Shared Memory in an Adverse Environment

Jared Saia
Joint with
Valerie King and Maxwell Young
Shared Memory: An Outsiders View

- Abstracts out messy details about communication thereby allowing simpler description of distributed algorithms
- Decouples 1) development of distributed algorithms and 2) memory management
Wireless Networks with Jamming

- Synchronous communication; time divided into slots
- All players commit to actions simultaneously
- Easy: Adversary can jam only; Hard: Adversary can spoof
Our Communication Model: Jammable Channel

- Alice has a message she wants to send to Bob
- Adversary doesn’t know random bits of Alice or Bob
- Each player pays a cost (mW) for each action (send, listen, or jam)
- Adversary has a finite but unknown budget, B
Goal

- Simulate a Single Writer, Multiple Reader (SWMR) shared register
Did he get it???
Costs

- Costs $S to send on channel
- Costs $L to listen on channel
- Costs $J to block channel
- Adv. spends $B
Costs - Sensors

- Costs $S to send on channel: 38mW
- Costs $L to listen on channel: 35mW
- Costs $J to block channel: >1mW
- Adv. spends $B: >5,000mW
Costs - Sensors

- Costs $S$ to send on channel $38\text{mW}$
- Costs $L$ to listen on channel $35\text{mW}$
- Costs $J$ to block channel $>1\text{mW}$
- Adv. spends $B$ $>5,000\text{mW}$

We assume $S$, $L$, and $J$ are $O(1)$ and $B$ is unknown but finite.
Key Assumption

- If Alice or Bob listen on channel when Adv. jams it, they can detect a “collision”
An Idea

- A round consists of $n$ slots
- Alice sends w/ prob $c/\sqrt{n}$
- Bob listens w/ prob $c/\sqrt{n}$
An Idea

- A round consists of $n$ slots
- Alice sends w/ prob $c/\sqrt{n}$
- Bob listens w/ prob $c/\sqrt{n}$

Assume Adv. blocks w/ prob $1/2$.

Then prob. a given slot is one where Alice sends and there is no jam is $\frac{c}{2\sqrt{n}}$
An Idea

A round consists of $n$ slots

- Alice sends w/ prob $\frac{c}{\sqrt{n}}$
- Bob listens w/ prob $\frac{c}{\sqrt{n}}$

Assume Adv. blocks w/ prob $\frac{1}{2}$.

Then prob. a given slot is one where Alice sends and there is no jam is $\frac{c}{2\sqrt{n}}$

\[
\text{Prob}(\text{Bob fails to get message}) \sim \left(1 - \frac{c}{2\sqrt{n}}\right)^{c\sqrt{n}}
\leq e^{-c^2/2}
\]
An Idea

- A round consists of $n$ slots
- Alice sends w/ prob $c/\sqrt{n}$
- Bob listens w/ prob $c/\sqrt{n}$

Assume Adv. blocks w/ prob $1/2$.

Then prob. a given slot is one where Alice sends and there is no jam is $\frac{c}{2\sqrt{n}}$

$\text{Prob}(\text{Bob fails to get message}) \sim \left(1 - \frac{c}{2\sqrt{n}}\right)^{c\sqrt{n}} \leq e^{-c^2/2}$

Bob can send an ACK in same way
An Idea

- A round consists of \( n \) slots
- Alice sends w/ prob \( \frac{c}{\sqrt{n}} \)
- Bob listens w/ prob \( \frac{c}{\sqrt{n}} \)

Assume Adv. blocks w/ prob 1/2.

Then prob. a given slot is one where Alice sends and there is no jam is \( \frac{c}{2\sqrt{n}} \)

\[
\text{Prob(Bob fails to get message)} \sim \left(1 - \frac{c}{2\sqrt{n}}\right)^{c\sqrt{n}} \\
\leq e^{-c^2/2}
\]

Bob can send an ACK in same way
After each failed round, \( n \) can double in size
Result

- If adversary can only jam, the previous algorithm creates a secure (but randomized) communication channel between Alice and Bob

- Cost of sending and ACK of a message in this channel is: $\tilde{O}(B^{1/2} + 1)$

- Note: Bob must know when to listen to determine if Alice is writing (seems unavoidable)
Idea: create such a channel between the writer for the register and each “server”, and each reader of the register and each server.

