Distributed Resilient Computing over MANET

Nitin Vaidya
University of Illinois at Urbana-Champaign

MURI Review
October 2012

Research supported in part by
MURI Contributions

How to build Robust Networks?

Upper layers
Transport
Network
Link
Physical Layer
MURI Contributions

Computing/coordination over networks?

- Upper layers
- Transport
- Network
- Link
- Physical Layer

How to build Robust Networks?
MURI Contributions

Computing/coordination over networks?

Distributed tasks difficult to implement

How to build Robust Networks?

Layered network model:
- Physical Layer
- Link
- Network
- Transport
- Upper layers
This Talk

Consensus / Agreement

Upper layers
  Coordination
  Transport
  Network
  Link
  Physical Layer
Many Faces of Consensus

- What time is it?

- Network of clocks …

agree on a common notion of time
Many Faces of Consensus

- What is the temperature?

- Network of sensors …

agree on current temperature
Many Faces of Consensus

- Should we trust 😊? 

- Web of trust …

agree whether 😊 is good or evil
Many Faces of Consensus

- Is he still in the network?

- Network of hosts…

agree on network membership
Consensus … one definition

- want to decide which way to fly
- Every proposes a direction
- Consensus $\approx$ decision in convex hull of inputs
Average Consensus

- want to decide which way to fly
- Every proposes a direction

- Consensus $\approx$ average of inputs
Our Previous Contributions

**Consensus + Broadcast in presence of adversaries**

- Network capacity-aware algorithms
- Algorithms linear in # nodes

**Applications:**

- Securely maintaining consistent (core) network state despite compromised nodes
This Talk … Iterative Algorithms

Lightweight local algorithms … global properties
This Talk … Iterative Algorithms

Lightweight local algorithms … global properties

- Average consensus + Lossy links
  - Application: Distributed sensing/optimization in MANET

- Consensus + Adversarial nodes
  - Application: Distributed clock synchronization in MANET
Review
Centralized Solution

- A leader collects inputs & disseminates decision
Distributed Iterative Solution … Local Computation

- Initial state $a, b, c = \text{input}$
Distributed Iterative Solution

□ State update (iteration)

\[ a = \frac{3a}{4} + \frac{c}{4} \]

\[ b = \frac{3b}{4} + \frac{c}{4} \]

\[ c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2} \]
\[
\begin{pmatrix}
a \\ b \\ c
\end{pmatrix}
:=
\begin{pmatrix}
3/4 & 0 & 1/4 \\
0 & 3/4 & 1/4 \\
1/4 & 1/4 & 1/2
\end{pmatrix}
\begin{pmatrix}
a \\ b \\ c
\end{pmatrix}
= M
\begin{pmatrix}
a \\ b \\ c
\end{pmatrix}
\]

\[
b = \frac{3b}{4} + \frac{c}{4}
\]

\[
c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2}
\]

\[
a = \frac{3a}{4} + \frac{c}{4}
\]
after 2 iterations

\[
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix} := M \begin{pmatrix}
a \\
b \\
c
\end{pmatrix} = M^2 \begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
\]

after 1 iteration

\[
\begin{pmatrix}
a \\
b \\
c
\end{pmatrix} := M \begin{pmatrix}
a \\
b \\
c
\end{pmatrix} = M^2 \begin{pmatrix}
a \\
b \\
c
\end{pmatrix}
\]

\[
b = \frac{3b}{4} + \frac{c}{4}
\]

\[
c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2}
\]

\[
a = \frac{3a}{4} + \frac{c}{4}
\]
after $k$ iterations

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := M^k \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$b = \frac{3b}{4} + \frac{c}{4}$$

$$c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2}$$

$$a = \frac{3a}{4} + \frac{c}{4}$$
Well-Known Results

- Consensus: at least one node can reach all others
- Average consensus: network strongly connected

with suitably chosen transition matrix $M$
\begin{align*}
\begin{pmatrix} a \\ b \\ c \end{pmatrix} & := M^k \begin{pmatrix} a \\ b \\ c \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \\
\end{align*}

