KAOS: a KLAIM Extension for Reasoning on Programmable QoS

Emilio Tuosto

etuosto@di.unipi.it

Dipartimento di Informatica

Universitá di Pisa

In cooperation with:

R. De Nicola, G. Ferrari, U. Montanari and R. Pugliese

Plan of the talk

Plan of the talk

- WAN Programming with QoS

Plan of the talk

- WAN Programming with QoS
- A representation of QoS parameters

Plan of the talk

- WAN Programming with QoS
- A representation of QoS parameters
- KAOS

Plan of the talk

- WAN Programming with QoS
- A representation of QoS parameters
- KAOS
- KAOS & Hypergraphs: reasoning on QoS issues

Plan of the talk

- WAN Programming with QoS
- A representation of QoS parameters
- KAOS
- KAOS & Hypergraphs: reasoning on QoS issues
- Details in International Symposium on Verification - Theory and Practice (LNCS)

Wide Area Network Programming Issues

- Absence of centralised control
- Administrative domains
- Interoperability
- “Mobility” (of resources and computation)
- *Network Awareness*
  - Applications are location dependent
  - Locations have different features
  - and allow multiple (security) policies
- Independently programmed in a distributed environment
- Service Level Agreement
- Security
- ...

Klaim

Klaim

- Multiple TS

Klaim

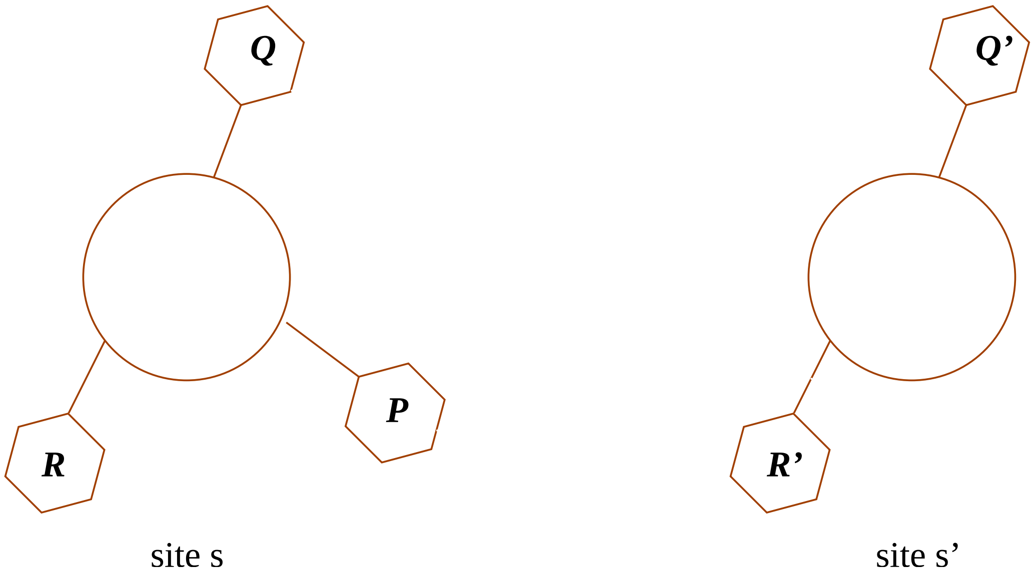
- Multiple TS
- Localities: first class citizens

