

Efficient and Accurate Protocols for Distributed Delaunay Triangulation under Churn

Dong-Young Lee and Simon S. Lam

Department of Computer Sciences

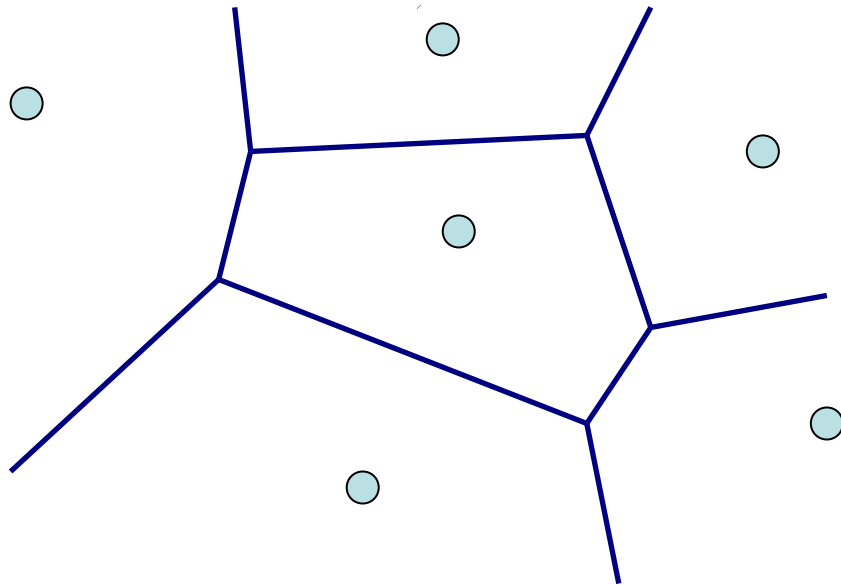
The University of Texas at Austin, USA



WHAT STARTS HERE CHANGES THE WORLD

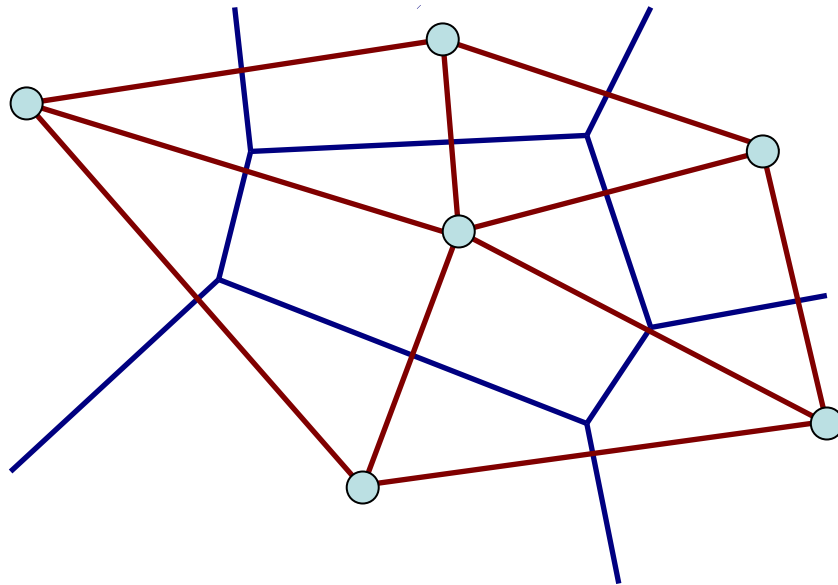
THE UNIVERSITY OF TEXAS AT AUSTIN

Voronoi Diagram



- Given a set of nodes S in a d -dimensional **Euclidean space**.
- The **Voronoi diagram** of S is a partitioning of the space into cells such that a node $u \in S$ is the closest node to all points within its Voronoi cell $VC_S(u)$.

Delaunay Triangulation



- The **Delaunay triangulation** of S is a graph on S where u and v have an edge between them if $VC_S(u)$ and $VC_S(v)$ share a facet.
 - denoted by $DT(S)$.
- DT and VD are the dual of each other.
- u and v are **neighbors** of each other.

