Aalto University

Distributed Algorithms 2024

5 CONGEST model: Bandwidth limitations

LOCAL model

=

port-numbering model
+ unique identifiers

Nodes have distinct labels from $\{1, 2, \ldots, \mathrm{poly}(n)\}$

CONGEST model

=

LOCAL model
+ bandwidth limitation

Messages at most $O(\log n)$ bits

LOCAL · unbounded messages

- everything trivial to solve in O(diameter) rounds: gather full input and solve locally

CONGEST · bounded messages

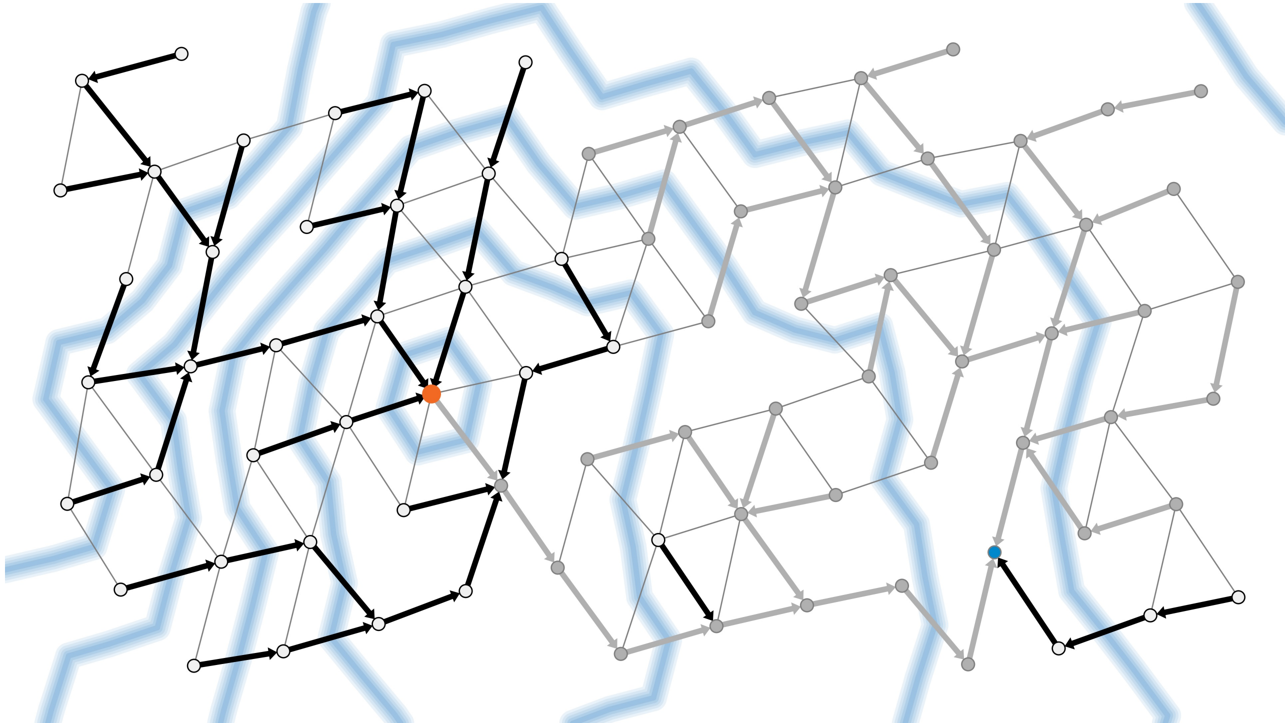
- gathering everything is way too expensive
- O(diameter) and $O(n)$ is nontrivial

Designing efficient algorithms in CONGEST model

Pipelining

Pipelining

- Multiple operations in progress ***simultaneously***
- Using ***different resources***
- In APSP algorithm:
  - multiple waves
  - using different communication links



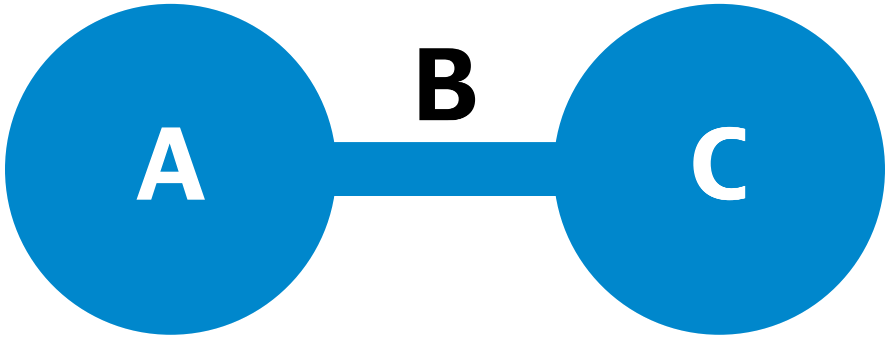
Pipelining

- ***Does not reduce the total number of messages***
  - only removes idle periods between messages
- If all communication links are already sending useful data every round, no room for pipelining

What kind of problems cannot be solved fast in CONGEST model?

