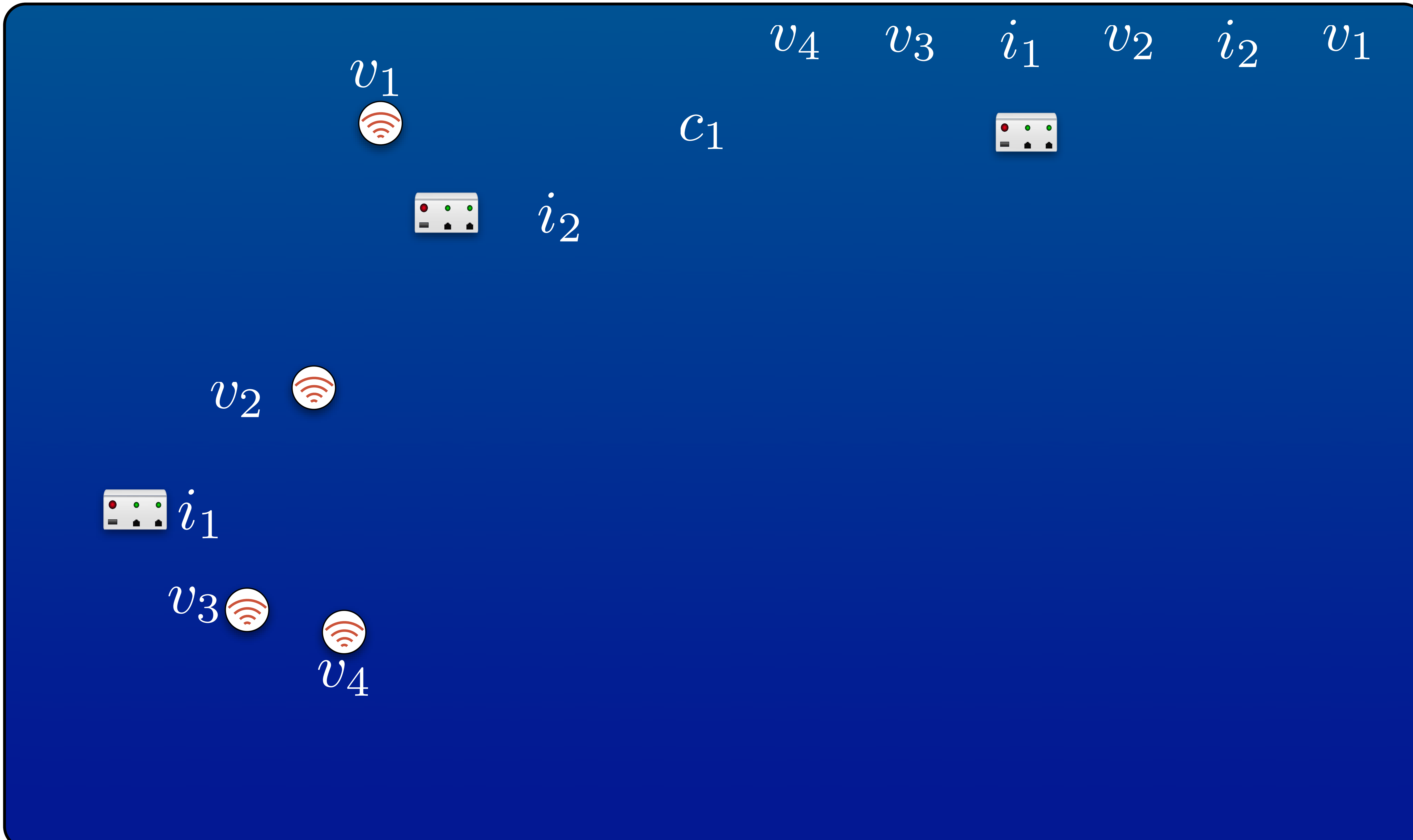
chain j contains router i (x_{ij})

node k belongs to chain j (y_{jk})

