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

Introduction: Continued

Introduction: Continued

Database types:

Introduction: Continued

Database types:

- Centralized database

Introduction: Continued

Database types:

- Centralized database
- Distributed database

Introduction: Continued

Database types:

- Centralized database
- Distributed database
- Mobile database

Introduction: Continued

Introduction: Continued

Problems with distributed databases:

Introduction: Continued

Problems with distributed databases:

- Design is difficult

Introduction: Continued

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

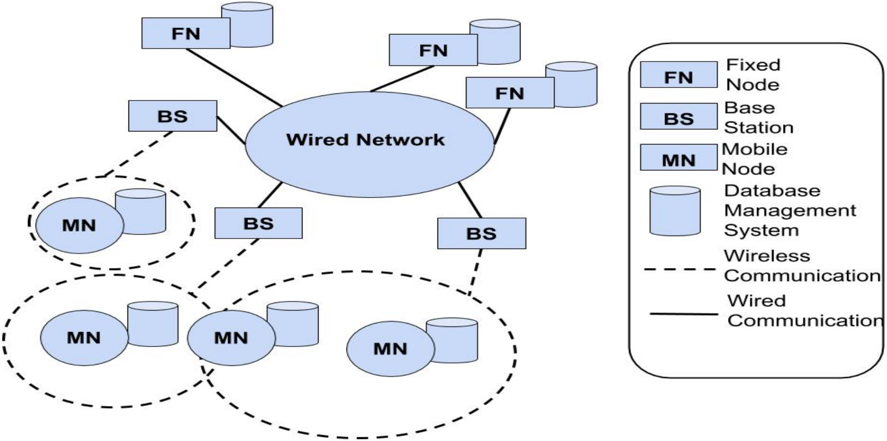
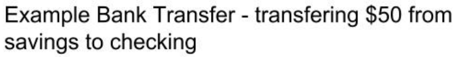


Figure 1 [3]

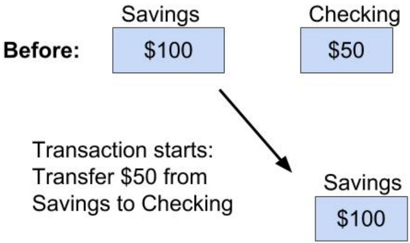
Transactions and Transaction Processing



Example Bank Transfer - transfering $50 from savings to checking

Transactions and Transaction Processing

Example Bank Transfer - transfering $50 from savings to checking



Before:

Savings

$100

Checking

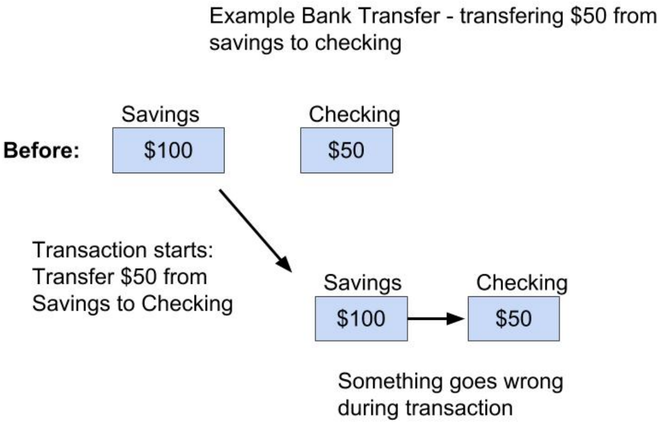
$50

Transaction starts: Transfer $50 from Savings to Checking

Savings

$100

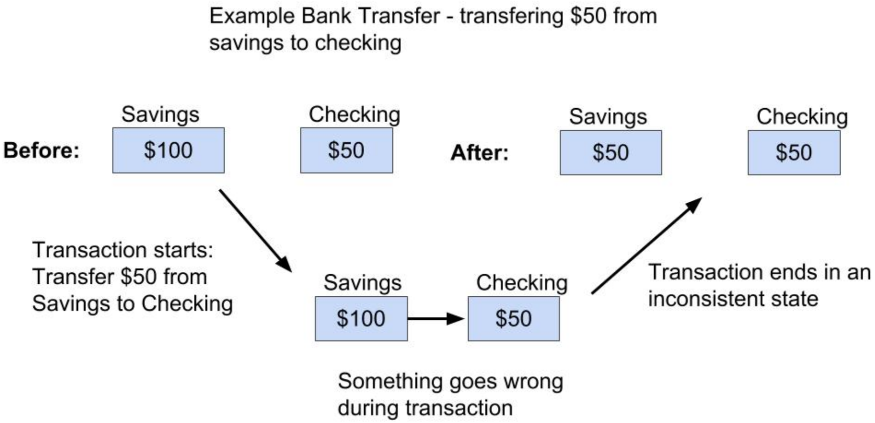
Transactions and Transaction Processing



Example Bank Transfer - transfering $50 from savings to checking

Before:
Savings
$100
Checking
$50
Transaction starts: Transfer $50 from Savings to Checking
Savings
$100
Checking
$50
Something goes wrong during transaction

Transactions and Transaction Processing


Example Bank Transfer - transfering $50 from savings to checking
Before:
Savings
$100
Checking
$50
After:
Savings
$50
Checking
$50
Transaction starts: Transfer $50 from Savings to Checking
Savings
$100
Checking
$50
Something goes wrong during transaction
Transaction ends in an inconsistent state

Perturbations

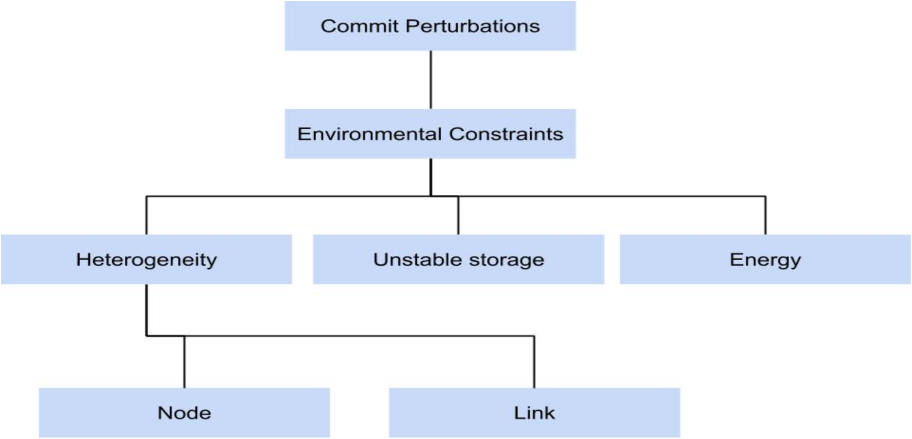


Figure 2 [3]

Perturbations

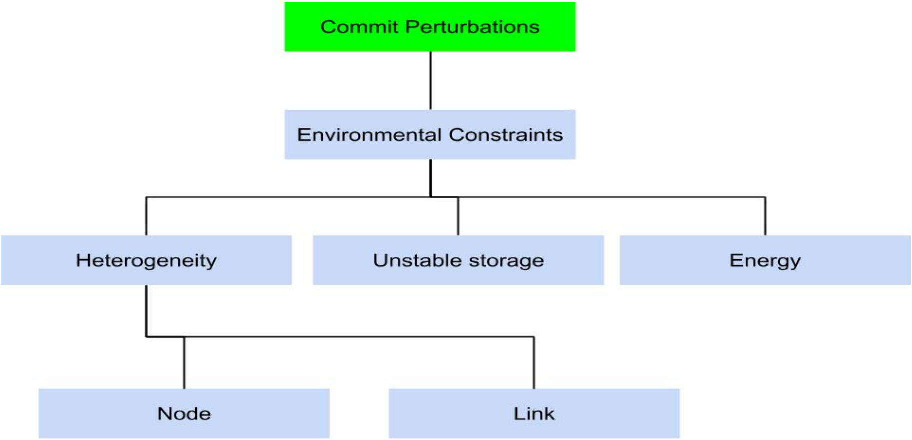


Figure 2 [3]

Perturbations

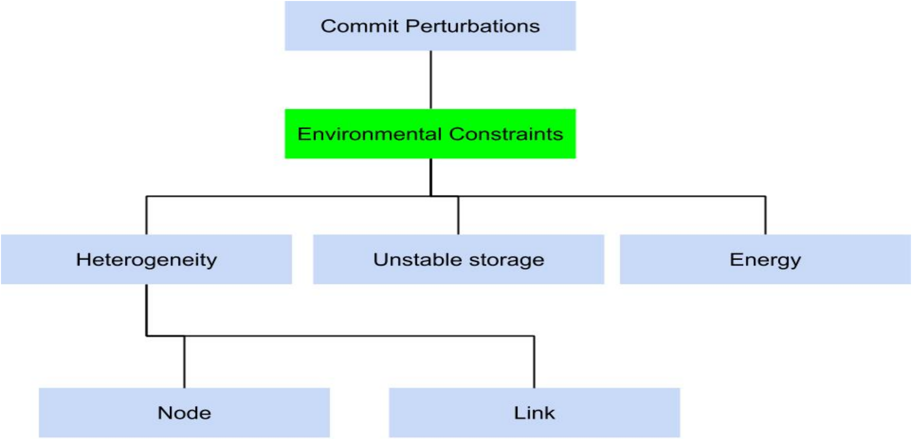


Figure 2 [3]

Perturbations

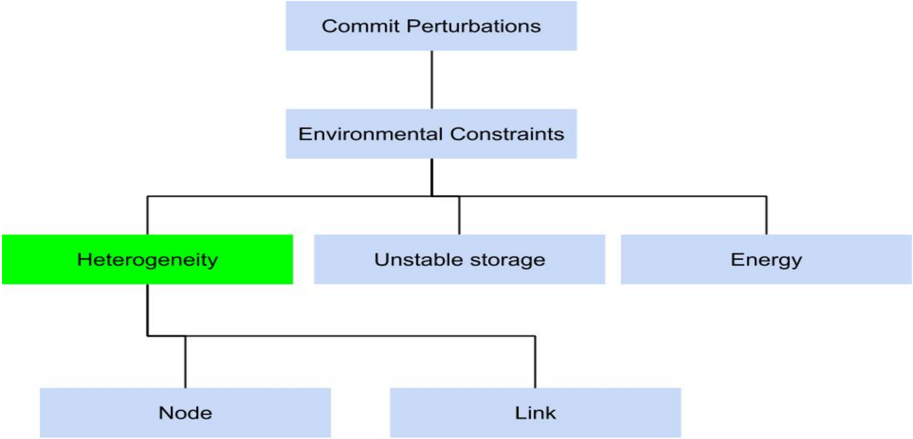


Figure 2 [3]

Perturbations

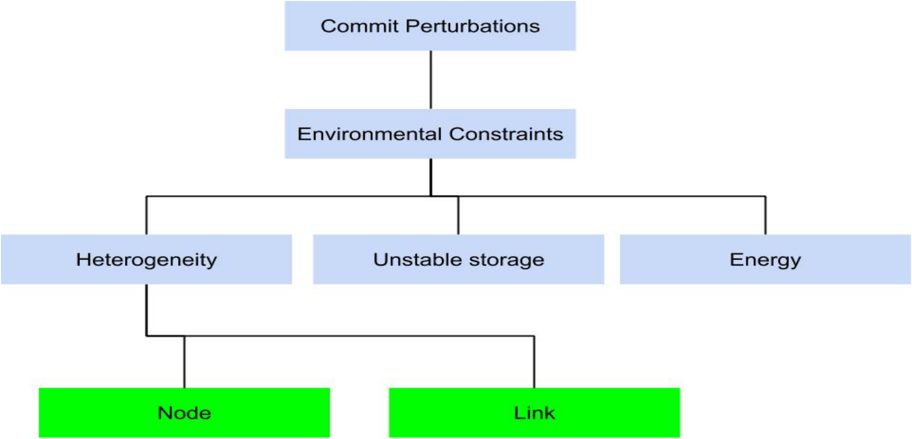


Figure 2 [3]

Perturbations

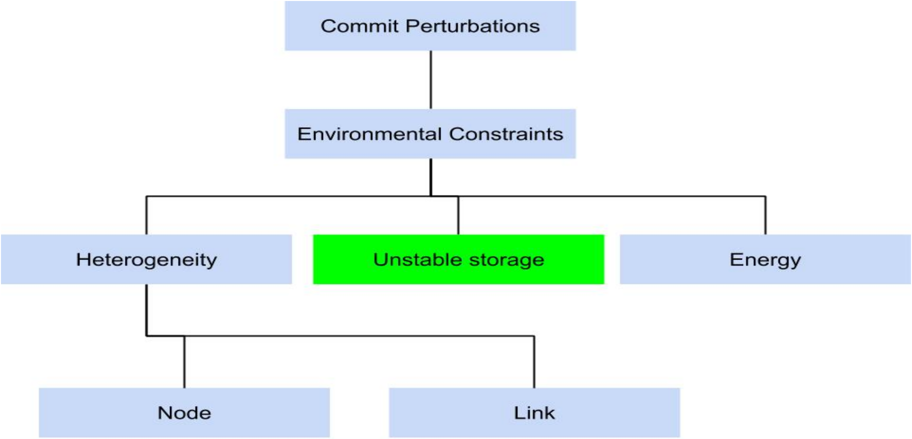


Figure 2 [3]

