Ad Hoc Networks: Pushing Mobile and Wireless Communication (Since 1970)

Roger Wattenhofer
Why Do You Study Ad Hoc Networks?
CHECKLIST

- important applications
- it’s fun
- for the money
CHECKLIST, really

- mobile
- wireless
- energy
Mobile Networks?
Distributed Control!
Complexity Theory

Can a Computer Solve Problem $P$ in Time $t$?
Distributed

Complexity Theory

Network

Can a Computer Solve
Problem \( P \) in Time \( t \)?
Network
Distributed
Complexity Theory

Network
Can a Computer Solve
Problem $P$ in Time $t$?
Distributed (Message-Passing) Algorithms

• Nodes are agents with unique ID’s that can communicate with neighbors by sending messages. In each synchronous round, every node can send a (different) message to each neighbor.
Distributed (Message-Passing) Algorithms

- Nodes are agents with unique ID’s that can communicate with neighbors by sending messages. In each synchronous round, every node can send a (different) message to each neighbor.

  each round:
  every node:
  1. send msgs
  2. rcv msgs
  3. compute

- Distributed (Time) Complexity: How many rounds does problem take?
An Example

each round:
every node:
1. send msgs
2. rcv msgs
3. compute
How Many Nodes in Network?

each round:
  every node:
    1. send msgs
    2. rcv msgs
    3. compute
How Many Nodes in Network?

each round:
every node:
1. send msgs
2. rcv msgs
3. compute
How Many Nodes in Network?

each round:
every node:
1. send msgs
2. rcv msgs
3. compute
How Many Nodes in Network?
How Many Nodes in Network?
How Many Nodes in Network?
How Many Nodes in Network?
How Many Nodes in Network?

With a simple flooding/echo process, a network can find the number of nodes in time $O(D)$, where $D$ is the diameter (size) of the network.
Diameter of Network?

- **Distance** between two nodes = Number of hops of shortest path
Diameter of Network?

- **Distance** between two nodes = Number of hops of shortest path
• **Distance** between two nodes = Number of hops of shortest path
• **Diameter** of network = Maximum distance, between any two nodes
Diameter of Network?
Diameter of Network?
Diameter of Network?
Diameter of Network?
Diameter of Network?
Diameter of Network?
Networks Cannot Compute Their Diameter in Sublinear Time!

(even if diameter is just a small constant)

Pair of rows connected neither left nor right? Communication complexity:
Transmit $\Theta(n^2)$ information over $O(n)$ edges $\rightarrow \Omega(n)$ time!

[Frischknecht, Holzer, W, 2012]
Distributed Complexity Classification

- $1 \leq \log^* n \leq \text{polylog } n \leq D \leq \text{poly } n$

1. $\log^* n$
   - e.g., dominating set approximation in planar graphs
   - MIS, approx. of dominating set, vertex cover, ...
   - various problems in growth-bounded graphs
   - count, sum, spanning tree, ...

2. polylog $n$
   - diameter, MST, verification of e.g. spanning tree, ...

3. $D$

4. poly $n$

e.g., [Kuhn, Moscibroda, W, 2014]
Sublinear Algorithms
Self-Assembly
Self-Stabilization
Distributed Complexity
Applications e.g. Multi-Core
Sublinear Algorithms
Dynamic (e.g. Ad Hoc) Networks
Sublinear Algorithms

Self-Stabilization

Distributed Complexity

Applications e.g. Multi-Core

Dynamic (e.g. Ad Hoc) Networks

Sublinear Algorithms

Self-Stabilization

Distributed Complexity

Applications e.g. Multi-Core

Dynamic (e.g. Ad Hoc) Networks
Wireless Communication?
Capacity!
Physical (SINR) Model
Signal-To-Interference-Plus-Noise Ratio (SINR) Formula

\[
\frac{P_u}{d(u,v)\alpha} \geq \frac{\sum_{w \in V \setminus \{u\}} \frac{P_w}{d(w,v)\alpha}}{N + \sum_{w \in V \setminus \{u\}} \frac{P_w}{d(w,v)\alpha}} \beta
\]

- Power level of sender \( u \)
- Received signal power from sender
- Path-loss exponent
- Noise
- Received signal power from all other nodes (= interference)
- Distance between two nodes
- Minimum signal-to-interference ratio
Example: Protocol vs. Physical Model

Assume a single frequency (and no fancy decoding techniques!)

Is spatial reuse possible?

