Commit Protocols in Mobile Databases

Kyle Foss

Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

April 16, 2018
What’s a database and why should we care?

Databases allow us to:
What’s a database and why should we care?

Databases allow us to:

- Store data, organize data, modify data, and share data
Database types:
• Centralized database
• Distributed database
• Mobile database
Introduction: Continued

Database types:

• Centralized database
• Distributed database
• Mobile database
Introduction: Continued

Database types:

- Centralized database
Database types:

- Centralized database
- Distributed database
Introduction: Continued

Database types:

- Centralized database
- Distributed database
- Mobile database
Problems with distributed databases:

• Design is difficult
• Communication is difficult
• Data can change in unexpected ways
Introduction: Continued

Problems with distributed databases:

• Design is difficult
• Communication is difficult
• Data can change in unexpected ways
Introduction: Continued

Problems with distributed databases:

- Design is difficult
Problems with distributed databases:

- Design is difficult
- Communication is difficult
Introduction: Continued

Problems with distributed databases:

- Design is difficult
- Communication is difficult
- Data can change in unexpected ways
Outline

1. Background
2. Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC)
3. Partition-tolerant atomic commit protocol (ParTAC)
4. Generalized mobile transaction commit (GMTC)
5. Conclusion
Outline

1 Background
   Architecture of Mobile Database System
   Transactions and Transaction Processing
   Perturbations
   Commit Protocols

2 Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC)

3 Partition-tolerant atomic commit protocol (ParTAC)

4 Generalized mobile transaction commit (GMTC)

5 Conclusion
Background: Architecture of Mobile Databases

Figure 1 [3]
Transactions and Transaction Processing

Example Bank Transfer - transferring $50 from savings to checking
Transactions and Transaction Processing

Example Bank Transfer - transferring $50 from savings to checking

Before:

<table>
<thead>
<tr>
<th>Savings</th>
<th>Checking</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100</td>
<td>$50</td>
</tr>
</tbody>
</table>

Transaction starts:
Transfer $50 from Savings to Checking

<table>
<thead>
<tr>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100</td>
</tr>
</tbody>
</table>
Transactions and Transaction Processing

Example Bank Transfer - transferring $50 from savings to checking

Before:

<table>
<thead>
<tr>
<th>Savings</th>
<th>Checking</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100</td>
<td>$50</td>
</tr>
</tbody>
</table>

Transaction starts:
Transfer $50 from Savings to Checking

Transaction goes wrong during execution:

<table>
<thead>
<tr>
<th>Savings</th>
<th>Checking</th>
</tr>
</thead>
<tbody>
<tr>
<td>$100</td>
<td>$50</td>
</tr>
</tbody>
</table>
Transactions and Transaction Processing

Example Bank Transfer - transferring $50 from savings to checking

Before:

Savings: $100
Checking: $50

After:

Savings: $50
Checking: $50

Transaction starts: Transfer $50 from Savings to Checking

Something goes wrong during transaction

Transaction ends in an inconsistent state
Perturbations

Commit Perturbations

Environmental Constraints

Heterogeneity

Unstable storage

Energy

Node

Link

Figure 2 [3]
Perturbations

Commit Perturbations

Environmental Constraints

Heterogeneity

Unstable storage

Energy

Node

Link

Figure 2 [3]
Perturbations

Commit Perturbations

Environmental Constraints

Heterogeneity

Unstable storage

Energy

Node

Link

Figure 2 [3]
Perturbations

Commit Perturbations

Environmental Constraints

Heterogeneity

Unstable storage

Energy

Node

Link

Figure 2 [3]
Figure 2 [3]
Figure 2 [3]
Perturbations

Commit Perturbations

Environmental Constraints

Heterogeneity

Unstable storage

Energy

Node

Link

Figure 2 [3]
Figure 2 [3]
Perturbations

Commit Perturbations

Failures

Node

MN

Transient

Predictable

FN

Permanent

Communication

Network disconnection

Transient

Unpredictable

Message loss

Figure 2 [3]
Perturbations

Commit Perturbations

Failures

Node

MN

Transient

Predictable

FN

Permanent

Communication

Network disconnection

Transient

Unpredictable

Message loss
Figure 2 [3]