Klaim

- Multiple TS
- Localities: first class citizens
- Process migration

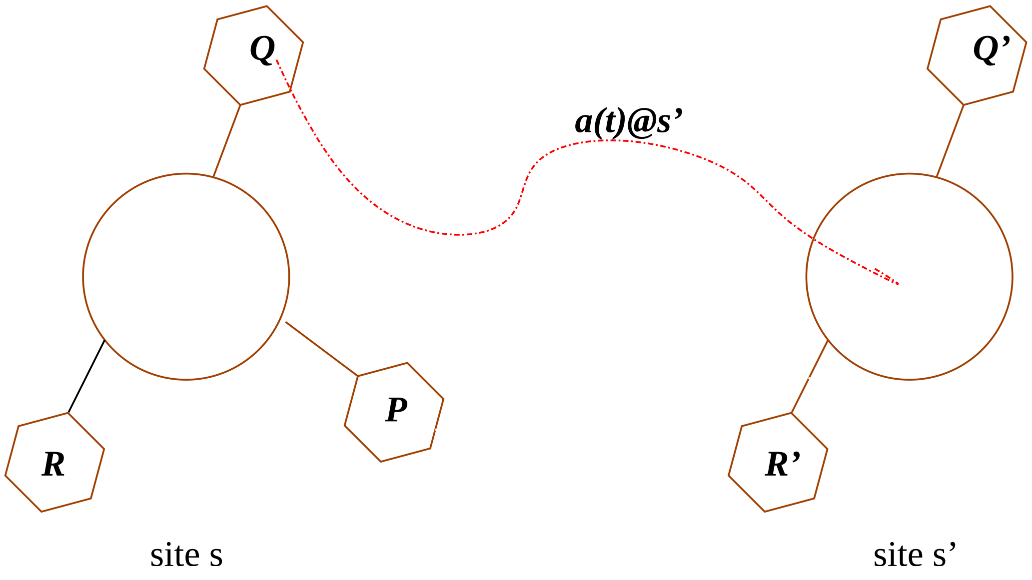
Klaim

- Multiple TS
- Localities: first class citizens
- Process migration



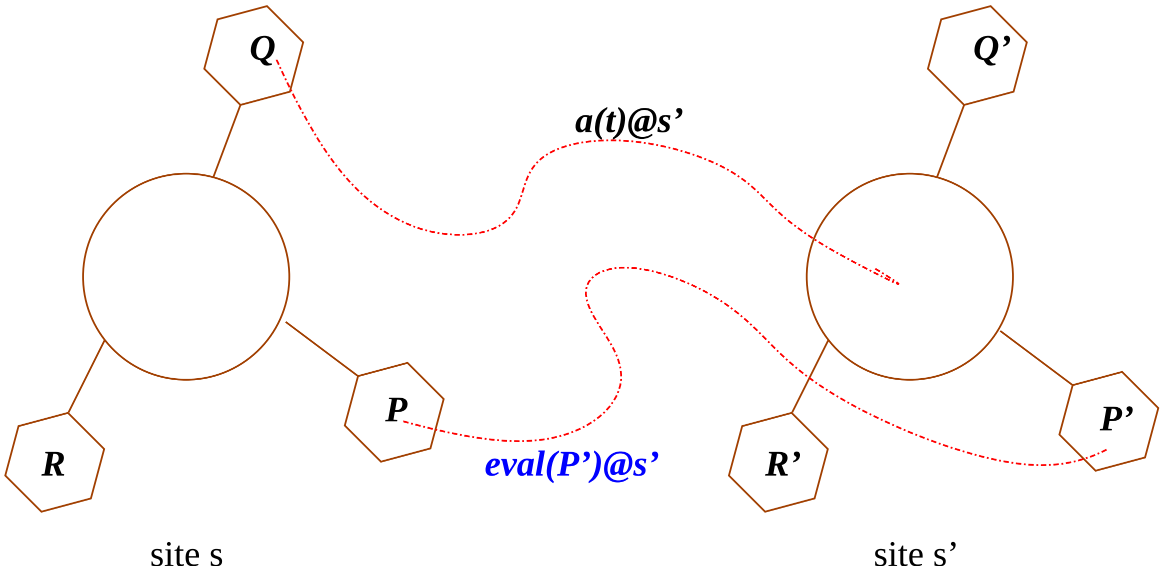
Klaim

- Multiple TS
- Localities: first class citizens
- Process migration



Klaim

- Multiple TS
- Localities: first class citizens
- Process migration



$$
\begin{array}{lcl}
P & ::= & \mathbf{nil} \\
 & \mid & \alpha.P \\
 & \mid & P_1 \mid P_2 \\
\alpha & ::= & a@s \\
a & ::= & \ldots \text{ // Klaim actions} \\
 & \mid & \varepsilon(P)
\end{array}
$$

KAOS: Gateways

KAOS: Gateways

- Coordinators (super processes)

KAOS: Gateways

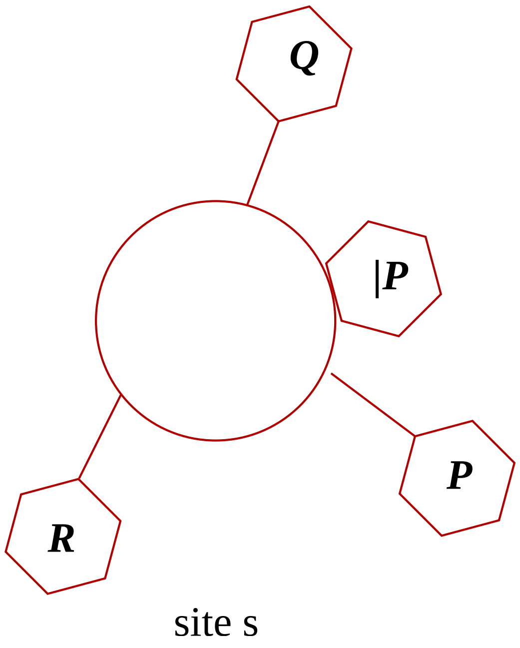
- Coordinators (super processes)
- Dynamic creation of sites

KAOS: Gateways

- Coordinators (super processes)
- Dynamic creation of sites
- Gateway connection management

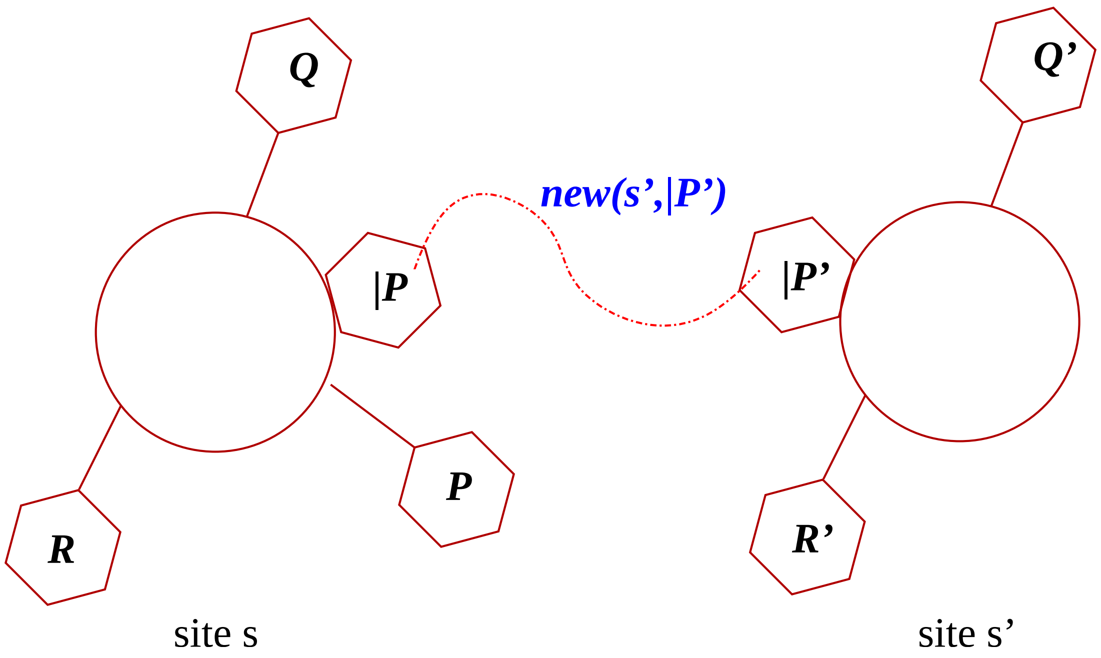
KAOS: Gateways

- Coordinators (super processes)
- Dynamic creation of sites
- Gateway connection management