Perturbations

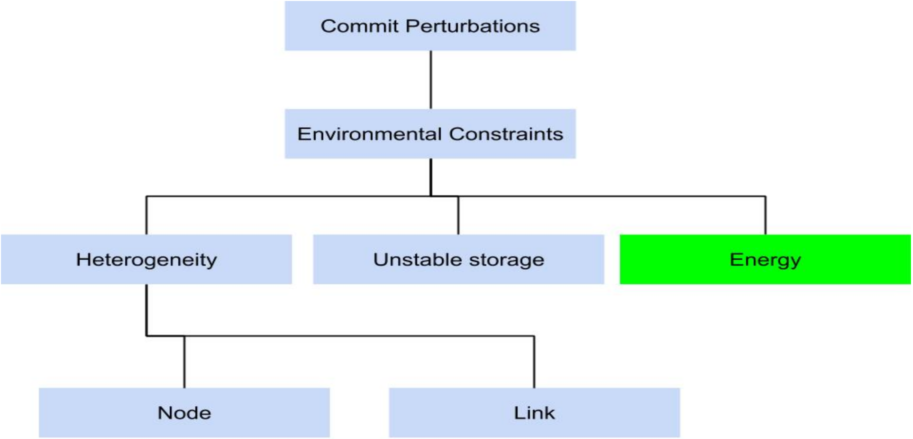


Figure 2 [3]

Perturbations

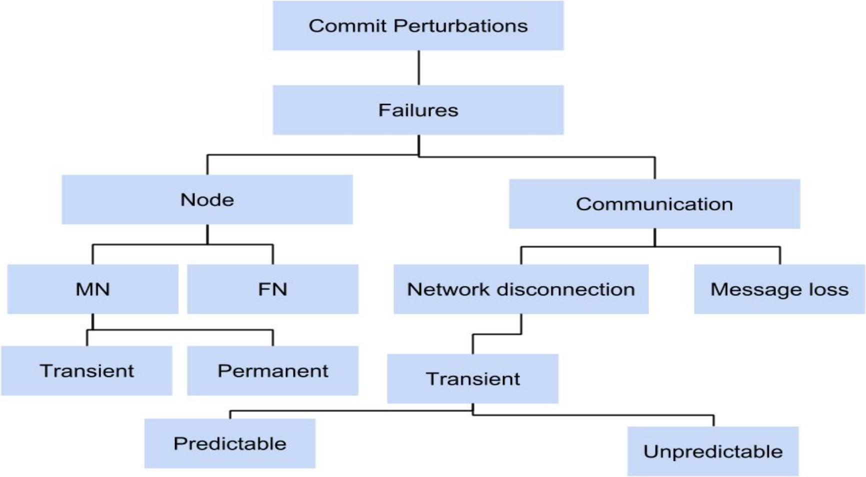


Figure 2 [3]

Perturbations

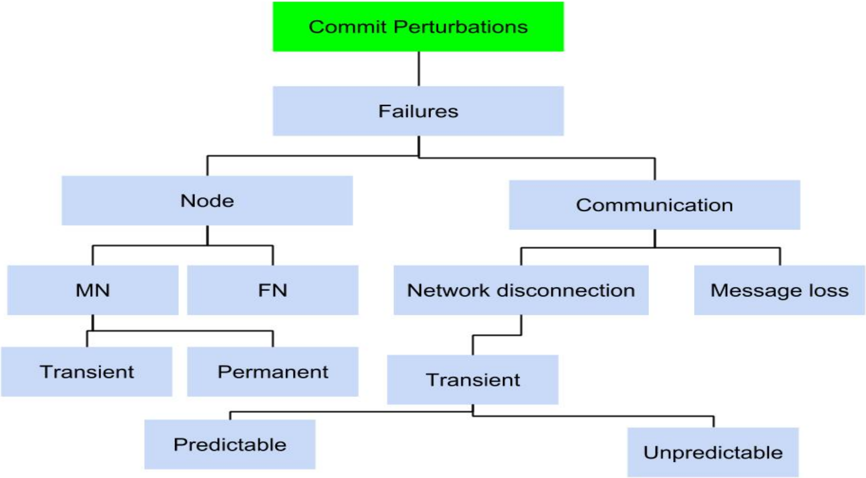


Figure 2 [3]

Perturbations

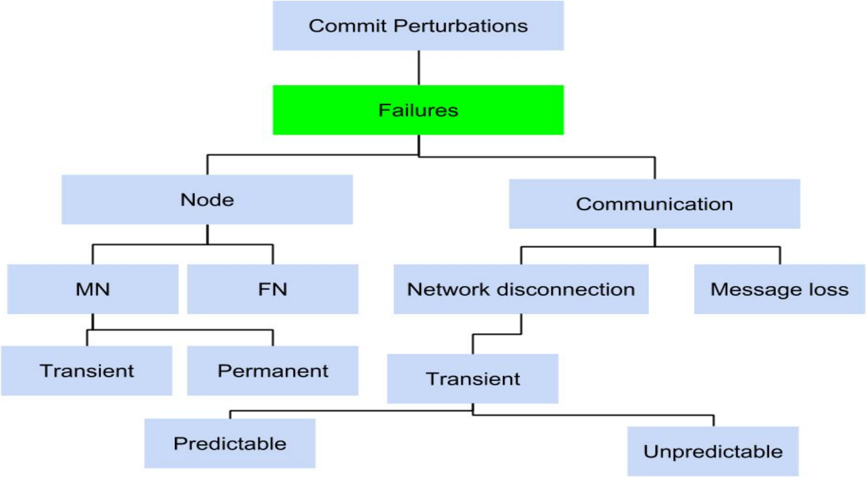


Figure 2 [3]

Perturbations

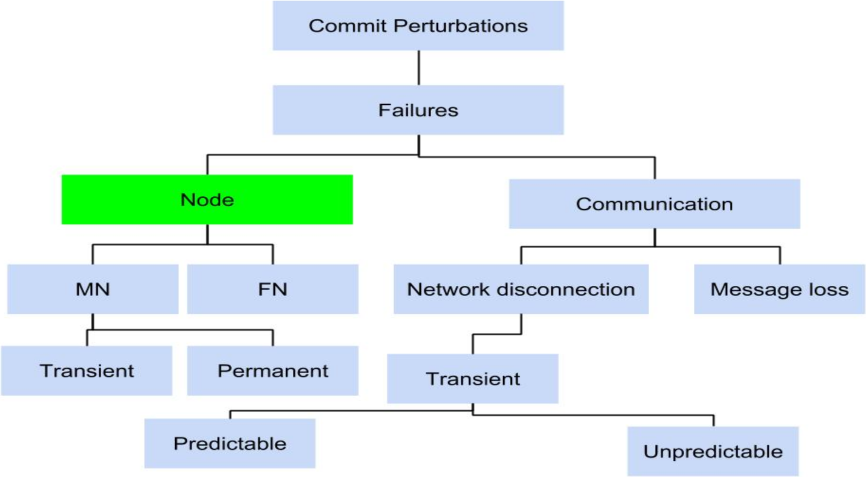


Figure 2 [3]

Perturbations

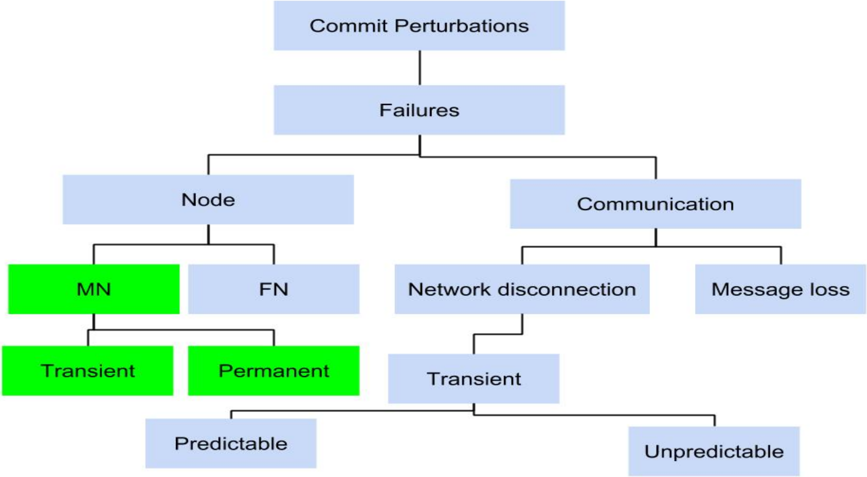


Figure 2 [3]

Perturbations

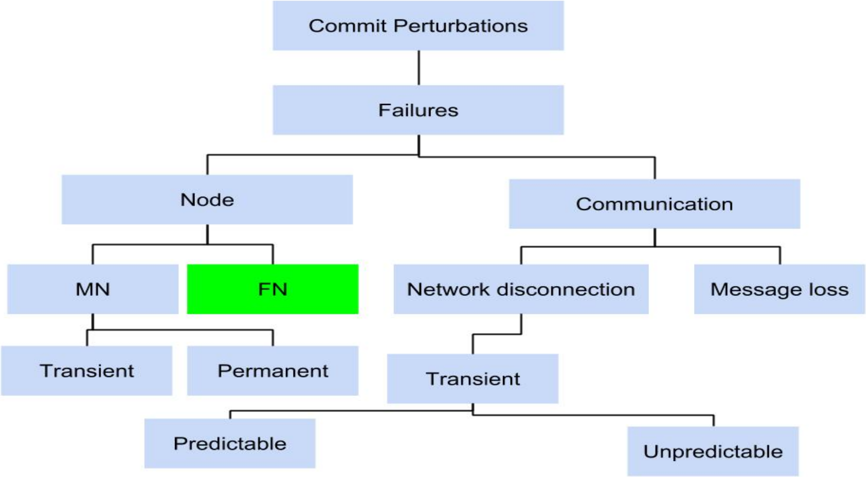


Figure 2 [3]

Perturbations

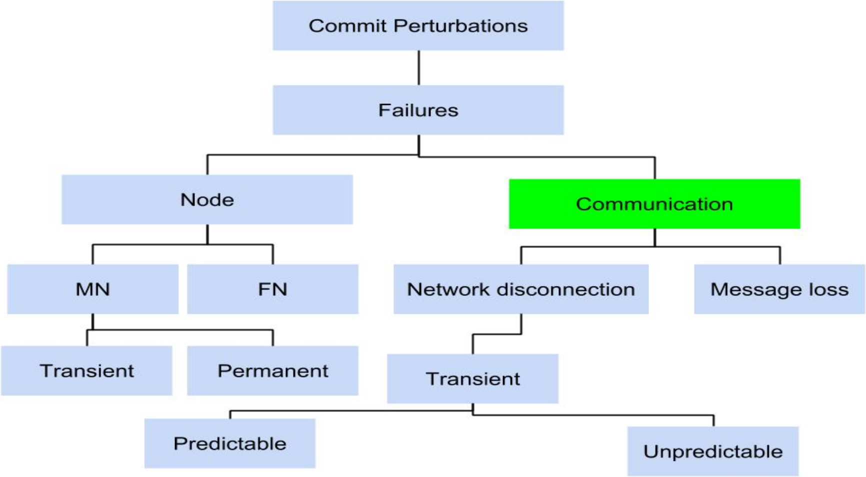


Figure 2 [3]

Perturbations

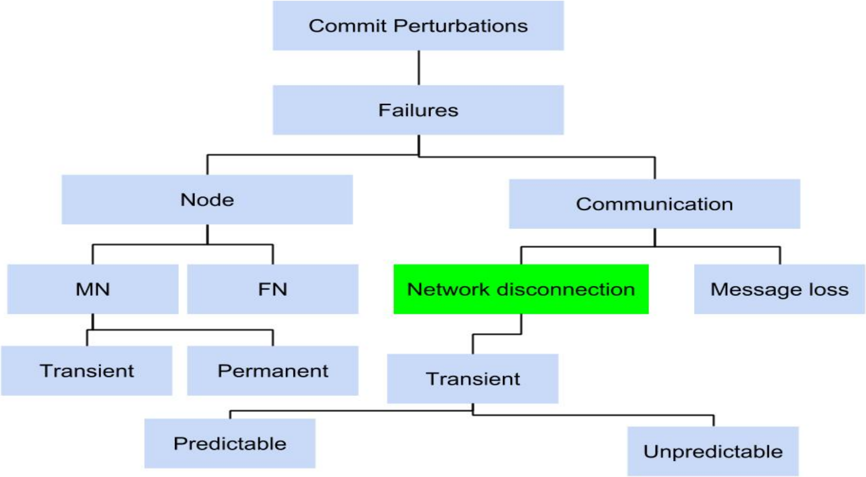


Figure 2 [3]