Networking Applications

- Greedy routing
 - Greedy routing always succeeds on DT [BM04].
- Finding a closest existing node
- Clustering
- Broadcast
 - Hypercast [LNS02].
- Geocast
 - Distributed virtual reality, on-line games,

Distributed Delaunay Triangulation

- Each node u keeps a set N_u of its neighbor nodes.
- A distributed DT for S is **correct** when, for every node $u \in S$, N_u is the same as the set of u 's neighbor nodes on the **global** $DT(S)$.

ACE Protocol Suite

- Protocols to construct and maintain a distributed DT in a **dynamic** set of nodes in a ***d*-dimensional** space.
 - Join, leave, failure, and maintenance protocols.
- Assume that nodes may join, leave, or fail at any time.
 - System **churn**.

ACE Protocol Suite

- Correctness
 - Proved to be correct for a **single** join, leave, and failure.
- Accuracy
 - Define an accuracy metric (100% accuracy = correct DT).
 - **Maintain high accuracy** during system churn and **recover 100% accuracy** after system churn stops.
 - Maintenance protocol.
- Efficiency
 - Use fewer messages than existing protocols.

Candidate-Set Approach

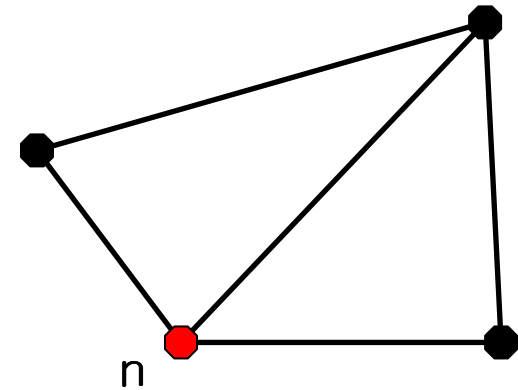
- Each node $u \in S$ knows a set C_u of nearby nodes (**candidate set**).
- u determines a set N_u of its neighbor nodes from $DT(C_u)$.

Correctness Condition

- How much **local** information (C_u) does u need to correctly determine its neighbors on the **global** $DT(S)$?
- Theorem 1: Let S be a set of nodes and for each node $u \in S$, u knows C_u , such that $u \in C_u \subset S$. The distributed DT of S is correct **if and only if**, for every $u \in S$, C_u includes all neighbor nodes of u on $DT(S)$.

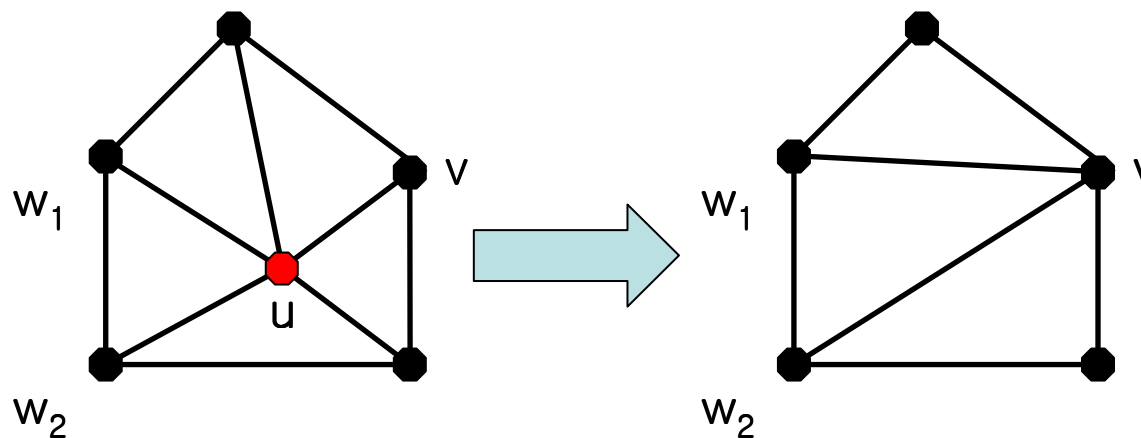
ACE Join Protocol

- A joining node n should discover all its neighbors on the global DT.
- In the old protocol, n recursively queries **all** of its neighbors to find a new neighbor.
- In the ACE protocol, n queries **one neighbor for each simplex** in $DT(C_n)$.
 - Queries (and replies) for several simplexes can be **combined**.