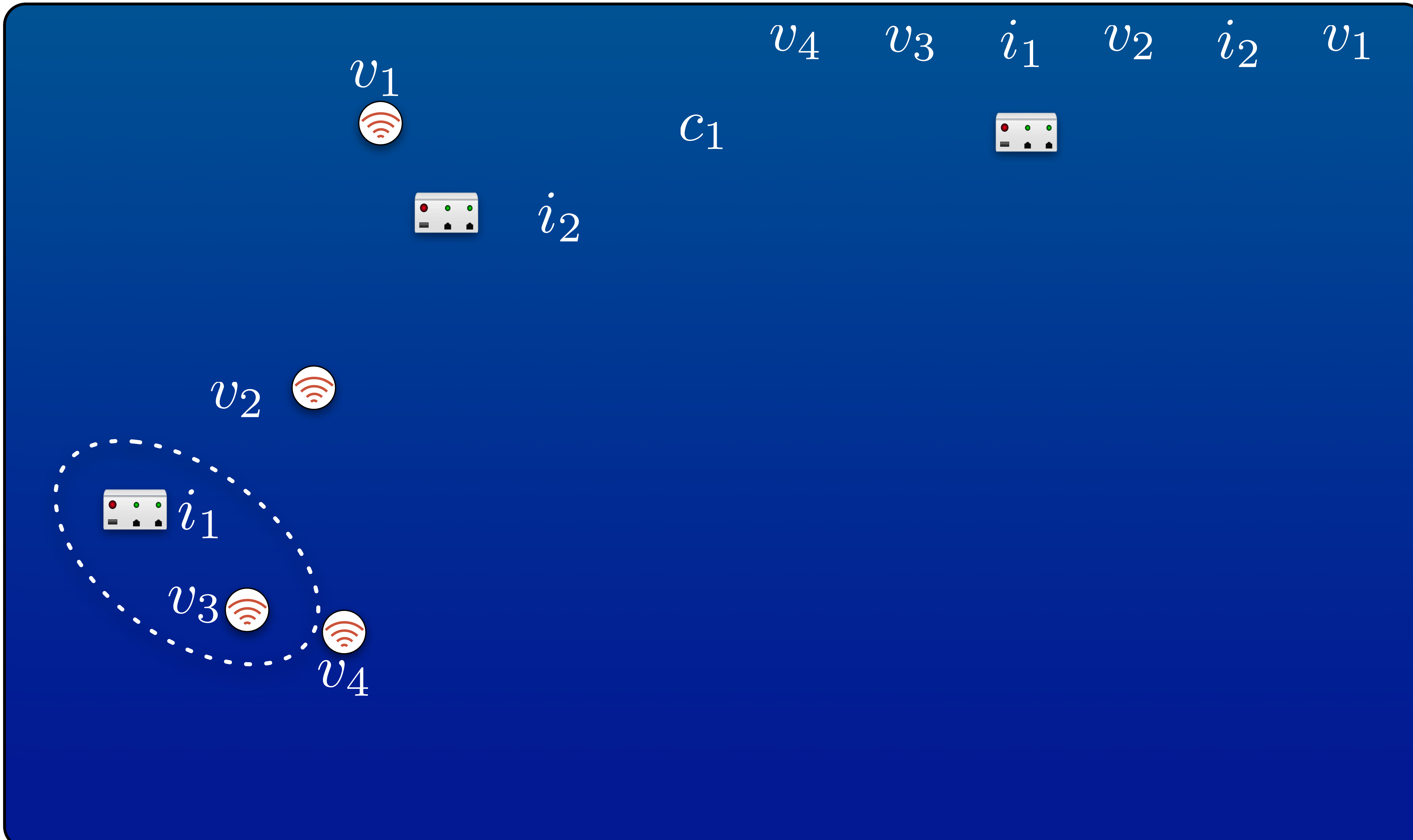
Algorithms: elcarO ehT



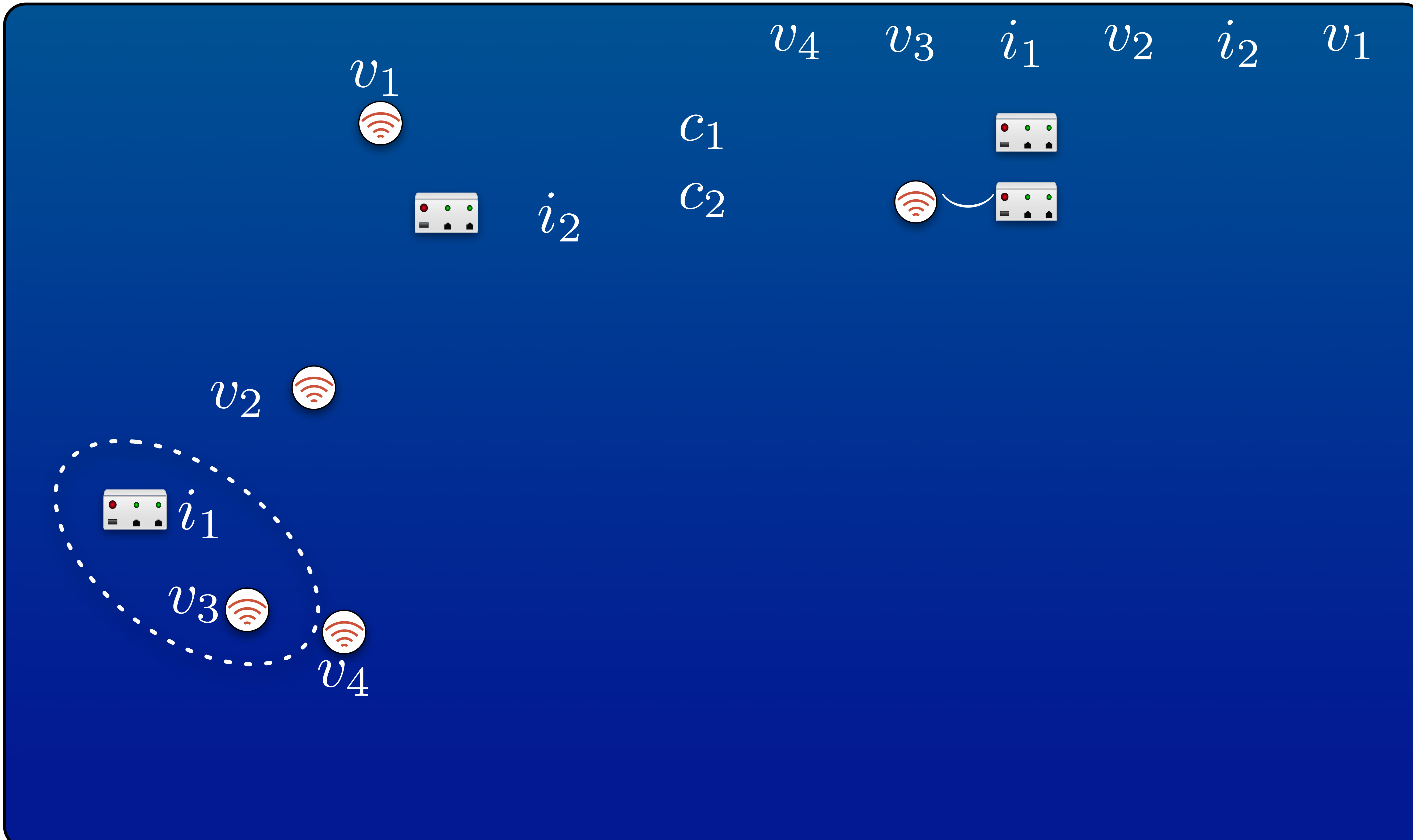
Algorithms: elcarO ehT



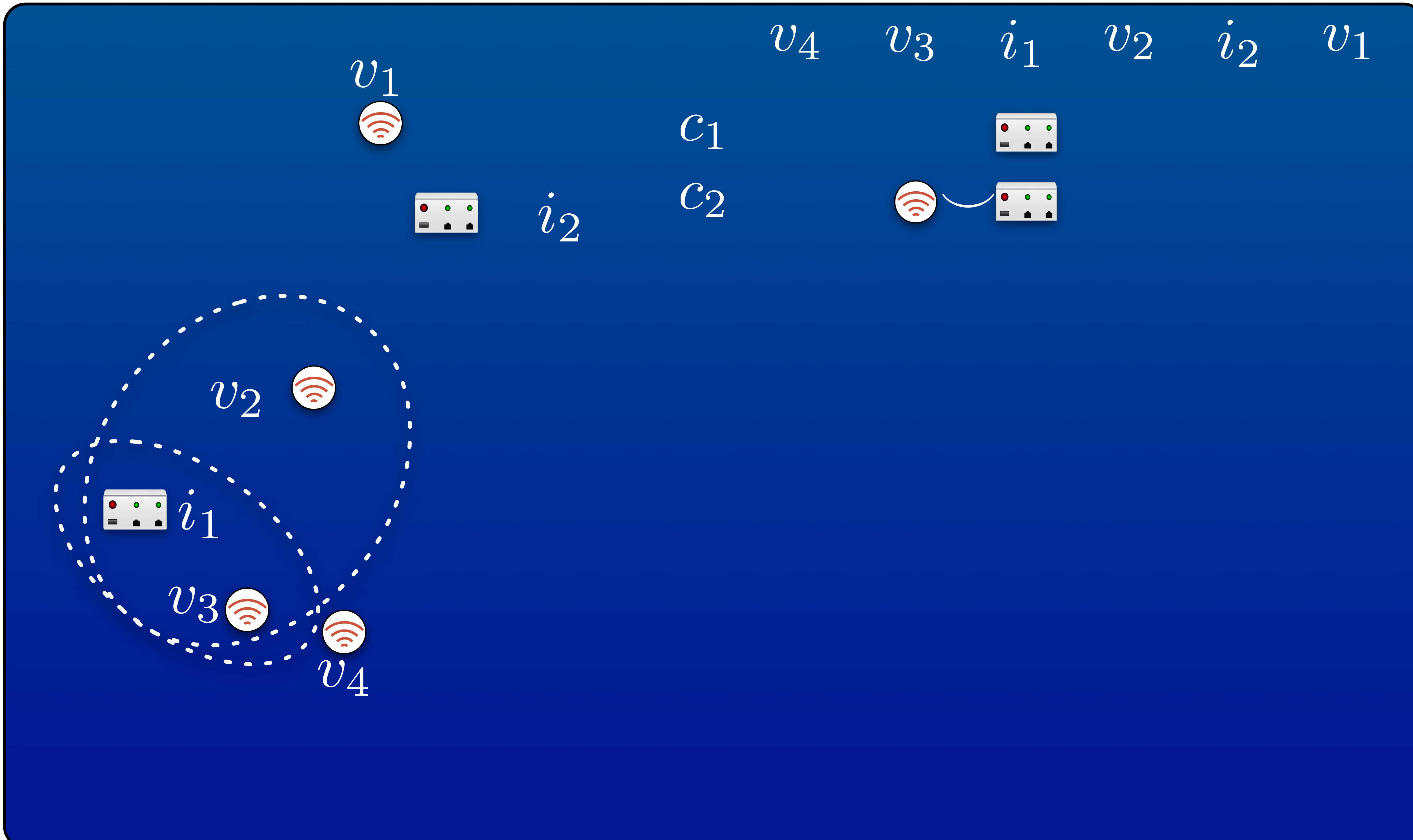
Algorithms: elcarO ehT



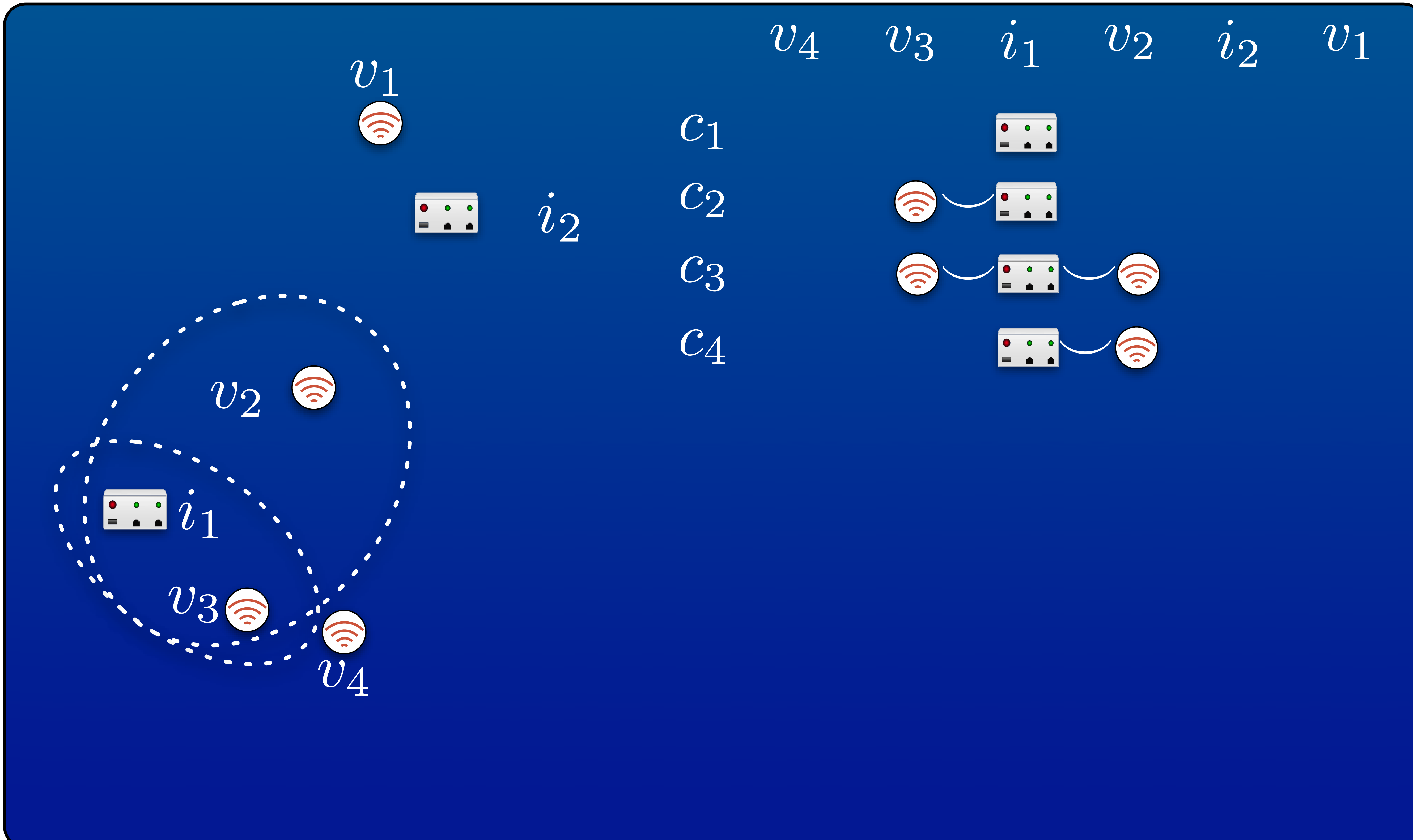
Algorithms: elcarO ehT



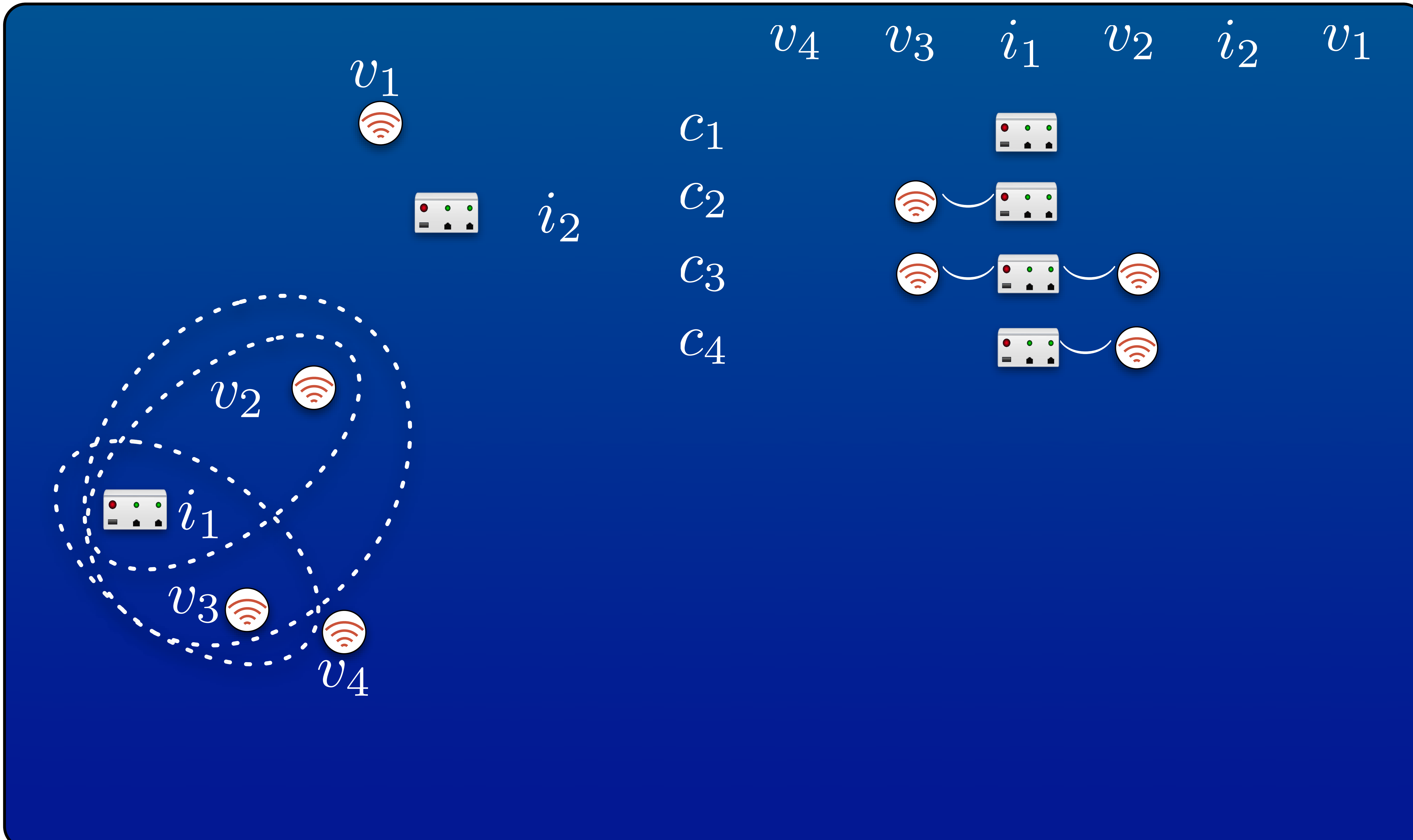
Algorithms: elcarO ehT



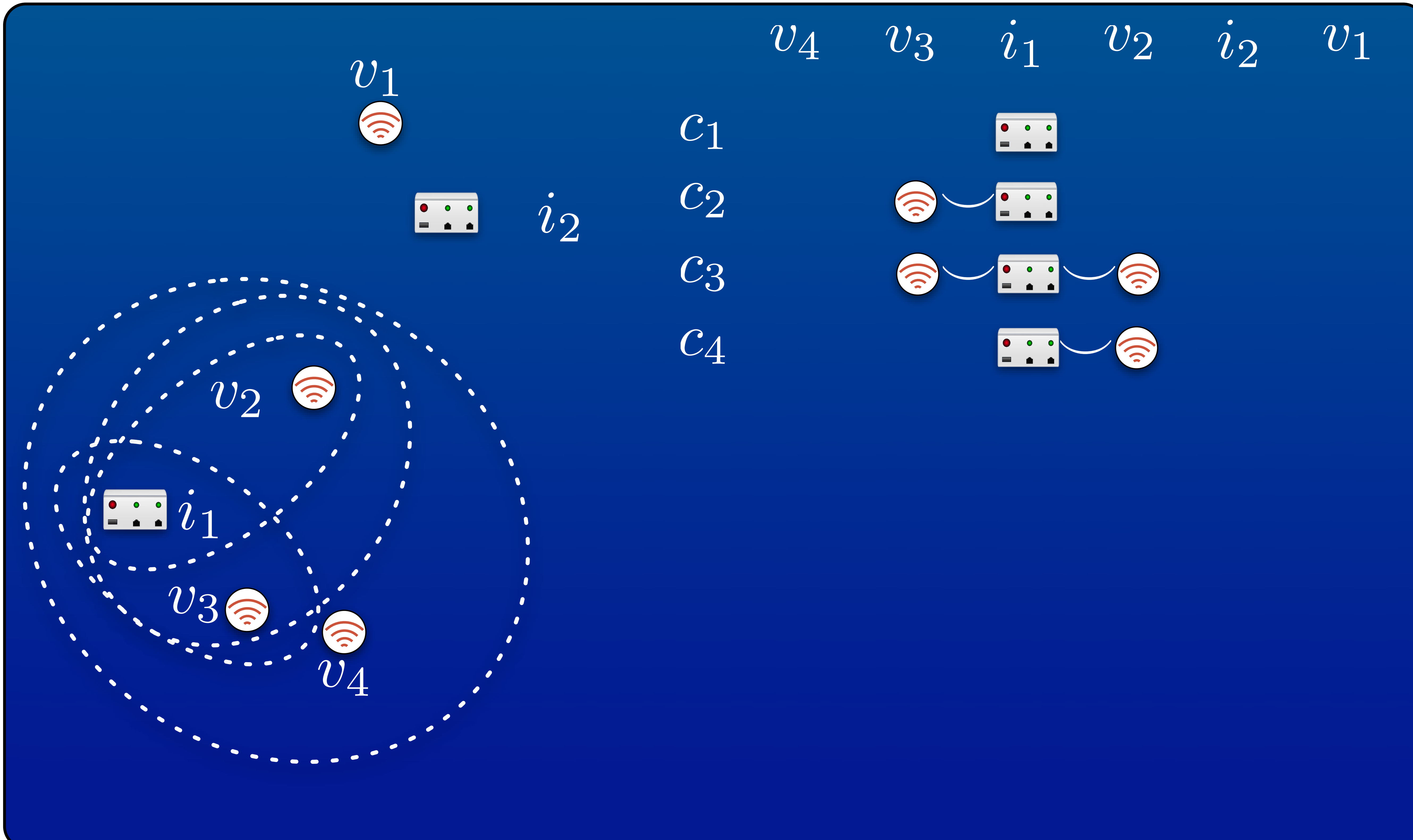
Algorithms: elcarO ehT



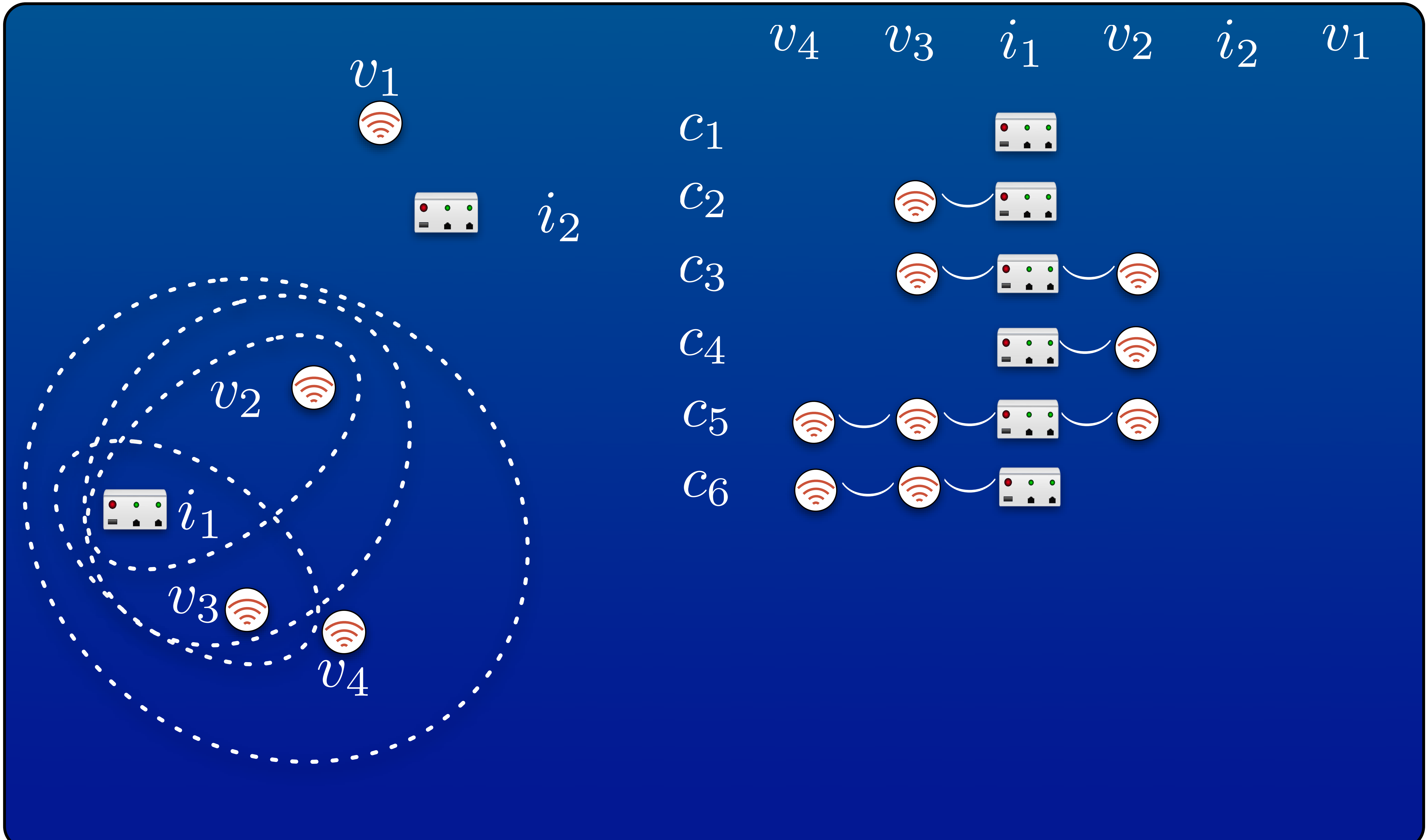
Algorithms: The Oracle



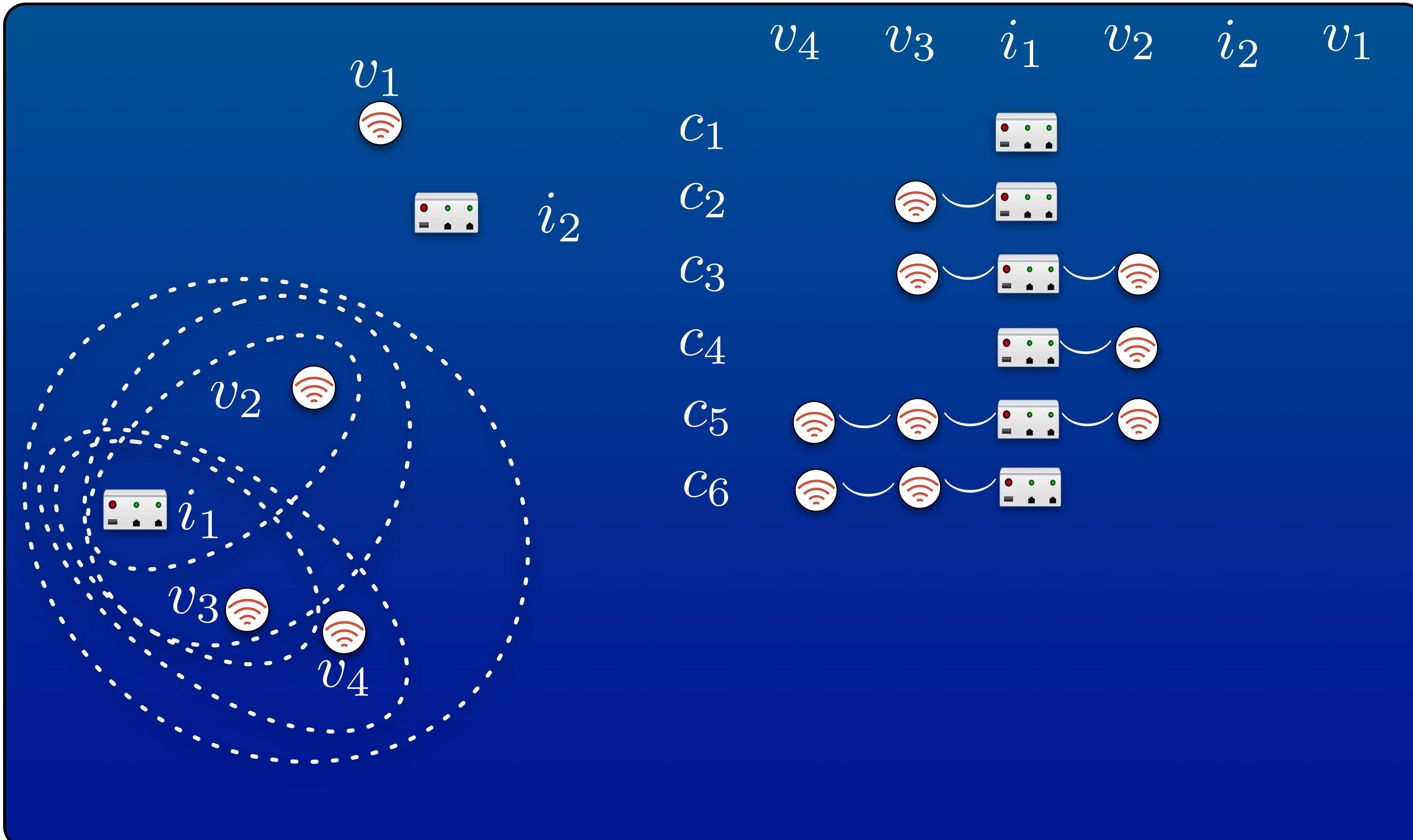
Algorithms: elcarO ehT



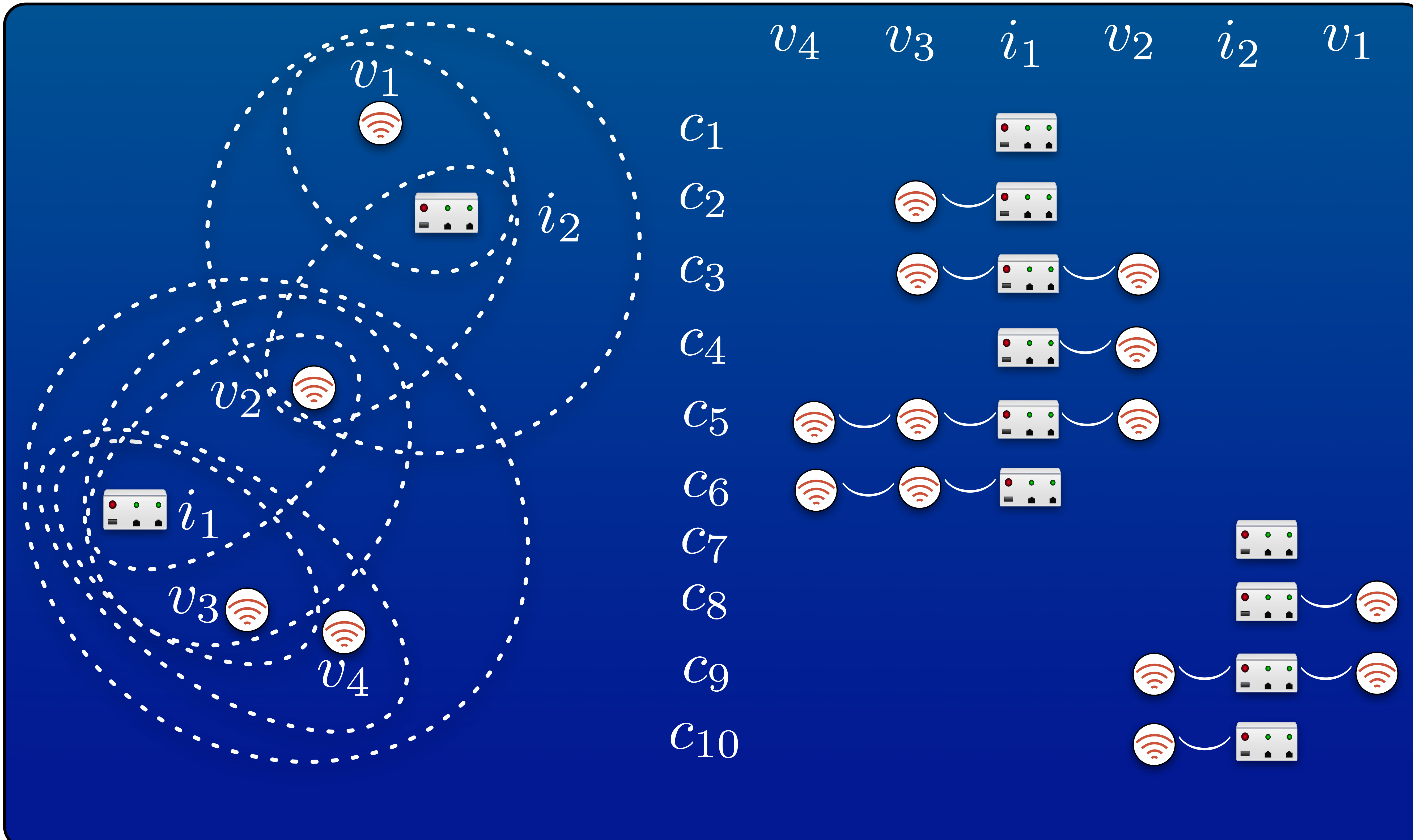
Algorithms: elcarO ehT



Algorithms: elcarO ehT



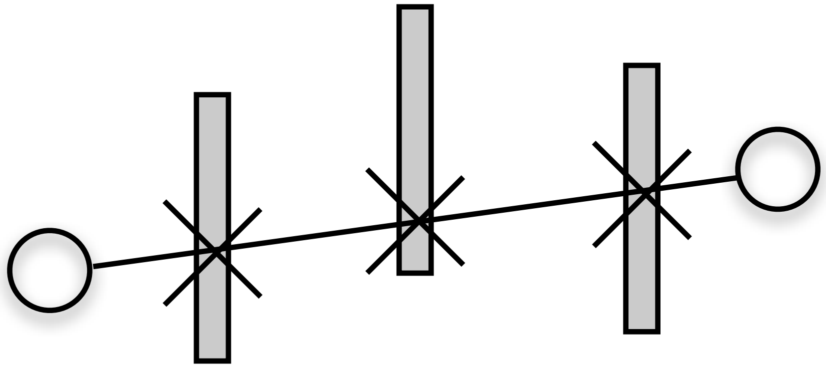
Algorithms: elcarO ehT



Modeling the ZigBee Protocol

Modeling the ZigBee Protocol

Physical Layer

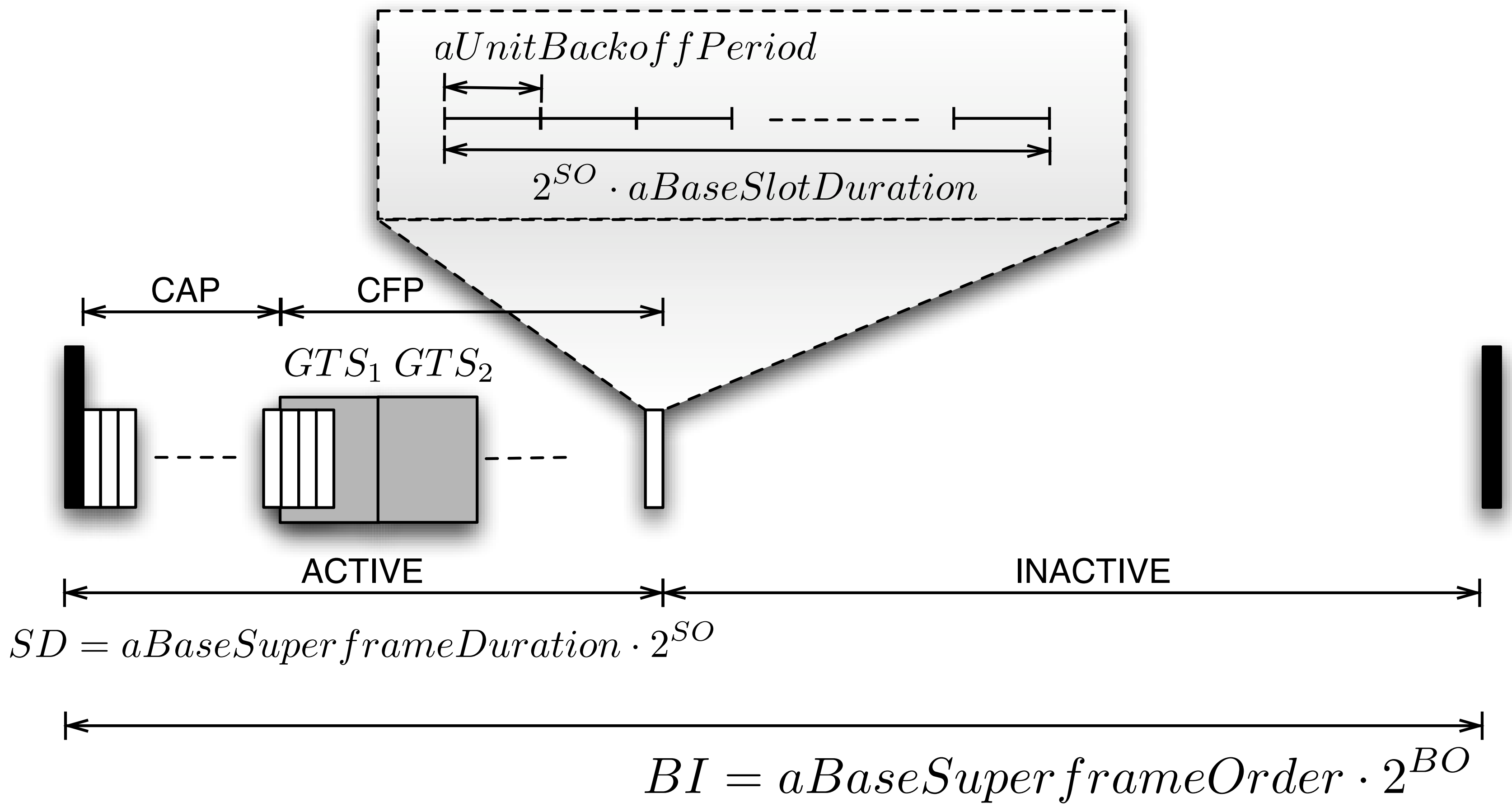


$$P_b = f(SNR)$$

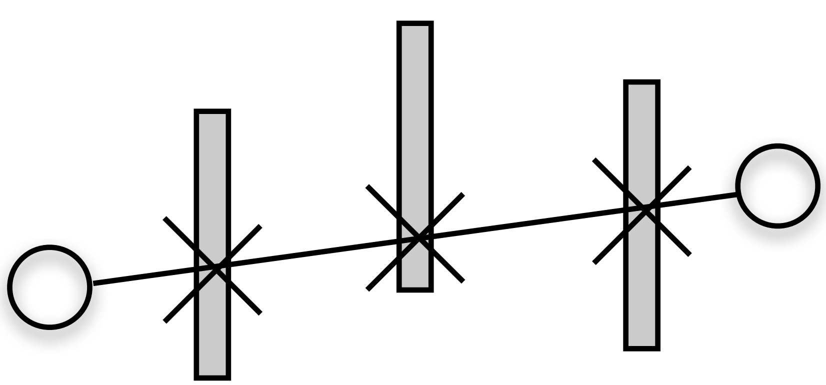
$$L = L_1 + 20\log_{10} r + n_f a_f + n_w a_w$$

Modeling the ZigBee Protocol

MAC layer



Physical Layer

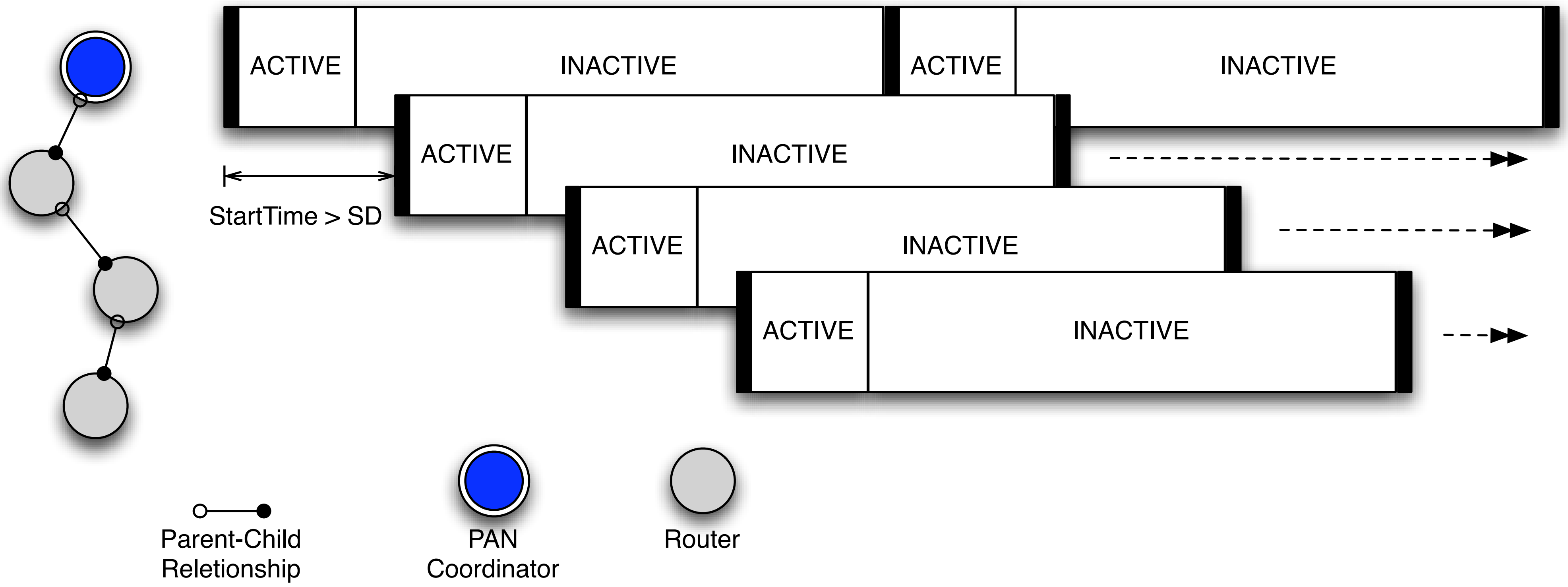


$$P_b = f(SNR)$$

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

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

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

j is the parent of i

$s.t.$

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

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.