Using standard tricks (e.g. [ABD ’89]) can implement a SWMR shared Register that tolerates fail-stop faults on the servers.
Problems

- Problem 1: What if the adversary can spoof in addition to jamming?
- Problem 2: What if there is only one channel for everyone?
Problem 1: Spoofing

- Now imagine Adv. can spoof Bob (but not Alice)
- Idea: Alice stops sending only if she hears a silent slot
- Problem: Adv. can keep sending fake requests and thereby bankrupt Alice
- Idea: Impose a larger cost to trigger a resend, to mitigate increased cost to Alice
Our Algorithm: Round $i$

Send Phase: For $2^ci$ slots do

- Alice sends with prob. $2/2^i$
- Bob listens with prob. $2/2^{(c-1)i}$

Req Phase: For $2^i$ slots do

- If Bob has not received $m$, Bob sends req message
- Alice listens with prob. $4/2^i$

If Alice listened in Req phase and detected no req message or collision then algorithm terminates.
Our Algorithm: Round $i$

Send Phase: For $2^{ci}$ slots do
  - Alice sends with prob. $2/2^i$
  - Bob listens with prob. $2/2^{(c−1)i}$

Req Phase: For $2^i$ slots do
  - If Bob has not received $m$, Bob sends req message
  - Alice listens with prob. $4/2^i$

If Alice listened in Req phase and detected no req message or collision then algorithm terminates

Analysis shows it’s best to set $c = \varphi$
Theorem: Our algorithm has the following properties:

- The expected cost to Alice and Bob is $O(B^{\varphi^{-1}} + 1) = O(B^{0.62} + 1)$.
- Alice and Bob terminate within $O(B^\varphi)$ slots in expectation.

Result
A note on $c$

- $c > 1$ since otherwise Bob listens with prob $> 1$
- $c < 2$ since otherwise, adversary can cause Alice to spend more than itself by causing repeated Req failures via jamming the entire Req phase.
Failure

- There are two ways that a stage can fail
  - **Send Failure**: Bob did not get the message in Send phase
  - **Req Failure**: Alice never listened to a silent slot in the Req phase. Note: this type of attack only makes sense after Bob terminates.
Jamming

- We call a round *send-jamming* if the adversary jams at least half the slots in the send phase.
- We call a round *req-jamming* if the adversary jams at least half the slots in the req phase.
Lemma 1

Lemma 1. Consider a round that is not send-jamming. The probability that Bob does not receive the message from Alice is less than $e^{-2}$. 
Lemma 1 proof

- Let $s = 2^{ci}$ be the number of slots in the Send Phase.
- Let $p_A$ be the probability that Player $A$ sends in a particular slot. Let $p_B$ be the probability that Player $B$ listens in a particular slot.
- Let $X_j = 1$ if the message is not delivered from Player $A$ to Player $B$ in the $j^{th}$ slot.
Lemma 1 Pf (Cont’d)

- Let \( q_i = 1 \) if adversary does not jam slot \( i \); 0 otherwise.

- Then \( Pr[X_i = 1 \mid X_1 X_2 \cdots X_{i-1} = 1] = 1 - p_A p_B q_i \)

- Thus \( Pr[m \text{ not delivered }] = \prod_{j=1}^{s}(1 - p_A p_B q_j) \leq e^{-p_A p_B \sum_{j=1}^{s} q_j} < e^{-2} \)

- Since \( p_A p_B \sum_{j=1}^{s} q_j > (2/2^i)(2/2^{(c-1)i})(s/2) > 2 \)
Lemma 2. Assume Bob has received $m$ by round $i$ and that round $i$ is non req-jamming. Then the probability that Alice retransmits $m$ in round $i + 1$ is less than $e^{-2}$. 
Proof

• Let $s = 2^i$ be number of slots in the req phase and $p = 4/2^i$ be the probability that Alice listens in a slot.

• For slot $j$, let $X_j = 1$ if Alice does not terminate and 0 otherwise.

• Let $q_j = 1$ if the adversary does not jam slot $j$ and 0 otherwise.

• Then $Pr[X_j = 1] = (1 - pq_j)$.

• Therefore, $Pr[X_1X_2 \cdots X_s = 1] \leq e^{-p\sum_{j=1}^{s} q_j} < e^{-2}.$
Lemma 3. Assume that Player B is correct and there are no send-jamming or ack-jamming rounds. Then, the expected cost of each player is $O(1)$. 
Proof

• Using Lemma 1, the expected cost to Alice is at most

\[
\sum_{i=2}^{\infty} e^{-2(i-2)} \cdot (2 \cdot 2^{(c-1)i} + 4) \leq \sum_{i=2}^{\infty} (e^{5-i} + 4 \cdot e^{-2(i-1)}) = O(1).
\]

• Using Lemma 2, expected cost to Bob is at most

\[
\sum_{i=2}^{\infty} e^{-2(i-2)} \cdot (2^{i+1} + 2^i) \leq \sum_{i=2}^{\infty} (e^{5-i} + e^{4-i}) = O(1).
\]
Lemma 4. Assume there is at least one send-jamming round. Then the expected cost to Alice is $O(B^{(c-1)/c} + B^{c-1})$ and the expected cost to Bob is $O(B^{1/c})$. 
Theorem: Our algorithm has the following properties:

- The expected cost to Alice and Bob is $O(B^{\varphi^{-1}} + 1) = O(B^{0.62} + 1)$.
- Alice and Bob terminate within $O(B^{\varphi})$ slots in expectation.
Proof (Adv. Cost)

• Let \( i \) be the last send send-jamming round; \( j \geq i \) be the last req-jamming round

• If no such req-jamming round exists \( j = 0 \)

• Then cost to the adversary, \( B \), is \( \Omega(2^{ci} + 2^j) \)
Proof (Alice’s cost)

- Using Lemma 1, the expected cost to Alice, prior to $m$ being delivered is

$$O(2^{(c-1)i}) + \sum_{k=1}^{\infty} e^{-2(k-1)}(2 \cdot 2^{(c-1)(i+k)} + 4) = O(2^{(c-1)i})$$

since $c < 2$

- Now, using Lemma 2, the expected cost to Alice after delivery is

$$O(2^{(c-1)j}) + \sum_{k=1}^{\infty} e^{-2(k-1)}(2 \cdot 2^{(c-1)(j+k)} + 4) = O(2^{(c-1)j})$$

since $c < 2$

- Therefore, the total expected cost to Alice is $O(2^{(c-1)i} + 2^{(c-1)j})$

- Since $B = \Omega(2^c + 2^j)$, this cost as a function of $B$ is $O(B^{(c-1)/c} + B^{(c-1)})$
Proof (Bob’s Cost)

• Using Lemma 1, Bob’s expected cost prior to receiving $m$ is $O(2^i) + \sum_{i=1}^{\infty} e^{-2(k-1)}(2 \cdot 2^{i+k} + 2^{i+k}) = O(2^i)$

• Thus, the expected cost for Bob as a function of $B$ is $O(B^{1/c})$
Proof

• By Lemma 4, the expected cost of Alice is $O(B^{(c-1)/c} + B^{(c-1)})$ and the expected cost of Bob is $O(B^{1/c})$

• Therefore, the exponents that control the cost ratios are $(c - 1)/c, c - 1, 1/c$

• Since $1 < c < 2$, we know that $1/c > (c - 1)/c$. Thus we solve for $c$ in $c - 1 = 1/c$

• This gives $c = (1 + \sqrt{5})/2$
What if the adversary tries to spoof Bob?

- The theorem still holds (analysis omitted)

- However Alice’s messages must be authenticated with e.g. digital signature.

- Seems inherent requirement
Problem 2

- What if there is only one channel?
- Want Alice to be able to communicate to multiple Bob’s simultaneously
Model

- In a given time slot, when a single player sends a message on the channel, all listening players hear the message.

- In a given time slot, when more than one players sends a message, all listening players hear a jam.
**Theorem:** There exists an algorithm for one sender and $n$ receivers that ensures the message is delivered to all receivers and has the following costs:

- The sender’s expected cost is $O(B^{\varphi^{-1}} \log n + \log^\varphi n)$
- The expected cost to any receiver is $O(B^{\varphi^{-1}} + \log n)$
- The worst case number of slots used is $O(((B + \log^{\varphi^{-1}})^{\varphi+1})$
New Conjecture

- There exists an algorithm for one sender and n receivers such that
- The message is received and acknowledged whp
- All players terminate with expected cost:
  \[ O\left(\frac{B}{n} + 1 \right) \log^2(B + n) \]
Candidate Algorithm

For $i = 0$ to $\infty$
    Repeat $i$ times:
        Set $S \leftarrow 4$
        Repeat for $2^i$ slots
            Send the message or noise if the message is not known with probability $S/2^i$.
            Listen with probability $iS/2^i$
        CASE:
        a) if a message is heard, then
            with probability $1/2$: terminate; else $S \leftarrow 2S$ for the next repetition
        b) if mostly empty slots are heard $S \leftarrow 2S$ for the next repetition
        c) if jamming or collision is heard $S \leftarrow S$ for the next repetition
Problems

- Assume we can enable broadcast and acknowledgement in $\tilde{O}(B/n + 1)$

- Want: no cost to anyone when the register doesn’t change. Seems hard.

- Q: Can we allow that there is only a “small” cost in rounds where the value of the register doesn’t change?
Adapt to DOS Attacks

- Consider a p2p system with a shared memory
- Are send, listen and jam costs similar?
- Estimates based on Amazon.com’s EC2
  - $.17 to send 1GB
  - $.9 to “listen” to 1GB (i.e. process)
What is a slot?

- a slice of time
- a unique communication channel
  - physical channel: wireless nodes broadcast at different frequencies
  - virtual channel: messages indexed by a channel id
Problems

- Connections to known shared memory results? Are there lower/upper bounds from the message passing model that apply to this model with jamming?

- Can we bound the competitive ratio achievable even if our algorithm is randomized?
Questions