ACE Leave Protocol

- A leaving node u should notify its neighbors that it is leaving.
- A neighbor v of u may get a new neighbor w , which previously was not v 's neighbor.
 - But w is a neighbor of $u \rightarrow u$ notifies v of w .



ACE Failure Protocol

- Old protocols
 - **Reactive** approach – the information to recover a failure is lost since it is in the failed node.
 - A node is redundantly probed by all of its neighbors.
- A **proactive** approach
 - A node u prepares a **contingency plan** for its failure.
 - u selects a monitor node and give the contingency plan to the monitor node.
 - **Only the monitor node probes u** to detect u 's failure.
 - In case a failure is detected, the contingency plan is executed.

ACE Maintenance Protocol

- ACE join, leave, and failure protocols are correct only for a single join, leave, and failure.
 - Enough when the system churn rate is low.
 - **Concurrent joins, leaves, and/or failures may result in an incorrect distributed DT.**
- A maintenance protocol is run periodically.
 - Find a new neighbor by querying neighbors.
 - Similar to ACE join protocol.
 - Detect leave or failure of a neighbor by message timeouts.
 - With **help of ACE failure protocol**, ACE maintenance protocol may be run less frequently.

Comparison of d -Dimensional DT Protocols

	Efficiency	Convergence to 100% accuracy after system churn
Simon et al.'s basic algorithms [SSB 05]	Medium	No
Simon et al.'s improved algorithms	High	No
Our old protocols [ICDCS 07]	Low	Yes
ACE protocols	Very high	Yes

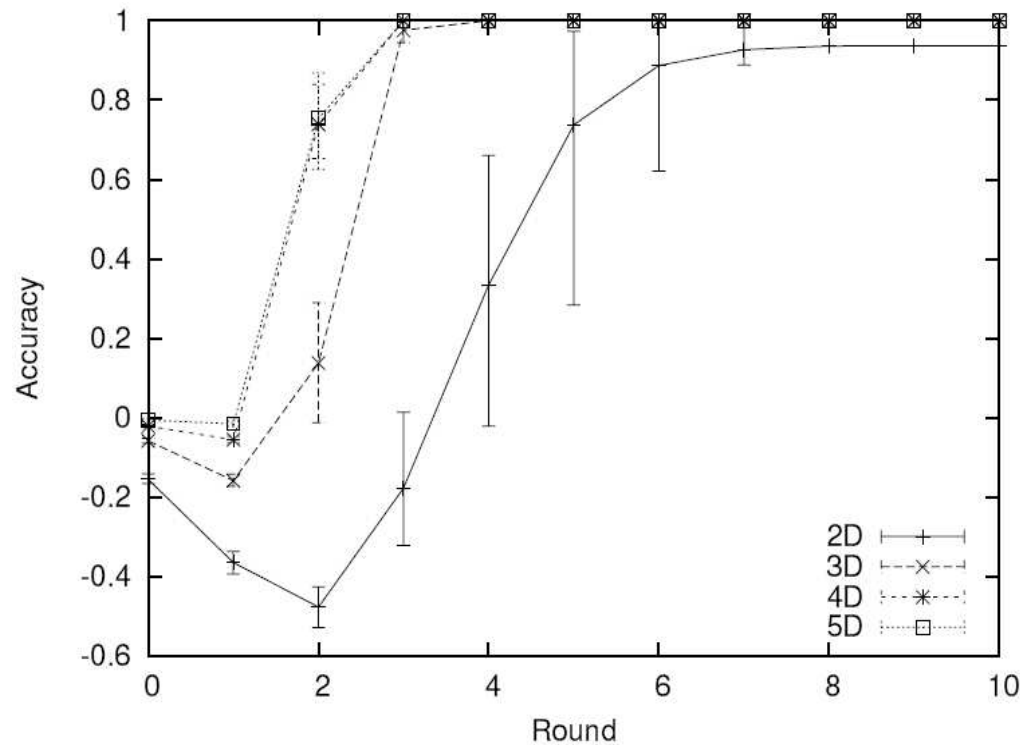
Accuracy Metric

$$\text{accuracy}(DDT_S) = \frac{|E_{correct}^D(DDT_S)| - |E_{wrong}^D(DDT_S)|}{2 \times |E(DT(S))|}$$