Perturbations

Commit Perturbations

Failures

Node

MN

Transient

Predictable

FN

Permanent

Communication

Network disconnection

Transient

Message loss

Unpredictable
Perturbations

Commit Perturbations

Failures

Node

MN

FN

Communication

Network disconnection

Message loss

Transient

Predictable

Permanent

Transplant

Unpredictable

Figure 2 [3]
Perturbations

Commit Perturbations

Failures

Node

MN

Transient

Predictable

FN

Permanent

Communication

Network disconnection

Transient

Unpredictable

Message loss
Figure 2 [3]
Figure 2 [3]
Perturbations

Commit Perturbations

Failures

Node

- MN
  - Transient
  - Predictable

- FN
  - Permanent

Communication

- Network disconnection
- Message loss
  - Transient
  - Unpredictable

Figure 2 [3]
Perturbations

Commit Perturbations

Failures

Node

MN

Transient

Predictable

FN

Permanent

Communication

Network disconnection

Transient

Message loss

Unpredictable

Figure 2 [3]
Perturbations

Commit Perturbations

Failures

Node

MN

Transient

Predictable

FN

Permanent

Communication

Network disconnection

Transient

Message loss

Unpredictable

Figure 2 [3]
Perturbations

Commit Perturbations

Failures

Node

MN

Transient

Predictable

FN

Permanent

Communication

Network disconnection

Transient

Unpredictable

Message loss

Figure 2 [3]
Commit Protocols

Commit Protocols:

• Used in transactions
• Ensure data consistency
• Ensure all participants agree (Atomicity)
• Data can return to a previous state
Commit Protocols

Commit Protocols:

- Used in transactions
Commit Protocols

Commit Protocols:

- Used in transactions
- Ensure data consistency
Commit Protocols

Commit Protocols:

- Used in transactions
- Ensure data consistency
- Ensure all participants agree (Atomicity)
Commit Protocols

Commit Protocols:

- Used in transactions
- Ensure data consistency
- Ensure all participants agree (Atomicity)
- Data can return to a previous state
Outline

1 Background

2 Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC)
   Protocol
   Results

3 Partition-tolerant atomic commit protocol (ParTAC)

4 Generalized mobile transaction commit (GMTC)

5 Conclusion
Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC) is aimed at:

• Improving transactions in infrastructure environments
• Tolerating environmental constraints (heterogeneity, unstable storage, and battery life)
• Tolerating network disconnections (predictable and unpredictable disconnections)
• Tolerating message loss (network congestion)
FT-PPTC is aimed at:

- Improving transactions in infrastructure environments
- Tolerating environmental constraints (heterogeneity, unstable storage, and battery life)
- Tolerating network disconnections (predictable and unpredictable disconnections)
- Tolerating message loss (network congestion)
Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC) Protocol

FT-PPTC is aimed at:

- Improving transactions in infrastructure environments
- Tolerating environmental constraints (heterogeneity, unstable storage, and battery life)
- Tolerating network disconnections (predictable and unpredictable disconnections)
- Tolerating message loss (network congestion)
Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC)

FT-PPTC is aimed at:

- Improving transactions in infrastructure environments
- Tolerating environmental constraints (heterogeneity, unstable storage, and battery life)
Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC) is aimed at:

- Improving transactions in infrastructure environments
- Tolerating environmental constraints (heterogeneity, unstable storage, and battery life)
- Tolerating network disconnections (predictable and unpredictable disconnections)
Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC) protocol is aimed at:

- Improving transactions in infrastructure environments
- Tolerating environmental constraints (heterogeneity, unstable storage, and battery life)
- Tolerating network disconnections (predicable and unpredictable disconnections)
- Tolerating message loss (network congestion)
FT-PPTC: Continued
FT-PPTC: Continued