Perturbations

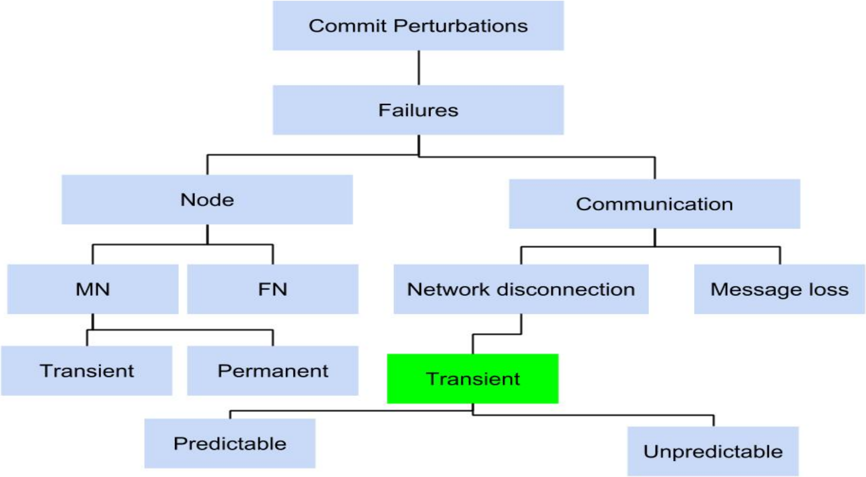
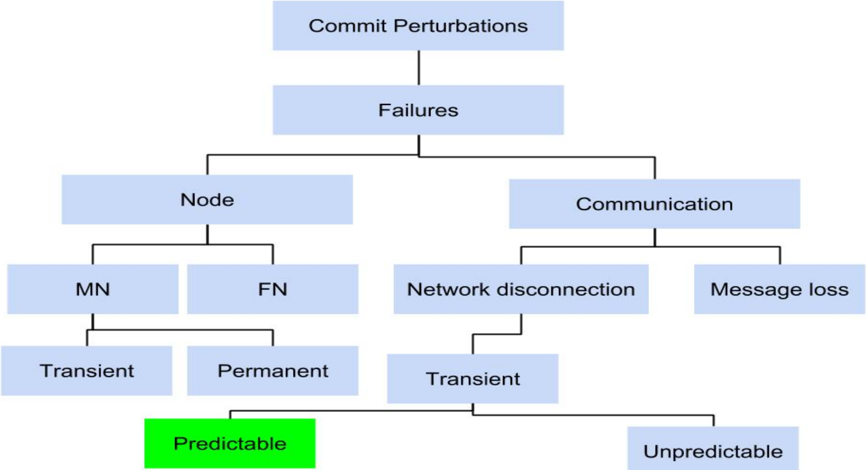


Figure 2 [3]

Perturbations


Commit Perturbations

Failures

Node

Communication

MN

FN

Network disconnection

Message loss

Transient

Permanent

Transient

Predictable

Unpredictable

Figure 2 [3]

Perturbations

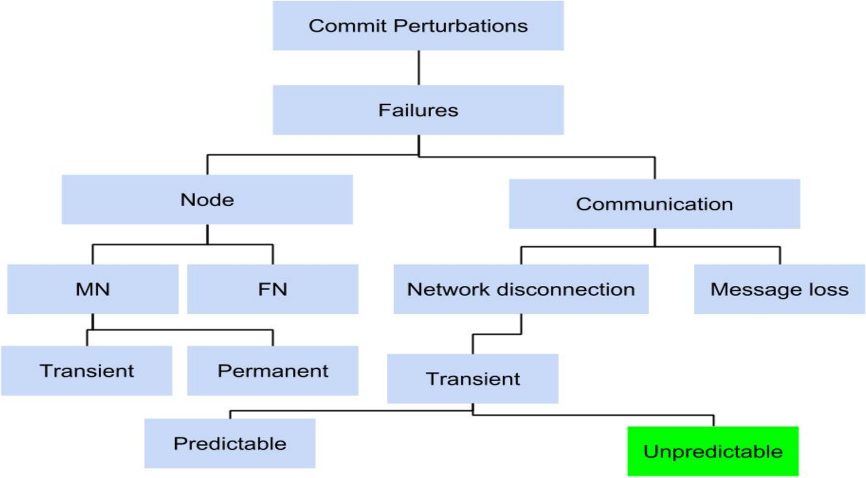


Figure 2 [3]

Perturbations

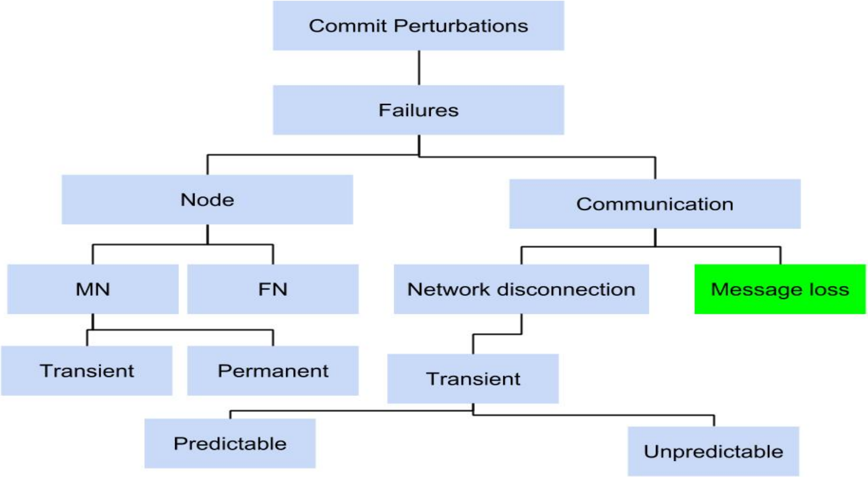


Figure 2 [3]

Commit Protocols

Commit Protocols:

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)

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

FT-PPTC is aimed at:

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

FT-PPTC is aimed at:

- Improving transactions in infrastructure environments

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)

FT-PPTC is aimed at:

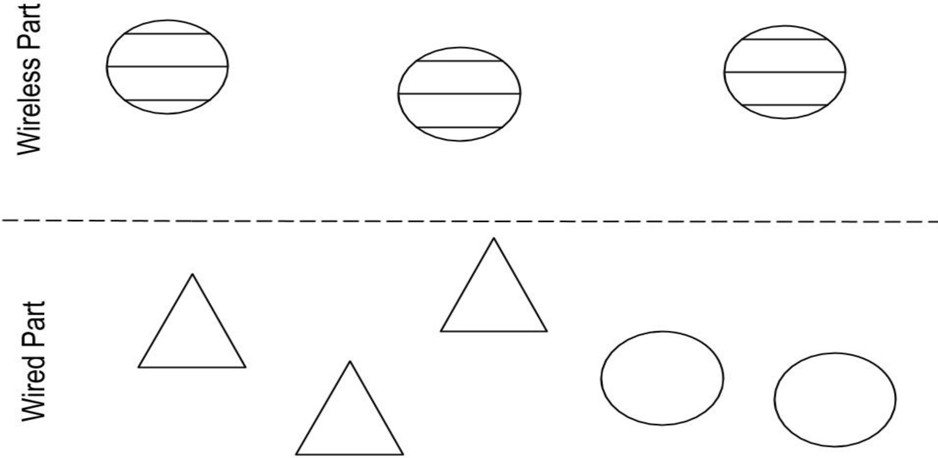
- Improving transactions in infrastructure environments
- Tolerating environmental constraints(heterogeneity, unstable storage, and battery life)
- Tolerating network disconnections (predicable and unpredictable disconnections)

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

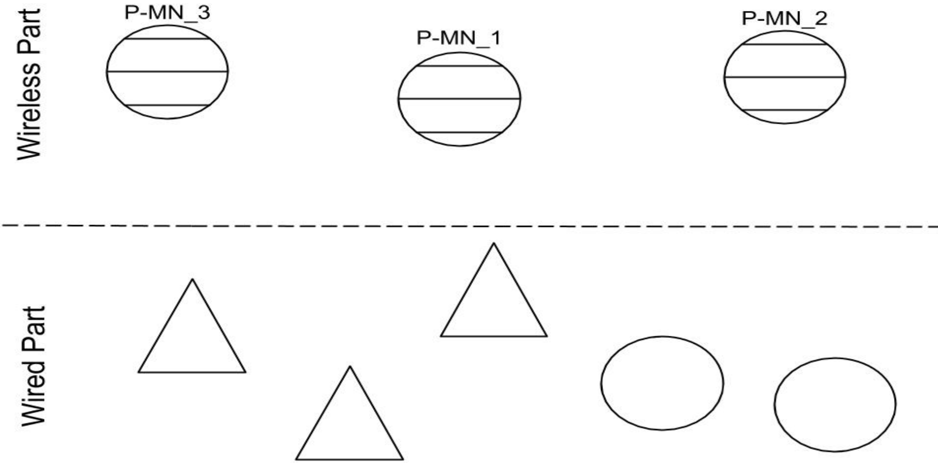
FT-PPTC is aimed at:

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

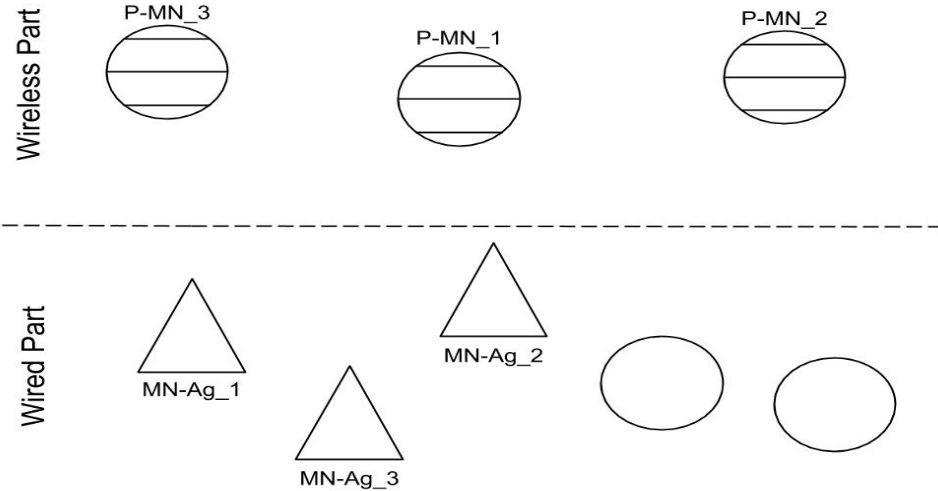
FT-PPTC: Continued



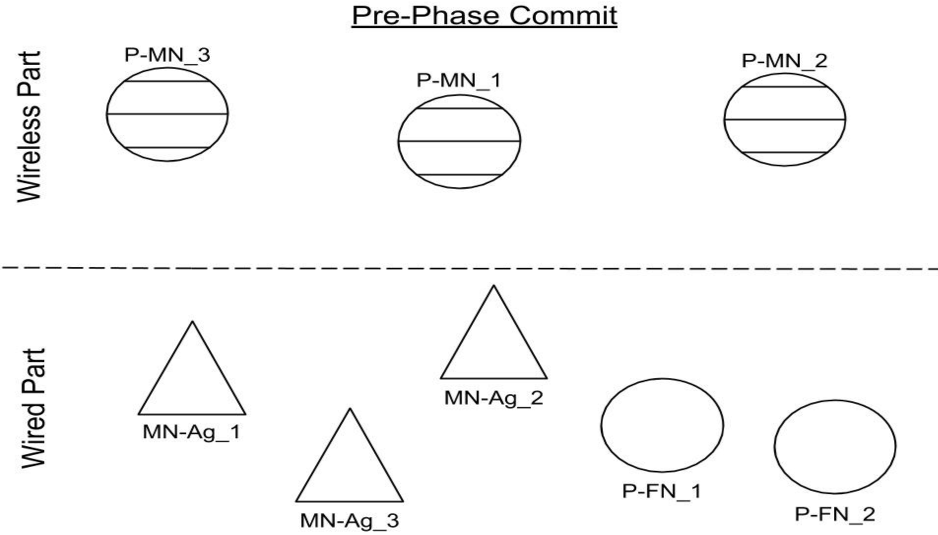
FT-PPTC: Continued



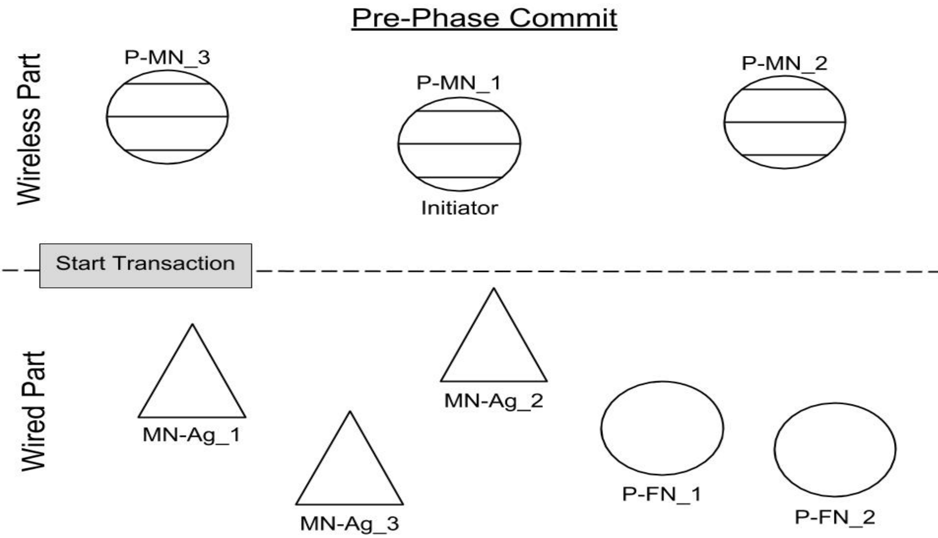
FT-PPTC: Continued



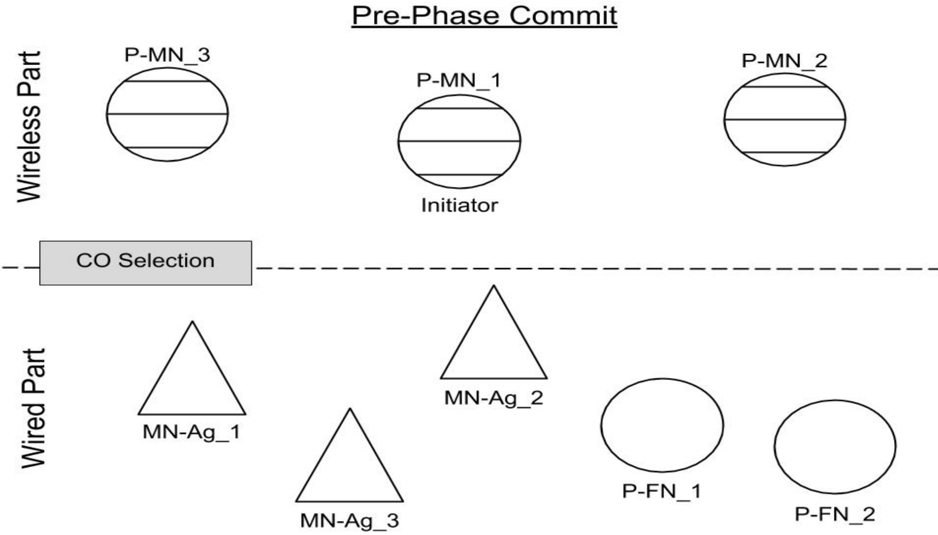
FT-PPTC: Continued



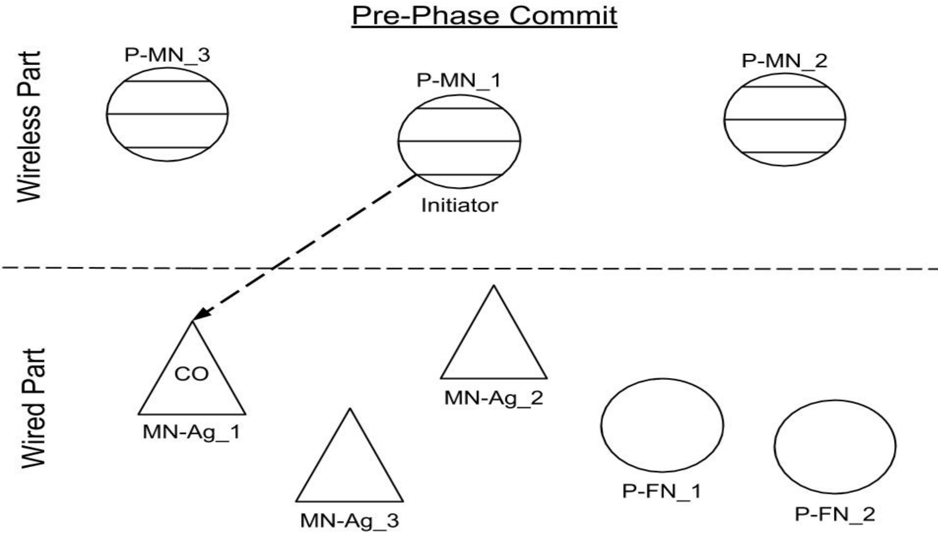
FT-PPTC: Continued



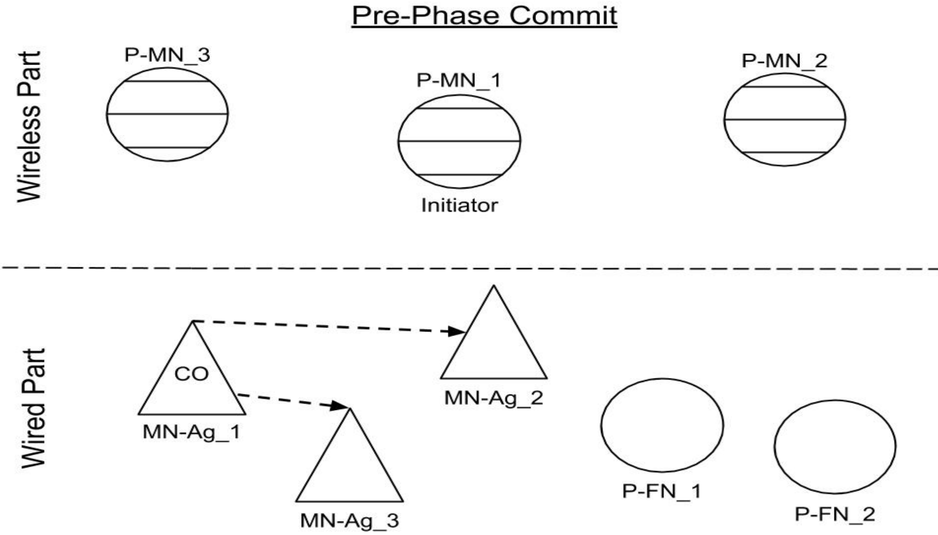
FT-PPTC: Continued



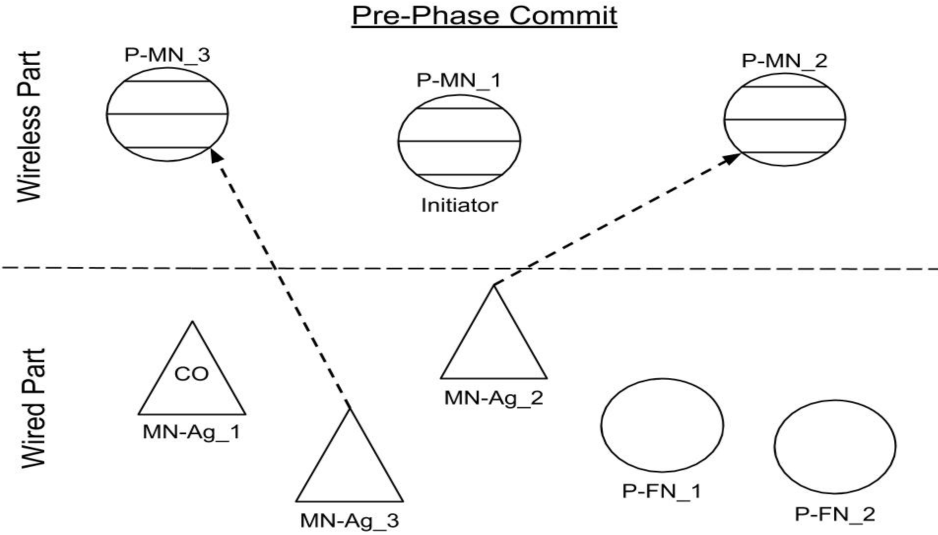
FT-PPTC: Continued



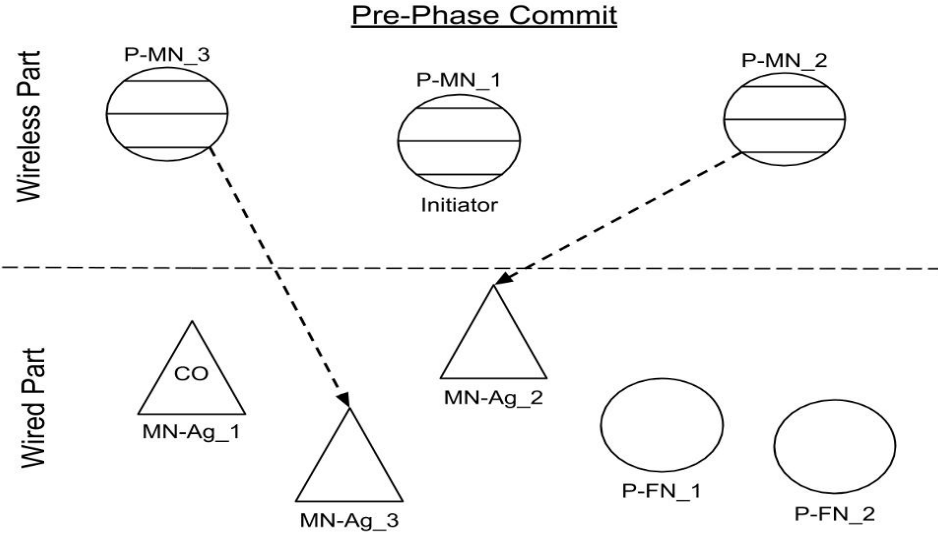
FT-PPTC: Continued



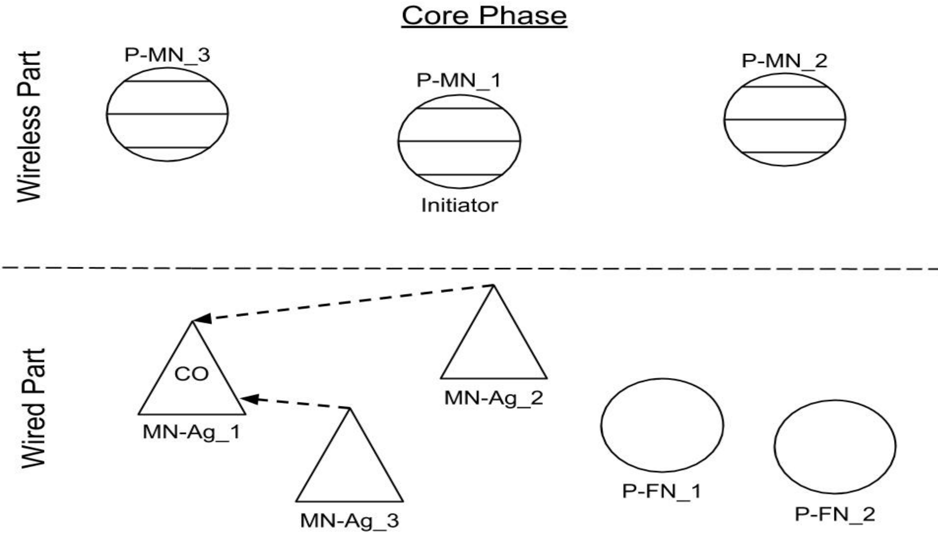
FT-PPTC: Continued



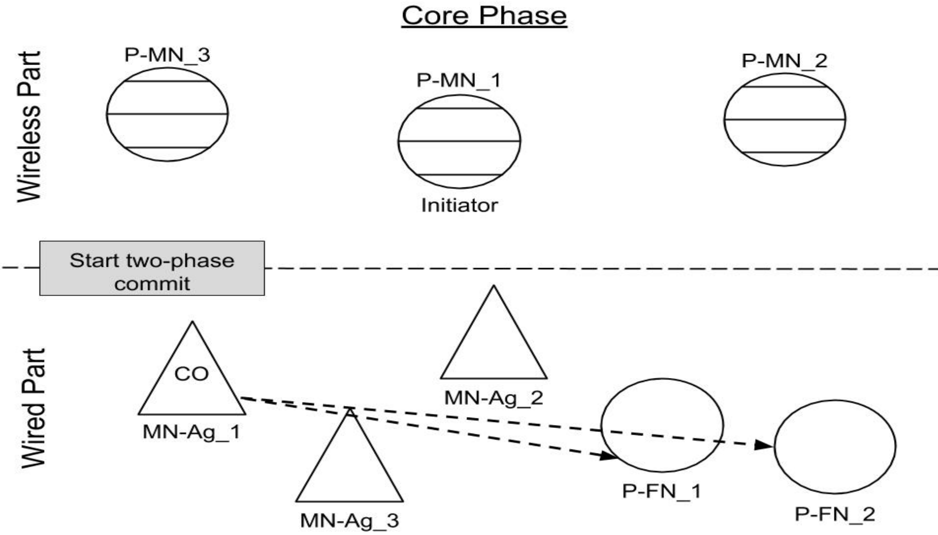
FT-PPTC: Continued



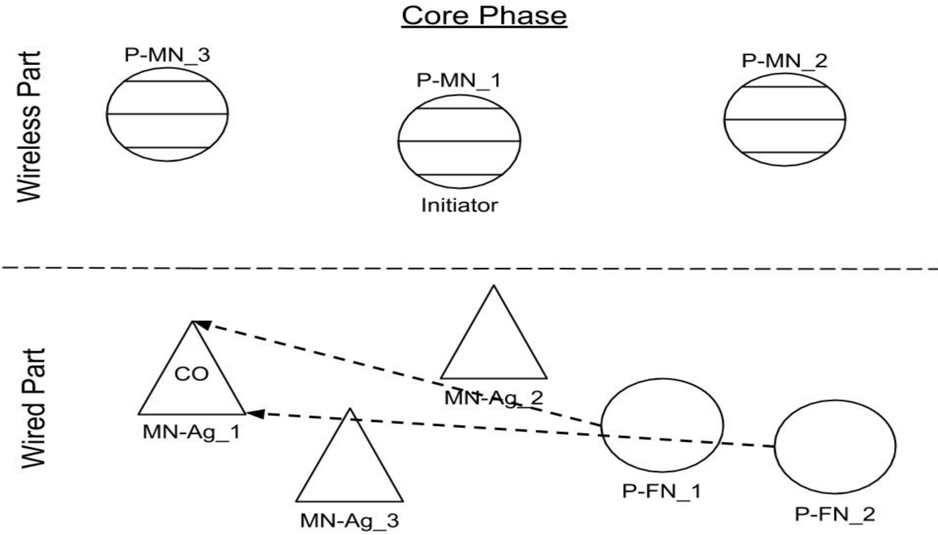
FT-PPTC: Continued



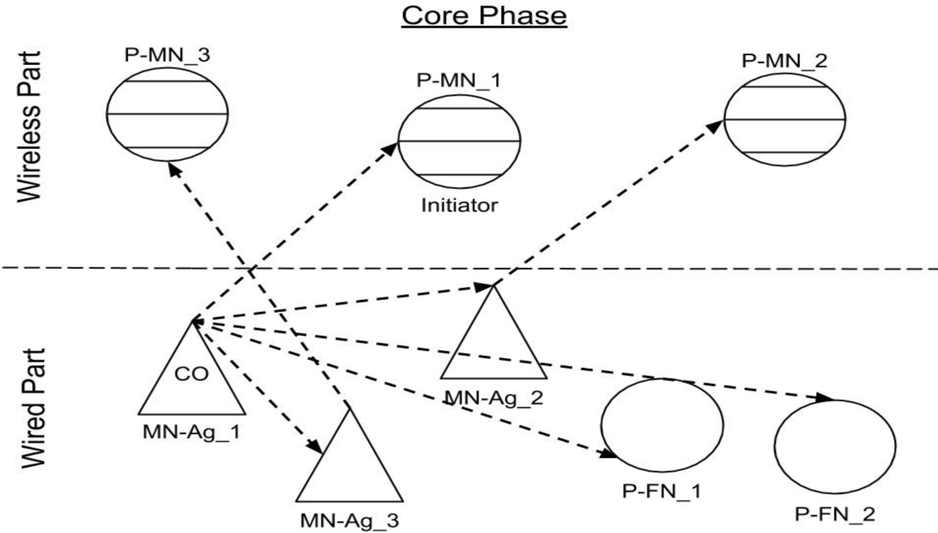
FT-PPTC: Continued



FT-PPTC: Continued



FT-PPTC: Continued

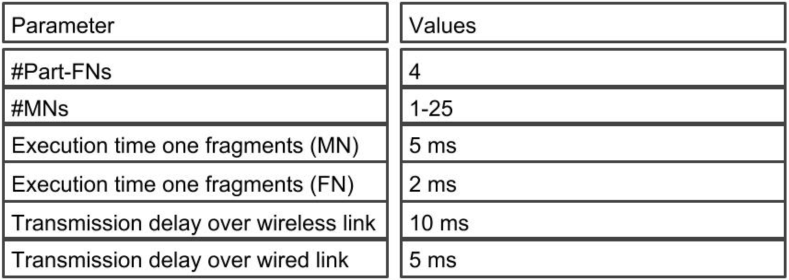


Results

Simulation Settings

Results