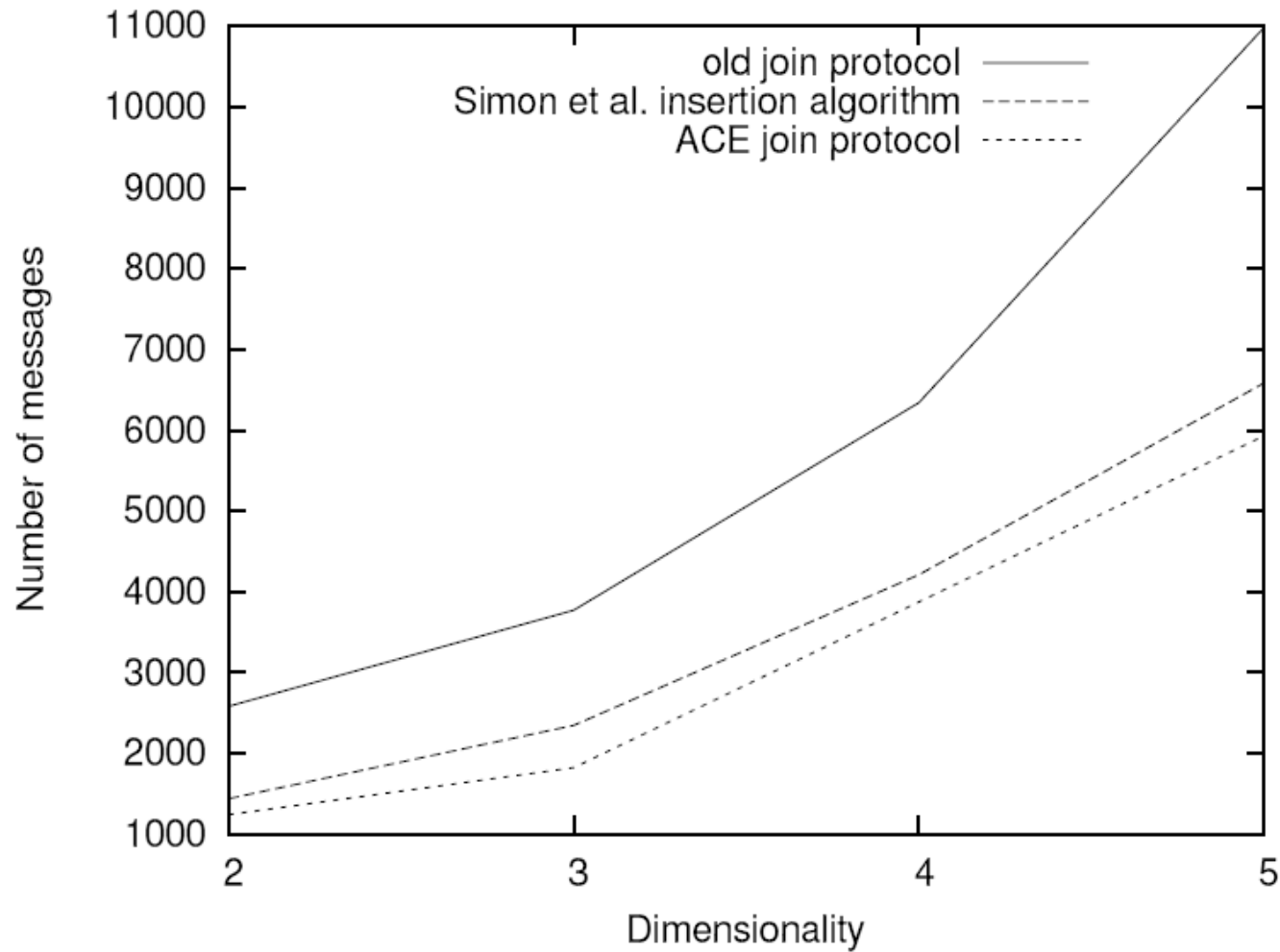
- The distributed DT is correct if and only if the accuracy value is 1.