Wireless Part

P-MN_3

P-MN_1

P-MN_2

Wired Part


FT-PPTC: Continued
FT-PPTC: Continued

Pre-Phase Commit

Wireless Part

P-MN_3

P-MN_1

P-MN_2

Wired Part

MN-Ag_1

MN-Ag_2

MN-Ag_3

P-FN_1

P-FN_2
FT-PPTC: Continued

Pre-Phase Commit

Wireless Part
- P-MN_3
- P-MN_1
- P-MN_2

Initiator

Start Transaction

Wired Part
- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
FT-PPTC: Continued

Pre-Phase Commit

Wireless Part
- P-MN_3
- P-MN_1
- P-MN_2

Initiator

CO Selection

Wired Part
- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
FT-PPTC: Continued

Pre-Phase Commit

Wireless Part
- P-MN_3
- P-MN_1
- P-MN_2

Initiator

Wired Part
- CO
- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
FT-PPTC: Continued

Pre-Phase Commit

Wireless Part

P-MN_1
Initiator

P-MN_3

P-MN_2

Wired Part

MN-Ag_1

MN-Ag_2

MN-Ag_3

CO

P-FN_1

P-FN_2
FT-PPTC: Continued

Pre-Phase Commit

Wireless Part

- P-MN_3
- Initiator

Wired Part

- CO
- MN-Ag_1
- MN-Ag_2
- P-FN_1
- P-MN_2
- P-FN_2
- MN-Ag_3
FT-PPTC: Continued

Pre-Phase Commit

Wireless Part

P-MN_3

P-MN_1
Initiator

P-MN_2

Wired Part

CO

MN-Ag_1

MN-Ag_3

MN-Ag_2

P-FN_1

P-FN_2
FT-PPTC: Continued

Core Phase

Wireless Part

P-MN_3

P-MN_1

P-MN_2

Initiator

Wired Part

CO

MN-Ag_1

MN-Ag_2

MN-Ag_3

P-FN_1

P-FN_2
FT-PPTC: Continued

Core Phase

Wireless Part

P-MN_3

P-MN_1

Initiator

P-MN_2

Start two-phase commit

Wired Part

CO

MN-Ag_1

MN-Ag_2

MN-Ag_3

P-FN_1

P-FN_2
FT-PPTC: Continued

Core Phase

Wireless Part

P-MN_3
P-MN_1
P-MN_2

Initiator

Wired Part

CO
MN-Ag_1
MN-Ag_2
MN-Ag_3

P-FN_1
P-FN_2
FT-PPTC: Continued

Core Phase

Wireless Part
- P-MN_3
- P-MN_1
- P-MN_2

Initiator

Wired Part
- CO
- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
Results

Simulation Settings
## Simulation Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Part-FNs</td>
<td>4</td>
</tr>
<tr>
<td>#MNs</td>
<td>1-25</td>
</tr>
<tr>
<td>Execution time one fragments (MN)</td>
<td>5 ms</td>
</tr>
<tr>
<td>Execution time one fragments (FN)</td>
<td>2 ms</td>
</tr>
<tr>
<td>Transmission delay over wireless link</td>
<td>10 ms</td>
</tr>
<tr>
<td>Transmission delay over wired link</td>
<td>5 ms</td>
</tr>
</tbody>
</table>
FT-PPTC: Results

Conclusion:

• Resource blocking time is reduced
• The number of MNs don’t affect the resource blocking time
• Transactions were processed faster.
FT-PPTC: Results

Conclusion:

- Resource blocking time is reduced
FT-PPTC: Results

Conclusion:

- Resource blocking time is reduced
- The number of MNs don’t affect the resource blocking time
FT-PPTC: Results

Conclusion:

- Resource blocking time is reduced
- The number of MNs don’t affect the resource blocking time
- Transactions were processed faster.
Outline