Simulation Settings



Parameter	Values
#Part-FNs	4
#MNs	1-25
Execution time one fragments (MN)	5 ms
Execution time one fragments (FN)	2 ms
Transmission delay over wireless link	10 ms
Transmission delay over wired link	5 ms

FT-PPTC: Results

Conclusion:

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)

Partition-tolerant atomic commit protocol (ParTAC)

ParTAC is aimed at:

Partition-tolerant atomic commit protocol (ParTAC)

ParTAC is aimed at:

- Improving transactions in mobile ad-hoc scenarios

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:

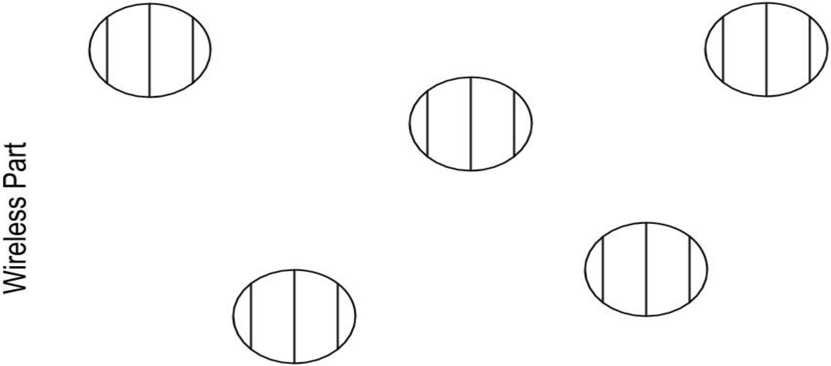
- Improving transactions in mobile ad-hoc scenarios
- Tolerating message loss (network congestion)
- Tolerating transient failures (network disconnections)

Partition-tolerant atomic commit protocol (ParTAC)

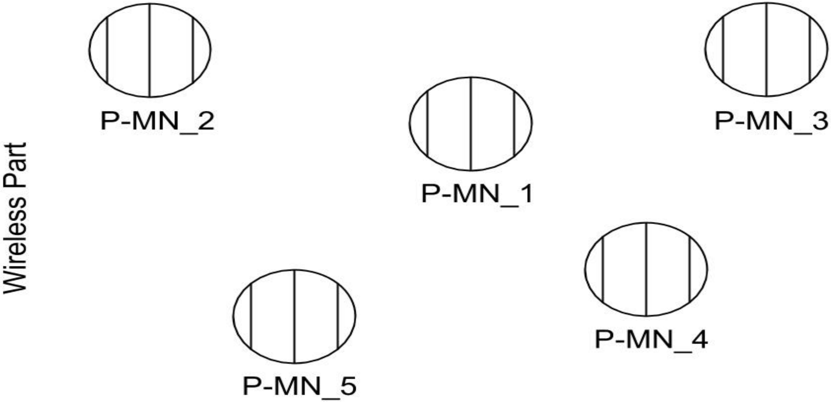
ParTAC is aimed at:

- 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)