Rules

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

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

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

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

i-th router

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

j is the parent of i

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

$s.t.$

Rules

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

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

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

$s.t.$

Rules

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

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

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.

Rules

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

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

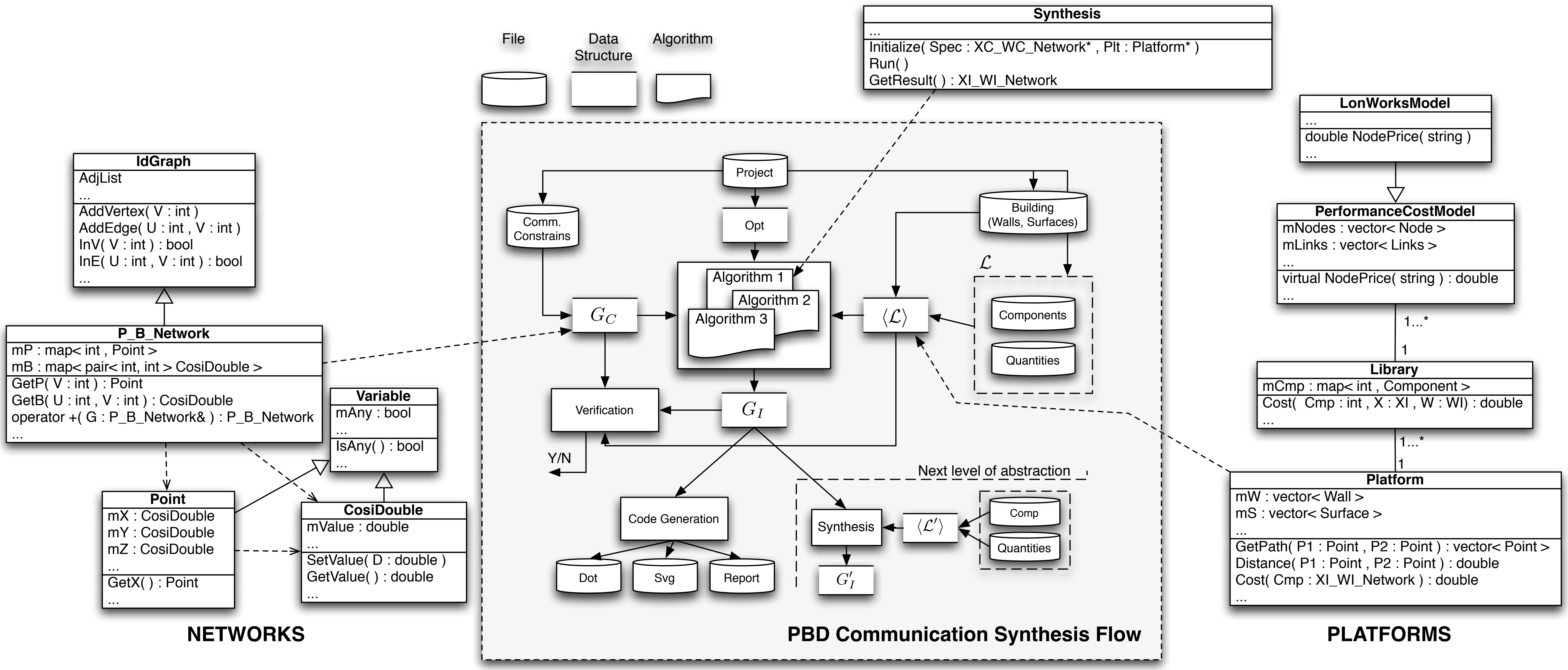
9. $\sum_q y_{ijq}(b_q + O) \leq b_{max} \quad \forall i, j \in N \cup M$
10. $\sum_{i \in M} x_i \leq n_{max}$

Latency constraints

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

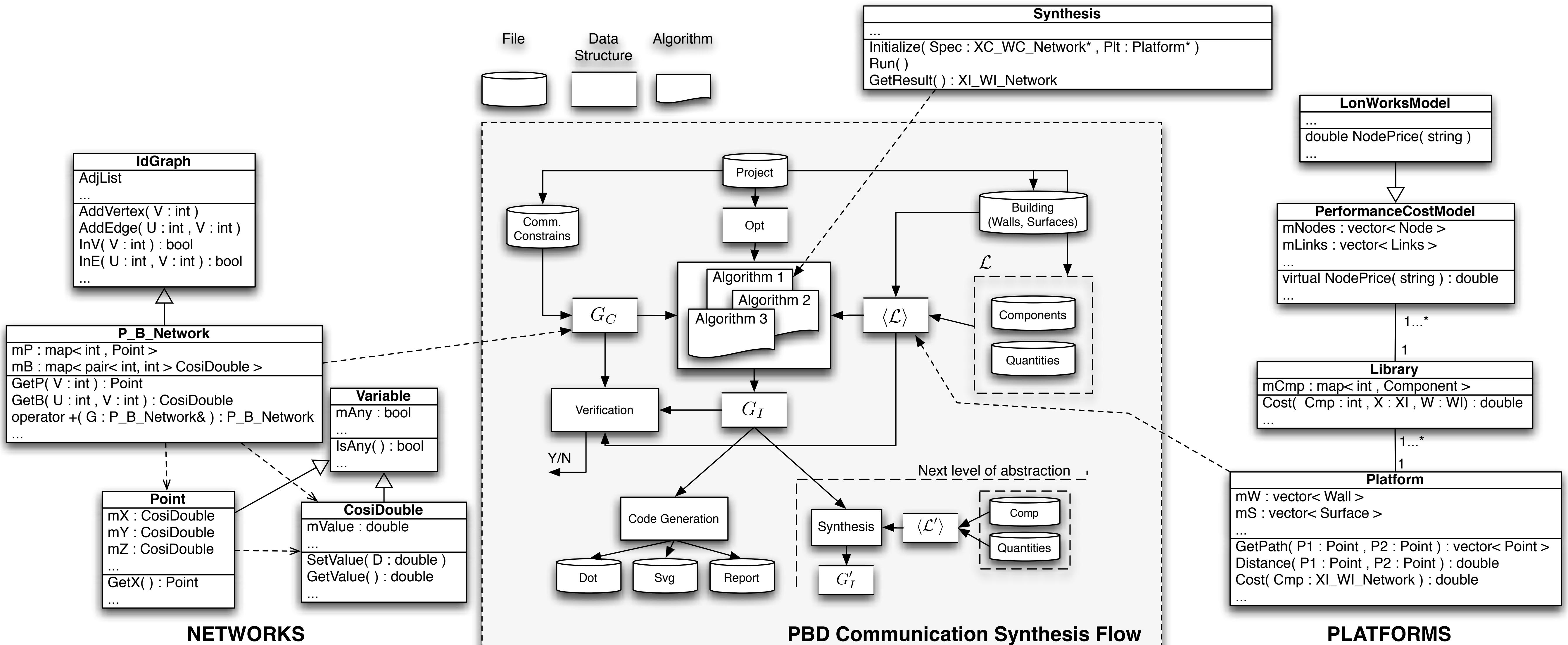
PER constraints

12. $\sum_{ij} y_{ijq} \log p'(i, j) \leq \log(1 - p_q) \quad \forall q \in Q$
13. $x_i,\; e_{ij},\; y_{ijq} \in \{0, 1\} \quad \forall i, j \in N \cup M, \forall q \in Q$



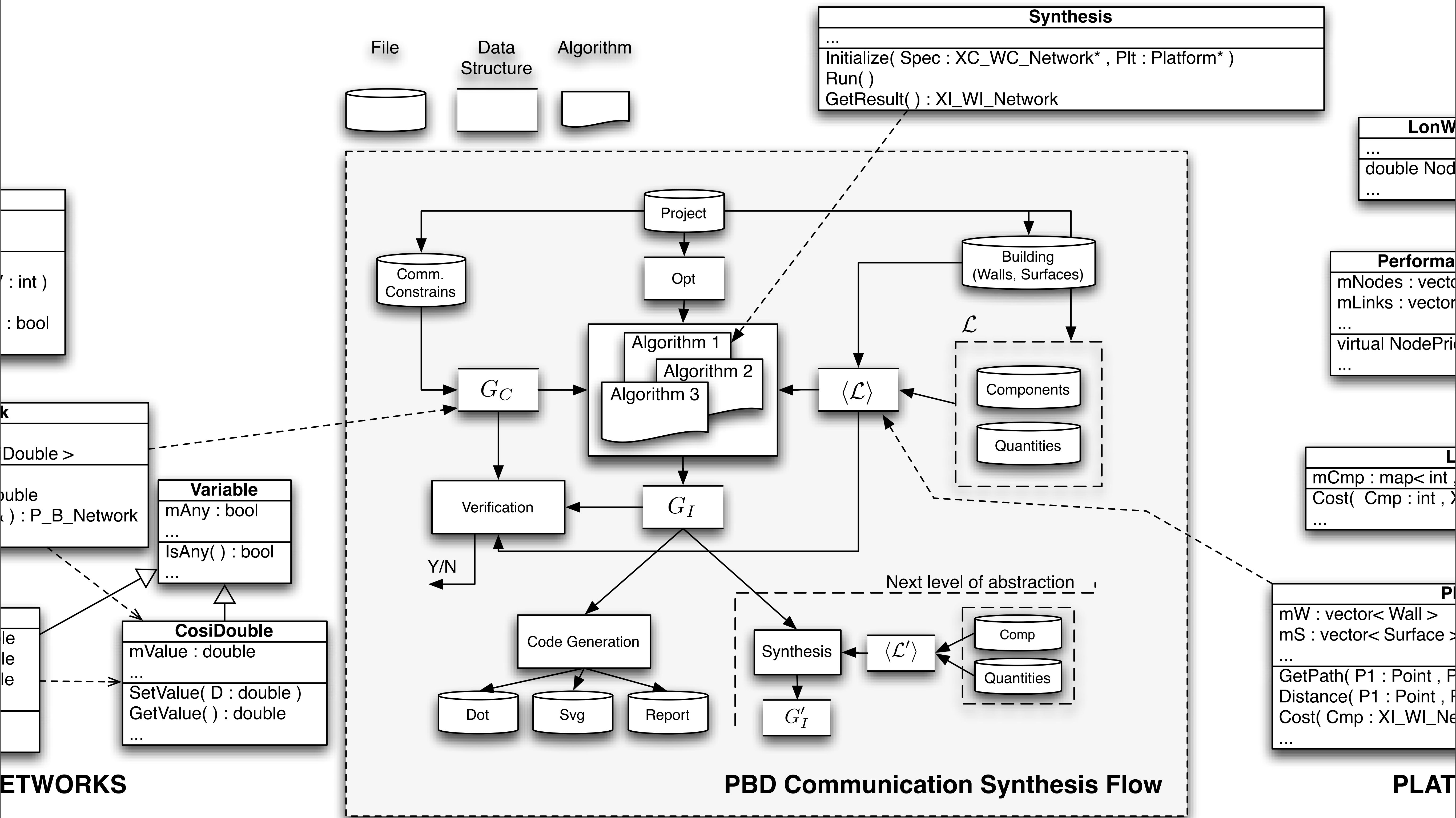
- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
- .Capture network components
- .Automatic synthesis

COSI-BAS



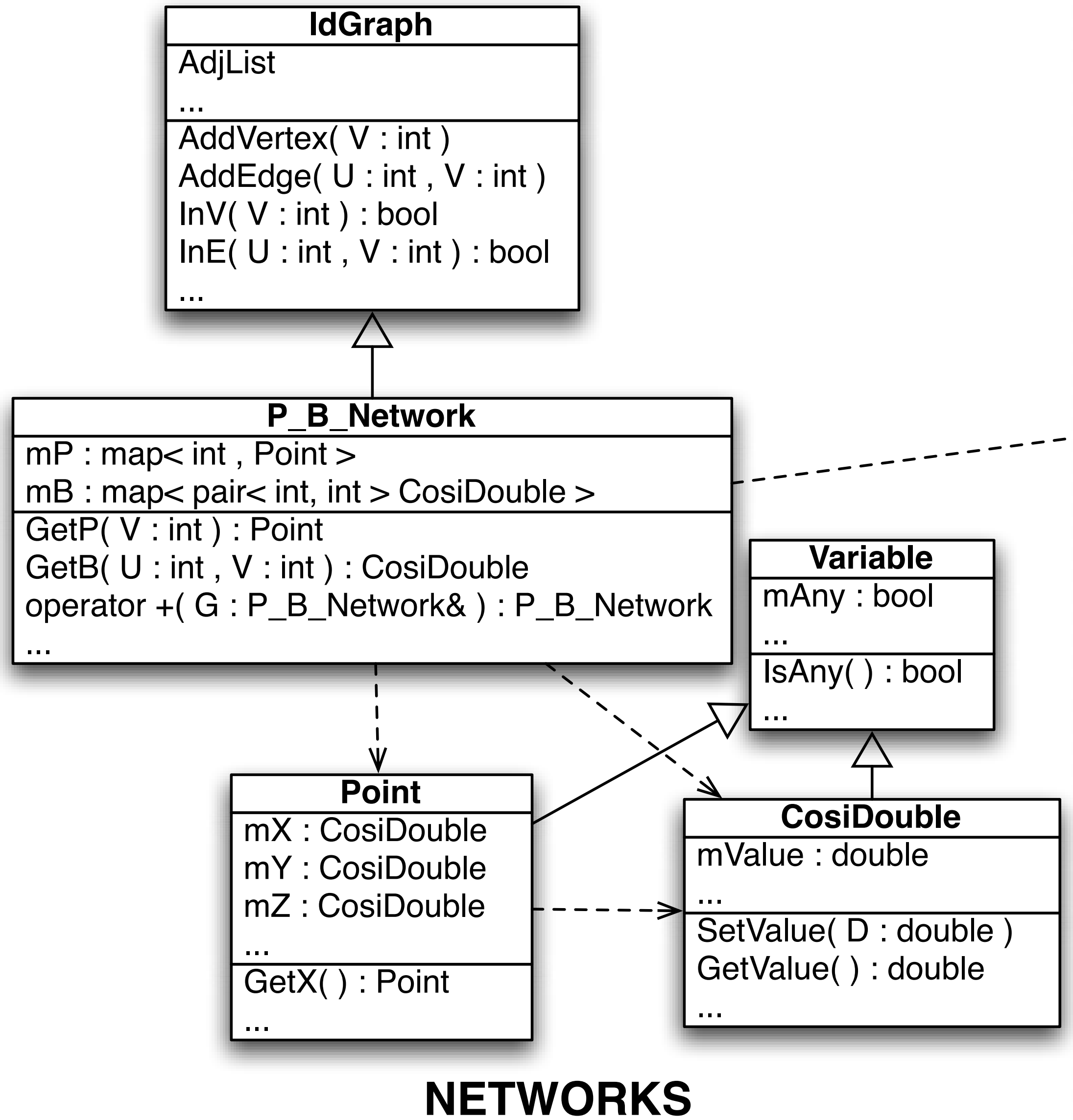
COSI-BAS

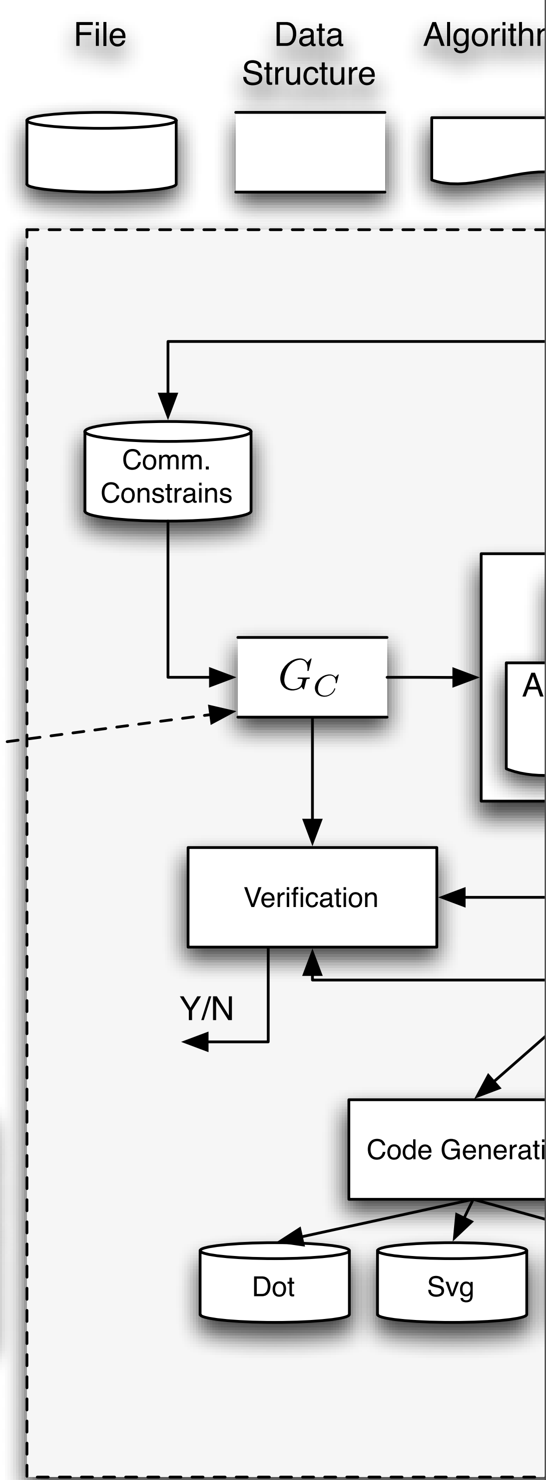
- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
- .Capture network components
- .Automatic synthesis



COSI-BAS

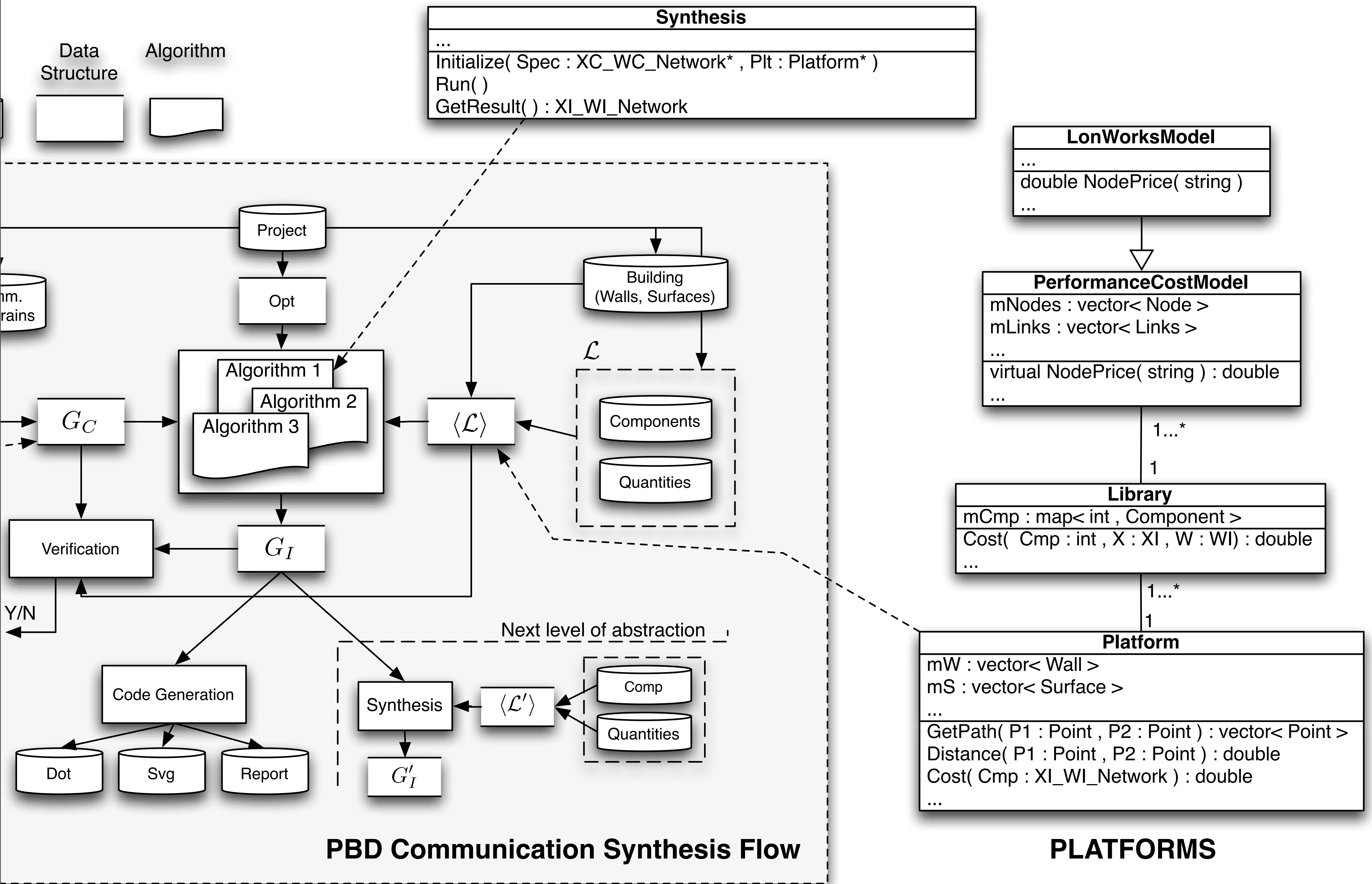
- .Platform-Based Methodology
- .Capture end-to-end QoS
- .Capture building structure
- .Capture network components
- .Automatic synthesis