Ring Scenario

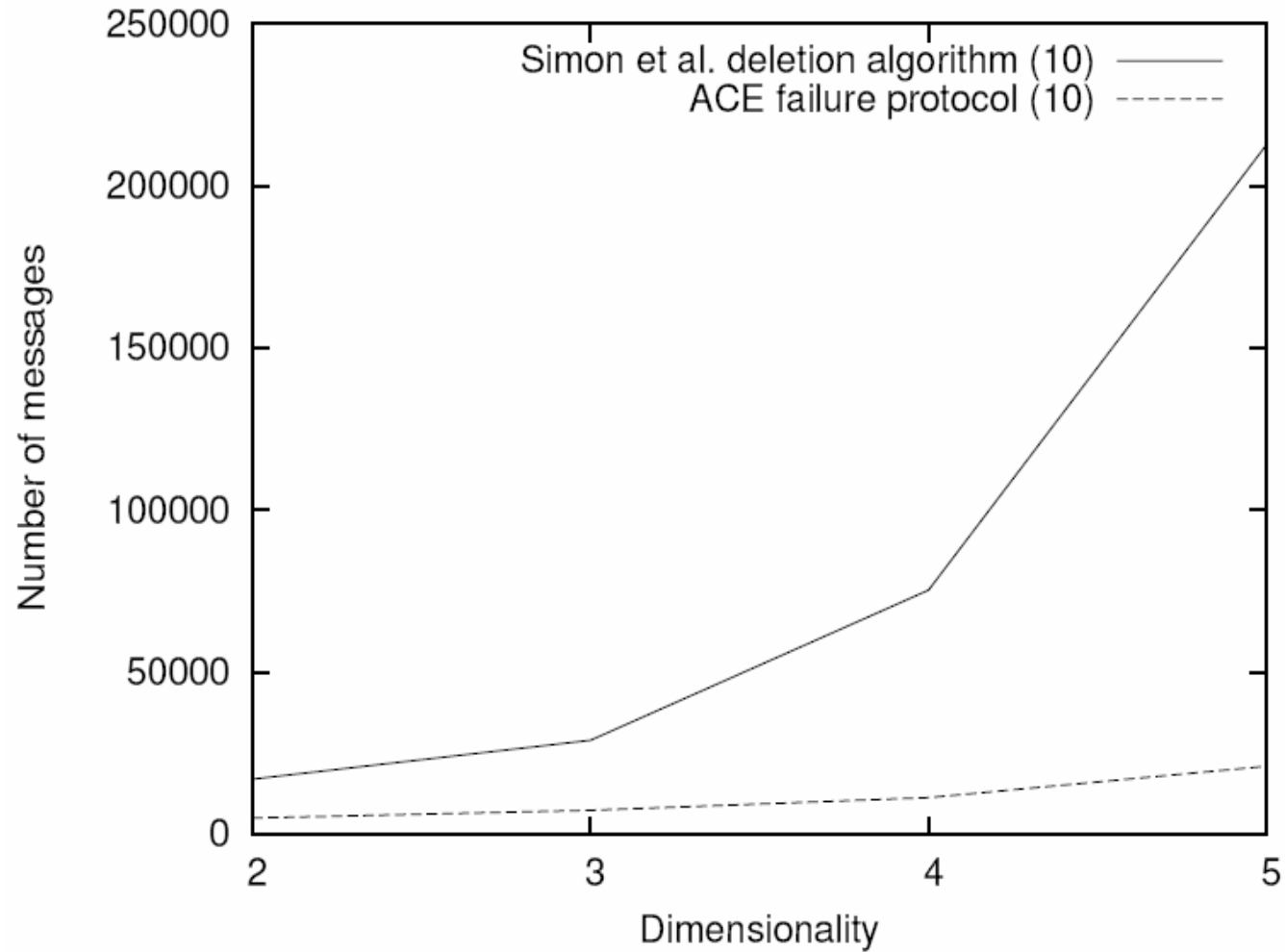
- Only the ACE maintenance protocol is run from an initial **unidirectional ring** configuration.