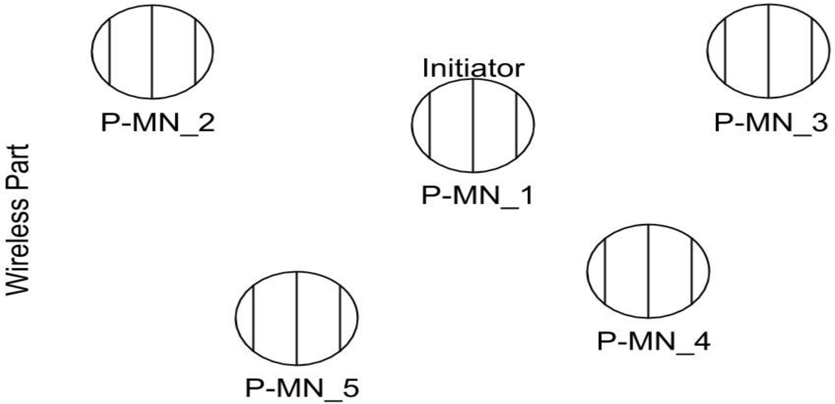
ParTAC: Continued



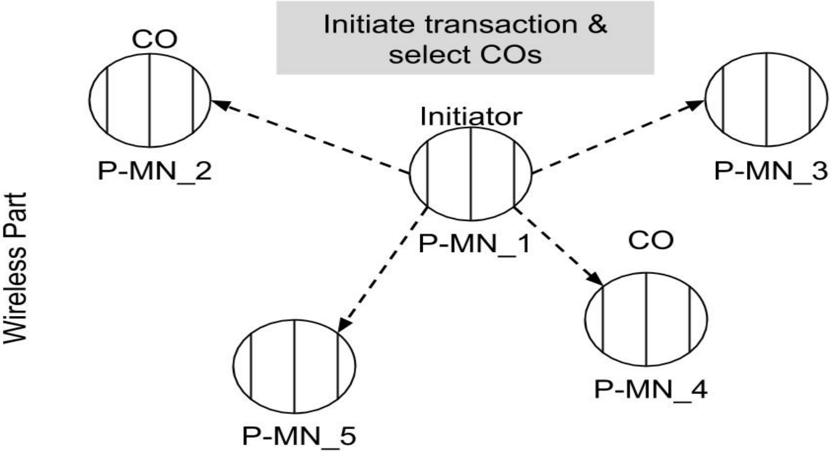
ParTAC: Continued



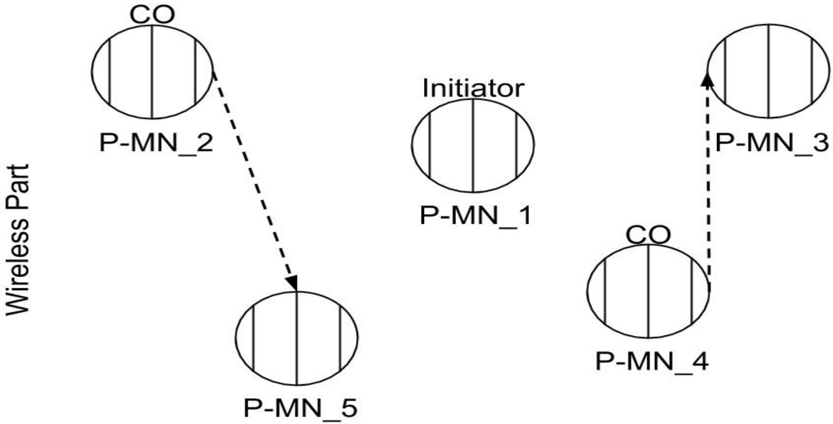
ParTAC: Continued



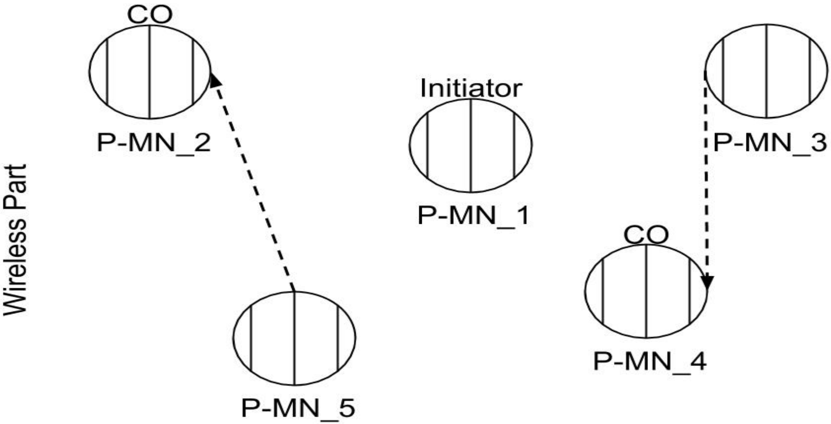
ParTAC: Continued



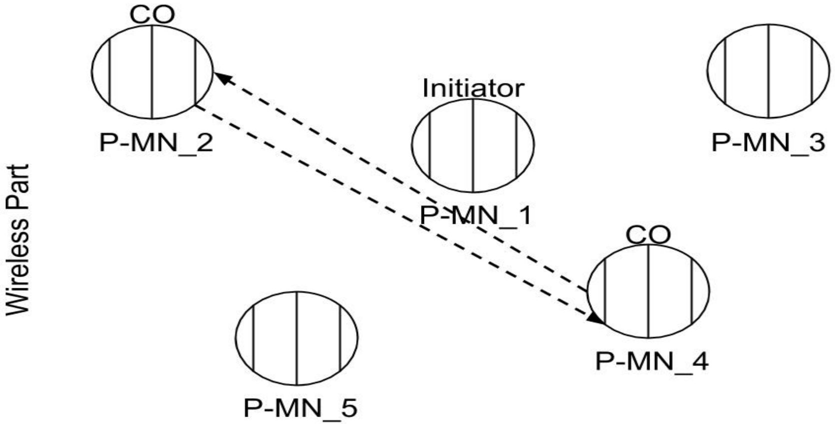
ParTAC: Continued



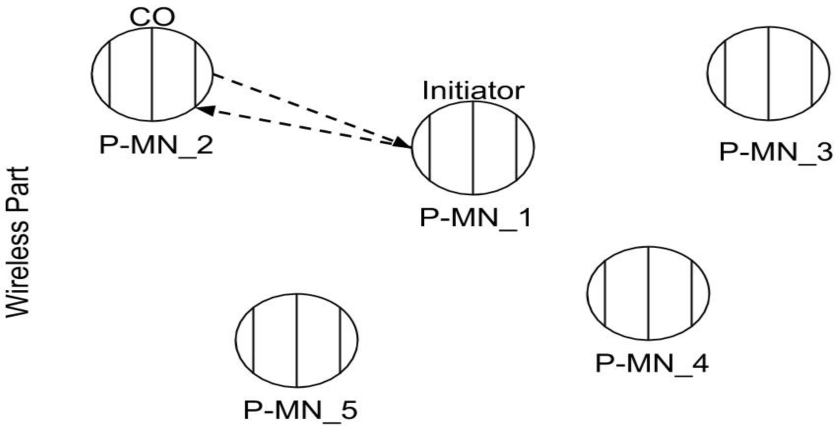
ParTAC: Continued



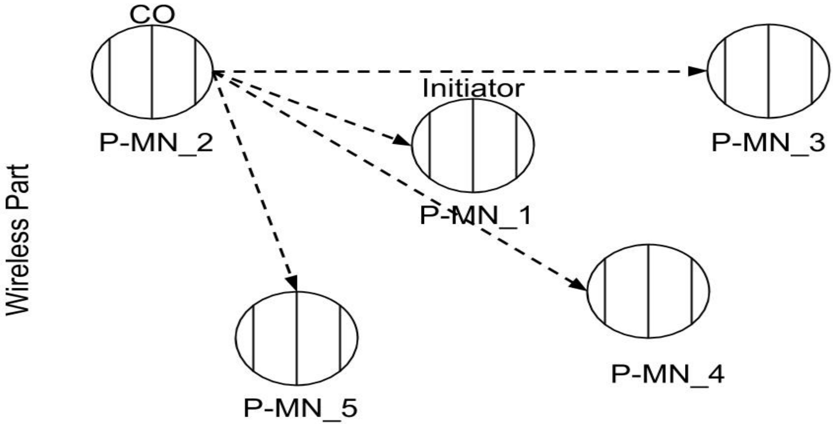
ParTAC: Continued



ParTAC: Continued



ParTAC: Continued

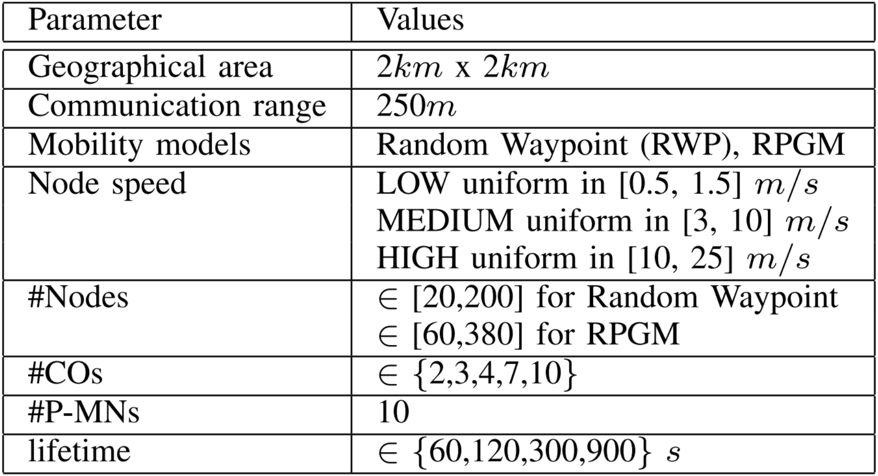


ParTAC: Results

Simulation Settings

ParTAC: Results

Simulation Settings



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

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

GMTC

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

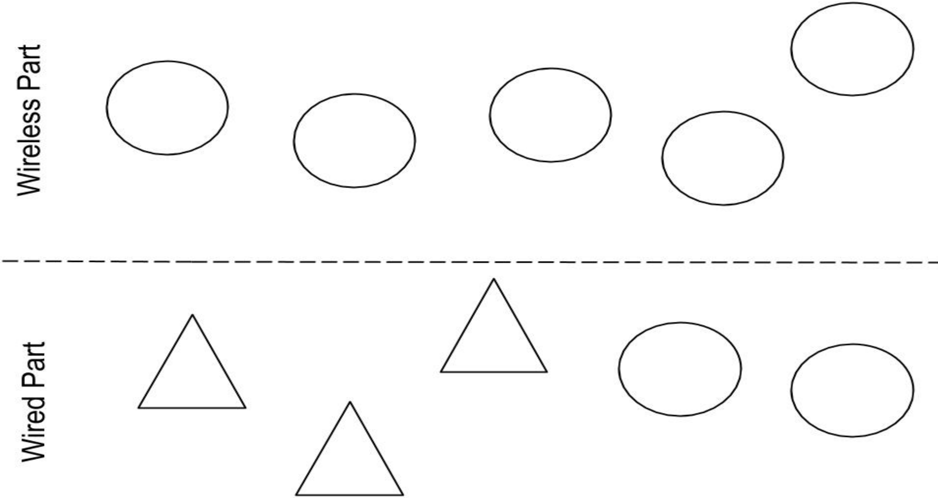
Why ParTAC and FT-PPTC fail in this environment:

- ParTAC very inefficient in environments with infrastructure
- Transactions may not involve any infrastructure so FT-PPTC won't work

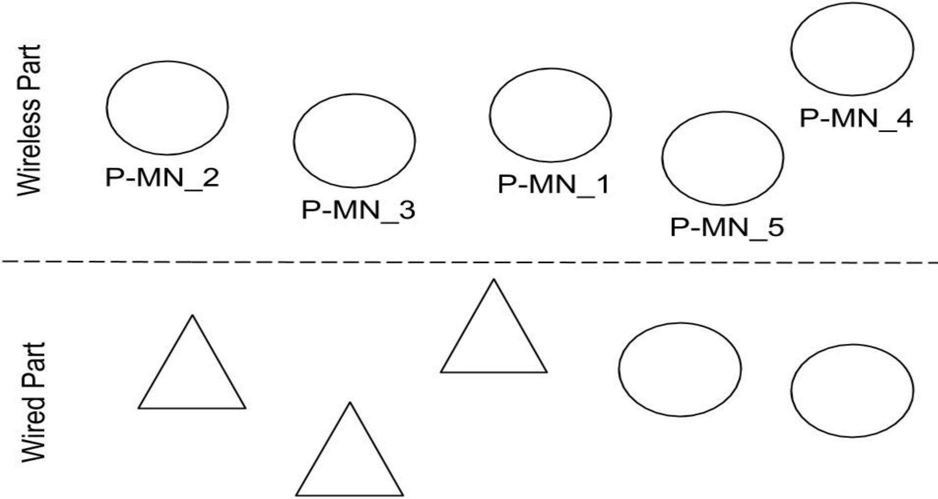
Improving transactions is accomplished by:

- Combining ParTAC and FT-PPTC
- Reducing resource blocking time
- Reduce transaction decision time