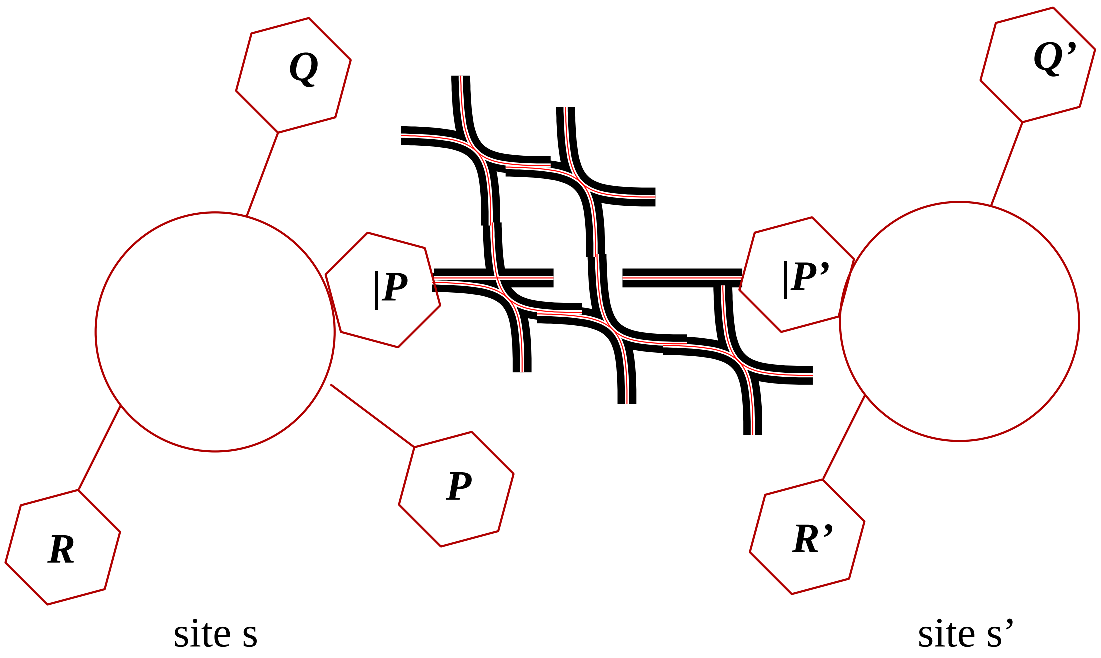
KAOS: Gateways

- Coordinators (super processes)
- Dynamic creation of sites
- Gateway connection management



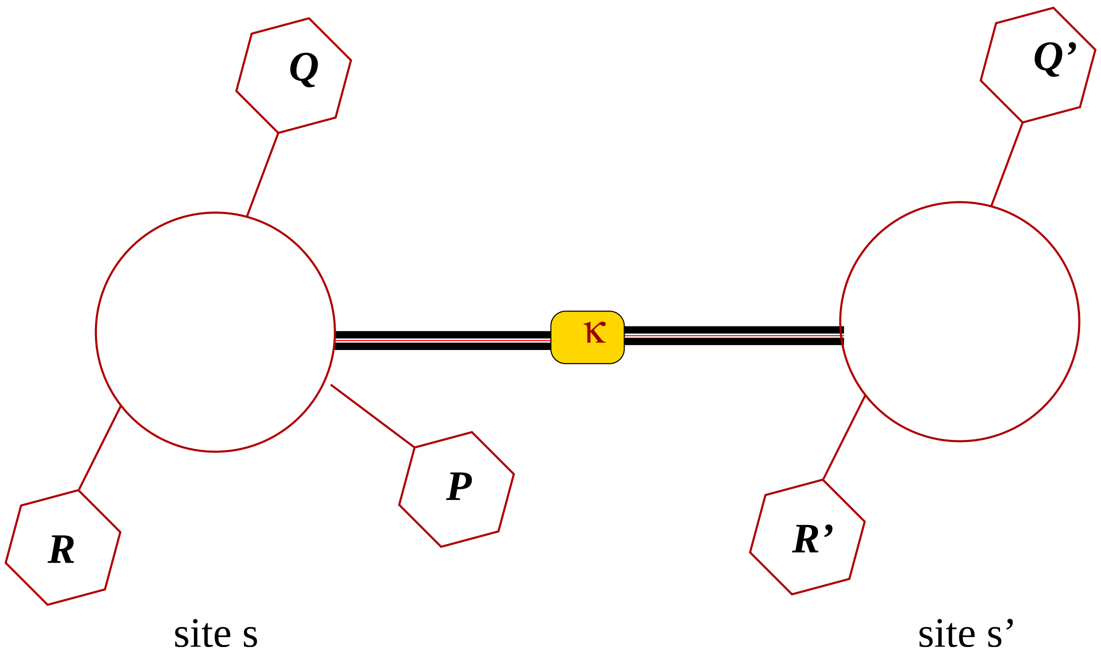
KAOS: Gateways

- Coordinators (super processes)
- Dynamic creation of sites
- Gateway connection management



KAOS: Gateways

- Coordinators (super processes)
- Dynamic creation of sites
- Gateway connection management



$$
\begin{array}{lcl}
P & ::= & \gamma.P \mid P_1 \mid P_2 \\
\gamma & ::= & \alpha \\
 & \mid & \nu(s \cdot \kappa) \\
 & \mid & \overset{\kappa}{\frown} s \\
 & \mid & s \overset{\kappa}{\smile} \\
 & \mid & \delta\, l
\end{array}
$$

Cost κ abstracts characteristics of connections according to many "dimensions" (bandwidth, latency, distance, access rights ...)