1. Background

2. Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC)

3. Partition-tolerant atomic commit protocol (ParTAC)
   - Protocol
   - Results

4. Generalized mobile transaction commit (GMTC)

5. Conclusion
Partition-tolerant atomic commit protocol (ParTAC)

ParTAC is aimed at:
• Improving transactions in mobile ad-hoc scenarios
• Tolerating message loss (network congestion)
• Tolerating transient failures (network disconnections)
• Tolerating network partitioning (groups of nodes separating from each other)
Partition-tolerant atomic commit protocol (ParTAC)

ParTAC is aimed at:

• Improving transactions in mobile ad-hoc scenarios
• Tolerating message loss (network congestion)
• Tolerating transient failures (network disconnections)
• Tolerating network partitioning (groups of nodes separating from each other)
Partition-tolerant atomic commit protocol (ParTAC)

ParTAC is aimed at:

- Improving transactions in mobile ad-hoc scenarios

Tolerating message loss (network congestion)

Tolerating transient failures (network disconnections)

Tolerating network partitioning (groups of nodes separating from each other)
Partition-tolerant atomic commit protocol (ParTAC)

ParTAC is aimed at:

- Improving transactions in mobile ad-hoc scenarios
- Tolerating message loss (network congestion)
Partition-tolerant atomic commit protocol (ParTAC)

ParTAC is aimed at:

- Improving transactions in mobile ad-hoc scenarios
- Tolerating message loss (network congestion)
- Tolerating transient failures (network disconnections)
ParTAC is aimed at:

- Improving transactions in mobile ad-hoc scenarios
- Tolerating message loss (network congestion)
- Tolerating transient failures (network disconnections)
- Tolerating network partitioning (groups of nodes separating from each other)
ParTAC: Continued
ParTAC: Continued

Wireless Part

P-MN_2
P-MN_1
P-MN_3
P-MN_4
P-MN_5
ParTAC: Continued

Wireless Part

P-MN_2

Initiator

P-MN_1

P-MN_3

P-MN_4

P-MN_5
ParTAC: Continued

Wireless Part

Initiate transaction & select COs
ParTAC: Continued

Wireless Part

CO
P-MN_2

Initiator
P-MN_1

P-MN_3

CO
P-MN_4

P-MN_5
ParTAC: Continued

Diagram of network connections involving nodes labeled CO, P-MN_2, P-MN_1, Initiator, P-MN_3, and P-MN_4. The diagram illustrates a wireless part of the network with points of connection and communication.
ParTAC: Continued

Wireless Part

CO
P-MN_2
P-MN_1
Initiator
P-MN_4
P-MN_3
P-MN_5

CO
P-MN_2
P-MN_1
Initiator
P-MN_4
P-MN_3
P-MN_5
ParTAC: Continued

Wireless Part

CO

P-MN_2

Initiator

P-MN_1

P-MN_3

P-MN_4

P-MN_5
ParTAC: Continued
ParTAC: Results

Simulation Settings
## ParTAC: Results

### Simulation Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical area</td>
<td>$2km \times 2km$</td>
</tr>
<tr>
<td>Communication range</td>
<td>$250m$</td>
</tr>
<tr>
<td>Mobility models</td>
<td>Random Waypoint (RWP), RPGM</td>
</tr>
<tr>
<td>Node speed</td>
<td>LOW uniform in $[0.5, 1.5] \text{ m/s}$</td>
</tr>
<tr>
<td></td>
<td>MEDIUM uniform in $[3, 10] \text{ m/s}$</td>
</tr>
<tr>
<td></td>
<td>HIGH uniform in $[10, 25] \text{ m/s}$</td>
</tr>
<tr>
<td>#Nodes</td>
<td>$\in [20, 200]$ for Random Waypoint</td>
</tr>
<tr>
<td></td>
<td>$\in [60, 380]$ for RPGM</td>
</tr>
<tr>
<td>#COs</td>
<td>$\in {2, 3, 4, 7, 10}$</td>
</tr>
<tr>
<td>#P-MNs</td>
<td>10</td>
</tr>
<tr>
<td>lifetime</td>
<td>$\in {60, 120, 300, 900} \text{ s}$</td>
</tr>
</tbody>
</table>