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

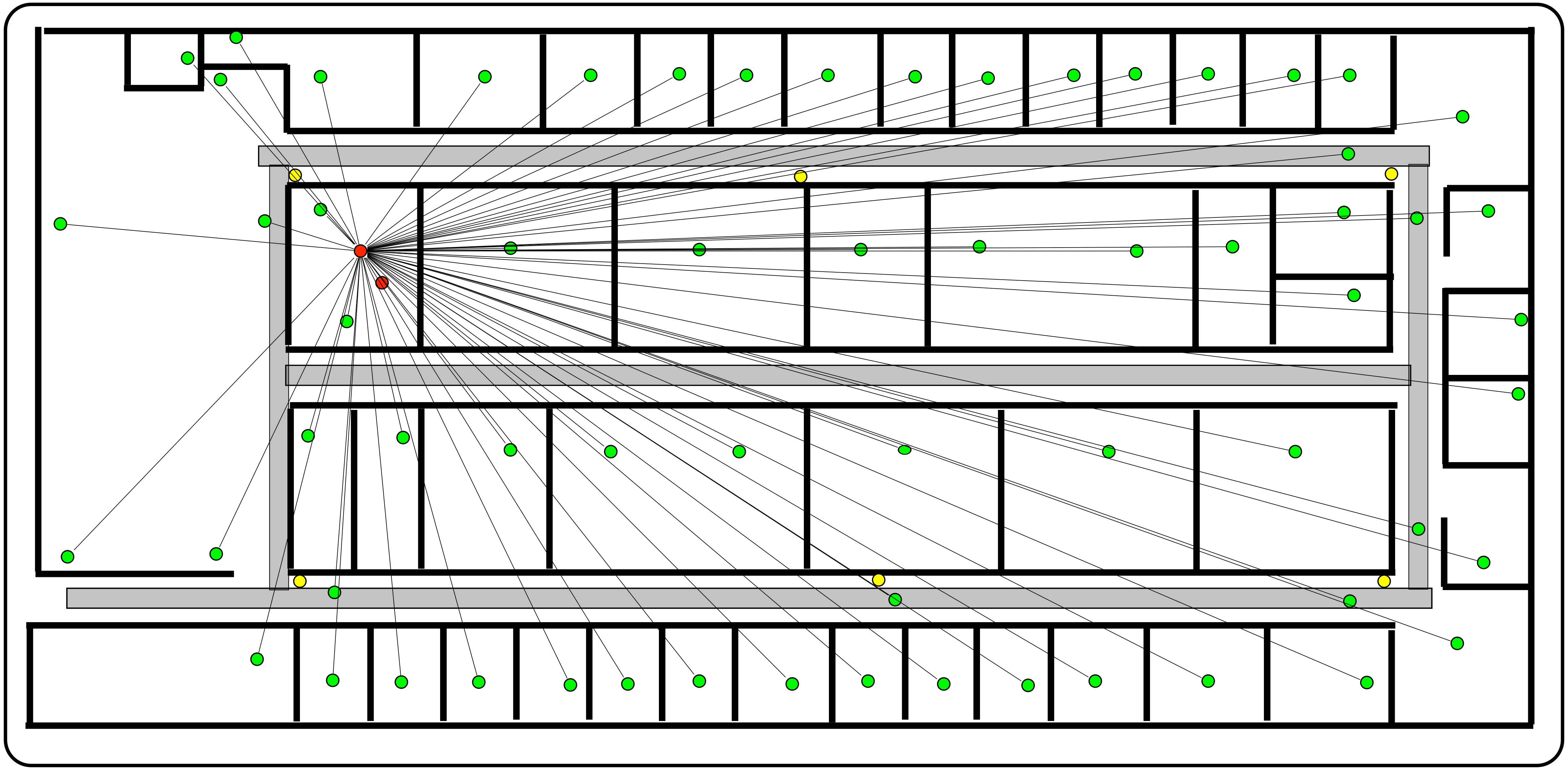
COSI-BAS



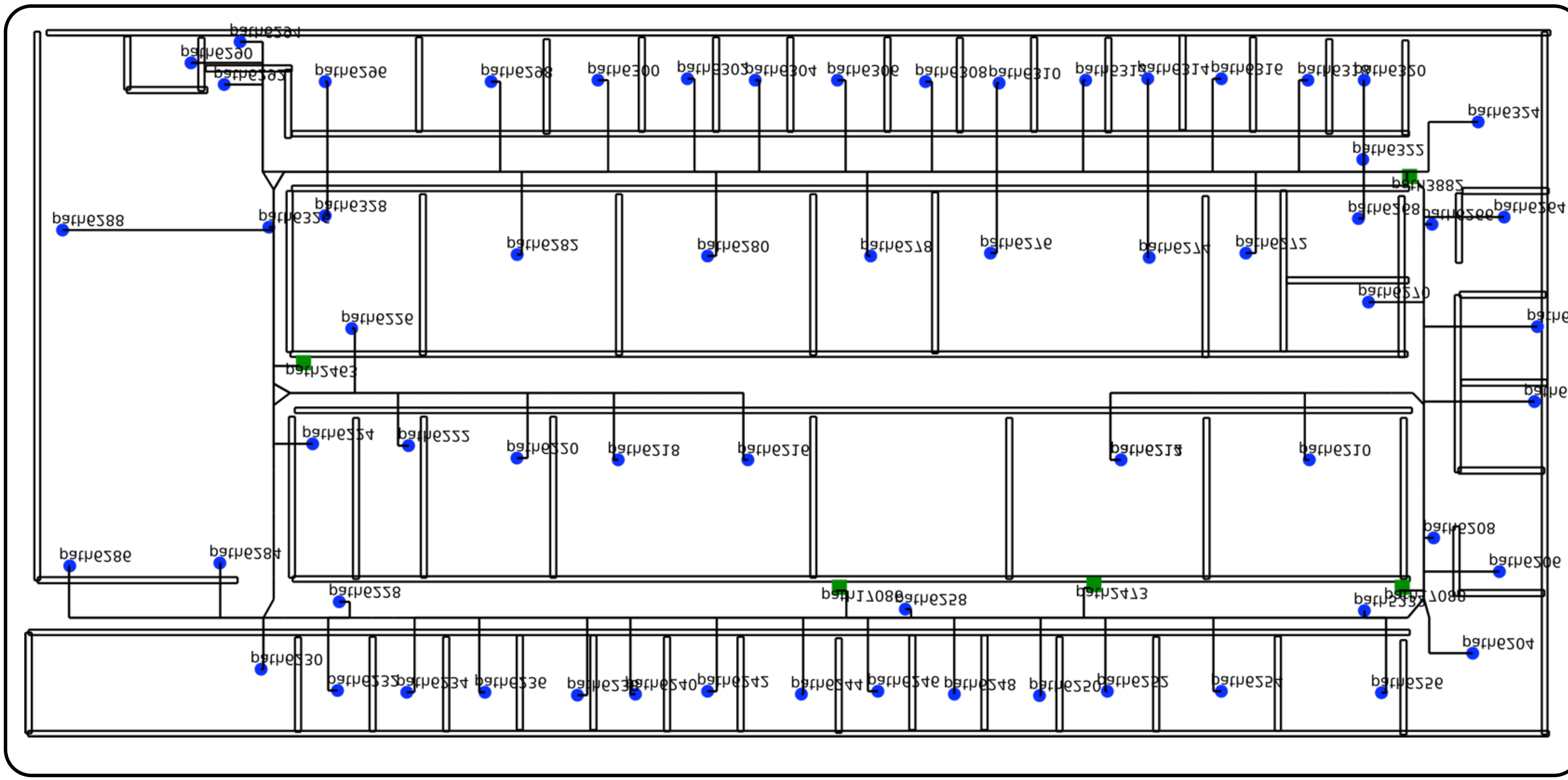
COSI-BAS

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

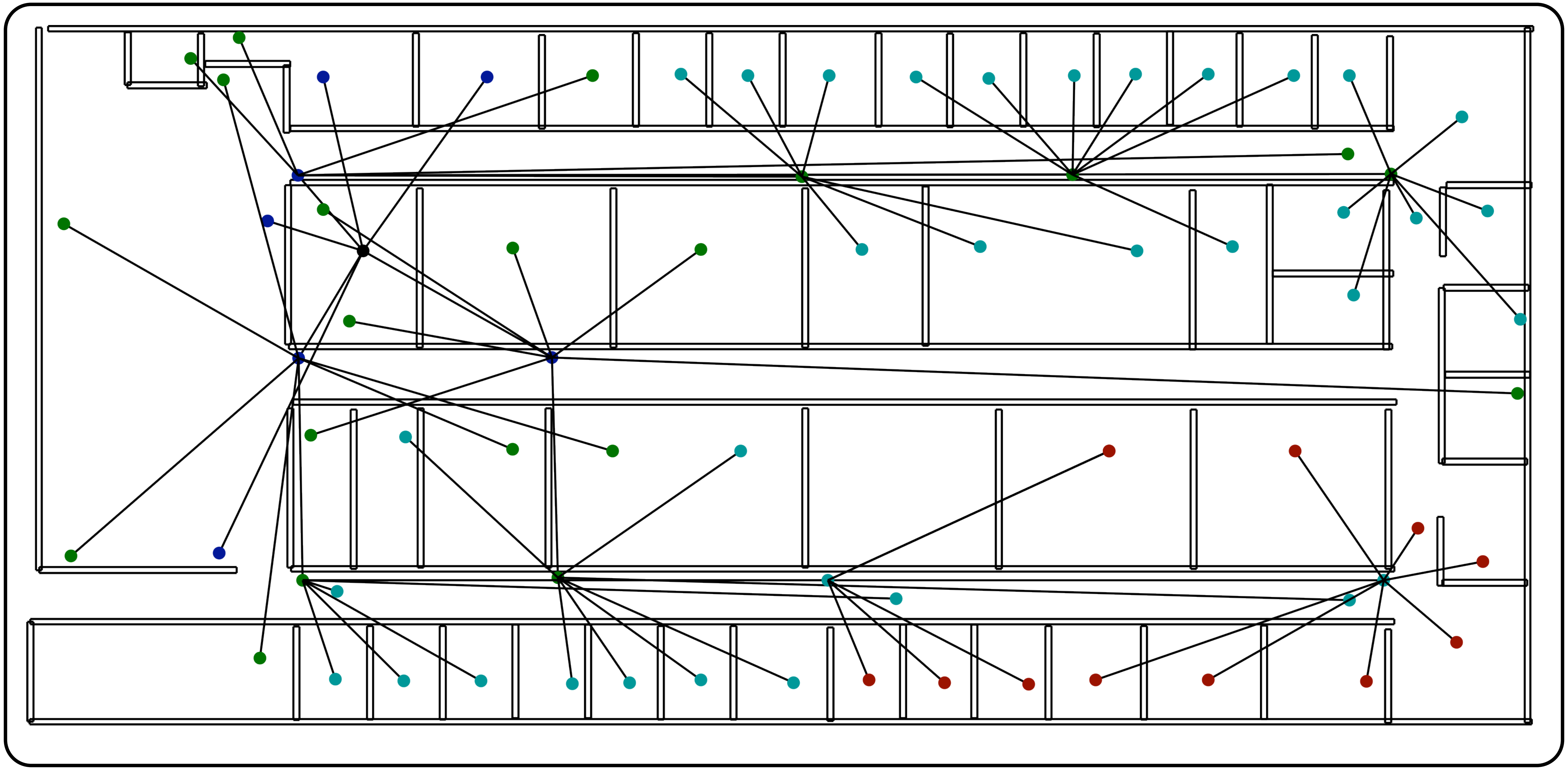
Examples



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

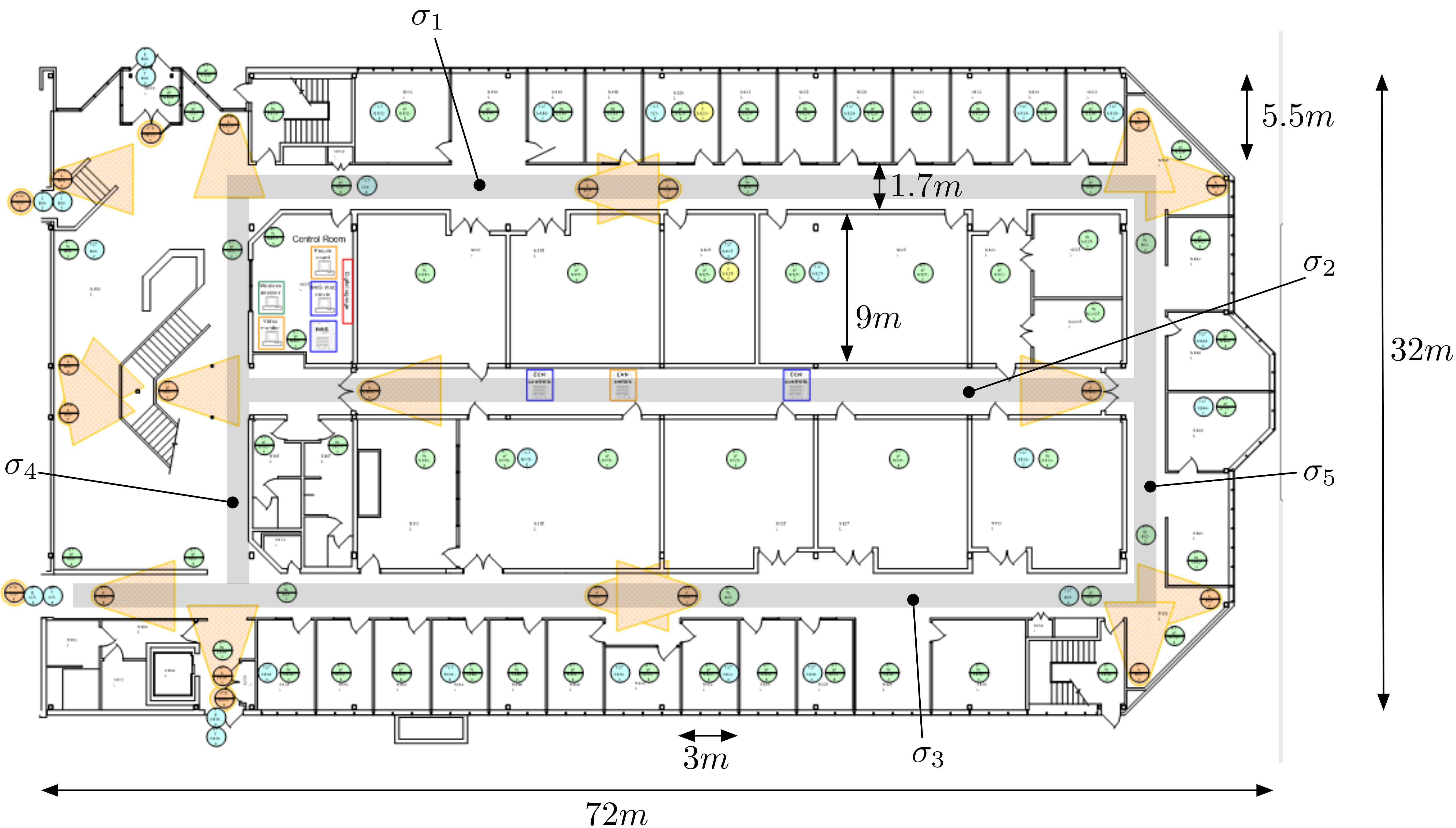
New

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

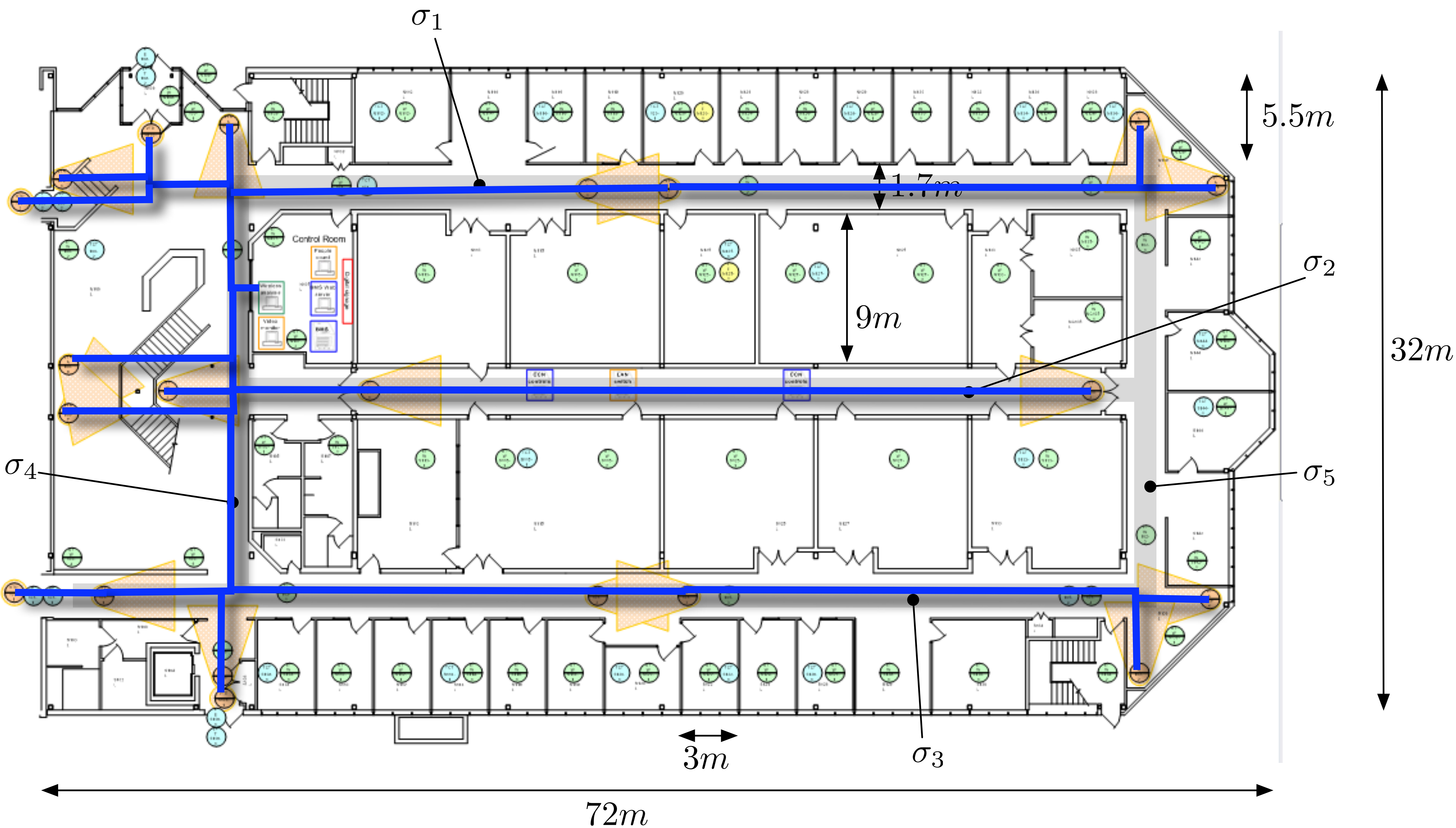
Retrofit

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

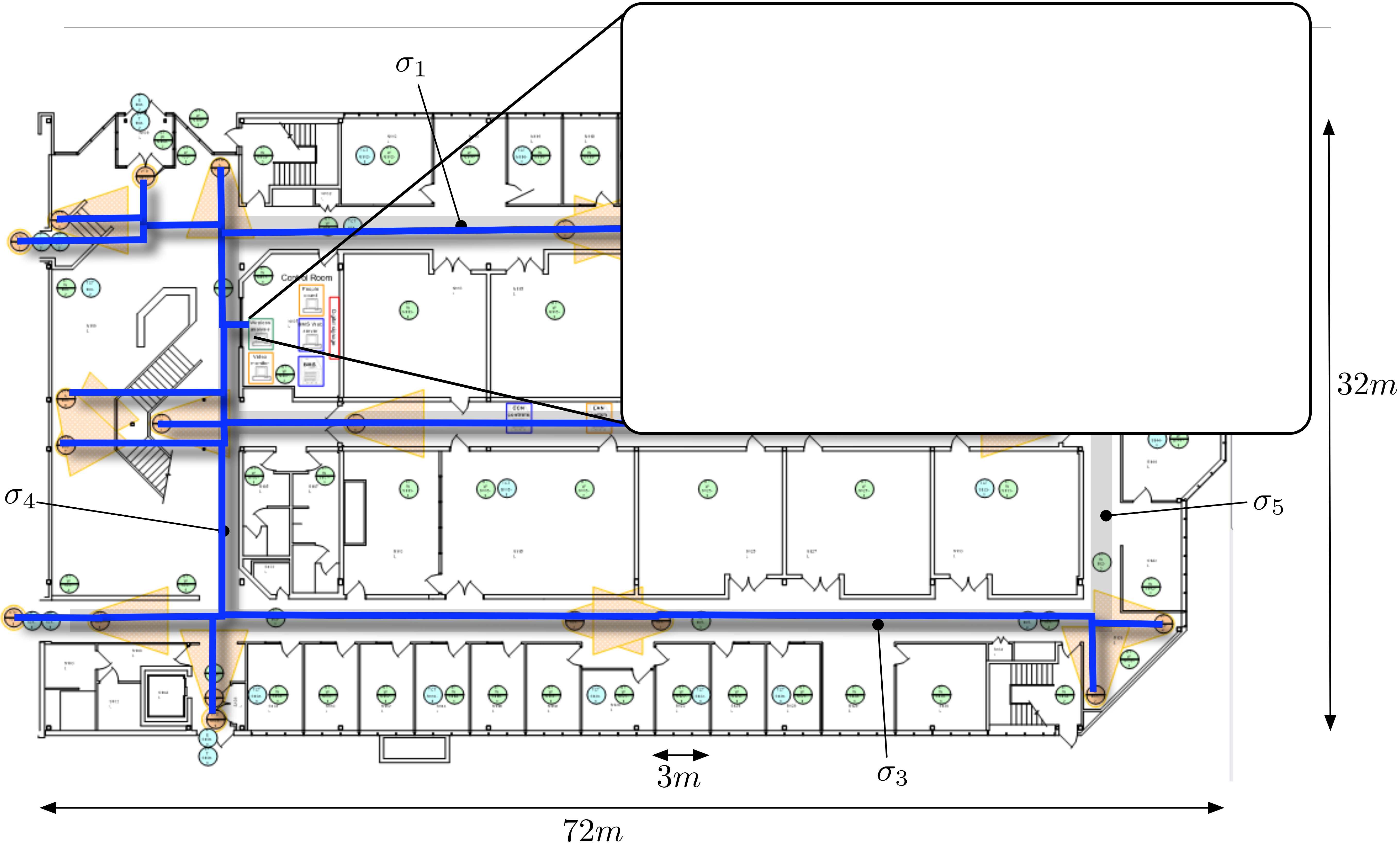
Keep Guaranteeing Properties: Constraint Propagation