Doubly stochastic $M$

\[
c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2}
\]

\[
b = \frac{3b}{4} + \frac{c}{4}
\]

\[
a = \frac{3a}{4} + \frac{c}{4}
\]
An Implementation: Mass Transfer + Accumulation

- Each node “transfers mass” to neighbors via messages
- Next state = Total received mass

\[ c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2} \]

\[ a = \frac{3a}{4} + \frac{c}{4} \]

\[ b = \frac{3b}{4} + \frac{c}{4} \]
An Implementation:
Mass Transfer + Accumulation

- Each node “transfers mass” to neighbors via messages
- Next state = Total received mass

\[ c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2} \]

\[ \frac{3b}{4} \]

\[ b = \frac{3b}{4} + \frac{c}{4} \]

\[ \frac{b}{4} \]

\[ \frac{3a}{4} \]

\[ a = \frac{3a}{4} + \frac{c}{4} \]
Conservation of Mass

- \( a+b+c \) constant after each iteration

\[
c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2}
\]

\[
a = \frac{3a}{4} + \frac{c}{4}
\]

\[
b = \frac{3b}{4} + \frac{c}{4}
\]
Outline

- Average consensus + Lossy links

- Consensus + Adversarial nodes
Wireless Transmissions Unreliable

\[
c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2}
\]

\[
b = 3\frac{b}{4} + \frac{c}{4}
\]

\[
a = 3\frac{a}{4} + \frac{c}{4}
\]
Impact of Unreliability

\[
\begin{pmatrix}
a \\
b \\
c \\
\end{pmatrix}
=
\begin{pmatrix}
3/4 & 0 & 1/4 \\
0 & 3/4 & 0 \\
1/4 & 1/4 & 1/2 \\
\end{pmatrix}
\begin{pmatrix}
a \\
b \\
c \\
\end{pmatrix}
\]

\[
a = \frac{3a}{4} + \frac{c}{4}
\]

\[
b = \frac{3b}{4} + \frac{c}{4}
\]

\[
c = \frac{a}{4} + \frac{b}{4} + \frac{c}{2}
\]
Conservation of Mass

\[
\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 3/4 & 0 & 1/4 \\ 0 & 3/4 & 0 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}
\]

\[a = 3a/4 + c/4\]
\[b = 3b/4 + c/4\]
\[c = a/4 + b/4 + c/2\]
Average consensus over lossy links?
Existing Solutions … Link Model

Common knowledge on whether a message is delivered
Existing Solutions … Link Model

Common knowledge on whether a message is delivered

S knows
R knows that S knows
S knows that R knows that S knows
R knows that S knows that R knows that ...

S ———> R
All models are wrong; some models are useful.

AND
SOME ARE JUST CUTE

-- George Box
Reality in Wireless Networks

Common knowledge on whether a message is delivered
Our Contribution

- Average consensus **without** common knowledge

- Trade-off … maintain additional per-neighbor state
Solution Sketch

- $S = \text{mass C wanted to transfer to node A in total so far}$
- $R = \text{mass A has received from node C in total so far}$
Solution Sketch

- Node C transmits quantity S
  .... message may be lost

- When it is received, node A accumulates (S-R)
What Does That Do?

- Implements virtual buffers
Does This Work?
Solution

- Run two iterations in parallel
  - First : original inputs
  - Second : \text{input} = 1
Technicalities

- Virtual buffers simulate common knowledge
- Non-doubly stochastic matrices with zeros on the diagonal
- Modulo above differences, algorithm similar to “push-sum”
Ideas have potential to be useful for other iterative computations over lossy links.
Outline

- Average consensus + Lossy links

- Consensus + Adversarial nodes
Adversary

- Compromised nodes

- Can behave entirely arbitrarily … worst-case behavior
Consensus … with adversarial nodes

- want to decide which way to fly
- Every proposes a direction

- Consensus \approx \text{decision in convex hull of good inputs}
Prior Related Work …

