Real-Time Simulation of the German Autobahn Network

Topics

- CA–Model of Traffic Simulation
- Routing
- Dynamic Load Balancing
- Demo
- Outlook

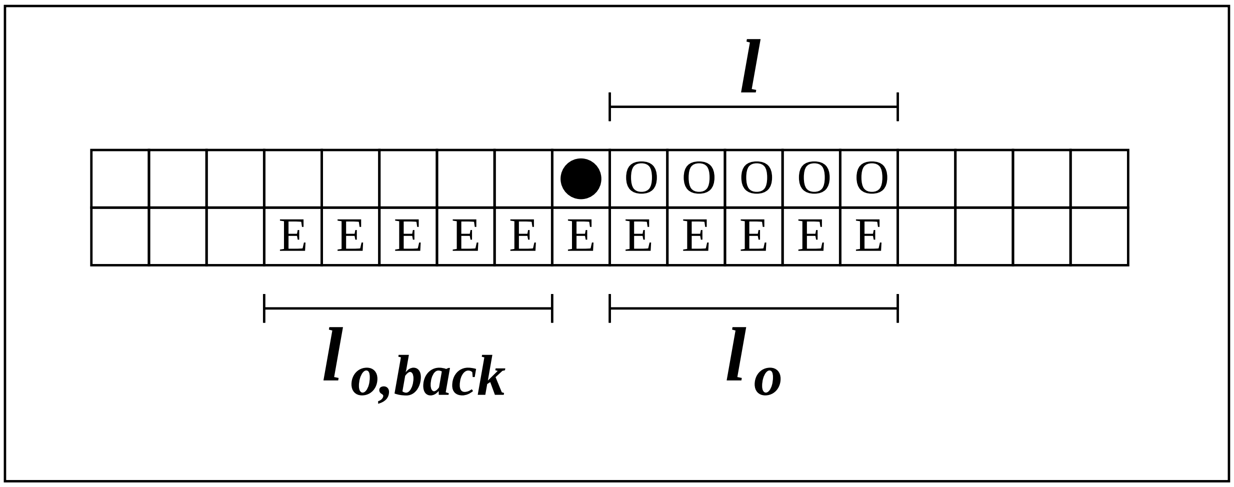
Marcus Rickert

email: mr@zpr.uni-koeln.de

ZPR Universität zu Köln
Zentrum für Paralleles Rechnen

Center for Parallel Computing
University of Cologne and
TSASA - Los Alamos National Lab

Lane Changing Rules for Timestep t_n



l	look ahead same lane
l_o	look ahead other lane
$l_{0,back}$	look back other lane

An Example: $\sim$ Latour III

symmetric asymm. $L \rightarrow R$	asymm. $R \rightarrow L$
$l = v + 1$ $l_o = v + 1$ $l_{o,back} = v_{max}$	- $l_o = v + 1$ $l_{0,back} = v_{max}$

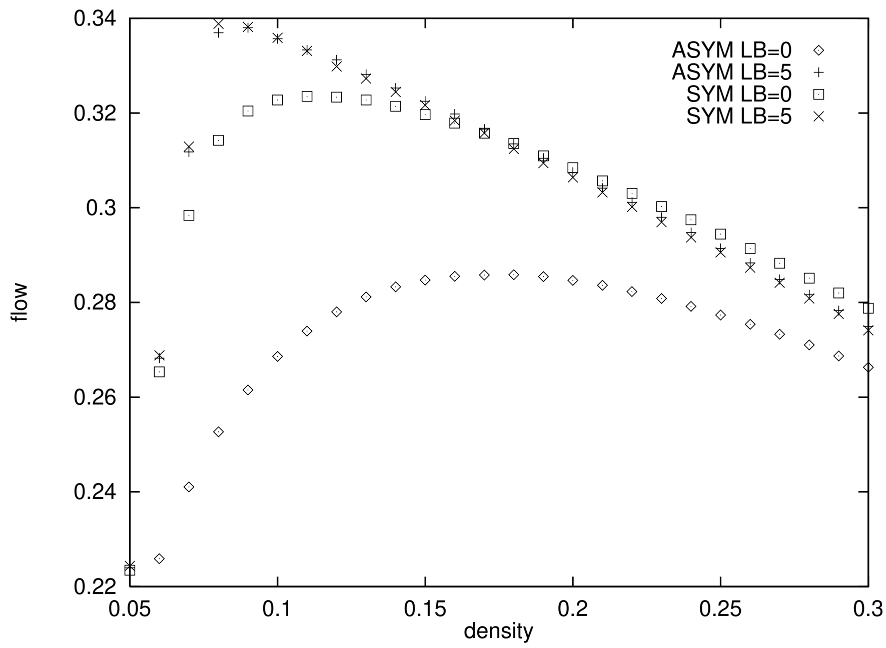
Observations

- $lookback > 0 +$ lane changing slightly improves throughput over single lane
- $lookback \sim 0$ separates symmetric and asymmetric case

Current work by *Nagel, Latour, Schreckenberg, and Rickert* will be published in Physica A.