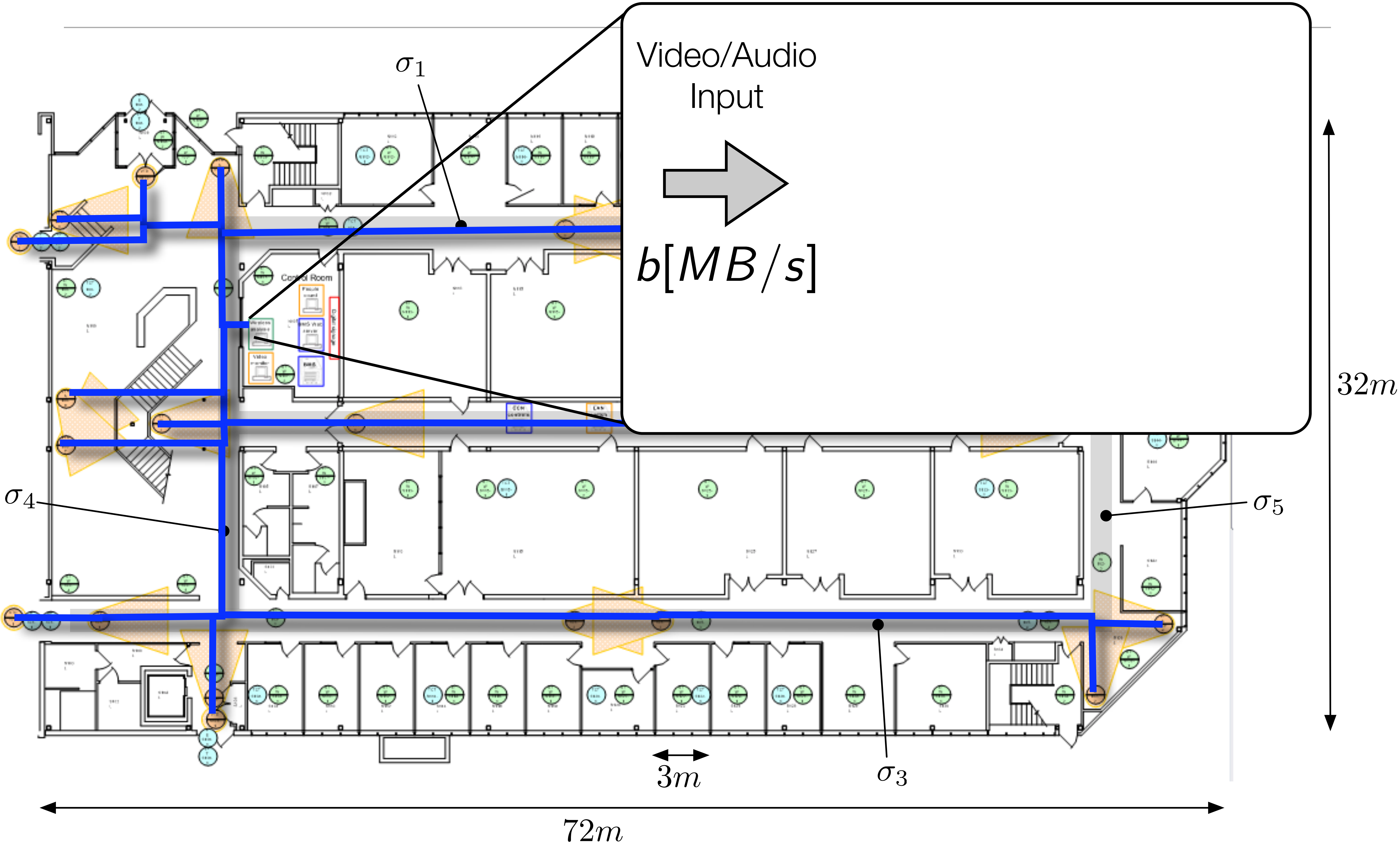
Keep Guaranteeing Properties: Constraint Propagation



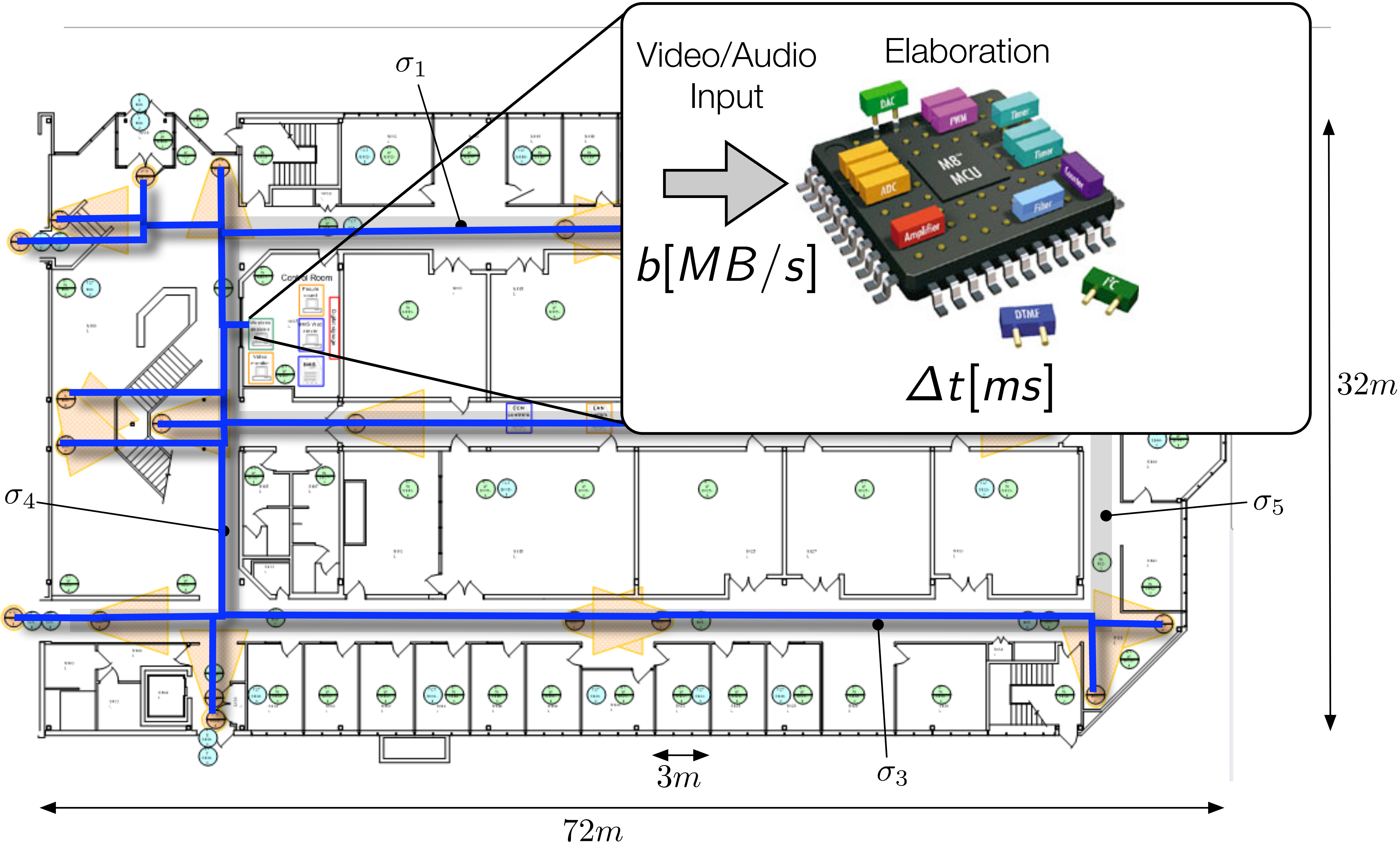
Keep Guaranteeing Properties: Constraint Propagation



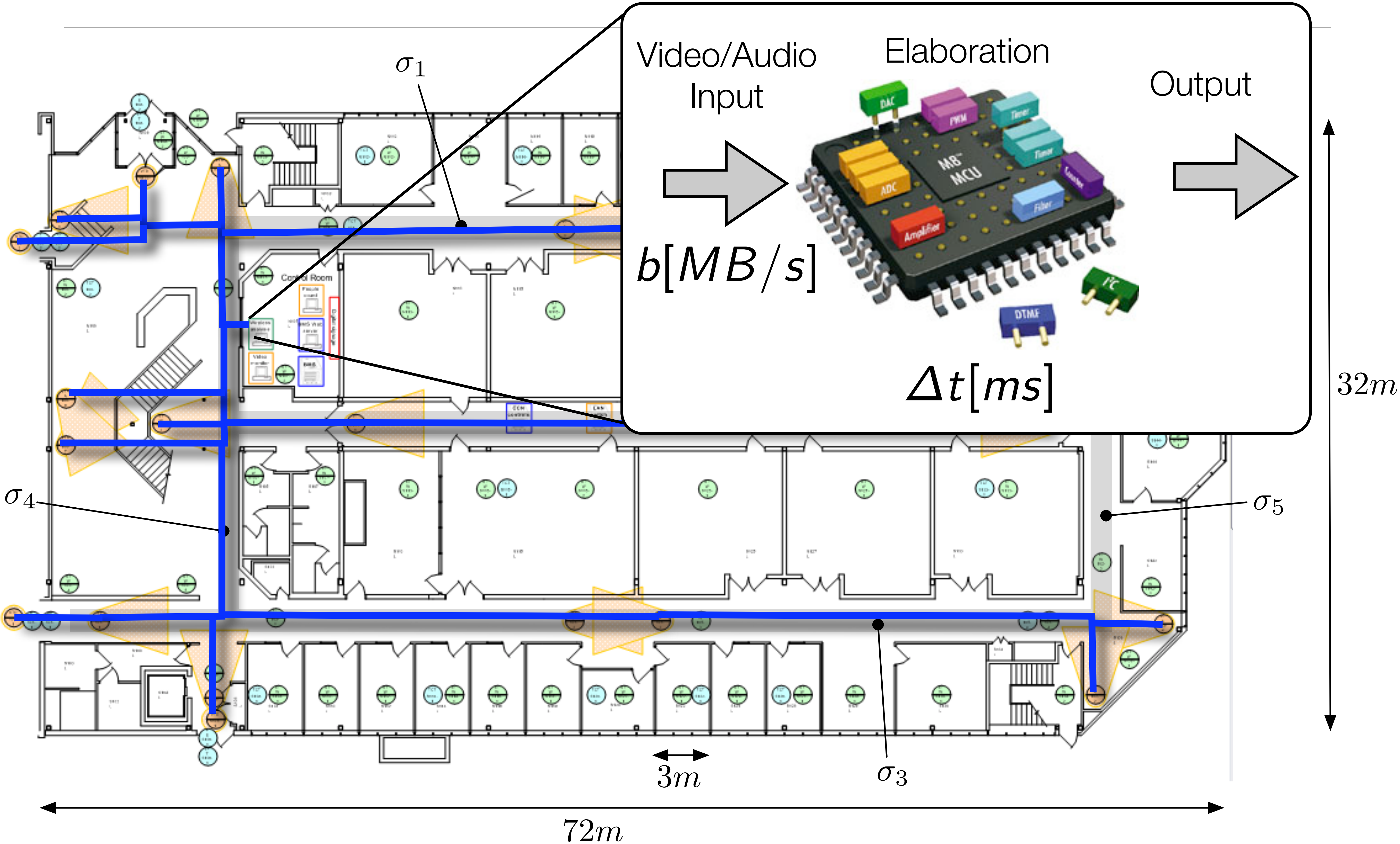
Keep Guaranteeing Properties: Constraint Propagation