Connection costs

Algebra on costs: c-semiring. $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ is a constraint semiring if A is a set ($\mathbf{0}, \mathbf{1} \in A$), and $+$ and $\times$ are binary operations on A that satisfy the following properties:

- $+$ is commutative, associative, idempotent, $\mathbf{0}$ is its unit element and $\mathbf{1}$ is its absorbing element;
- $\times$ is commutative, associative, distributes over $+$, $\mathbf{1}$ is its unit element, and $\mathbf{0}$ is its absorbing element.

The additive operation of a c-semiring induces a partial order on A defined as $a \leq_A b \iff \exists c : a + c = b$. The minimal element is thus $\mathbf{0}$ and the maximal $\mathbf{1}$

Examples of connection costs

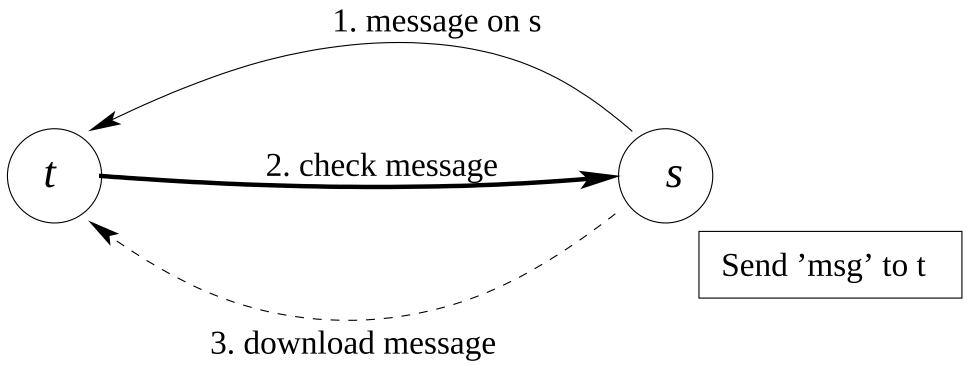
$\boxed{\langle N, min, +, +\infty, 0\rangle}$, the c-semiring of natural numbers N where

- the additive operation is min
- the multiplicative operation is the sum over natural numbers

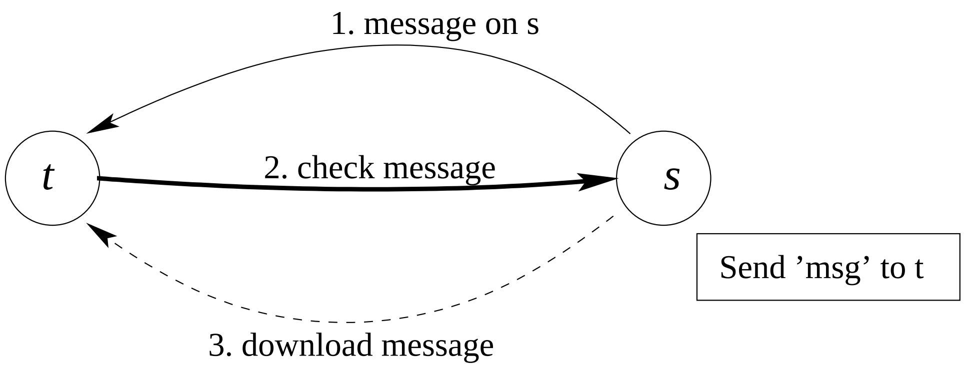
$\boxed{\langle \wp(\{A\}), \cup, \cap, A, A\}\rangle}$, c-semiring of capabilities A where

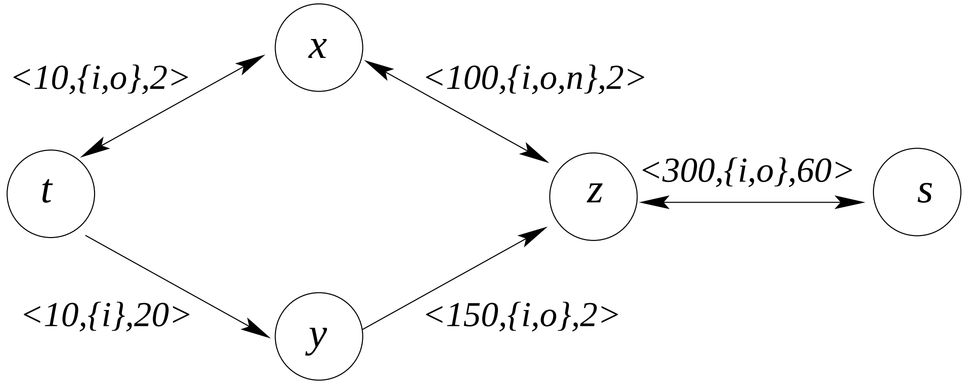
- the additive operation is $\cup$
- the multiplicative operation is $\cap$