Importance of Lookback (I)

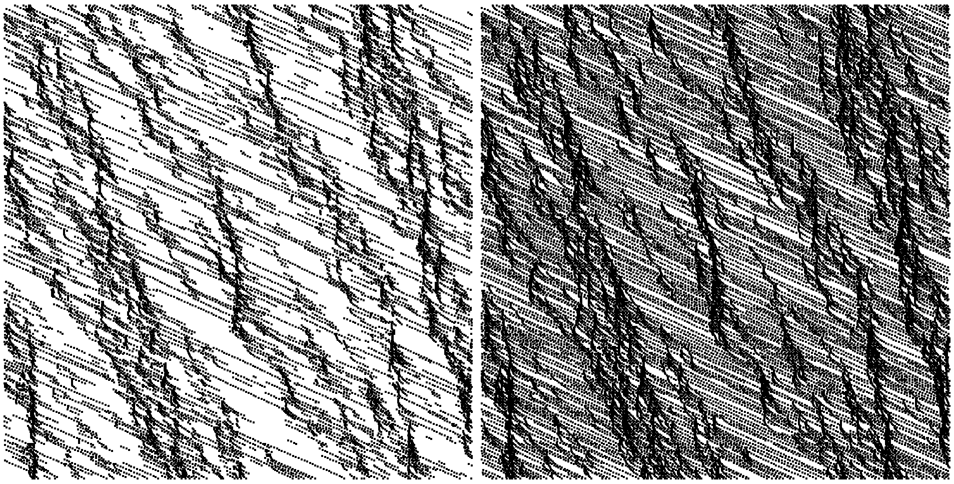
$l_{0,back} > 0$ improves flow for both symmetric and asymmetric rule sets.



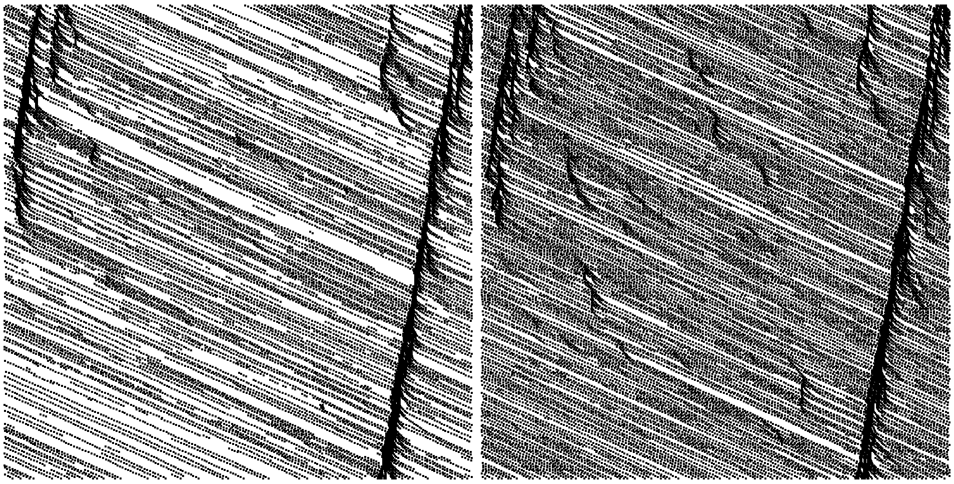
Importance of Lookback (II)

Time Space Plots

$l_{0,back} = 0$



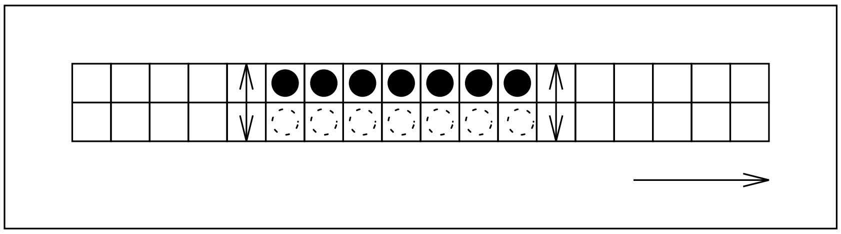
$l_{o,back} = 5$



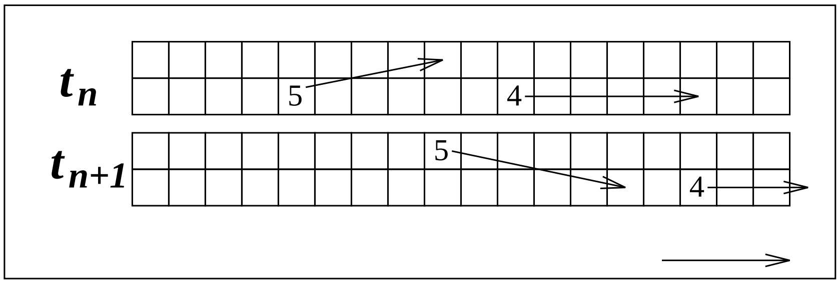
Artifacts

Ping Pong Lane Changes

At *high densities* vehicles cluster:



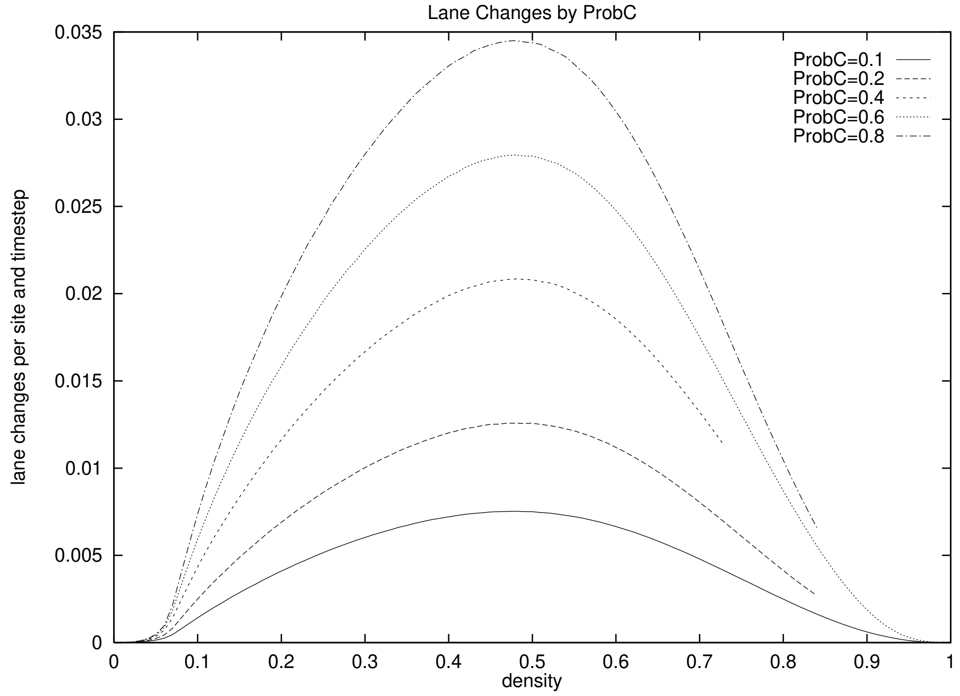
At free flow densities (asymmetric) passing often fails:

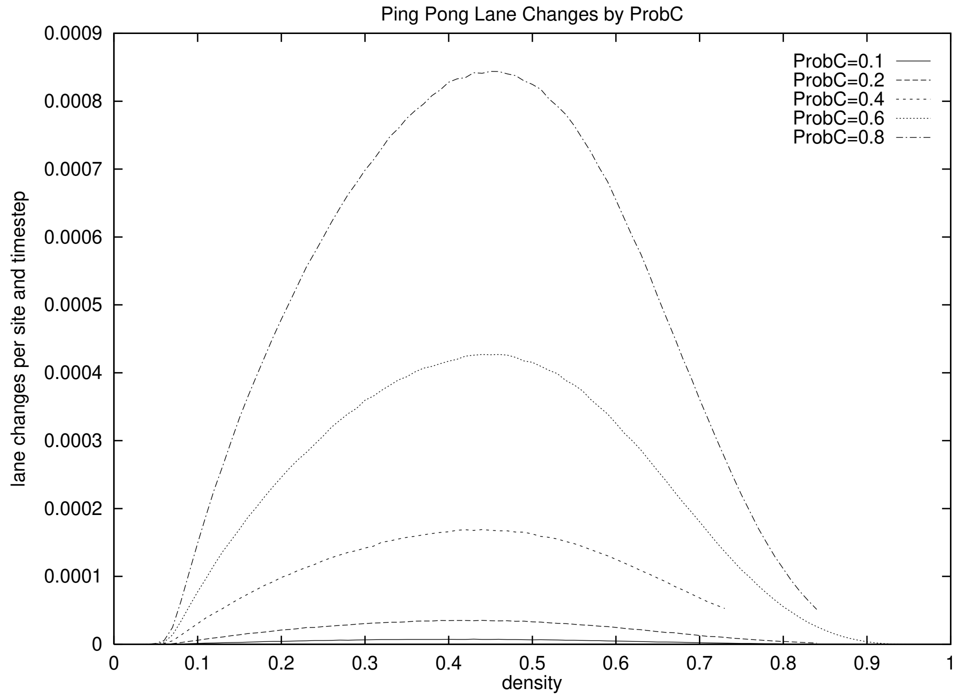


Introduction of stochasticity for lane changing considerably reduces ping pong lane changes!

Network Traffic Simulation

Ping Pong Lane Changes





Universität zu Köln
Zentrum für Paralleles Rechnen

Center for Parallel Computing
University of Cologne and
TSASA - Los Alamos National Lab

Problems of CA

- lane changing probability is too high
- lane densities are not modelled correctly ($\varrho_l < \varrho_r$ for $\varrho > \varrho_c$)
  $\rightarrow$ Wagner, Gawron