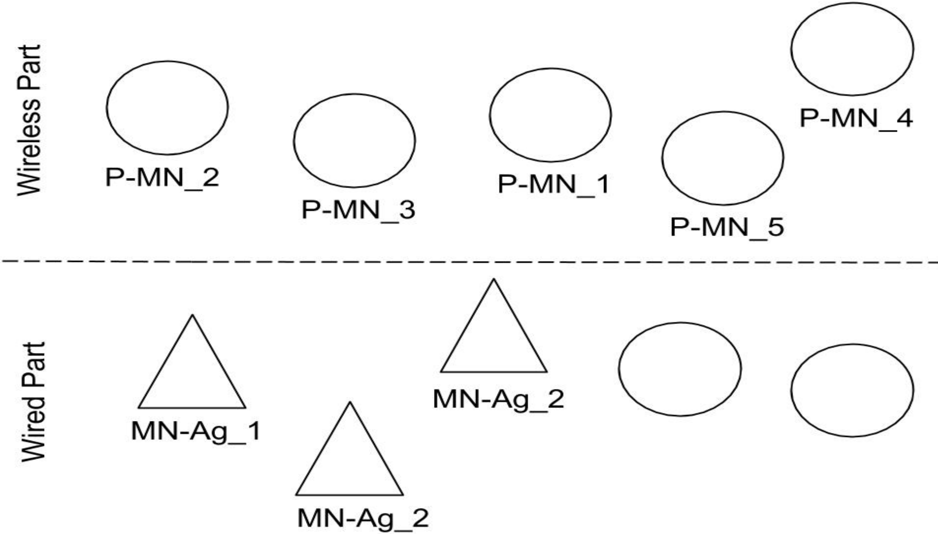
GMTC: Continued



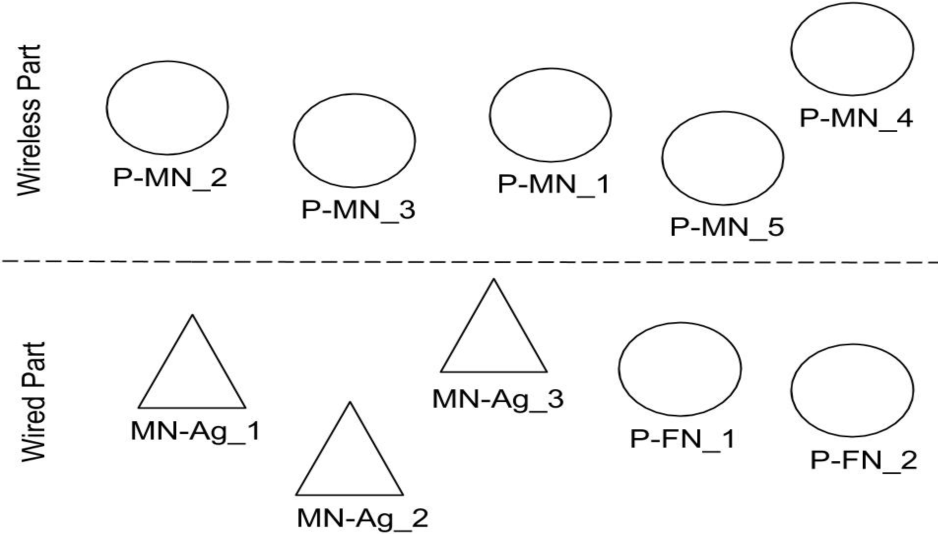
GMTC: Continued



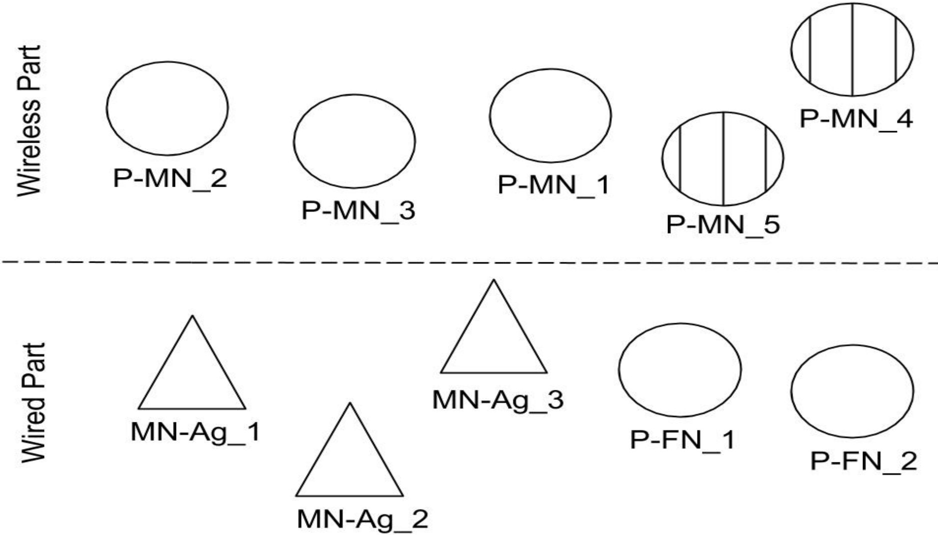
GMTC: Continued



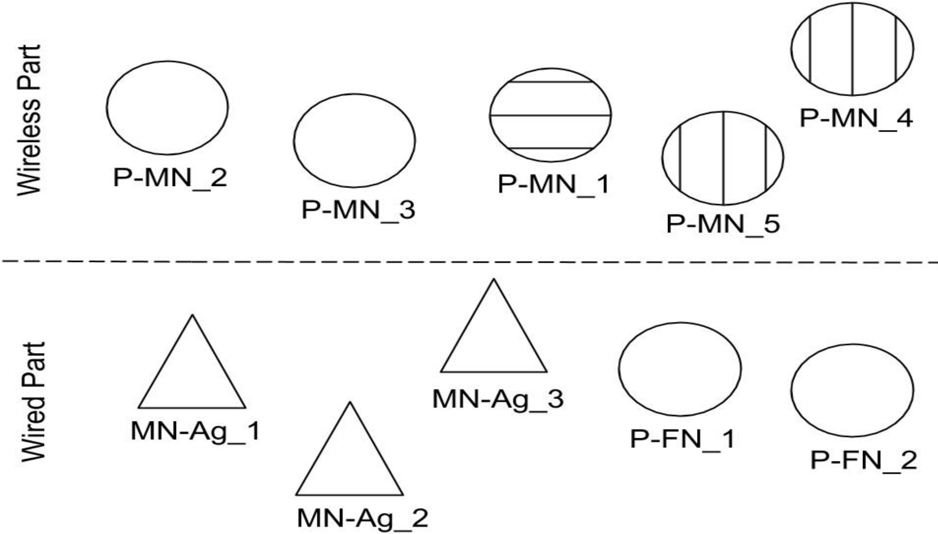
GMTC: Continued



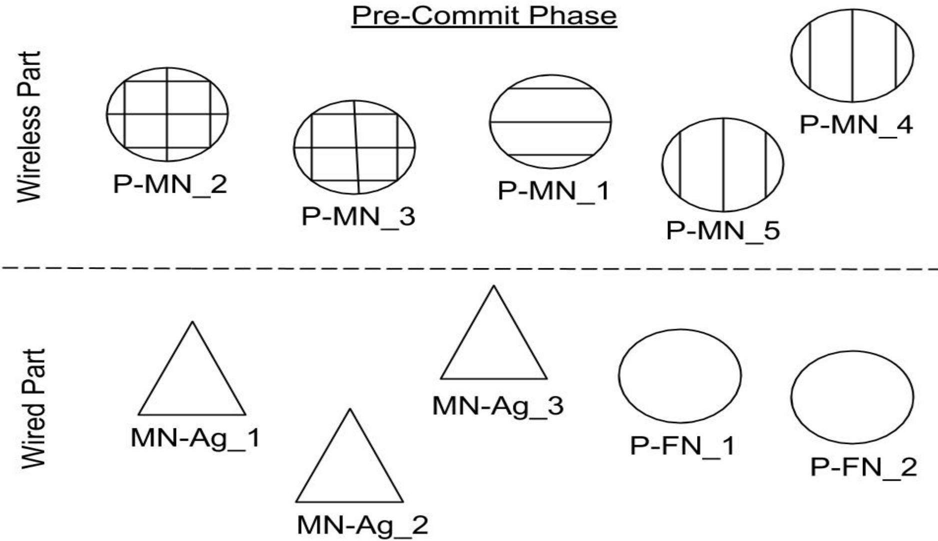
GMTC: Continued



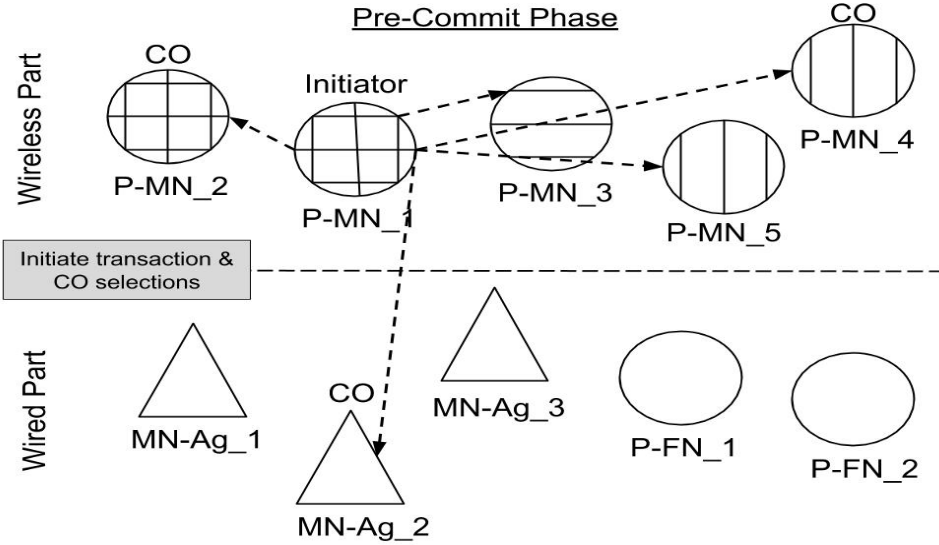
GMTC: Continued



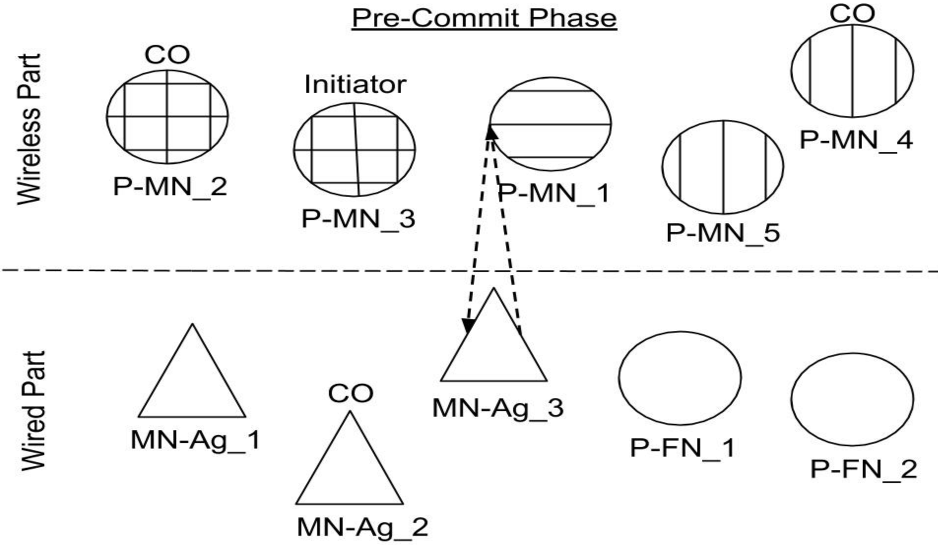
GMTC: Continued



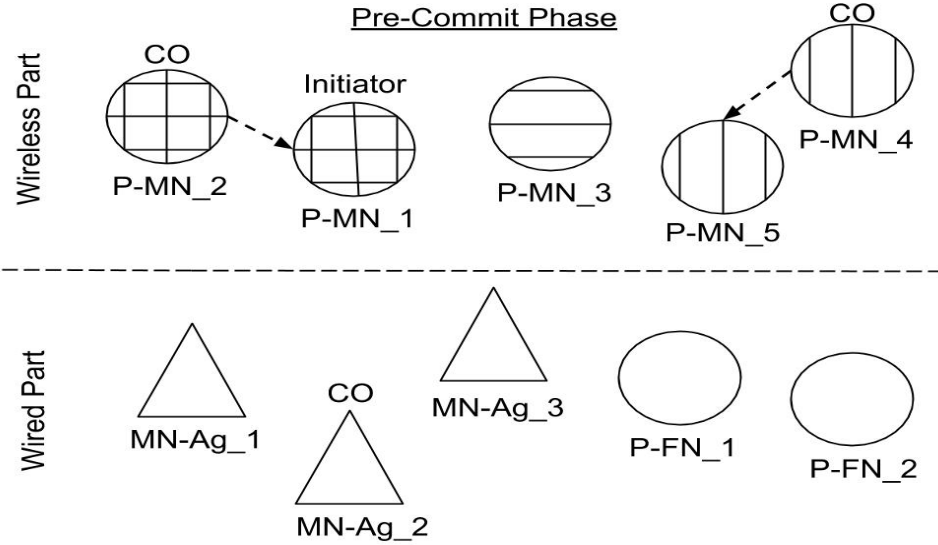
GMTC: Continued



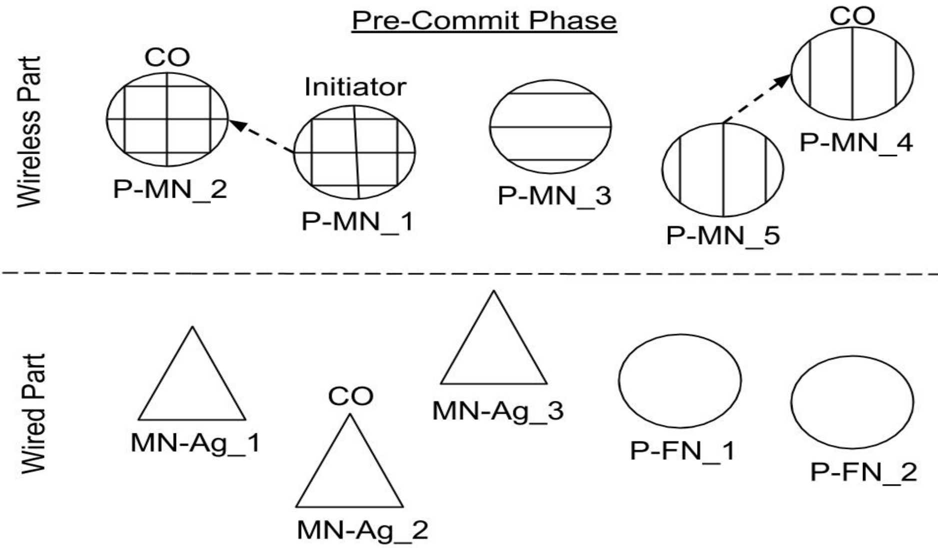
GMTC: Continued



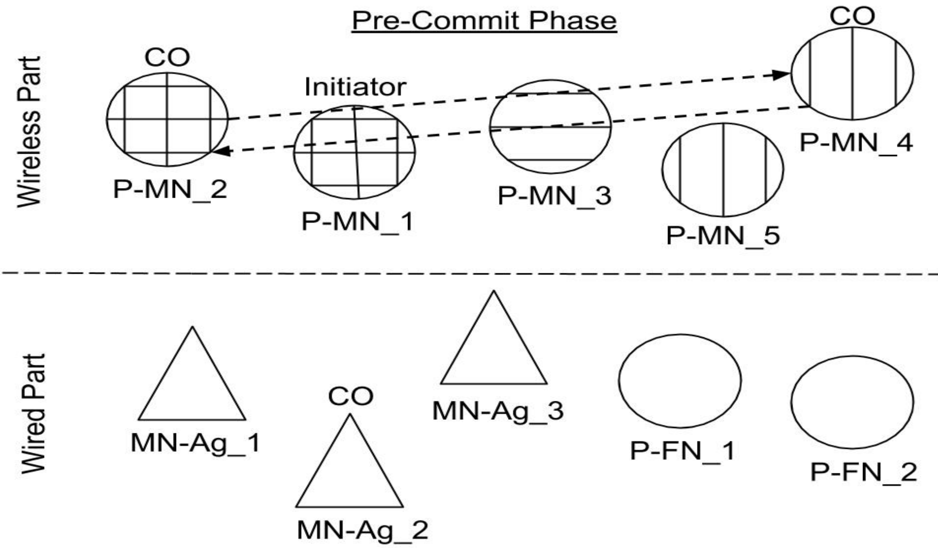
GMTC: Continued



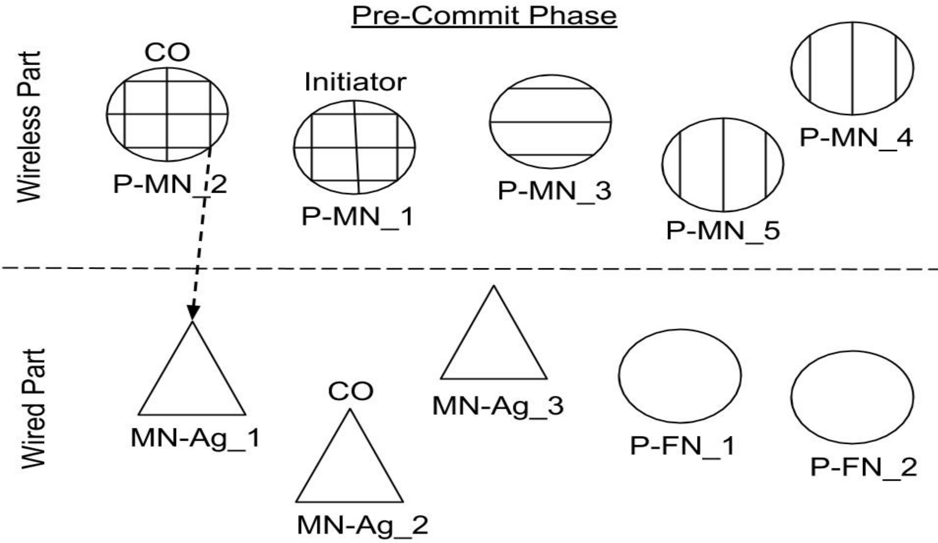
GMTC: Continued



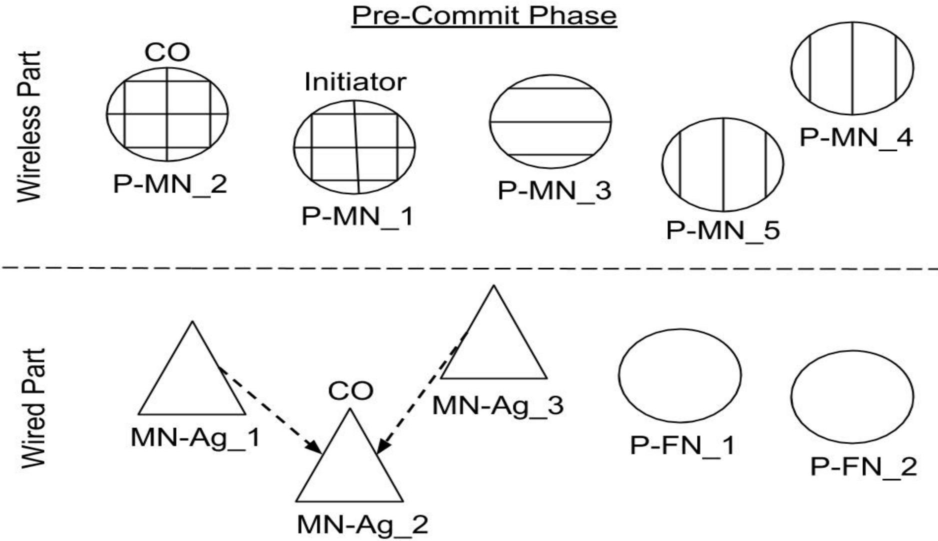
GMTC: Continued



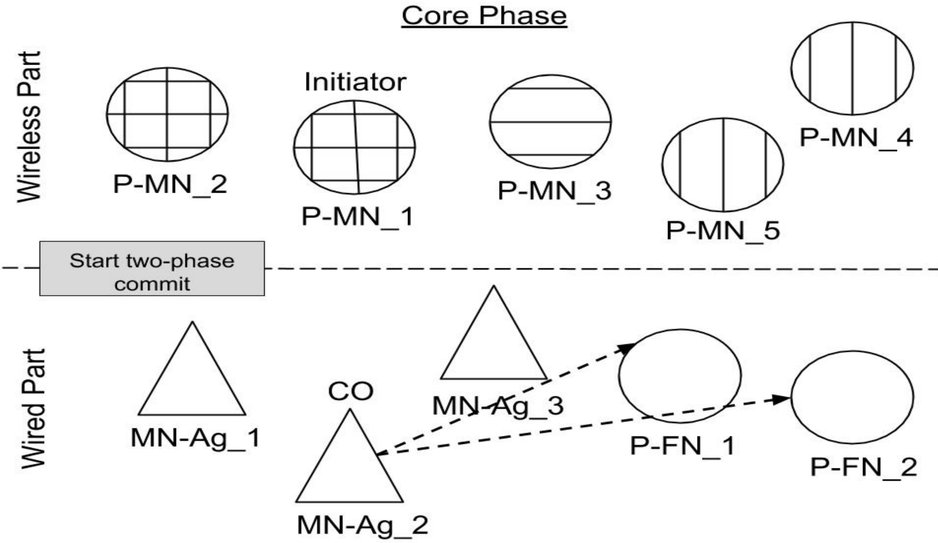
GMTC: Continued



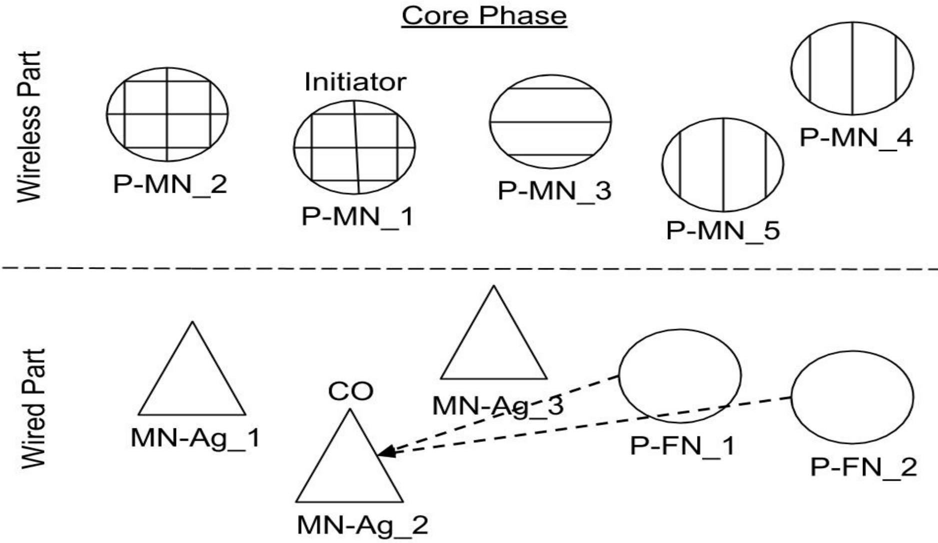
GMTC: Continued



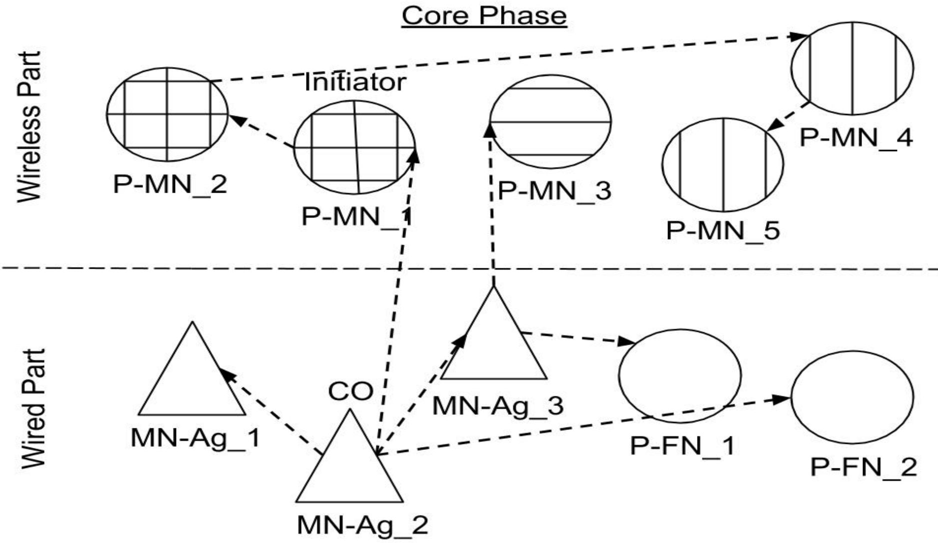
GMTC: Continued



GMTC: Continued

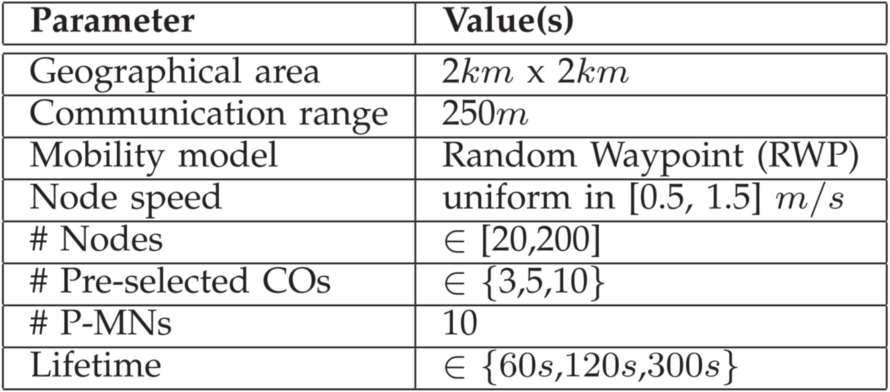


GMTC: Continued



Results

Simulation Settings