Table 2 [2]
ParTAC: Results

Conclusion:
ParTAC: Results

Conclusion:

- Commit rate increases in both mobility models
ParTAC: Results

Conclusion:

- Commit rate increases in both mobility models
- Commit rate is lowered when the amount of partitioning increases
ParTAC: Results

Conclusion:

- Commit rate increases in both mobility models
- Commit rate is lowered when the amount of partitioning increases
- Decision time decreases as number of MNs increase in some cases
Outline

1. Background

2. Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC)

3. Partition-tolerant atomic commit protocol (ParTAC)

4. Generalized mobile transaction commit (GMTC)
   - Protocol
   - GMTC: Results

5. Conclusion
Generalized mobile transaction commit (GMTC) Protocol

**GMTC**

**Goal of GMTC:**
- Improve transactions in a hybrid environment

**Why ParTAC and FT-PPTC fail in this environment:**
- ParTAC very inefficient in environments with infrastructure
- Transactions may not involve any infrastructure so FT-PPTC won’t work

**Improving transactions is accomplished by:**
- Combining ParTAC and FT-PPTC
- Reducing resource blocking time
- Reduce transaction decision time

Kyle Foss (U of Minn, Morris)
Commit Protocols in Mobile Databases
April 16, 2018 64 / 91
GMTC

Goal of GMTC:

- Improve transactions in a hybrid environment
GMTC

Goal of GMTC:
- Improve transactions in a hybrid environment

Why ParTAC and FT-PPTC fail in this environment:
- ParTAC very inefficient in environments with infrastructure
- Transactions may not involve any infrastructure so FT-PPTC won’t work
GMTC

Goal of GMTC:
- Improve transactions in a hybrid environment

Why ParTAC and FT-PPTC fail in this environment:
- ParTAC very inefficient in environments with infrastructure
- Transactions may not involve any infrastructure so FT-PPTC won’t work

Improving transactions is accomplished by:
- Combining ParTAC and FT-PPTC
- Reducing resource blocking time
- Reduce transaction decision time
GMTC: Continued
GMTC: Continued
GMTC: Continued

Wireless Part

P-MN_2
P-MN_3
P-MN_1
P-MN_4
P-MN_5

Wired Part

MN-Ag_1
MN-Ag_2
MN-Ag_2
GMTC: Continued

Wireless Part

- P-MN_2
- P-MN_3
- P-MN_1
- P-MN_4
- P-MN_5

Wired Part

- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
GMTC: Continued

Wireless Part

- P-MN_2
- P-MN_3
- P-MN_1
- P-MN_5
- P-MN_4

Wired Part

- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
GMTC: Continued

Wireless Part
- P-MN_2
- P-MN_3
- P-MN_1
- P-MN_4
- P-MN_5

Wired Part
- MN-Ag_1
- MN-Ag_3
- MN-Ag_2
- P-FN_1
- P-FN_2
GMTC: Continued

Pre-Commit Phase

Wireless Part

- P-MN_2
- P-MN_3
- P-MN_1
- P-MN_5
- P-MN_4

Wired Part

- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
GMTC: Continued

Diagram showing the Generalized mobile transaction commit (GMTC) Protocol stages for wireless and wired parts.
GMTC: Continued

Pre-Commit Phase

Wireless Part
- CO
- P-MN_2
- Initiator
- P-MN_3
- P-MN_1
- P-MN_4
- P-MN_5

Wired Part
- MN-Ag_1
- CO
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
GMTC: Continued