CA Simulation Network

Building Blocks

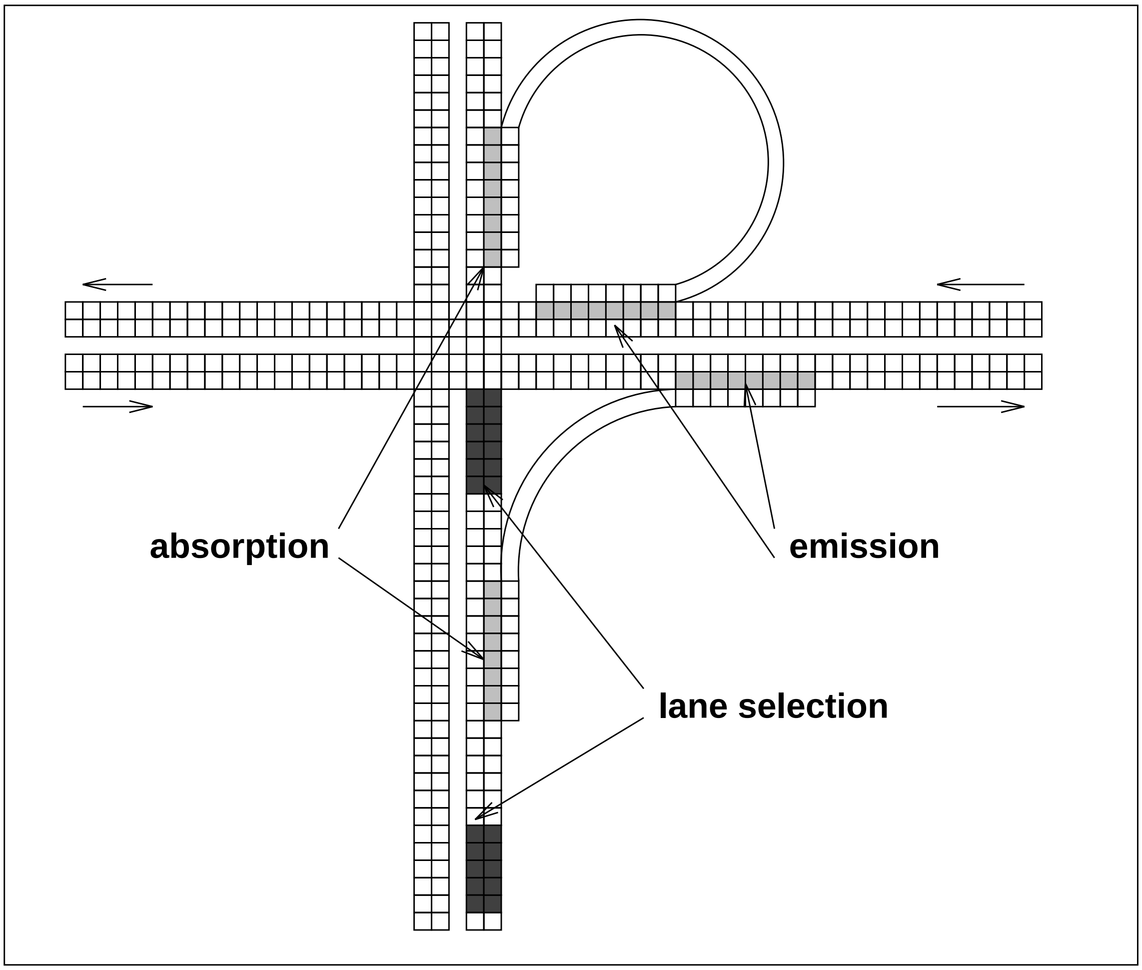
- multilane CA
- insertion, absorption point
- boundary
- source, sink

Composite Elements

- net terminator
- ramp
- intersection

CA Routeplan Execution

Vehicles are assigned individual route plans which influence the CA behavior near intersections and ramps.



Routing

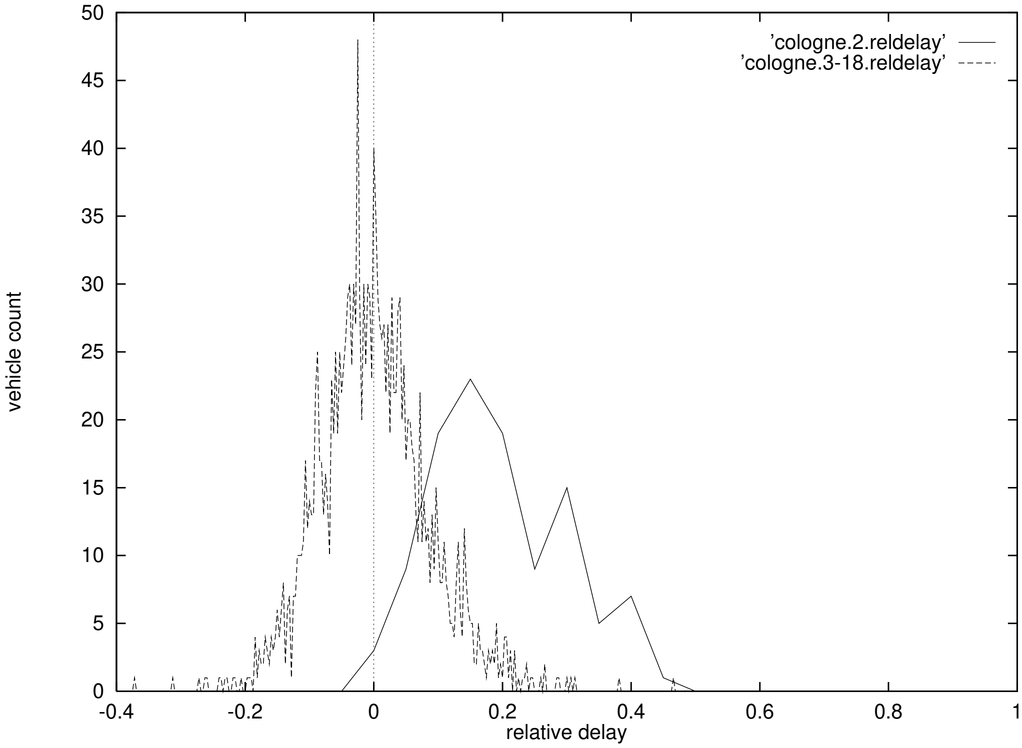
Normally an OD-matrix should be given as input. But since this data is still not available, an 'arbitrary' matrix is used guiding vehicles from each source to each sink of the network.