KAOS: Example

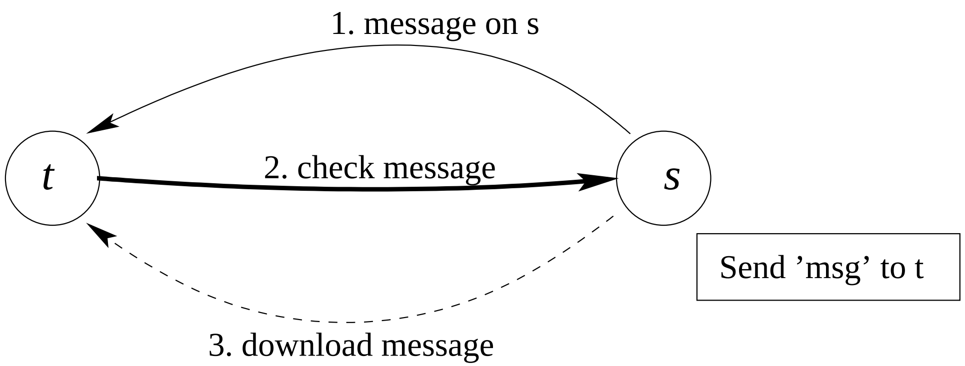


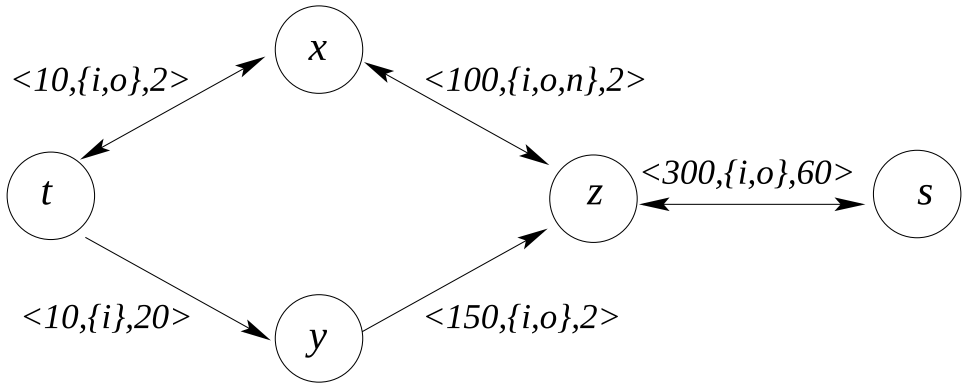
KAOS: Example





KAOS: Example





Costs are triples $\kappa = \langle d, C, p \rangle$ where

1. d is the geographical distance (in Km);
2. $C \subseteq \{i, o, n\}$ are the capabilities, where i, o and n stand for *i*nput, *o*utput and creation of *n*ew nodes, respectively;
3. p is the price (in euros).

KAOS: Example

Costs are elements of the cartesian product of

1. $(N, min, +, +\infty, 0)$,
2. $(\wp(\{i, o, n\}), glb, \cap, \{i, o, n\}, \{i, o, n\})$,
3. $(Q, min, +, +\infty, 0)$.

proved to be a c-semiring [BMR97]
Operations $\times$ and $+$ are

$$\langle d, C, p\rangle \times \langle d', C', p'\rangle = \langle d + d', C \cap C', p + p'\rangle$$
$$\langle d, C, p\rangle + \langle d', C', p'\rangle = \langle min\{d, d'\}, glb\{C, C'\}, \min\{p, p'\}\rangle.$$

Accordingly, the neutral elements of $\times$ and $+$, respectively are defined as $1 = \langle 0, \{i, o, n\}, 0\rangle$ and $0 = \langle +\infty, \emptyset, +\infty\rangle$.

Hypergraphs Programming model^2

Tackling new *non-functional* computational phenomena of systems using SHR.
The metaphor is

- “WAN systems *as* Hypergraphs”
- “WAN computations *as* SHR”

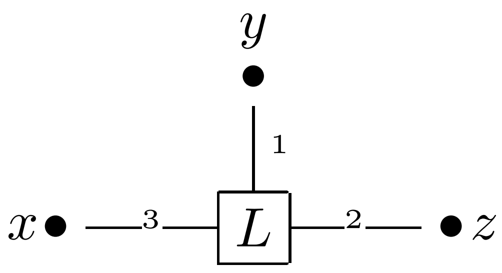
In other words:

- Components are represented by hyperedges
- Systems are *bunches* of (connected) hyperedges
- Computing means to rewrite hyperedge...
- ...according to a synchronisation policy

Hyperedges and Hypergraphs Syntax

A hyperedge generalises edges: It connects more than two nodes

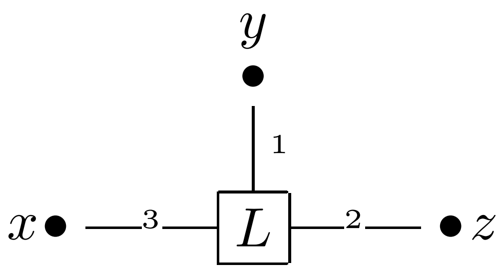
$$L : 3, \quad L(y, z, x),$$



Hyperedges and Hypergraphs Syntax

A hyperedge generalises edges: It connects more than two nodes

$$L : 3, \quad L(y, z, x),$$

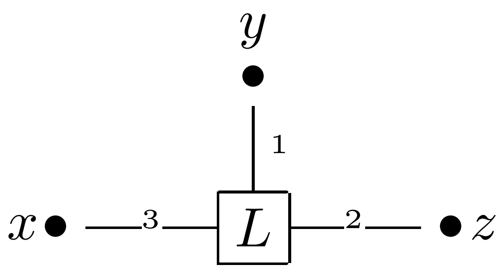


$$\begin{array}{rcl} G & ::= & nil \mid \nu\, y.G \\ & \mid & L(\vec{x}) \mid G|G \end{array}$$

Hyperedges and Hypergraphs Syntax

A hyperedge generalises edges: It connects more than two nodes

$L : 3, \quad L(y, z, x),$



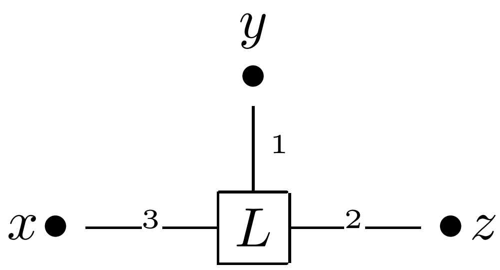
$$G \;::=\; nil \;\big|\; \nu\, y.G \;\big|\; L(\vec{x}) \;\big|\; G|G$$