Keep Guaranteeing Properties: Constraint Propagation

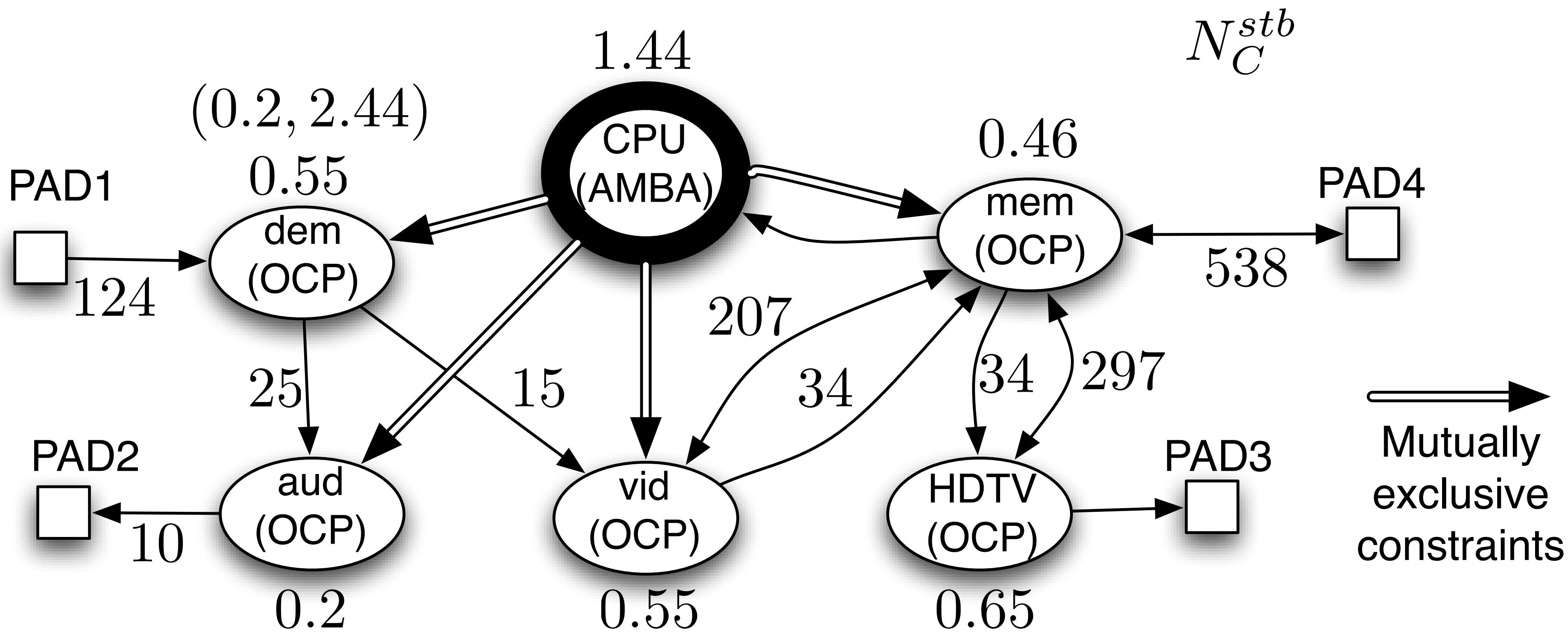


Keep Guaranteeing Properties: Constraint Propagation

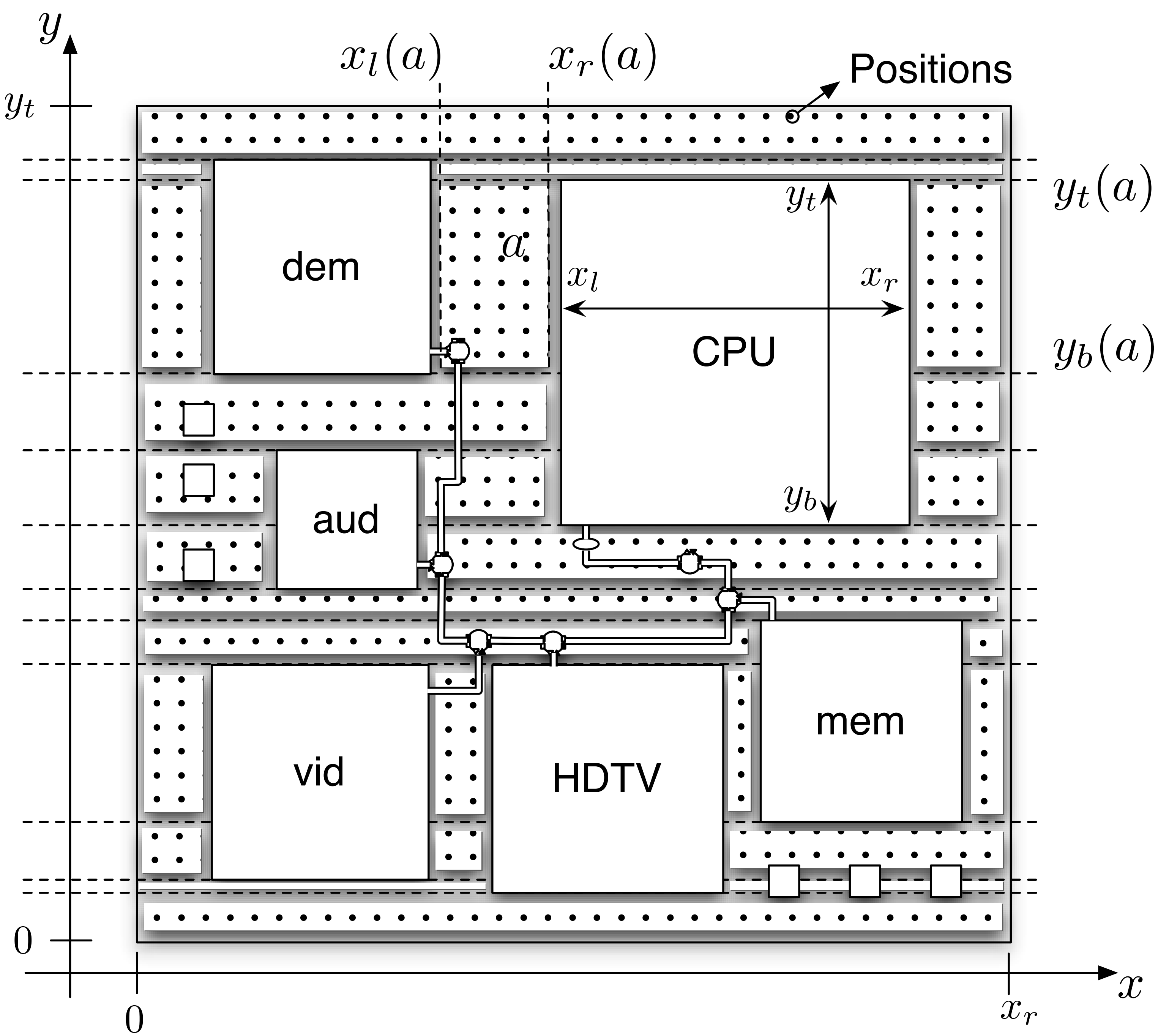


Communication Synthesis for On-Chip Communication

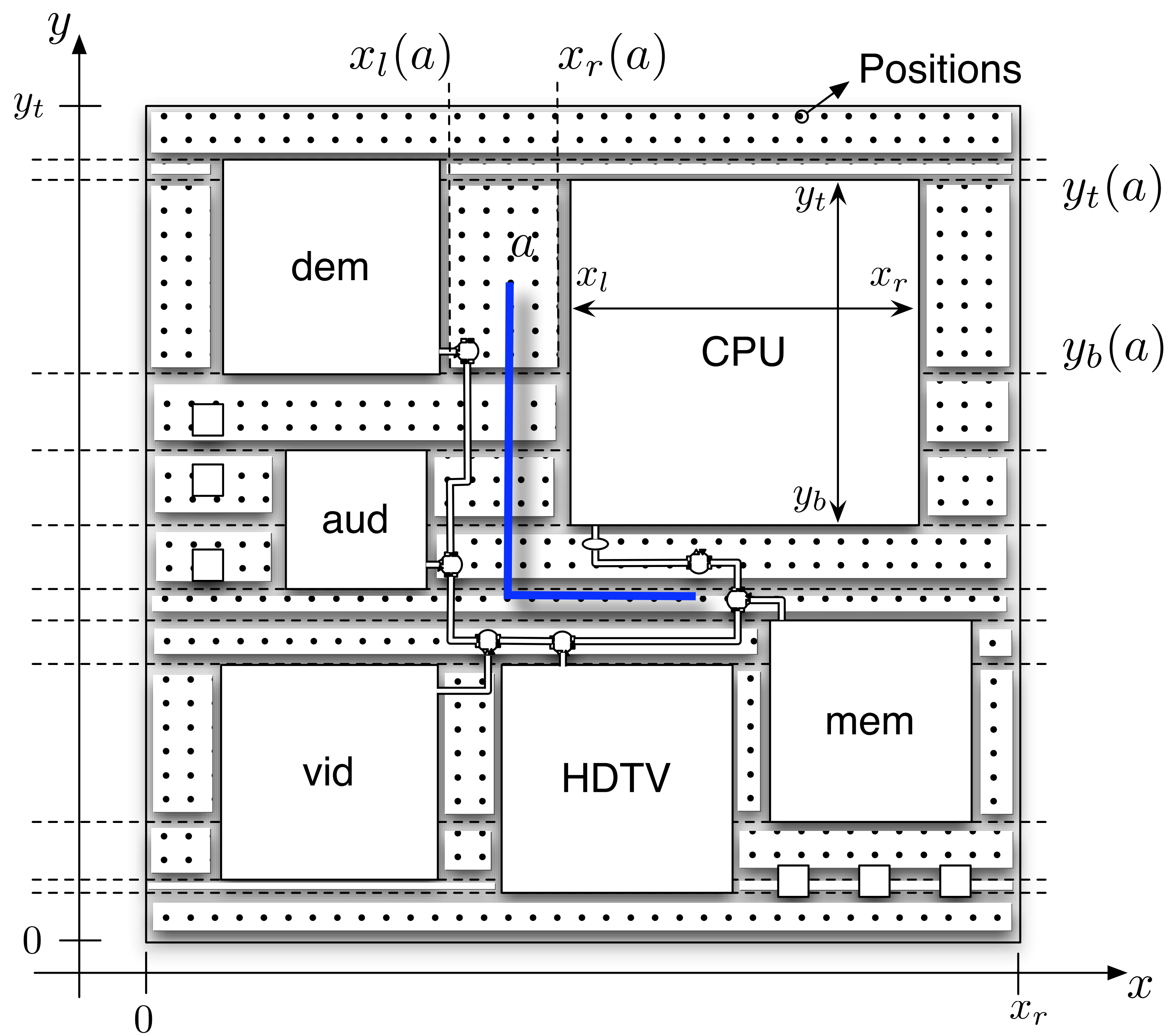
Physical Aspects



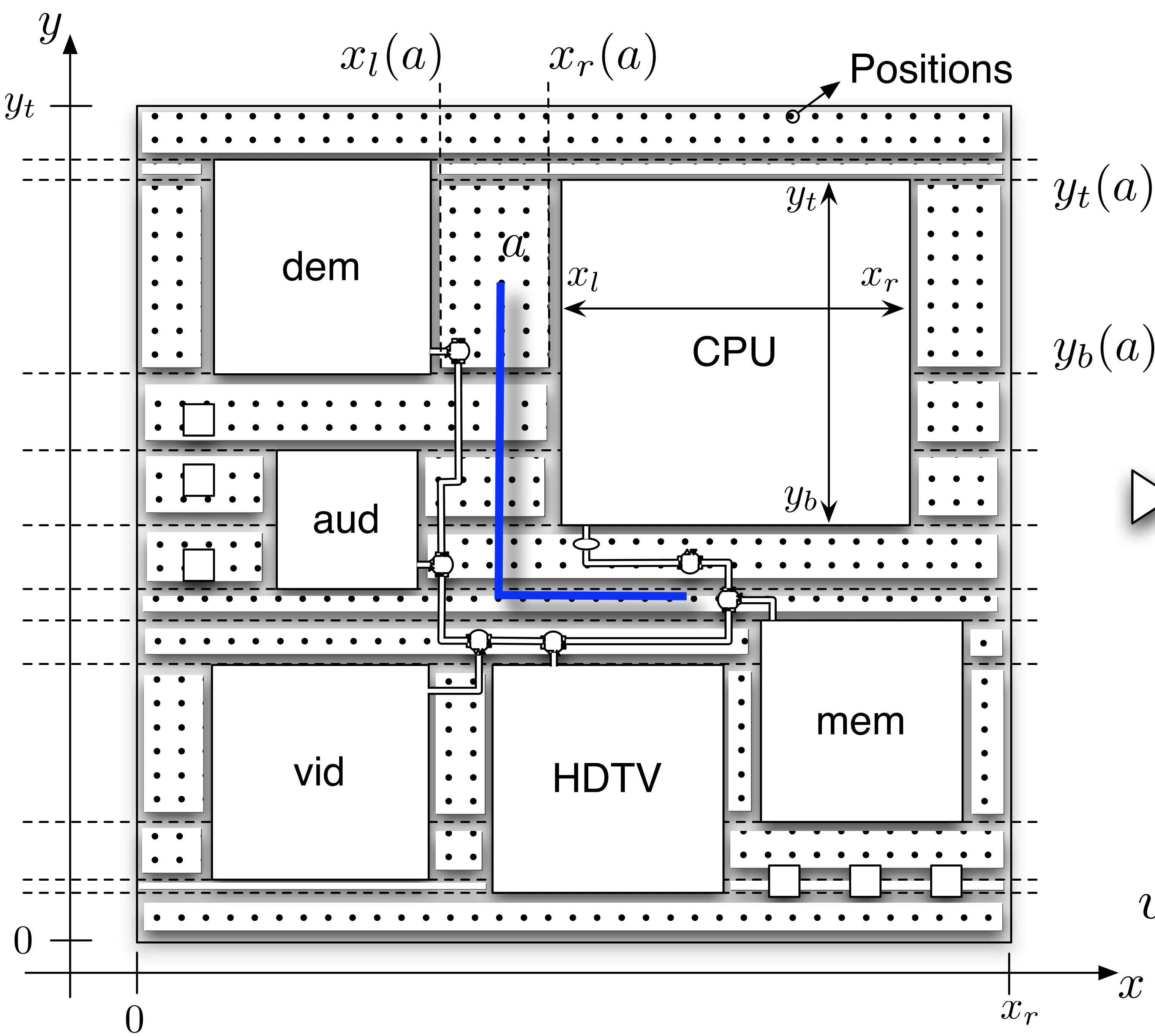
Physical Aspects

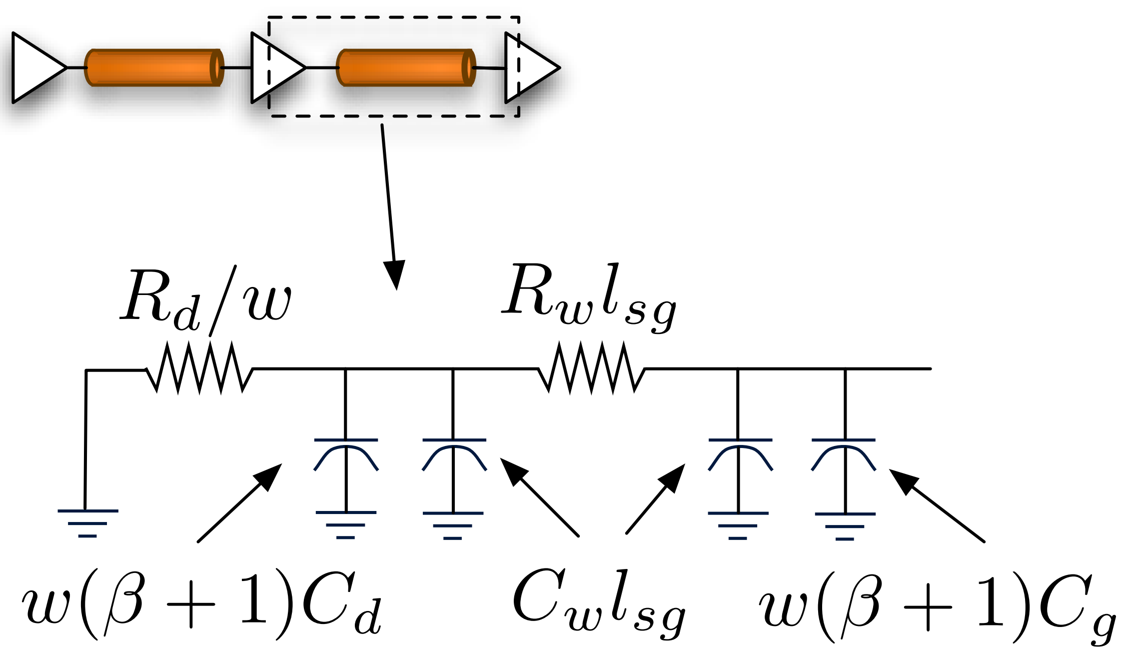


Physical Aspects

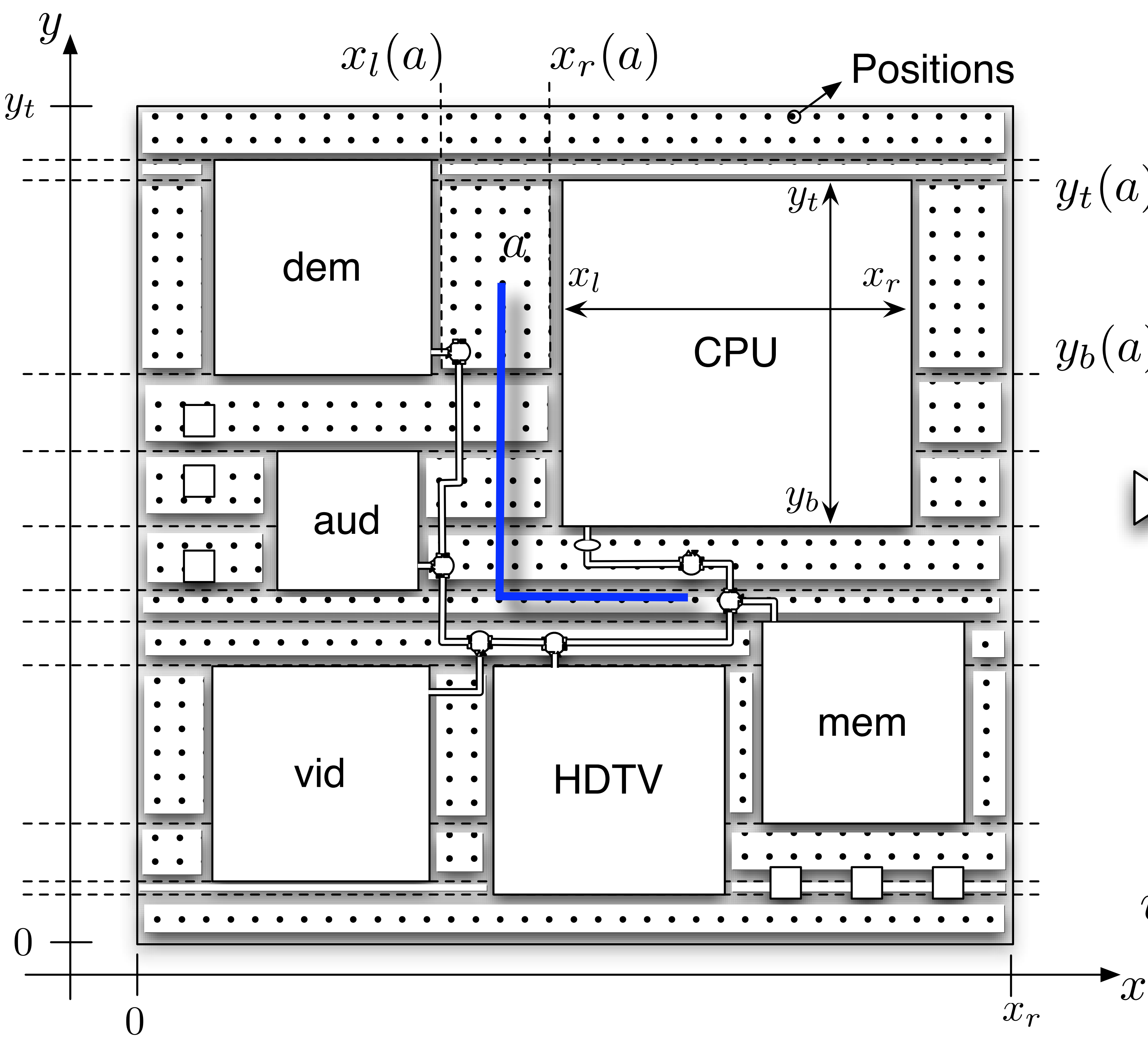


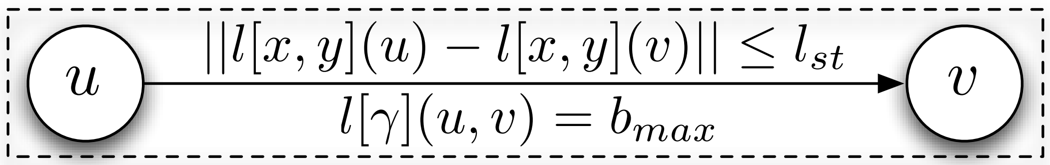
Physical Aspects

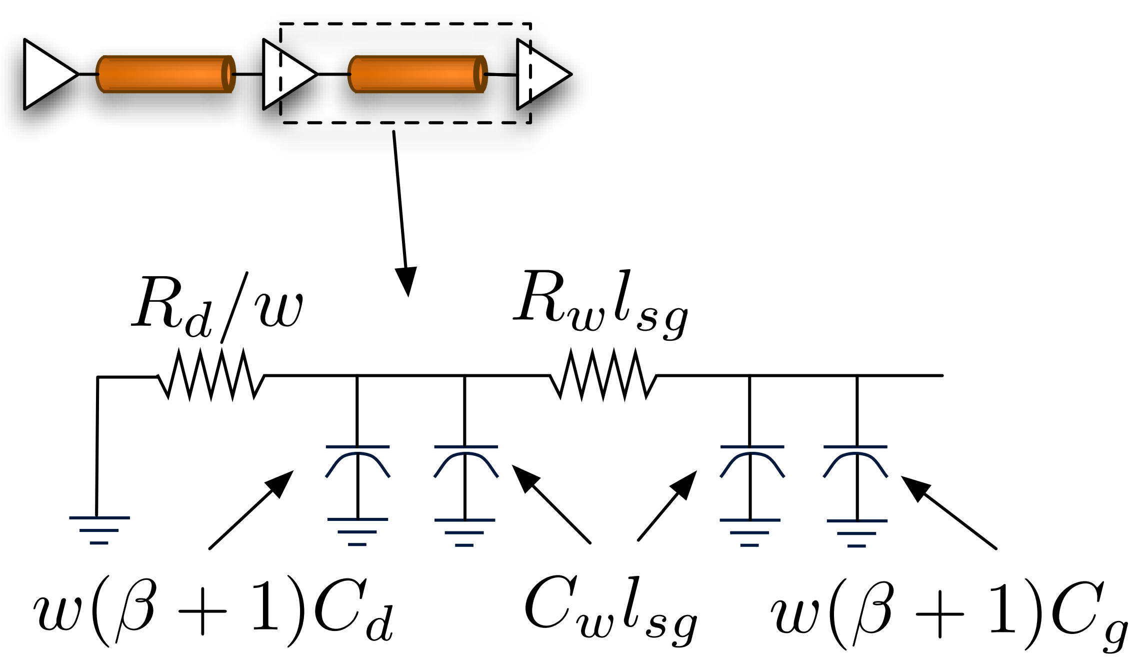




Physical Aspects





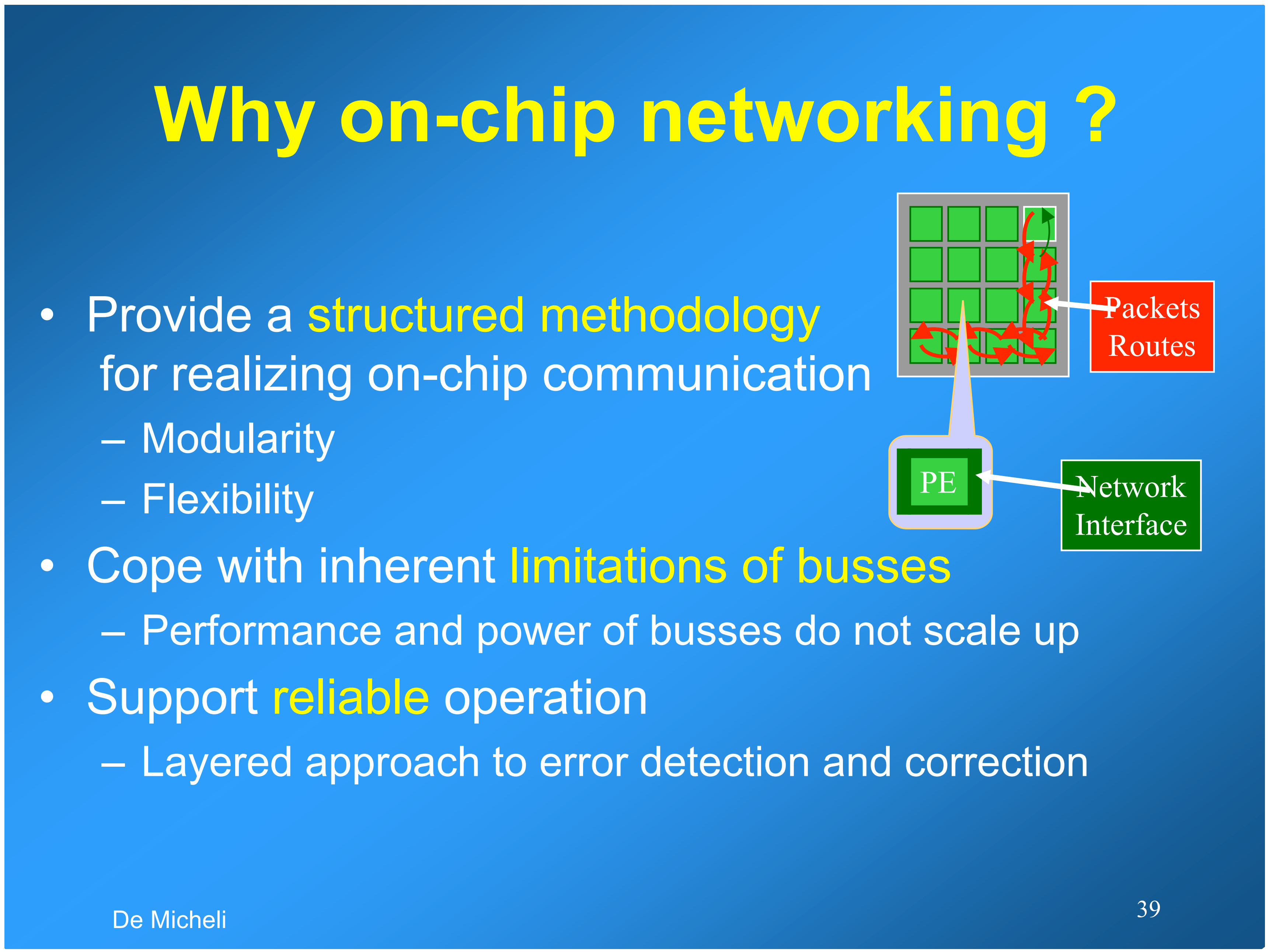


Controversial Views

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

Controversial Views

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



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

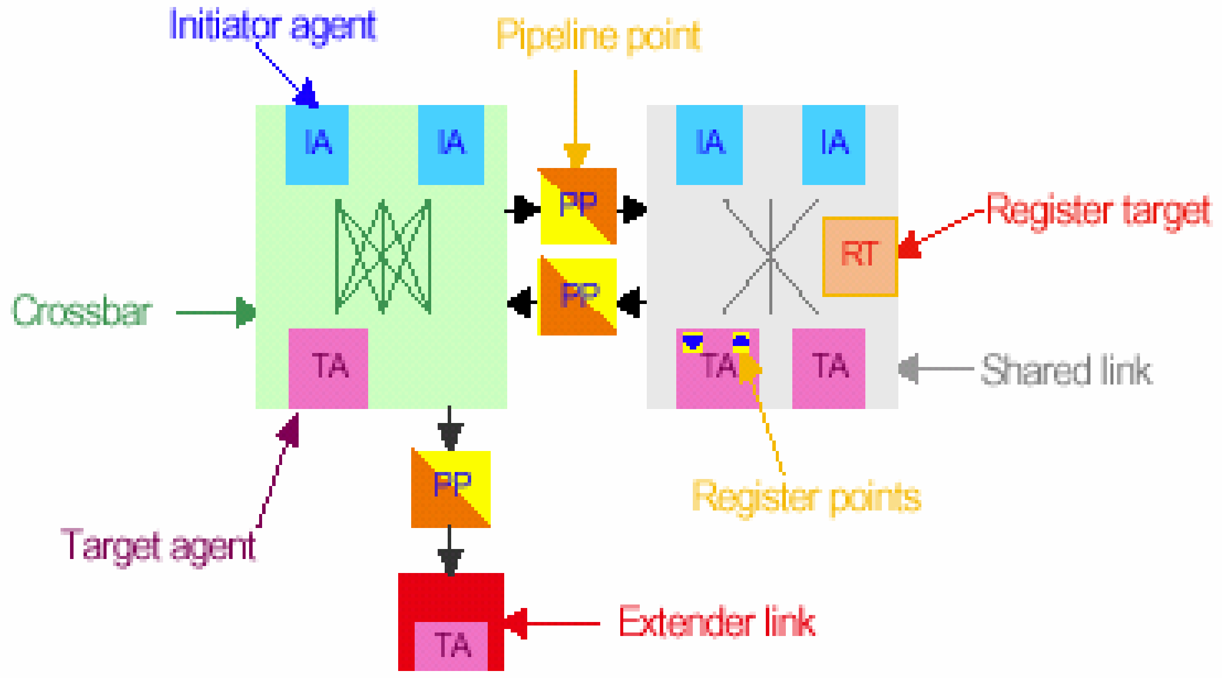
Go slower, wider, and simpler

Shekhar Borkar [OCIN06]