Iteration of Route Planing

1. preload edge weights with time independant free–flow–velocity
2. compute route plans
3. execute route plans and store actual time dependant edge weights
4. goto 2

Routing example

For low densities $\varrho = 0.05$ the process converges after the first iteration.



Question

For which densities and ratios of routed/unrouted vehicles does the iteration converge? Quality?

Distribution of Network

The CA traffic network is assigned to a graph of vertices and edges which are handled by the Parallel Toolbox.

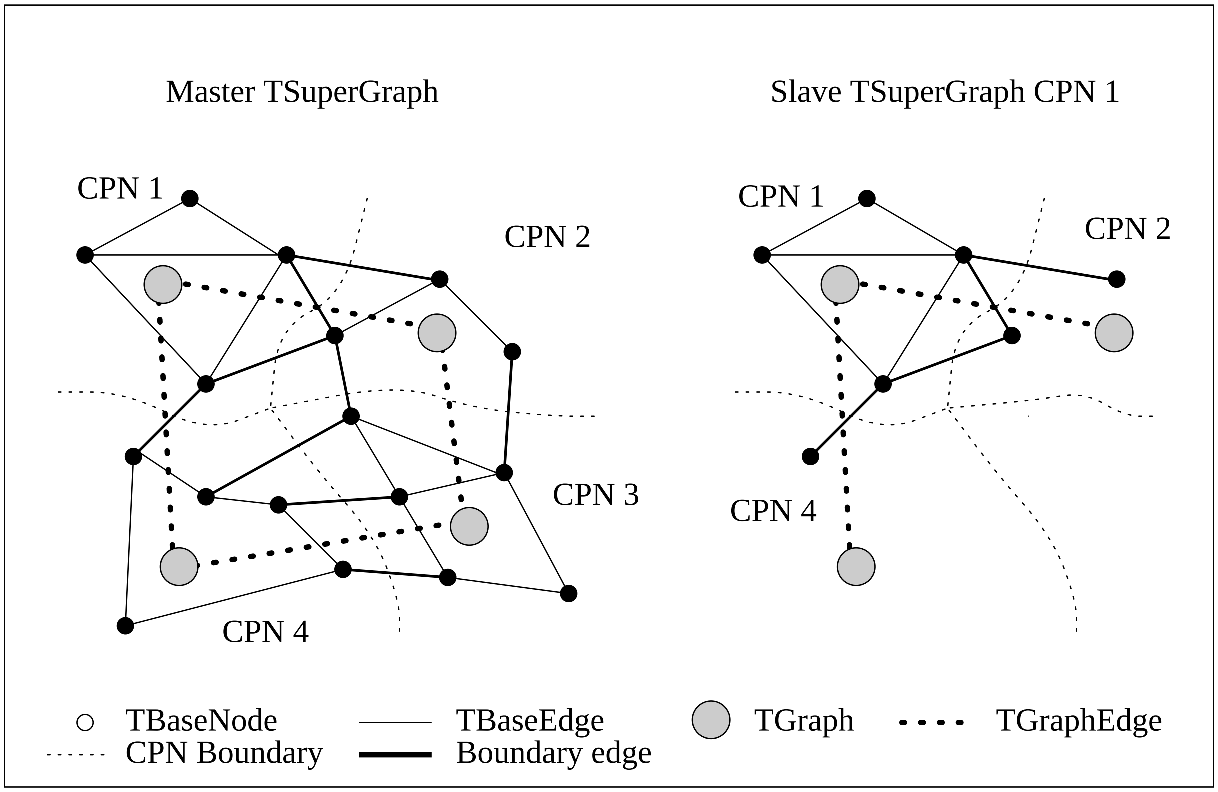
Vertices correspond to terminators, ramps, and intersections + transfer lanes.

Edges correspond to birectional CA multilane segments.

During the initial distribution (done by the master CPN) the vertices are geometrically distributed. Inter-CPN edges are handled by exchange of boundaries.