Syntactic Judgement $\quad \Gamma \vdash G, \quad fn(G) \subseteq \Gamma$

Hyperedges and Hypergraphs Syntax

A hyperedge generalises edges: It connects more than two nodes

$$L:3, \quad L(y,z,x),$$



$$G \;::=\; nil \;\mid\; \nu\, y.G \;\mid\; L(\vec{x}) \;\mid\; G|G$$

Syntactic Judgement $\quad \Gamma \vdash G, \qquad fn(G) \subseteq \Gamma$

An example:

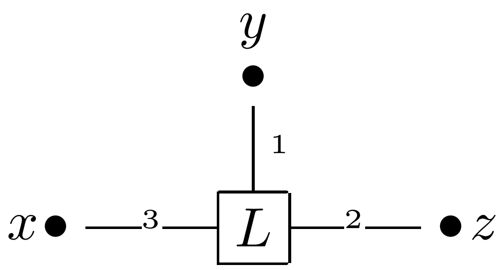
$$L:3, \quad M:2$$

$$x, y \vdash \nu\, z.(L(y,z,x)|M(y,z))$$

Hyperedges and Hypergraphs Syntax

A hyperedge generalises edges: It connects more than two nodes

$L : 3, \quad L(y, z, x),$



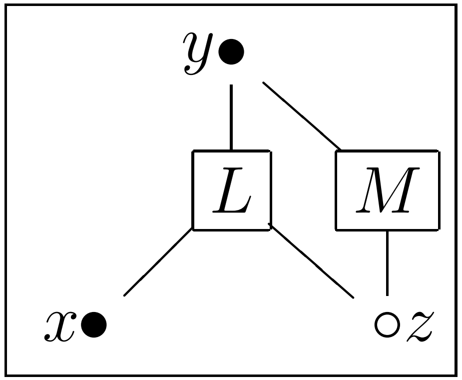
$$G \;::=\; nil \;\mid\; \nu\, y.G \;\mid\; L(\vec{x}) \;\mid\; G|G$$

Syntactic Judgement $\quad \Gamma \vdash G, \qquad fn(G) \subseteq \Gamma$

An example:

$L : 3, \quad M : 2$

$x, y \vdash \nu\; z.(L(y, z, x)|M(y, z))$

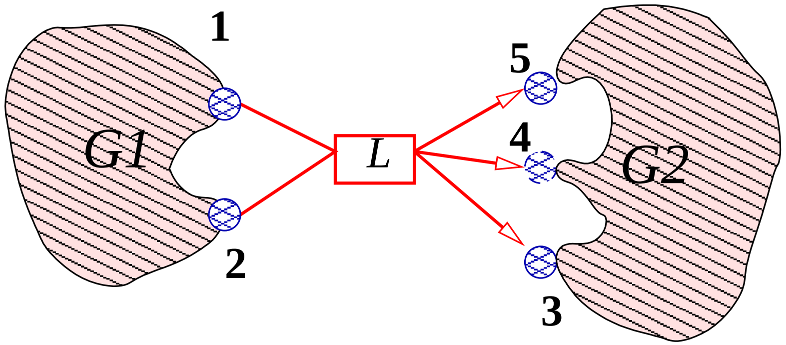


Replacement of Hyperedges

$L \rightarrow G$

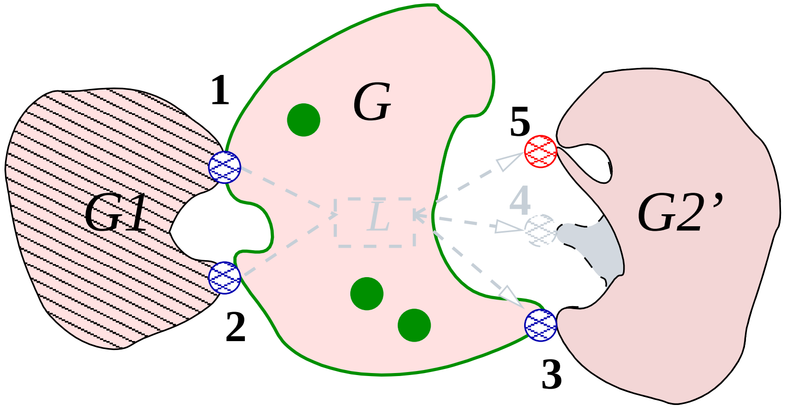
Replacement of Hyperedges

$L \longrightarrow G$



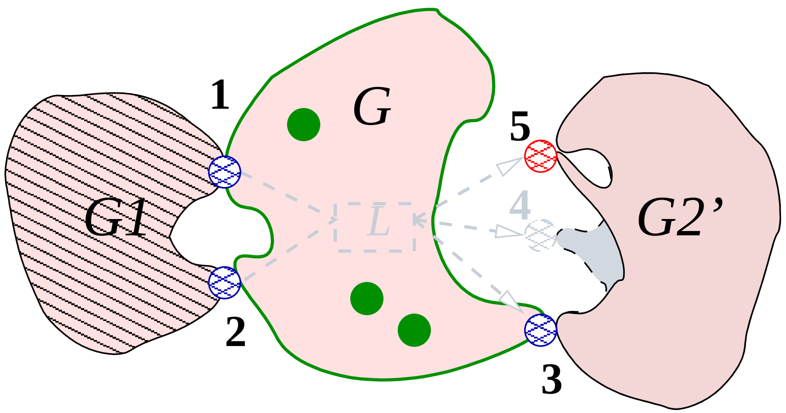
Replacement of Hyperedges

$L \longrightarrow G$



Replacement of Hyperedges

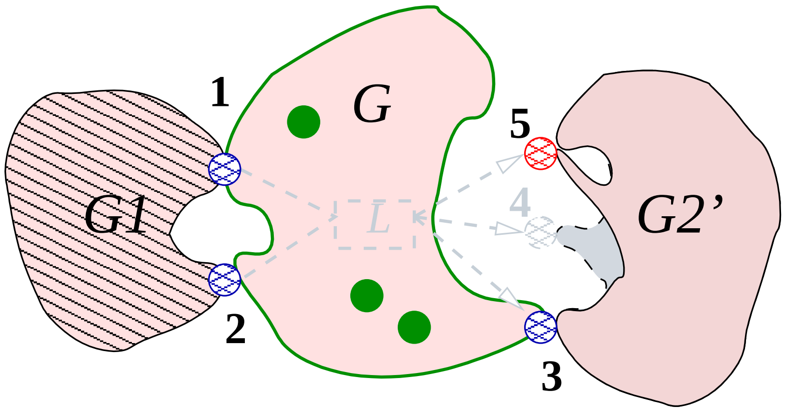
$L \rightarrow G$



- Edge replacement: local
- Synchronisation as distributed constraint solving
- New node creation
- Node fusion: mobility model

Replacement of Hyperedges

$L \longrightarrow G$



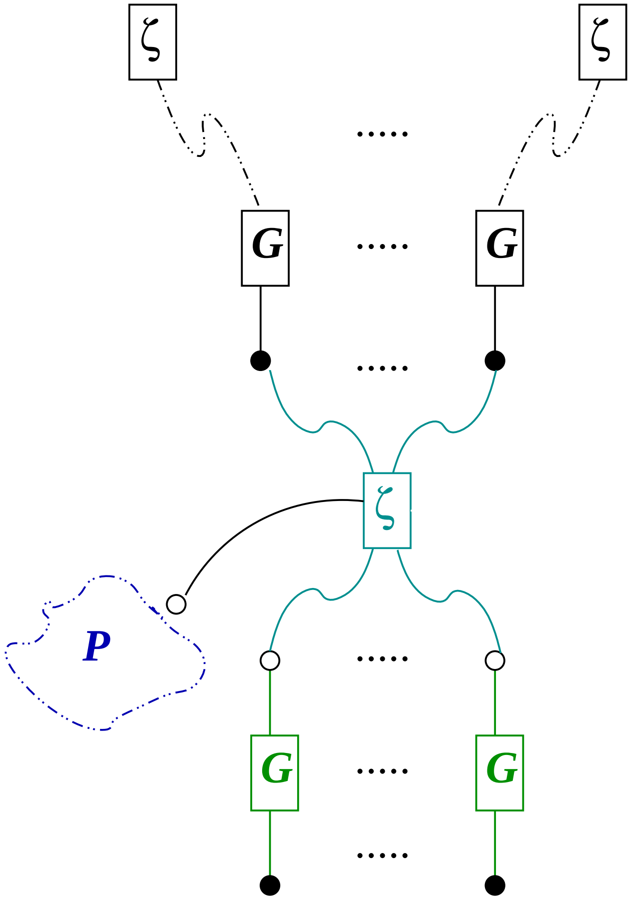
- Edge replacement: local
- Synchronisation as distributed constraint solving
- New node creation
- Node fusion: mobility model

Benefi ts:

- Powerful model of system composition (π, π-I, fusion)
- LTS for Ambient ...
- ...and for Klaim
- and *path reservation* for KAOS

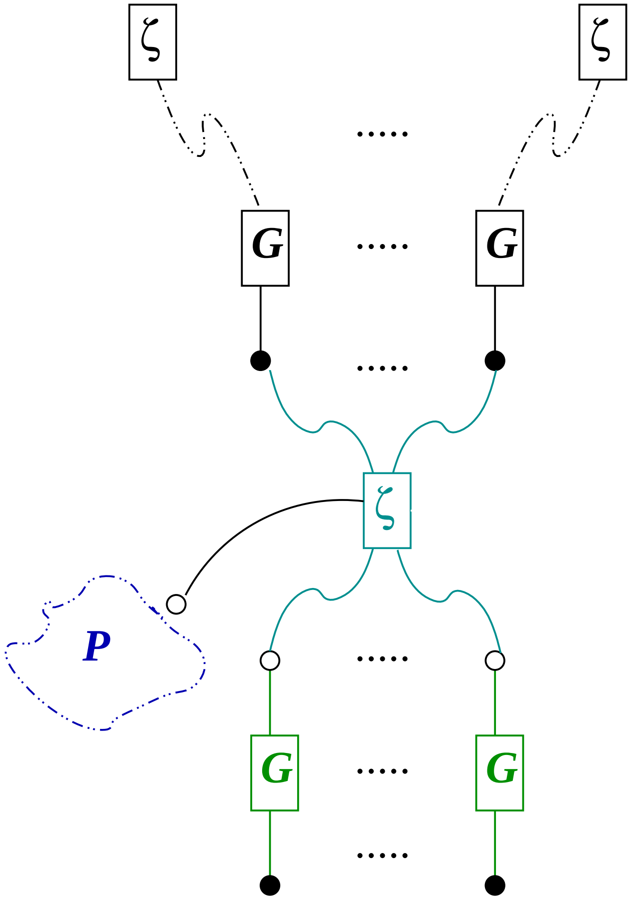
KAOS & Hypergraphs

$$[\![\ s ::^L P\ ]\!] \ = \ \Gamma \vdash (\nu\ \vec{x}, p)([\![\ P\ ]\!]_p \mid \mathfrak{S}^s{}_{m,n}(\vec{u}, \vec{x}, p) \mid \prod_{j=1}^{n} G_{t_j}^{\kappa_j}(x_j, v_j))$$