Controversial Views

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

SonicsMX

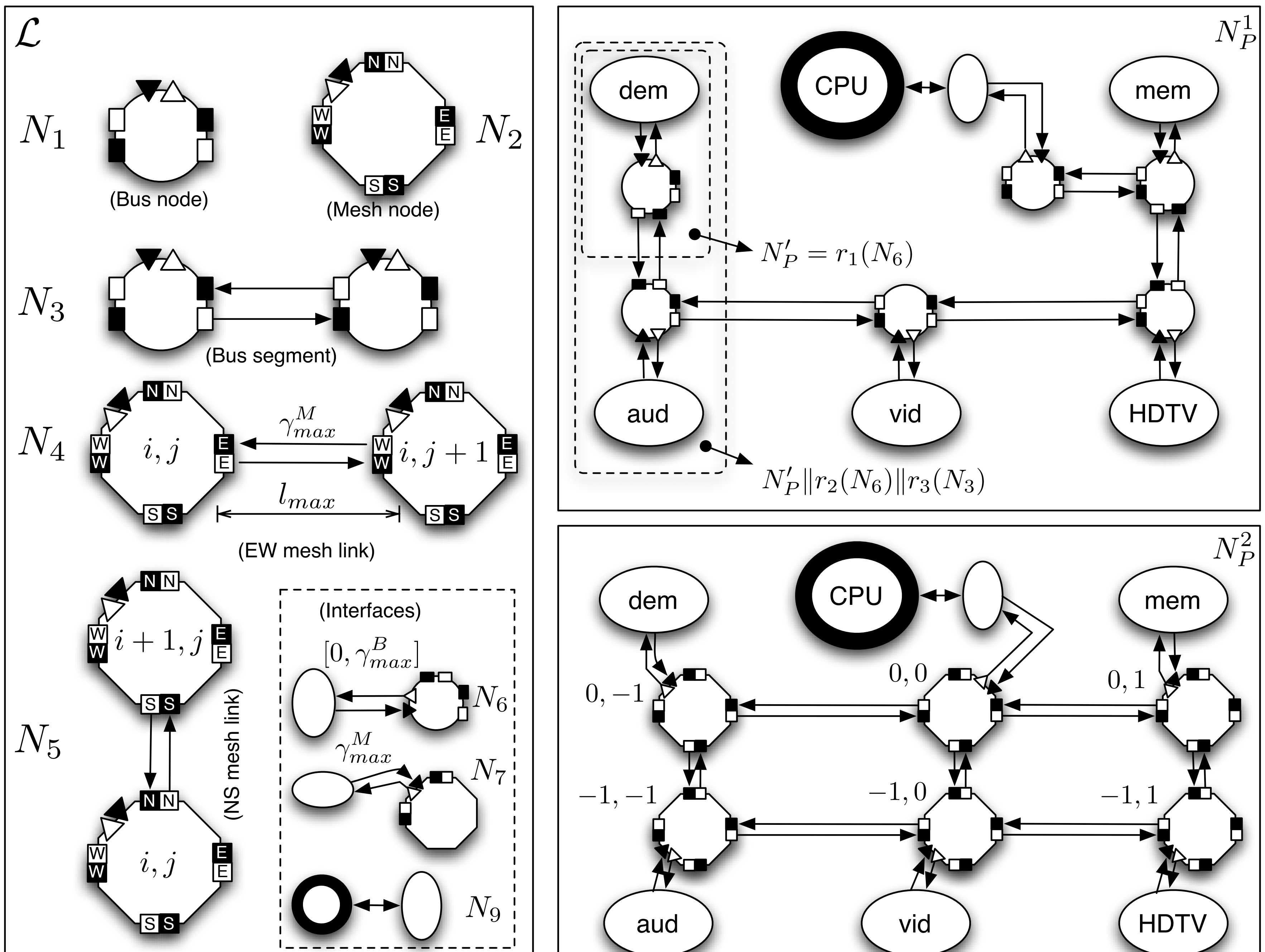


Controversial Views

- 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

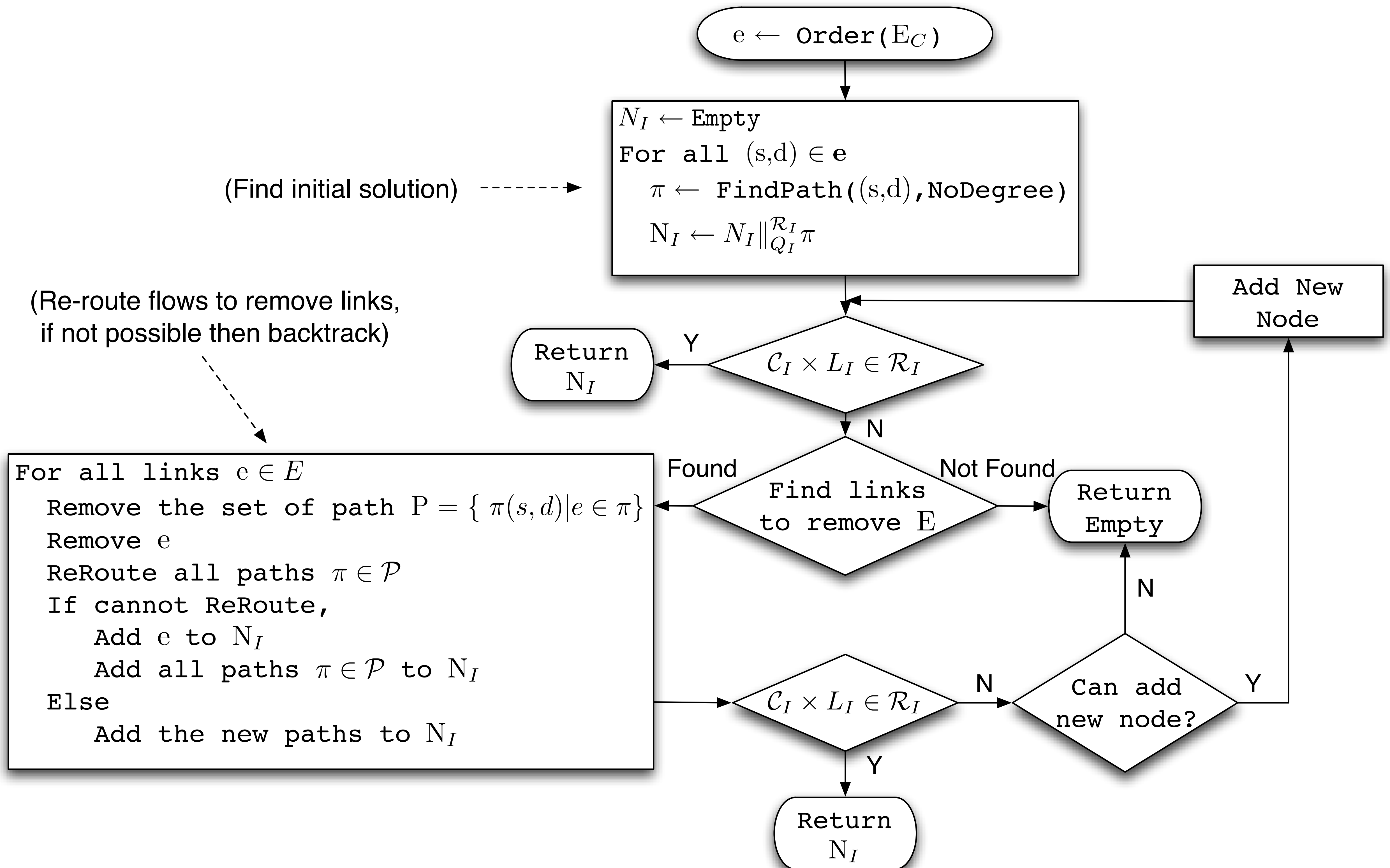
Rule 3: Mesh segment only between (i,j) and (i,j+1) or (i,j) and (i+1,j)

Hardness Results for Network Design

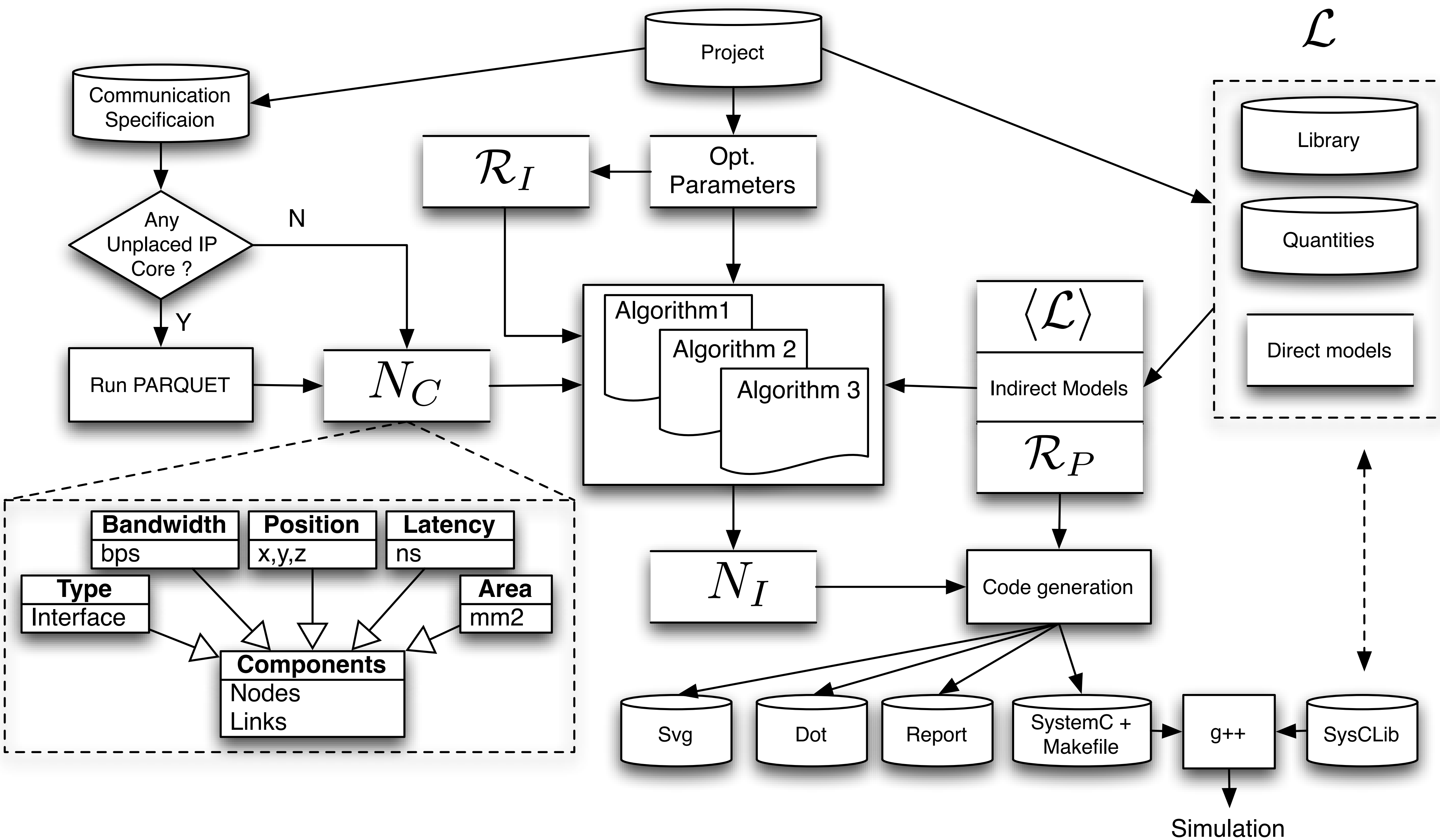
- Consider the bicriteria network design problem $(\mathbf{A}, \mathbf{B}, \mathcal{S})$
  - $\mathbf{A}$ is a budget requirement and $\mathbf{B}$ a minimization objective
  - $\mathcal{S}$ is a membership requirement in a class of subgraphs
- Consider $\mathbf{A}$ a fixed node degree, and $\mathbf{B}$ 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

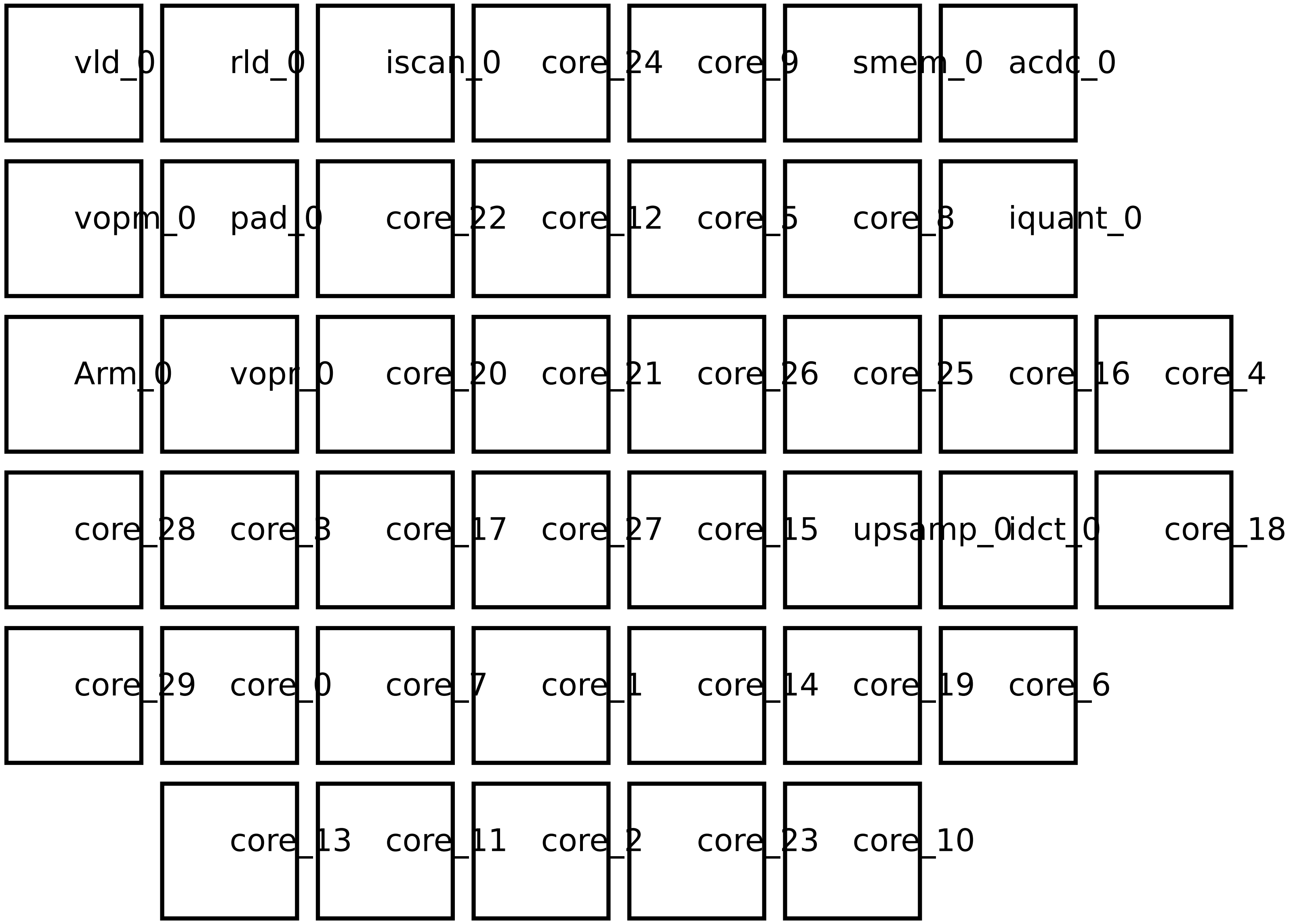


COSI-OCC

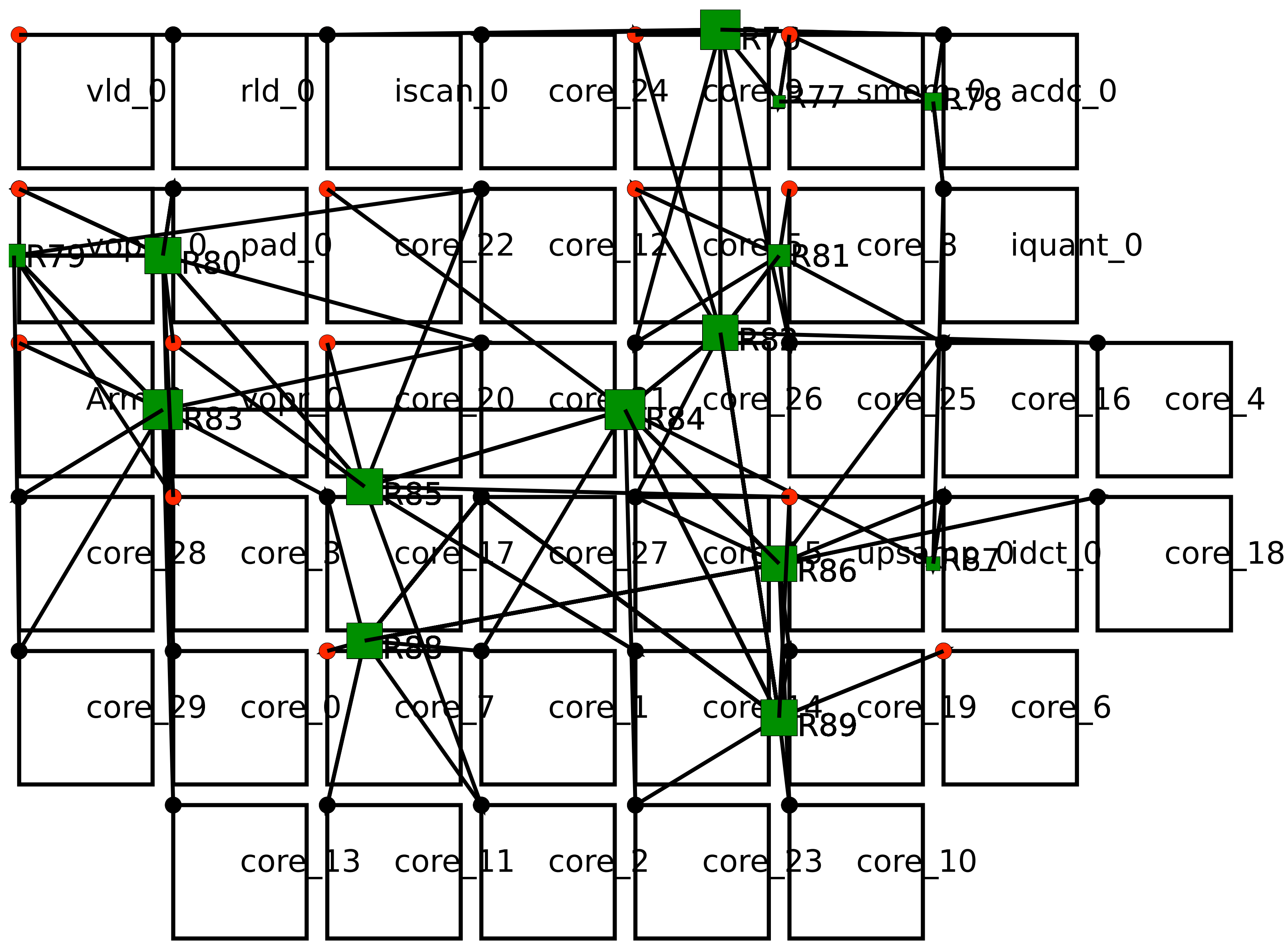


Project
$\mathcal{L}$
Communication Specification
$\mathcal{R}_I$
Opt. Parameters
Library
Quantities
Any Unplaced IP Core ?
N
Y
Algorithm1
Algorithm 2
Algorithm 3
$\langle\mathcal{L}\rangle$
Indirect Models
$\mathcal{R}_P$
Direct models
Run PARQUET
N_C
Bandwidth
bps
Position
x,y,z
Latency
ns
Type
Interface
Area
mm2
Components
Nodes
Links
N_I
Code generation
Svg
Dot
Report
SystemC + Makefile
g++
SysCLib
Simulation

COSI-OCC

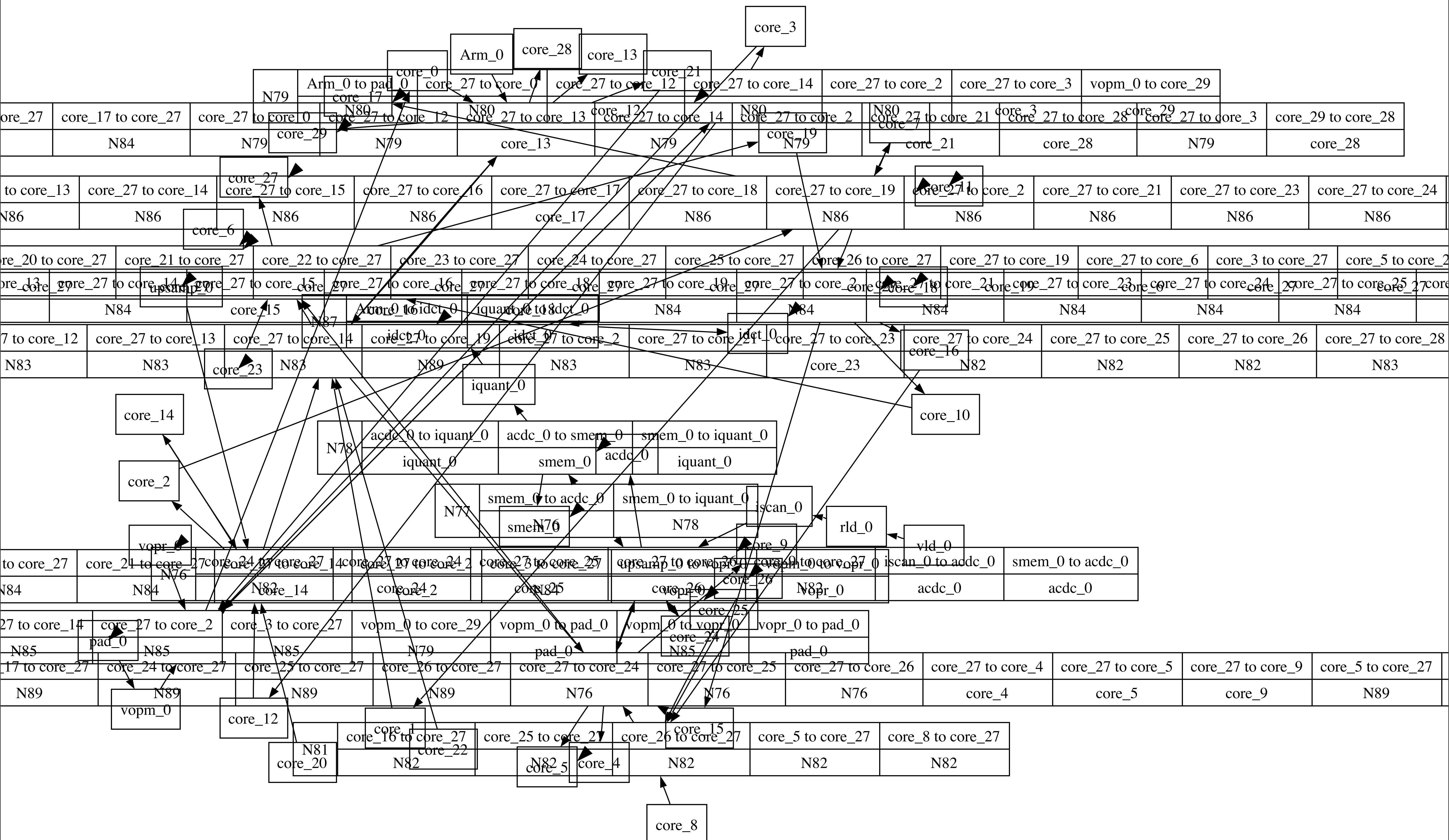

vld_0
rld_0
iscan_0
core_24
core_9
smem_0
acdc_0
vopm_0
pad_0
core_22
core_12
core_5
core_8
iquant_0
Arm_0
vopr_0
core_20
core_21
core_26
core_25
core_16
core_4
core_28
core_3
core_17
core_27
core_15
upsamp_0
idct_0
core_18
core_29
core_0
core_7
core_1
core_14
core_19
core_6
core_13
core_11
core_2
core_23
core_10

COSI-OCC



vld_0
rld_0
iscan_0
core_24
R77
R78
acdc_0
R79
R80
pad_0
core_22
R81
core_8
iquant_0
R83
core_20
R84
core_26
core_25
core_16
core_4
R85
core_28
core_3
core_17
core_27
R86
idct_0
core_18
R88
core_29
core_0
core_7
core_1
R89
core_19
core_6
core_13
core_11
core_2
core_23
core_10