Pre-Commit Phase

Wireless Part
- CO
- P-MN_2
- Initiator
- P-MN_1
- P-MN_3
- P-MN_4
- P-MN_5

Wired Part
- CO
- MN-Ag_1
- MN-Ag_3
- P-FN_1
- P-FN_2
- MN-Ag_2
GMTC: Continued

Pre-Commit Phase

Wireless Part

CO
P-MN_2

Initiator
P-MN_1

P-MN_3
P-MN_4
P-MN_5

Wired Part

MN-Ag_1

CO
MN-Ag_2

MN-Ag_3

P-FN_1
P-FN_2
GMTC: Continued
GMTC: Continued

Pre-Commit Phase

Wireless Part

- CO
- P-MN_1
- P-MN_2
- P-MN_3
- P-MN_4
- P-MN_5

Initiator

Wired Part

- MN-Ag_1
- CO
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
GMTC: Continued

**Pre-Commit Phase**

**Wireless Part**
- CO
- Initiator
- P-MN_1
- P-MN_2
- P-MN_3
- P-MN_4
- P-MN_5

**Wired Part**
- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
GMTC: Continued

Core Phase

Wireless Part

P-MN_2

P-MN_1

P-MN_3

P-MN_4

P-MN_5

Start two-phase commit

Wired Part

MN-Ag_1

CO

MN-Ag_2

MN-Ag_3

P-FN_1

P-FN_2
GMTC: Continued

Core Phase

Wireless Part
- P-MN_2
- P-MN_1
- P-MN_3
- P-MN_4
- P-MN_5

Wired Part
- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- P-FN_1
- P-FN_2
GMTC: Continued

Core Phase

Wireless Part

- P-MN_2
- P-MN_1
- P-MN_3
- P-MN_4
- P-MN_5

Initiator

Wired Part

- MN-Ag_1
- MN-Ag_2
- MN-Ag_3
- CO
- P-FN_1
- P-FN_2
Results

Simulation Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical area</td>
<td>$2km \times 2km$</td>
</tr>
<tr>
<td>Communication range</td>
<td>$250m$</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Random Waypoint (RWP)</td>
</tr>
<tr>
<td>Node speed</td>
<td>uniform in $[0.5, 1.5] \ m/s$</td>
</tr>
<tr>
<td># Nodes</td>
<td>$\in [20, 200]$</td>
</tr>
<tr>
<td># Pre-selected COs</td>
<td>$\in {3, 5, 10}$</td>
</tr>
<tr>
<td># P-MNs</td>
<td>10</td>
</tr>
<tr>
<td>Lifetime</td>
<td>$\in {60s, 120s, 300s}$</td>
</tr>
</tbody>
</table>

Table 3 [4]
Results

Figure 3 [4]
Results

Figure 4 [4]
Results: Conclusion
Results: Conclusion

- Commit rate increased
Results: Conclusion

- Commit rate increased
- Decision time reduced in some cases
Outline

1 Background

2 Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC)

3 Partition-tolerant atomic commit protocol (ParTAC)

4 Generalized mobile transaction commit (GMTC)

5 Conclusion
   Conclusion
Conclusion
Conclusion

- FT-PPTC
Conclusion

- FT-PPTC
- ParTAC
Conclusion

- FT-PPTC
- ParTAC
- GMTC
Acknowledgements

I would like to thank KK, Elena Machkasova, and Thomas Hagen for helping me with my senior seminar.
Questions??????
References I

B. Ayari, A. Khelil, and N. Suri.
Ft-pptc: An efficient and fault-tolerant commit protocol for mobile environments.

B. Ayari, A. Khelil, and N. Suri.
Partac: A partition-tolerant atomic commit protocol for manets.

B. Ayari, A. Khelil, and N. Suri.
On the design of perturbation-resilient atomic commit protocols for mobile transactions.

B. Ayari, A. Khelil, and N. Suri.
Gmtc: A generalized commit approach for hybrid mobile environments.
References II

C. Date.

An Introduction to Database Systems.