- Iterative consensus on complete networks

- Incomplete networks … open problem since mid-80s
Our Contribution

Consensus with adversaries

- Incomplete networks
- Directed links
- Lossy links
Our Contribution

- Necessary and sufficient condition on network topology for consensus with adversaries

  • Provides hints for topology control
Necessary & Sufficient Condition

\[ f \geq f + 1 \]

\[ \geq f + 1 \]
What Next?
The Way Forward

- How to make MANET more “usable” for complex distributed tasks?
The Way Forward

- How to make MANET more “usable” for complex distributed tasks?

- Need efficient distributed primitives for MANET
  - Lossy links
  - Dynamic topology
  - Capacity constraints
  - Energy constraints
  - Security requirements
  - …
The Way Forward
The Way Forward

Need **meaningful contact** between

Networking  ←  Distributed computing
Emphasis on “exact” performance metrics

Constants matter

Information transfer (typically “raw” info)
Distributed Computing

Emphasis on "exact" performance metrics

Constants matter

Information transfer (typically "raw" info)

Communications / Networking

Black box networks

Emphasis on order complexity

Computation affects communication
Thanks !
Thanks!
Additional slides on convergence of consensus in presence of adversaries

Matrix representation
Matrix Representation

- $s = \text{state vector of all nodes}$

$$s = M \times s$$

- Faulty nodes can behave arbitrarily
Matrix Representation

- $s = \text{state vector of \textsc{Fault-Free nodes}}$

\[ s = M[t] * s \]

where $M[t]$ depends on behavior of faulty nodes in $t$-th iteration
Matrix Representation

- $s = \text{state vector of } \text{FAULT-FREE nodes}$

$$s = M[t] \times s$$

where $M[t]$ depends on the behavior of faulty nodes in $t$-th iteration

Many such $M[t]$ exist
Matrix Representation

- Many such $M[t]$ exist

- When the graph satisfies the necessary condition, possible to choose $M[t]$ with adequate “connectivity”

- Results in correct decision
Additional slides on average consensus over lossy links
Existing Solution

When mass not transferred to neighbor,

keep it to yourself
Convergence ... if nodes intermittently connected

\[
\begin{pmatrix}
  a \\
  b \\
  c
\end{pmatrix}
= 
\begin{pmatrix}
  3/4 & 0 & 1/4 \\
  0 & 3/4 & 0 \\
  1/4 & 1/4 & 3/4
\end{pmatrix}
\begin{pmatrix}
  a \\
  b \\
  c
\end{pmatrix}
\]

\[
c = a/4+b/4+c/2+c/4
\]

\[
a = 3a/4 + c/4
\]

\[
b = 3b/4 + c/4
\]
Dynamic Topology

- When C→B transmission unreliable, mass transferred to buffer (d)

- \( d = d + c/4 \)
Dynamic Topology

- When C→B transmission unreliable, mass transferred to buffer (d)

- $d = d + \frac{c}{4}$

No loss of mass even with message loss
Dynamic Topology

- When $C \rightarrow B$ transmission **reliable**, mass transferred to $b$

- $b = \frac{3b}{4} + \frac{c}{4} + d$

No loss of mass even with message loss
Why Doesn’t it Work?

- After $k$ iterations, state of each node has the form

  $$z(k) \times \text{sum of inputs}$$

  where $z(k)$ changes each iteration ($k$)

- Does not converge to average
Solution

- Run two iterations in parallel
  - First: original inputs
  - Second: input = 1

- After k iterations ...
  - First algorithm: \( z(k) \times \text{sum of inputs} \)
  - Second algorithm: \( z(k) \times \text{number of nodes} \)
Solution

- Run two iterations in parallel
  - First: original inputs
  - Second: input = 1

After k iterations …

- first algorithm: \( z(k) \times \text{sum of inputs} \)
- second algorithm: \( z(k) \times \text{number of nodes} \)

\[
\text{ratio} = \text{average}
\]