Cost of Join Protocols

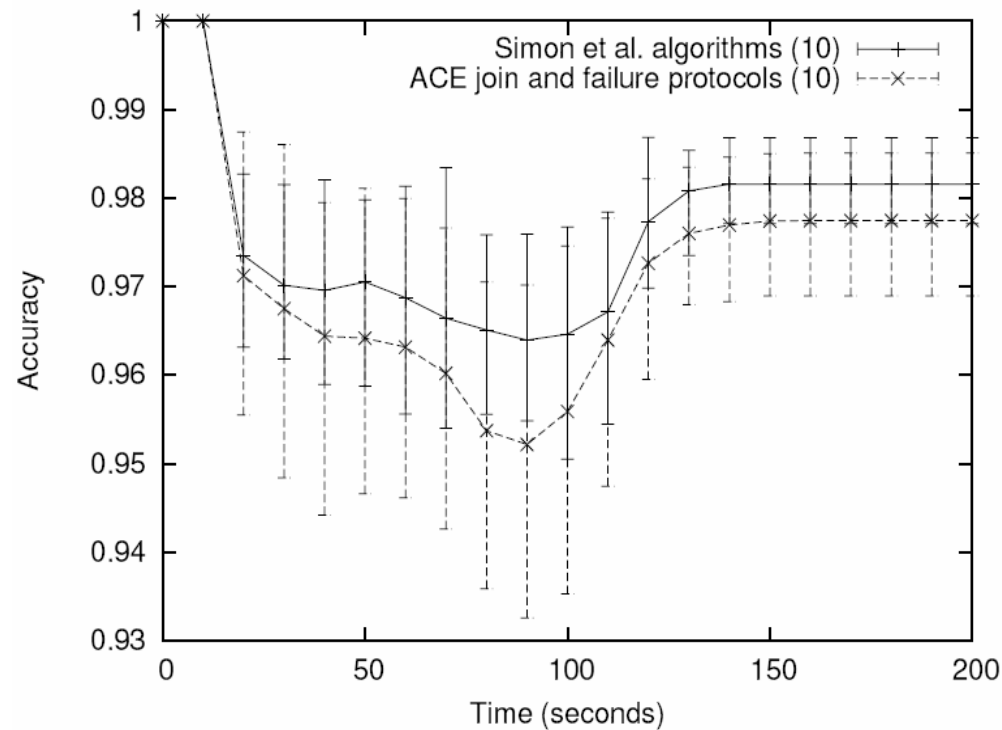


Cost of Failure Protocols



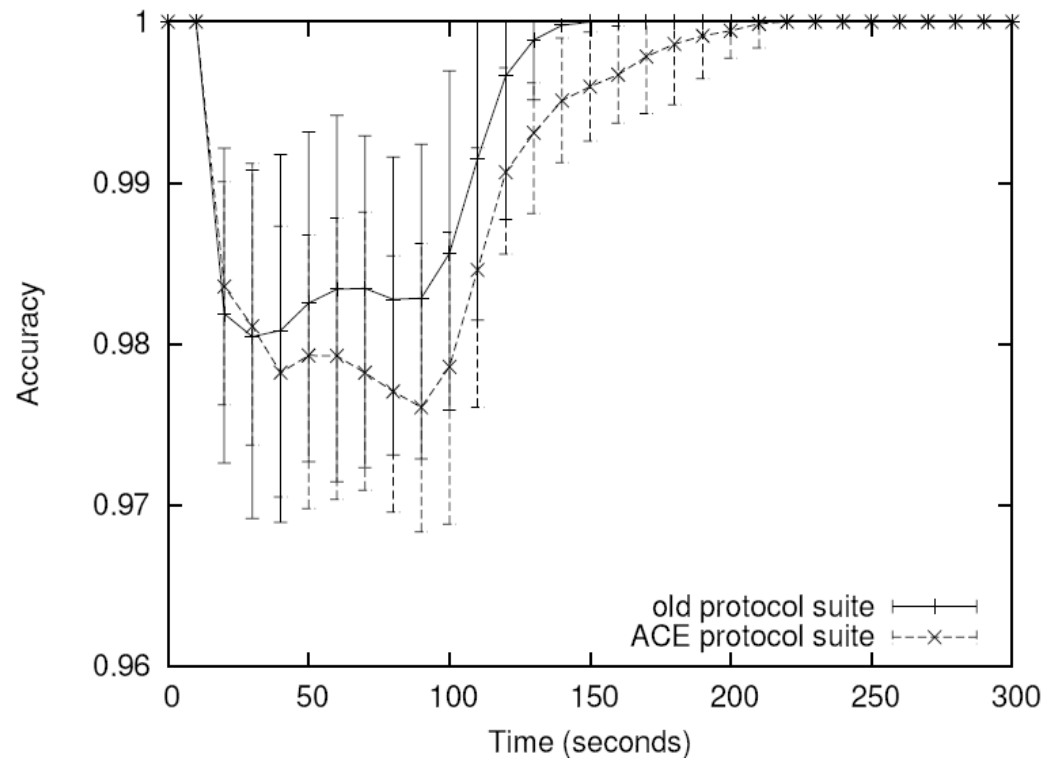
Accuracy without a Maintenance Protocol under Churn