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

Table 3 [4]

Results

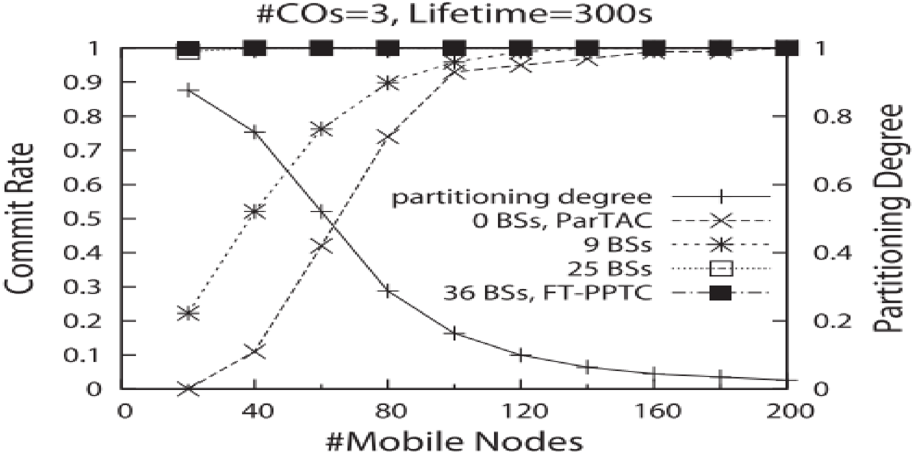


Figure 3 [4]

Results

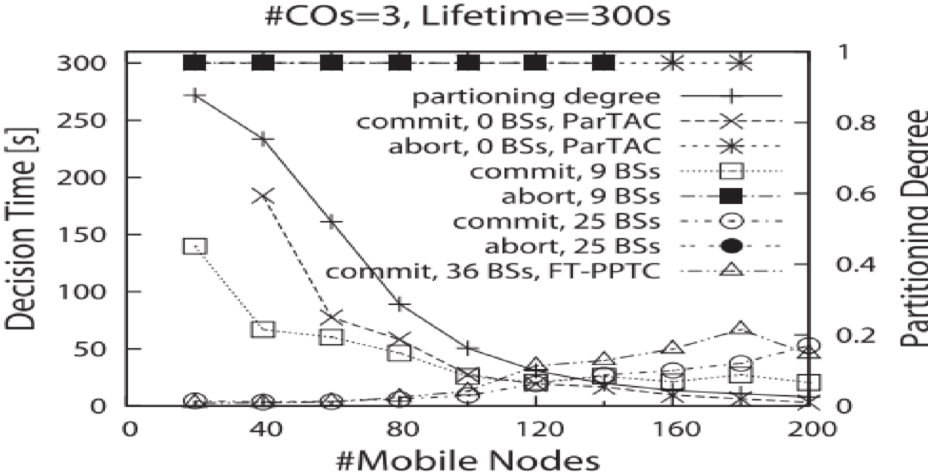


Figure 4 [4]

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.
In *2006 25th IEEE Symposium on Reliable Distributed Systems (SRDS'06)*, pages 96–105, Oct 2006.

B. Ayari, A. Khelil, and N. Suri.
Partac: A partition-tolerant atomic commit protocol for manets.
In *2010 Eleventh International Conference on Mobile Data Management*, pages 135–144, May 2010.

B. Ayari, A. Khelil, and N. Suri.
On the design of perturbation-resilient atomic commit protocols for mobile transactions.
ACM Trans. Comput. Syst., 29(3):7:1–7:36, Aug. 2011.

B. Ayari, A. Khelil, and N. Suri.
Gmtc: A generalized commit approach for hybrid mobile environments.
IEEE Transactions on Mobile Computing, 12(12):2399–2411, Dec 2013.

References II

C. Date.
An Introduction to Database Systems.
Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 8 edition, 2003.