NO  Protocol Model

YES  With power control

Let $\alpha=3$, $\beta=3$, and $N=10nW$

Transmission powers: $P_B=-15$ dBm and $P_A=1$ dBm

SINR of A at D:
$$\frac{1.26mW/(7m)^3}{0.01\mu W + 31.6\mu W/(3m)^3} \approx 3.11 \geq \beta$$

SINR of B at C:
$$\frac{31.6\mu W/(1m)^3}{0.01\mu W + 1.26mW/(5m)^3} \approx 3.13 \geq \beta$$
This works in practice

... even with very simple hardware

Time for transmitting 20’000 packets:

<table>
<thead>
<tr>
<th></th>
<th>Time required</th>
<th></th>
<th></th>
<th>Messages received</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>standard MAC</td>
<td>“SINR-MAC”</td>
<td></td>
<td>standard MAC</td>
</tr>
<tr>
<td>Node $u_1$</td>
<td>721s</td>
<td>267s</td>
<td>Node $u_4$</td>
<td>19999</td>
</tr>
<tr>
<td>Node $u_2$</td>
<td>778s</td>
<td>268s</td>
<td>Node $u_5$</td>
<td>18784</td>
</tr>
<tr>
<td>Node $u_3$</td>
<td>780s</td>
<td>270s</td>
<td>Node $u_6$</td>
<td>16519</td>
</tr>
</tbody>
</table>

Speed-up is almost a factor 3

Possible Application – Hotspots in WLAN
Possible Application – Hotspots in WLAN
The Capacity of a Network

(How many concurrent wireless transmissions can you have)
## Convergecast Capacity in Wireless (Sensor) Networks

<table>
<thead>
<tr>
<th>Topology</th>
<th>Max. rate in arbitrary, worst-case deployment</th>
<th>Max. rate in random, uniform deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol Model</td>
<td>$\Theta(1/n)$</td>
<td>$\Theta(1/\log n)$</td>
</tr>
<tr>
<td>Physical Model (power control)</td>
<td>$\Omega(1/\log^3 n)$</td>
<td>$\Omega(1/\log n)$</td>
</tr>
</tbody>
</table>

- [Moscibroda, W, 2006] (Worst-Case Capacity)
- [Giridhar, Kumar, 2005] (Classic Capacity)
Capacity of a Network

Real Capacity

- How much information can be transmitted in any network?

Classic Capacity
- How much information can be transmitted in nice networks?

Worst-Case Capacity
- How much information can be transmitted in nasty networks?
Core Capacity Problems

Given a set of arbitrary communication links

One-Shot Problem
Find the maximum size feasible subset of links

Scheduling Problem
Partition the links into fewest possible slots, to minimize time
Open problem: Only $O(\log n)$ approximation using the one-shot subroutine
Energy Efficiency?
Clock Synchronization!
Clock Synchronization Example: Dozer

- Multi-hop sensor network with duty cycling
- 10 years of network life-time, mean energy consumption: 0.066mW
- High availability, reliability (99.999%)
- Many different applications use Dozer: TinyNode, PermaSense, etc.

[Burri, von Rickenbach, W, 2007]
Problem: Physical Reality

\[ \text{clock rate} \]

\[ \begin{align*}
1 + \varepsilon \\
1 - \varepsilon
\end{align*} \]

message delay

![Graph showing message delay distribution](image_url)
Clock Synchronization in Theory?

Given a communication network

1. Each node equipped with hardware clock with drift
2. Message delays with jitter

Goal: Synchronize Clocks ("Logical Clocks")

- Both global and local synchronization!
Time Must Behave!

- Time (logical clocks) should not be allowed to stand still or jump
Local Skew

Tree-based Algorithms
e.g. FTSP

Neighborhood Algorithms
e.g. GTSP

Bad local skew
Synchronization Algorithms: An Example ("A^\text{max}\)"

- **Question:** How to update the logical clock based on the messages from the neighbors?

- **Idea:** Minimizing the skew to the **fastest** neighbor
  - Set clock to **maximum** clock value you know, forward new values immediately

- **First all messages are slow (1), then suddenly all messages are fast (0)!”

![Diagram showing time progression and clock values](image)
Local Skew: Overview of Results

- Everybody’s expectation, 10 years ago („solved“)
- Lower bound of \( \log D / \log \log D \) [Fan & Lynch, PODC 2004]
- Kappa algorithm [Lenzen et al., FOCS 2008]
- Tight lower bound [Lenzen et al., PODC 2009]
- Blocking algorithm [Locher et al., DISC 2006]
- All natural algorithms [Locher et al., DISC 2006]
- Dynamic Networks! [Kuhn et al., SPAA 2009]
- Dynamic Networks! [Kuhn et al., PODC 2010]
- Together [JACM 2010]
Experimental Results for Global Skew

FTSP

PulseSync

[Lenzen, Sommer, W, 2009]
Experimental Results for Global Skew

FTSP

PulseSync

[Lenzen, Sommer, W, 2009]
Summary
Thank You!

Questions & Comments?

Thanks to my co-authors, mostly
Silvio Frischknecht
Magnus Halldorsson
Stephan Holzer
Christoph Lenzen
Thomas Moscibroda
Philipp Sommer