- Protocols cannot converge to a correct distributed DT **without a maintenance protocol.**



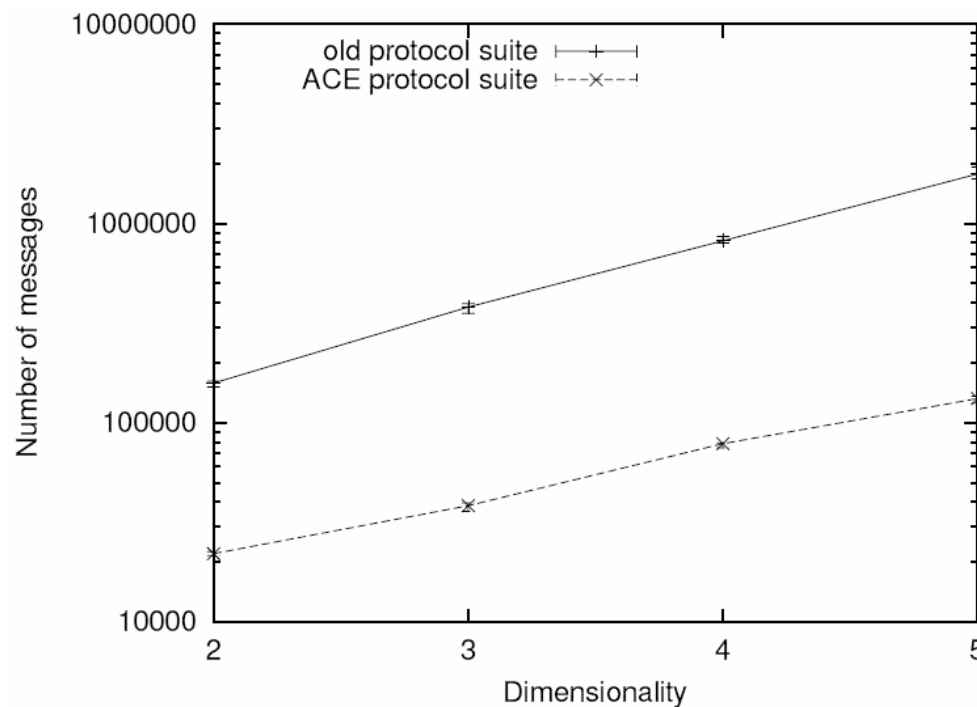
Accuracy of the old and ACE Protocols under Churn

- Our old and ACE protocols converge to 100% accuracy after system churn stops.



Cost of the old and ACE protocols

- ACE protocol suite is about **an order of magnitude more efficient** than our old protocol suite.



Conclusions

- Identify a necessary and sufficient condition for a distributed DT to be correct.
- Design and evaluate ACE protocol suite.
 - Join, leave, failure, and maintenance protocols.
 - Proved to be correct for a single join, leave, and failure.
 - Converge to a correct distributed DT after system churn stops.
 - About an order of magnitude more efficient than our old protocol suite.

Thank You

Questions?