Local Network

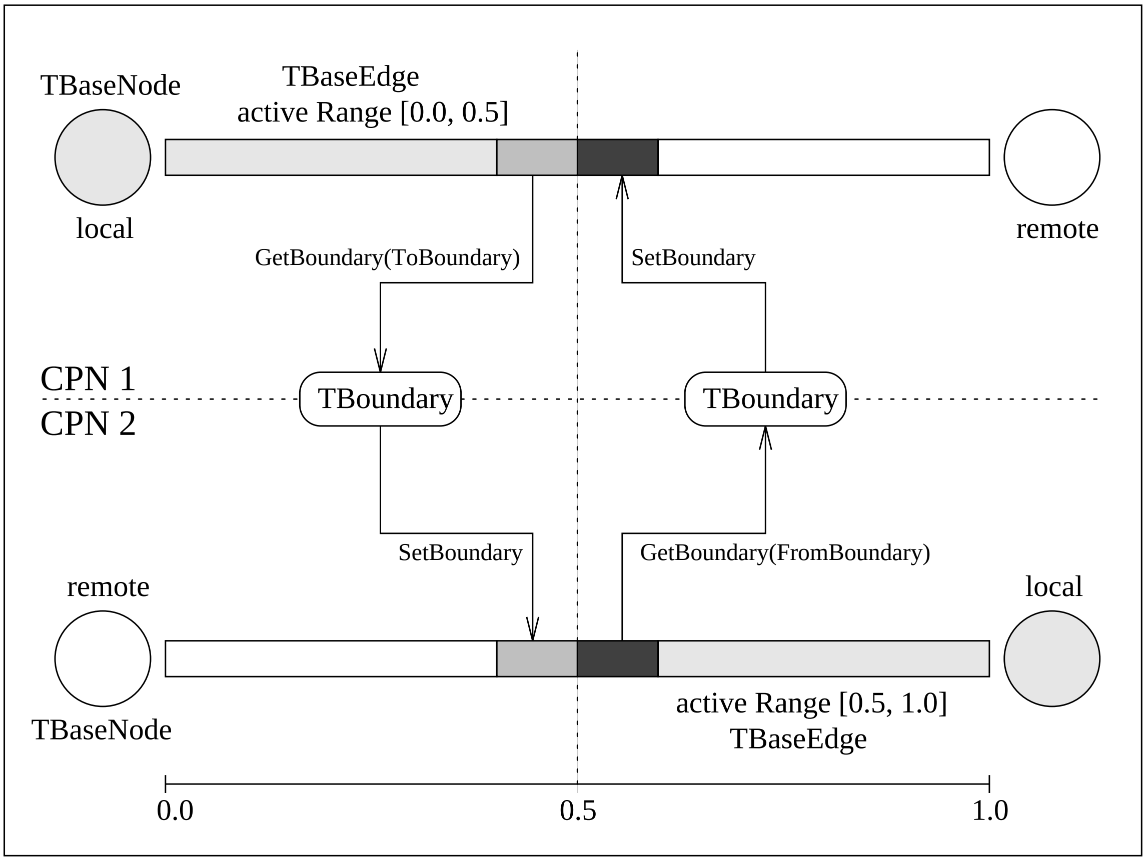
The master CPN has a full copy of the inactive network (mainly for graphics). Each slave CPN has an local active sub network + some inactive dummies.



Network Traffic Simulation

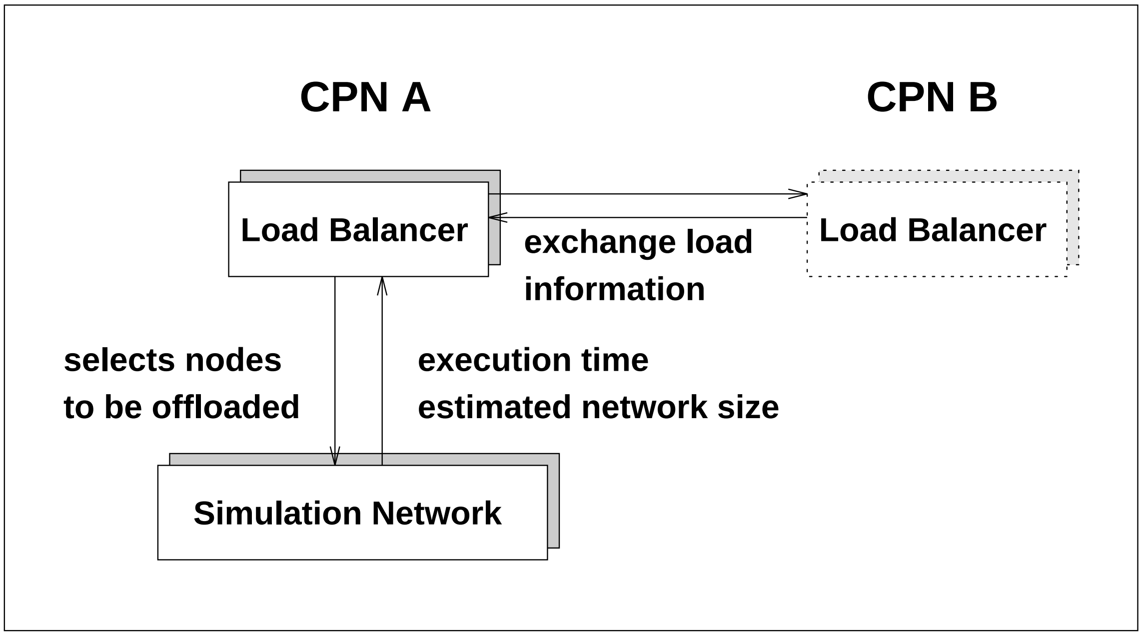
Boundaries

An inter-CPN edge is duplicated with different active ranges on either CPN it refers to. Boundary information is requested from the edges, encoded and passed to the remote CPN.



Load Balancing (I)

Each CPN compares its load with those of its neighbors. If neccessary it initiates a local synchronization and offloads topology to its most idle neighbor.