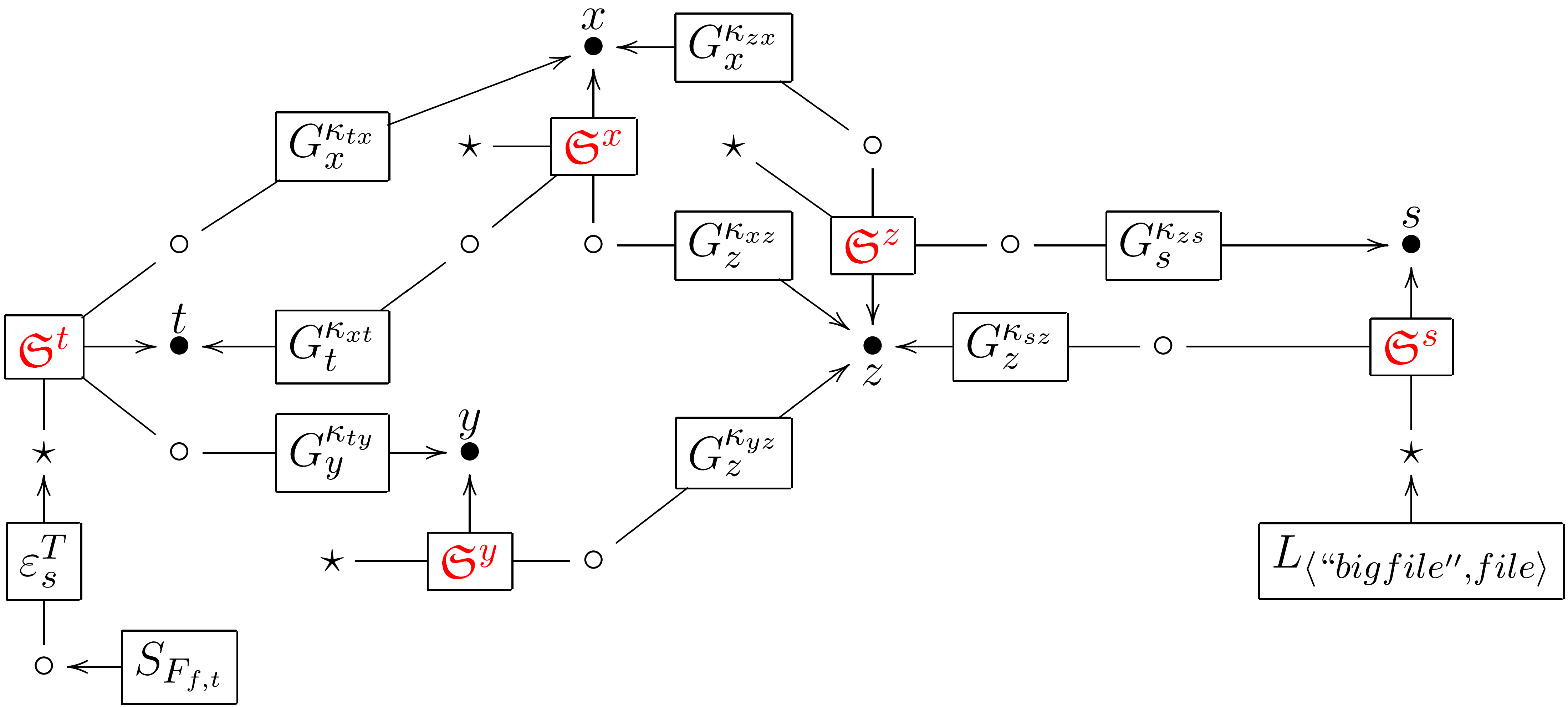
KAOS & Hypergraphs

$$[\![s ::^L P]\!] = \Gamma \vdash (\nu\, \vec{x}, p)([\![P]\!]_p \mid \mathfrak{S}^s{}_{m,n}(\vec{u}, \vec{x}, p) \mid \prod_{j=1}^{n} G_{t_j}^{\kappa_j}(x_j, v_j))$$



$$
\begin{aligned}
[\![\mathbf{nil}]\!]_p &= nil \\
[\![\mathbf{o}\langle t \rangle]\!]_p &= L_{\mathbf{o}\langle t \rangle}(p) \\
[\![\gamma.P]\!]_p &= L_{\gamma.P}(p) \\
[\![\varepsilon(P)@s]\!]_p &= (\nu\, u)(\varepsilon_s^{T(P)}(u, p) \mid S_P(u)) \\
[\![P_1 \mid P_2]\!]_p &= [\![P_1]\!]_p \mid [\![P_2]\!]_p \\
[\![rec\, X.\, P]\!]_p &= [\![P[^{rec\, X.\, P}/_X]]\!]_p.
\end{aligned}
$$

Graph for the message example



KAOS Graph semantics: pros & cons

KAOS Graph semantics: pros & cons

– Many productions (recently reduced :-)

KAOS Graph semantics: pros & cons

– Many productions (recently reduced :-)

+ Determines the “optimal” path (also KAOS)

KAOS Graph semantics: pros & cons

– Many productions (recently reduced :-)

\+ Determines the “optimal” path (also KAOS)

\+ Path reservation

KAOS Graph semantics: pros & cons

– Many productions (recently reduced :-)

+ Determines the “optimal” path (also KAOS)

+ Path reservation

+ Path routing

KAOS Graph semantics: pros & cons

– Many productions (recently reduced :-)
+ Determines the “optimal” path (also KAOS)
+ Path reservation
+ Path routing

Theorem KAOS remote actions are routed on paths with minimal cost (wrt the c-semiring operations)