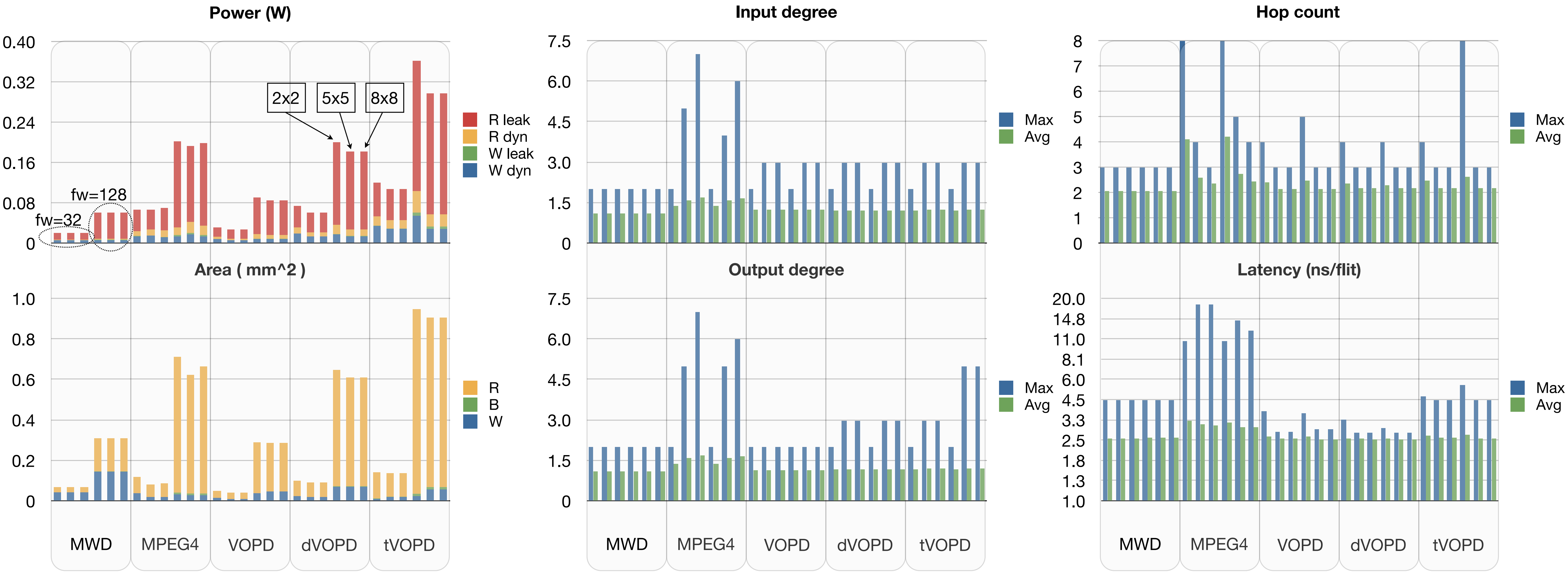
COSI-OCC



core_3
Arm_0
core_28
core_13
core_21
core_0
Arm_0 to pad_0
core_27 to core_0
core_27 to core_12
core_27 to core_14
core_27 to core_2
core_27 to core_3
vopm_0 to core_29
core_17 to core_27
core_27 to core_13
core_27 to core_21
core_27 to core_28
core_29 to core_28
core_27 to core_15
core_27 to core_16
core_27 to core_17
core_27 to core_18
core_27 to core_19
core_27 to core_23
core_27 to core_24
core_21 to core_27
core_22 to core_27
core_23 to core_27
core_24 to core_27
core_25 to core_27
core_26 to core_27
core_27 to core_6
core_3 to core_27
core_27 to core_25
core_27 to core_26
iquant_0
core_10
core_14
core_2
acdc_0 to iquant_0
acdc_0 to smem_0
smem_0 to iquant_0
smem_0 to acdc_0
iscan_0
rld_0
vld_0
iscan_0 to acdc_0
smem_0 to acdc_0
vopm_0 to core_29
vopm_0 to pad_0
vopr_0 to pad_0
core_27 to core_4
core_27 to core_5
core_27 to core_9
core_5 to core_27
core_16 to core_27
core_8 to core_27
core_12
core_20
core_15
core_4
core_8
vopm_0

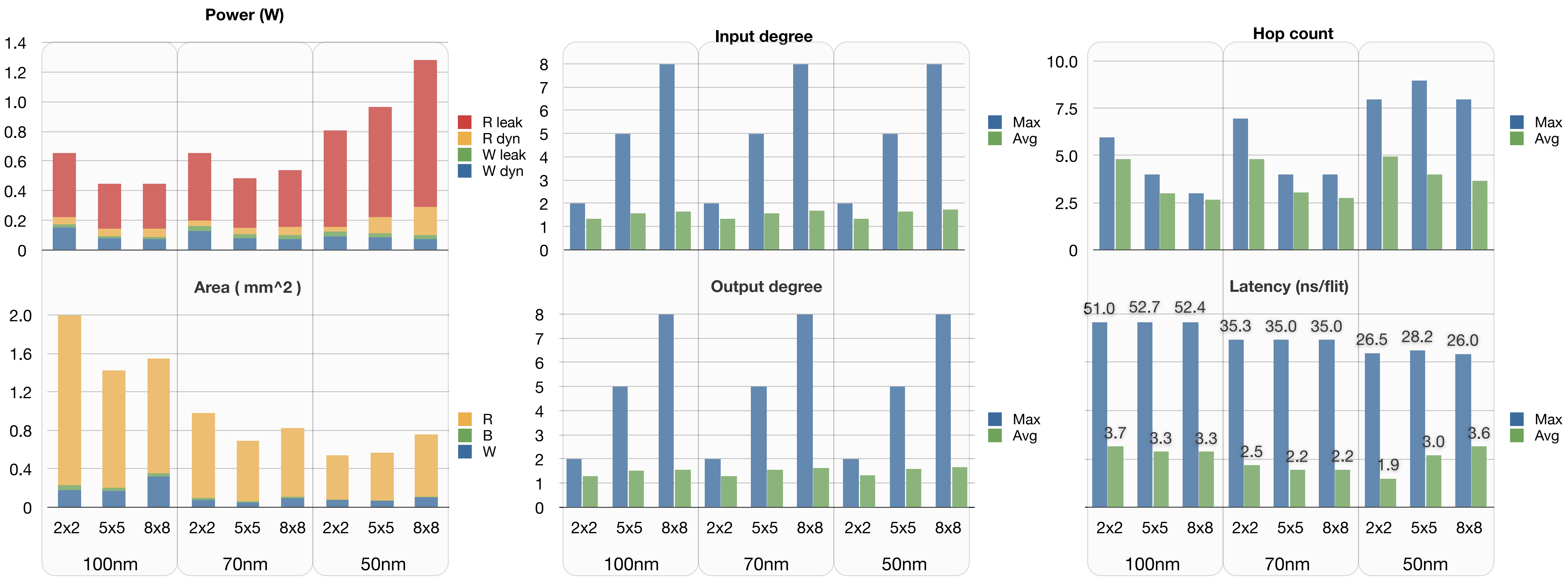
Results

Results



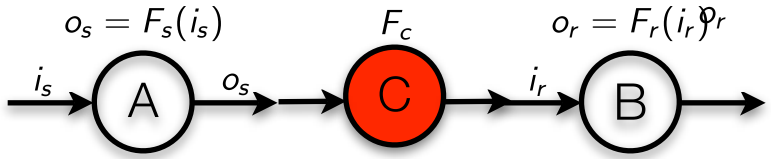
Power (W)
0.40
0.32
0.24
0.16
0.08
0
fw=32
fw=128
2x2
5x5
8x8
R leak
R dyn
W leak
W dyn
Area (mm^2)
1.0
0.8
0.6
0.4
0.2
0
R
B
W
MWD
MPEG4
VOPD
dVOPD
tVOPD
Input degree
7.5
6.0
4.5
3.0
1.5
0
Max
Avg
Output degree
7.5
6.0
4.5
3.0
1.5
0
Max
Avg
MWD
MPEG4
VOPD
dVOPD
tVOPD
Hop count
8
7
6
5
4
3
2
1
0
Max
Avg
Latency (ns/flit)
20.0
14.8
11.0
8.1
6.0
4.5
3.3
2.5
1.8
1.3
1.0
Max
Avg
MWD
MPEG4
VOPD
dVOPD
tVOPD

Results

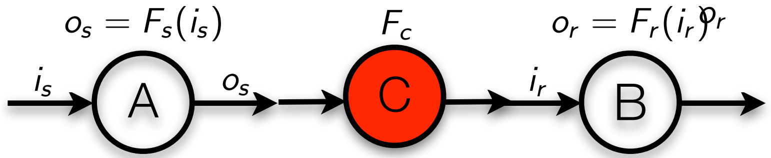


Power (W)
R leak
R dyn
W leak
W dyn
Area (mm^2)
R
B
W
Input degree
Output degree
Max
Avg
Hop count
Latency (ns/flit)
51.0
52.7
52.4
35.3
35.0
35.0
26.5
28.2
26.0
3.7
3.3
3.3
2.5
2.2
2.2
1.9
3.0
3.6
2x2 5x5 8x8 2x2 5x5 8x8 2x2 5x5 8x8
100nm 70nm 50nm

CONCLUSIONS RESEARCH DIRECTIONS

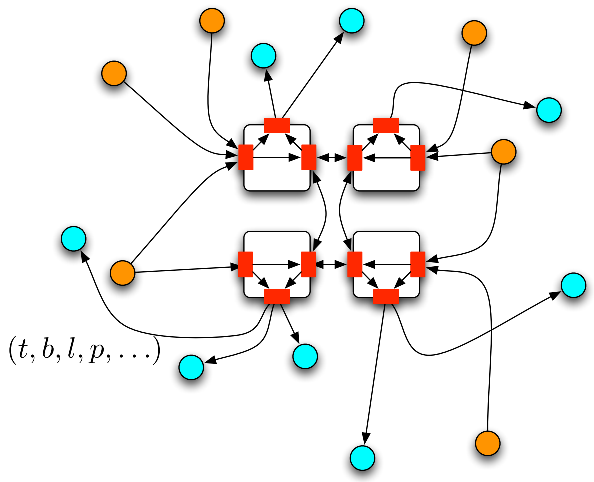


Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness



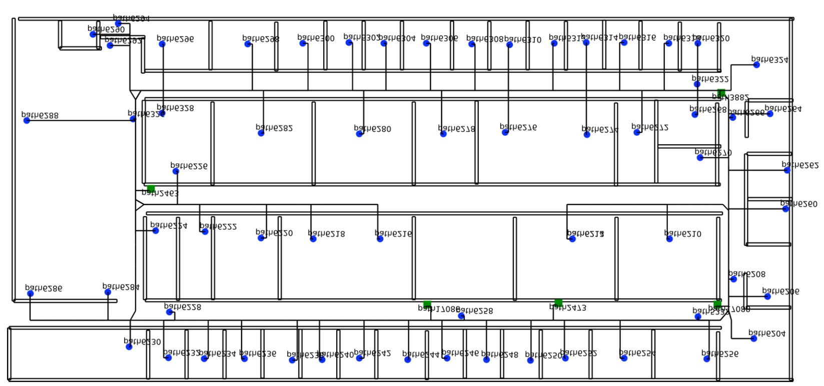
Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness

Sensor, actuators, controllers, links...

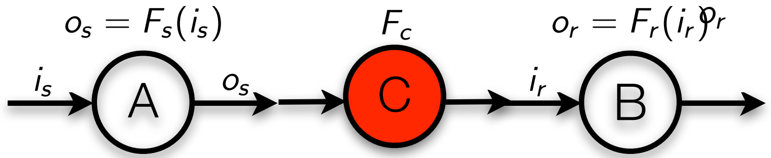


Compose sensors, actuators, controllers. Define accuracy and stability of the distributed control [preserve functionality, infer QoS]

Buses, Routers, Gateways...

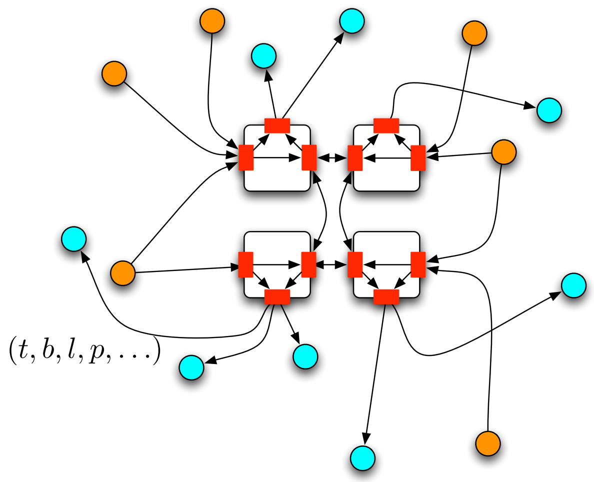


Compose communication nodes and links to minimize installation and operation cost [preserve QoS, infer HW/SW performance]



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

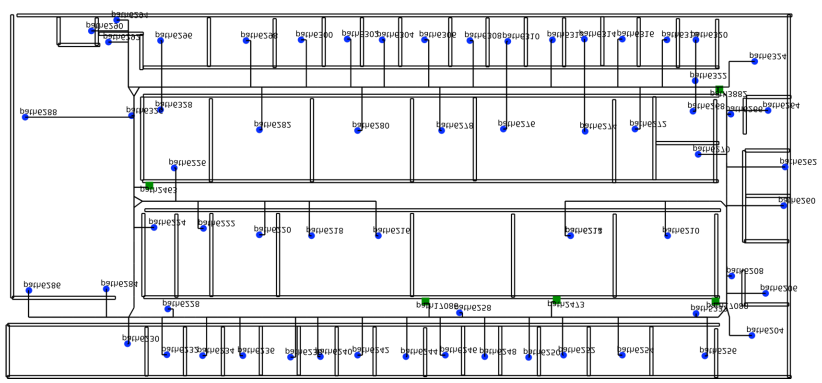
Sensor, actuators, controllers, links...

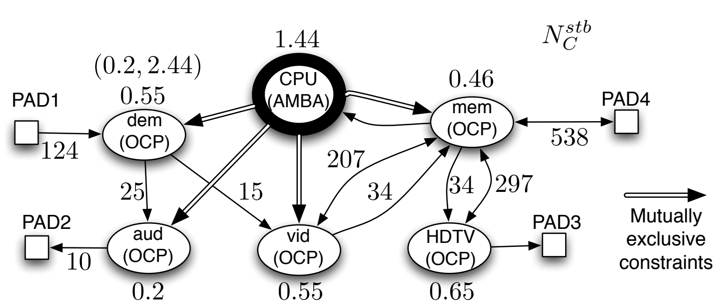


Compose sensors, actuators, controllers. Define accuracy and stability of the distributed control [preserve functionality, infer QoS]

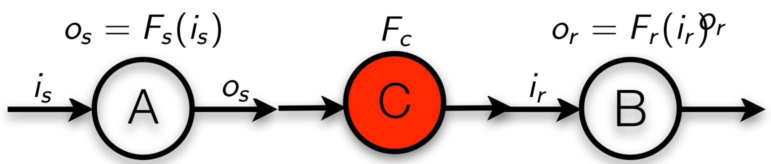
Buses, Routers, Gateways...

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

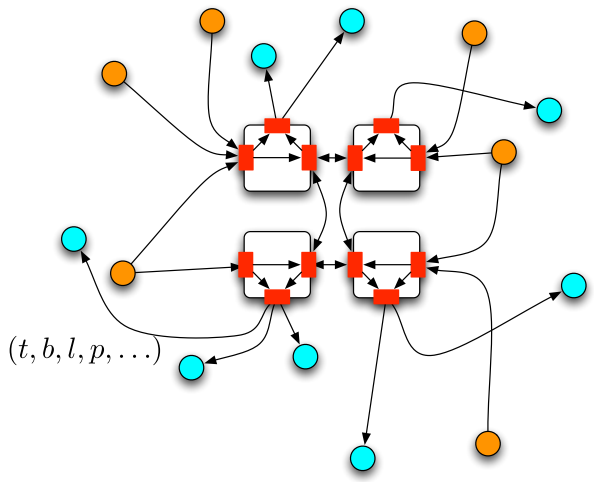




Compose platform [preserve performance, infer QoS]



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

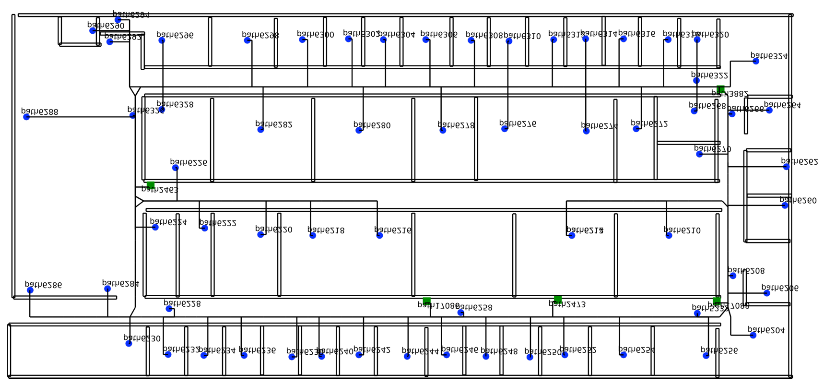


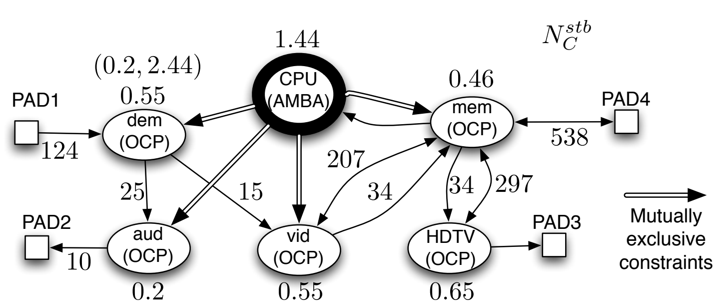
Sensor, actuators, controllers, links...

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]

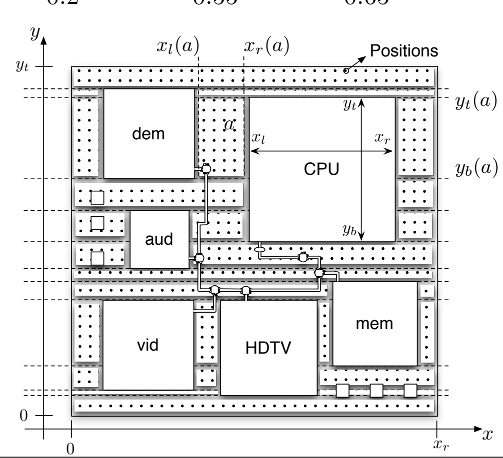


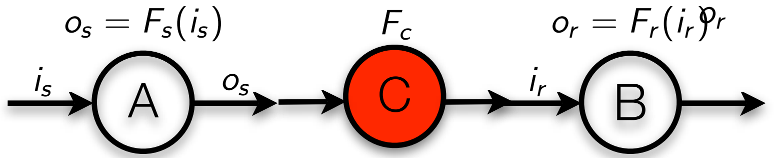


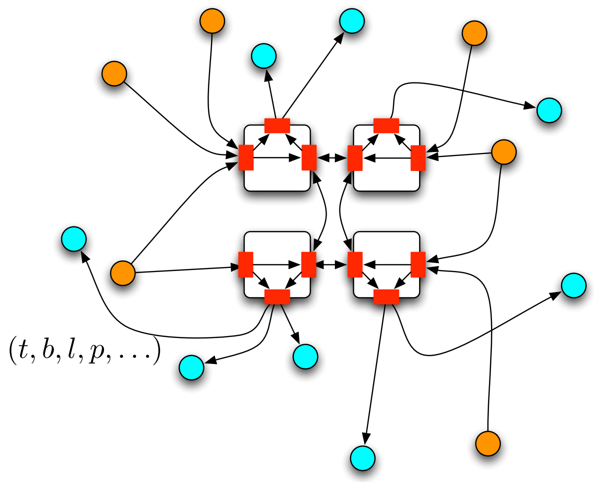
Buses, Rings, Mesh, Routers, Interfaces...

Compose platform [preserve performance, infer QoS]

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



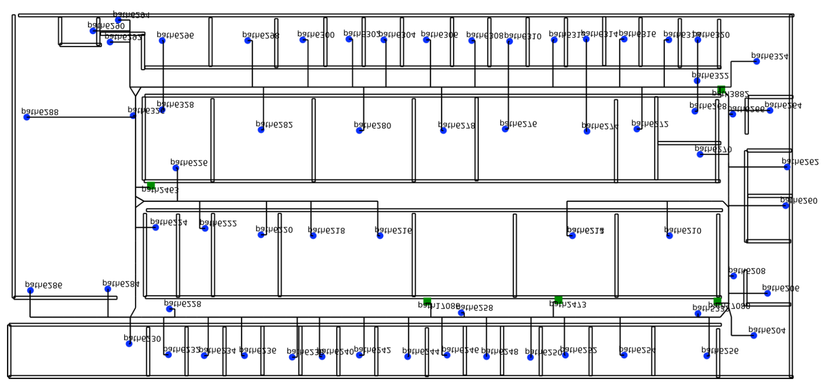




Sensor, actuators, controllers, links...

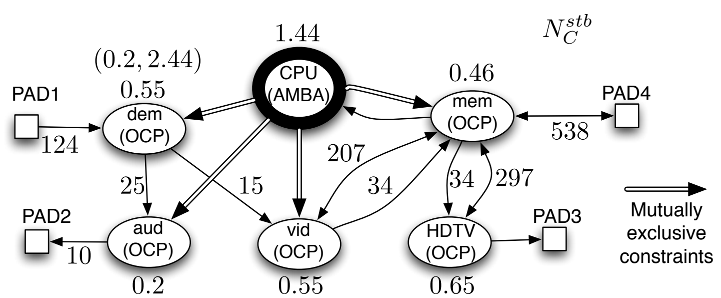
Specify the system, use adaptors for heterogeneous composition [preserve semantics], verify algorithm correctness

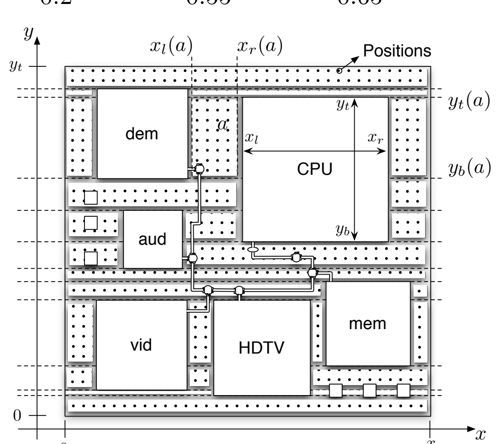
Compose sensors, actuators, controllers. Define accuracy and stability of the distributed control [preserve functionality, infer QoS]



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