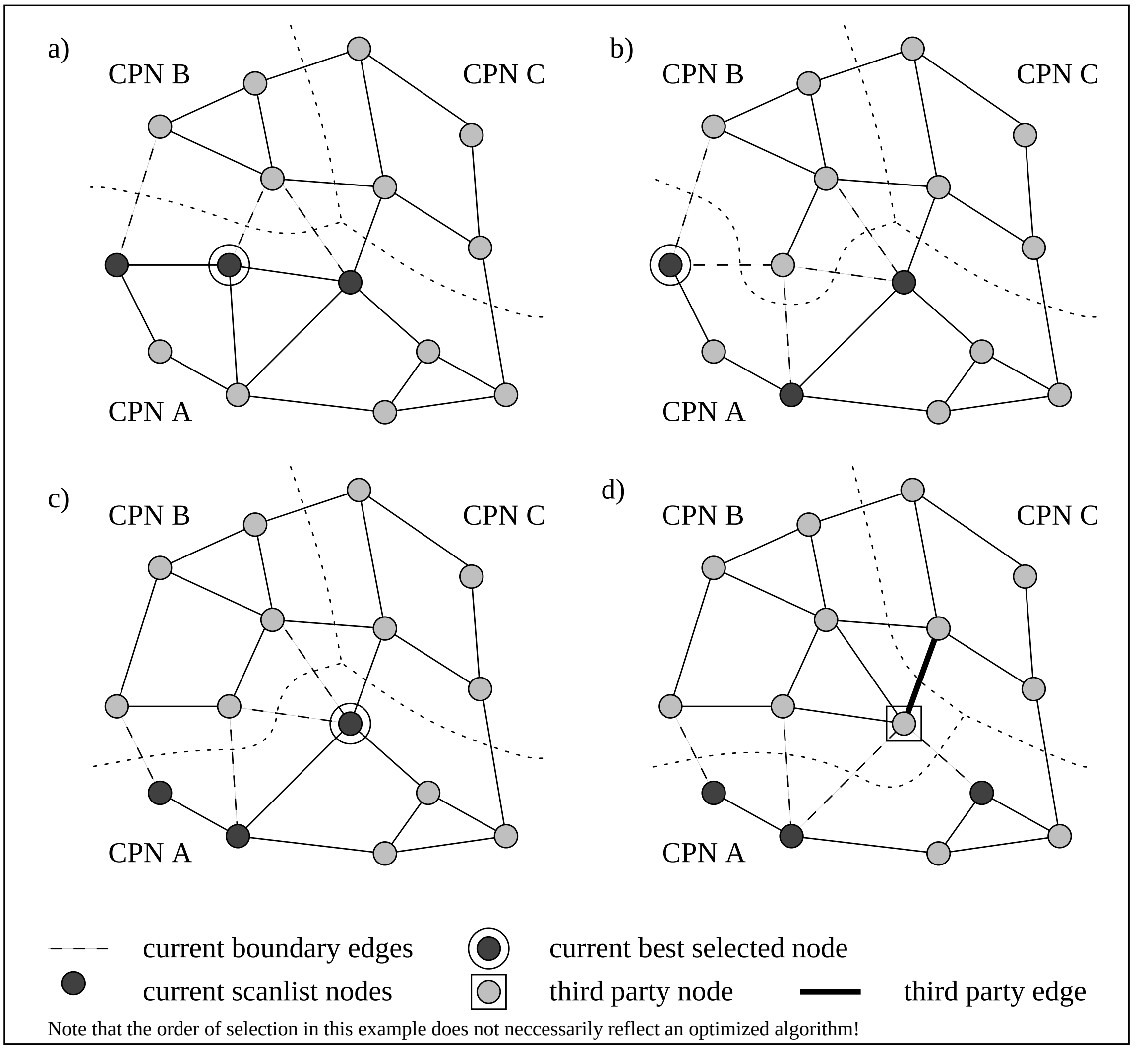
Load Balancing (II)

Topology is offloaded along the common boundaries of the participating CPNs. Vertices located furthest from the center of gravity are selected first which keeps the local sub network 'nicely' shaped (not optimal!).

Optionally, the system can try to maintain one connectivity component on each sub network which reduces communication, but increases granularity.

Load Balancing (III)

Offloading topology along the boundaries



Note that the order of selection in this example does not neccessarily reflect an optimized algorithm!

Load Balancing (IV)