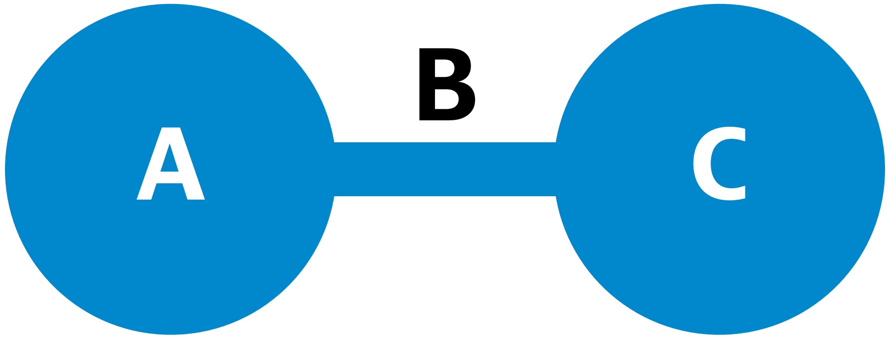
Typical hard problems

- **A:** complicated, lots of information
- **B:** **bottleneck**
  - can only send small number of bits per round from A to C
- **C:** need to know A



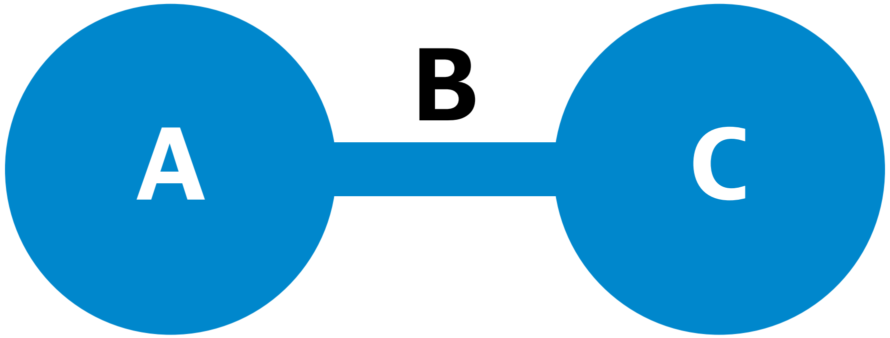
Proving hardness

- Counting argument
- *Many* possible inputs in A
- *Few* possible messages across bottleneck B



Proving hardness

- Counting argument
- *Many* possible inputs in A
- *Few* possible messages across bottleneck B
- Contradiction:
  - ***different*** inputs in A
  - ***same*** messages across B
  - ***same*** output in C



Aalto University

Distributed Algorithms 2024

6 Randomized algorithms

Recap

- **Deterministic algorithms in PN model**
  - $\mathrm{init}_d(\ldots)$, $\mathrm{send}_d(\ldots)$, $\mathrm{receive}_d(\ldots)$
- **Deterministic algorithms in LOCAL model**
  - add ***unique identifiers***
- **Deterministic algorithms in CONGEST model**
  - add ***bandwidth constraints***

Randomized algorithms

- **Randomized algorithms in PN model**
  - $\text{init}_d(\ldots)$, $\text{receive}_d(\ldots)$: ***probability distribution***
- **Randomized algorithms in LOCAL model**
  - add unique identifiers
- **Randomized algorithms in CONGEST model**
  - add bandwidth constraints

Guarantees

- **Monte Carlo**
  - *guaranteed* running time
  - probabilistic output quality
- **Las Vegas**
  - probabilistic running time
  - *guaranteed* output quality

Guarantees

- **Monte Carlo**
  - *guaranteed* running time
  - probabilistic output quality
- **Las Vegas**
  - probabilistic running time
  - *guaranteed* output quality
- **"With high probability"** (w.h.p.)

Role of randomness

- Sometimes randomness is the only way to design fast distributed algorithms
- Example: **sinkless orientation**
  - deterministic LOCAL: $O(\log n)$ is best possible
  - randomized LOCAL: $O(\log \log n)$ w.h.p. is best possible

Role of randomness

- Sometimes randomness is just one of many ways to break symmetry
- Example:
  - ***PN model*** + randomness + knowledge of n: you can construct ***unique identifiers*** w.h.p.

Video

Pretty simple idea:

- nodes are ***active*** with probability 1/2
- only active nodes try to pick a ***random free color***
- stop if successful

Simplest possible idea:

- everyone tries to pick a ***random free color***
- stop if successful

Aalto University

Distributed Algorithms 2024

1–6 Recap

Models of computing

- **PN**
- **LOCAL** — unique identifiers
- **CONGEST** — bandwidth constraints
- Deterministic and **randomized** algorithms

Canonical problems

- **Vertex coloring**
  - coloring = schedule
  - coloring breaks symmetry
- Used to solve many other problems
- Coming later: used to show that other problems are hard
- Demonstrates different algorithm design ideas

Algorithm ideas

- **Conflict avoidance & coordination**
- Process nodes by color classes
- Send proposals one by one
- Random subset of nodes is active
- Pipelining
- Algebraic techniques

New kinds of challenges

- **Unknown systems**
  - algorithms that work in any network
- **Partial information**
  - making decisions based on local information
- **Parallelism**
  - many nodes act simultaneously