The load of workstations varies over time. The performance

$$P_i = \frac{\text{estimated load}}{\text{execution time}} = \frac{l_i}{t_i}$$

and its minimum over a certain time

$$P_i^{min} = \min_{history} P_i.$$

are used to estimate a safe work load

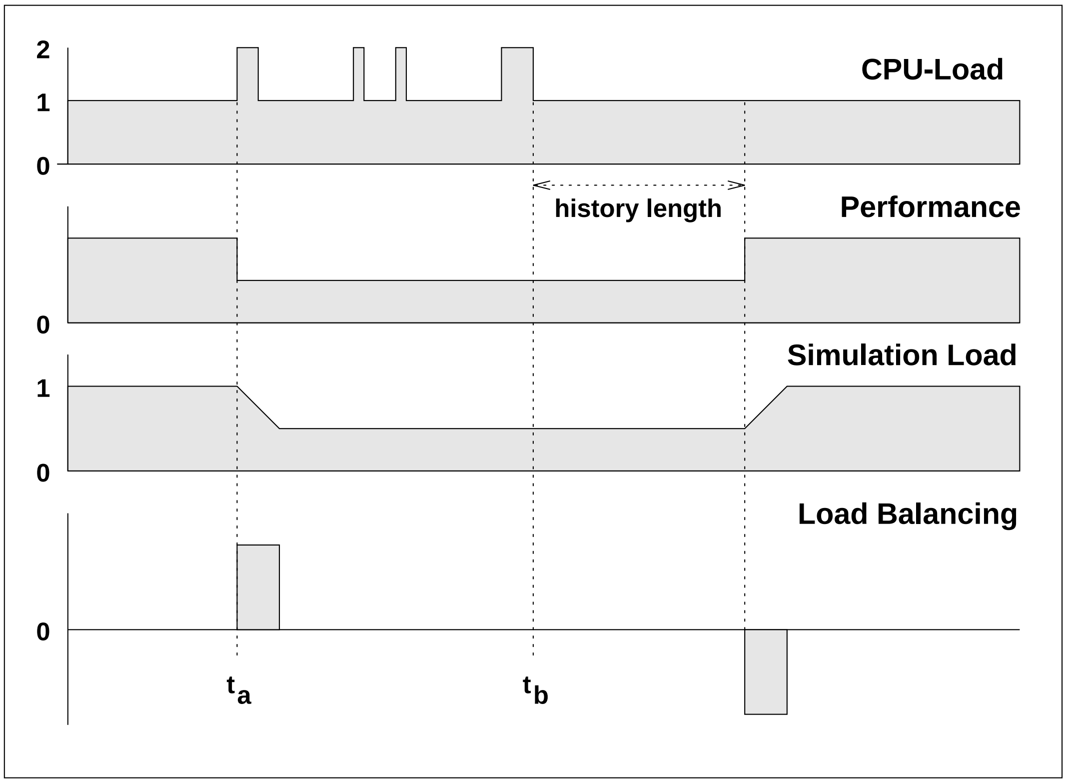
$$L = \frac{l_i}{P_i^{min}}.$$

An optimal transfer amounts to

$$l_t = \frac{l_0 P_j^{min} - l_j P_0^{min}}{P_0^{min} + P_j^{min}}$$

Example Run

The simulation adapts to the additional load between t_a and t_b.



Current Application Structure

Micro Simulation

(descendant C++ classes)

Parallel Toolbox 1.0

(C++ base classes)

PVM

Message Passing Architectures

- **Single CPN**
- **Workstation Cluster**
- **Distributed Memory**
- **Shared Memory (emulated)**

Network Traffic Simulation

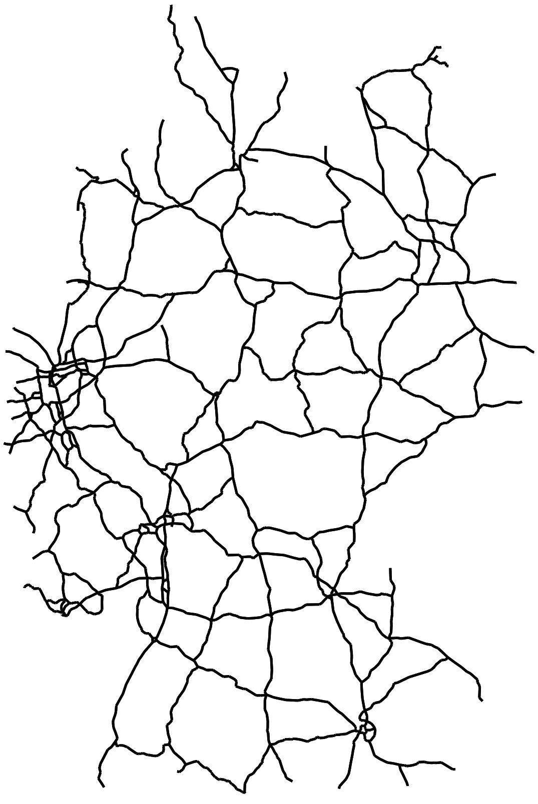
Testbed
German Autobahn Network

3300 nodes, 3400 edges

$\sim 75,000$ kilometers (lane corrected)

10,000,000 sites

1,000,000 vehicles



Center for Parallel Computing
University of Cologne and
TSASA - Los Alamos National Lab

Demos

Start simulation on SGI (ARCH SGI5) with 16 CPNs. It will (hopefully) reach realtime after some dynamic load balancing to compensate for the insufficient initial distribution.

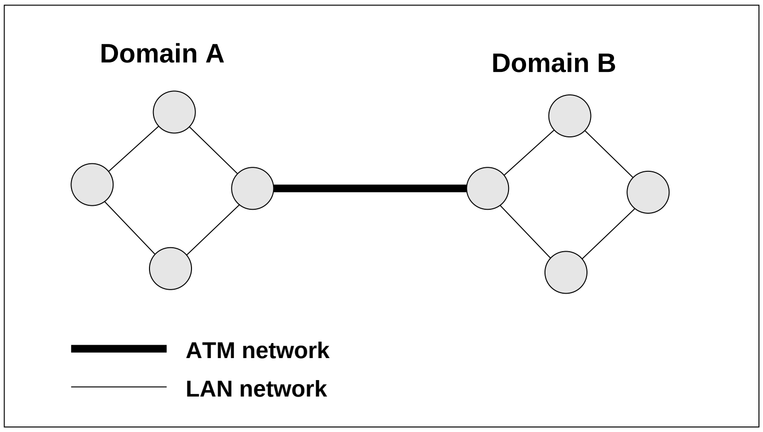
Start simulation on homogeneous CPN cluster with master *Jukebox* (ARCH SUN4), *ParC04*, and 2 CPNs on *Sysiphos* (ARCH SUN4SOL2).

Add *Speedy* (ARCH LINUX) (on the fly) which makes the topology heterogeneous.

Remove *Speedy* again (on the fly).

Outlook

- Reduce number of boundaries
- Improve mapping of simulation network to CPN network:



- Include global correction of load balancing to reduce static load gradients (especially on dedicated systems)
- Investigate best monitoring parameters in workstation clusters

Outlook (II)

- Include Shared Memory to allow for hybrid computer networks