On the Cost of Database Clusters Reconfiguration

R. Vilaça, J. Pereira ,R. Oliveira
Universidade do Minho
Portugal

J.E. Armendáriz-Iñigo, J.R. González de Mendívil
Universidad Pública de Navarra
Spain

Online Database Recovery

- Reconfiguration of a database cluster required to face load spikes or to restore the resilience of the system
- Update new replicas to the most current database state online
- Minimize time required to finish recovery
- Minimize impact on resource usage and performance of the cluster as a whole

Motivation

- Prominent practical problem
  - Often addressed before (eg., [Kemme et al. , 2001], [Jiménez-Peris et al. , 2002], [Armendáriz-Íñigo et al. , 2007]
  - Applied to consistent database replication
  - Existing refined techniques to improve performance
- Existing work
  - Assumes simplified models without taking into account system limitations such as I/O and CPU
  - Does not provide a detailed evaluation under representative workload scenarios

Goals

- Combine the proposed techniques in a single protocol
- Systematically benchmark different reconfiguration scenarios
- Assess each technique's performance impact and overhead on the clustered database service
- Determine fundamental limits to cluster reconfiguration
- Discuss the relative merits of each approach

Recovery Protocol

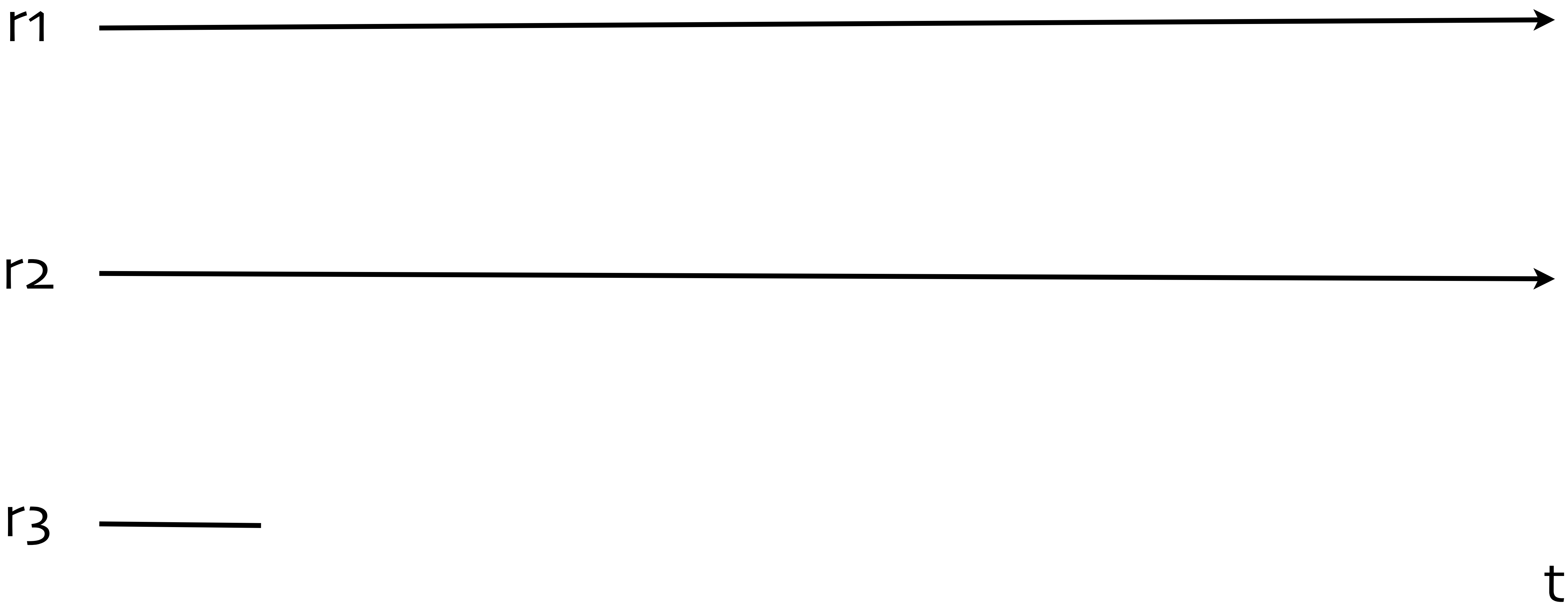
- Based on the algorithm presented in [Kemme et al., 2001]
- Parallel Recovery [Jiménez-Peris et al., 2002]
- Convergence Phases [Armendáriz-Íñigo et al. , 2007]
- Includes several obvious optimizations, eg:
  - purging of redundant data changes
  - data compression
  - parallel and batch applier for received recovery data at the recovering replica.
- Configurable: number of donors for Parallel Recovery and number of Convergence Phases

Recovery Protocol

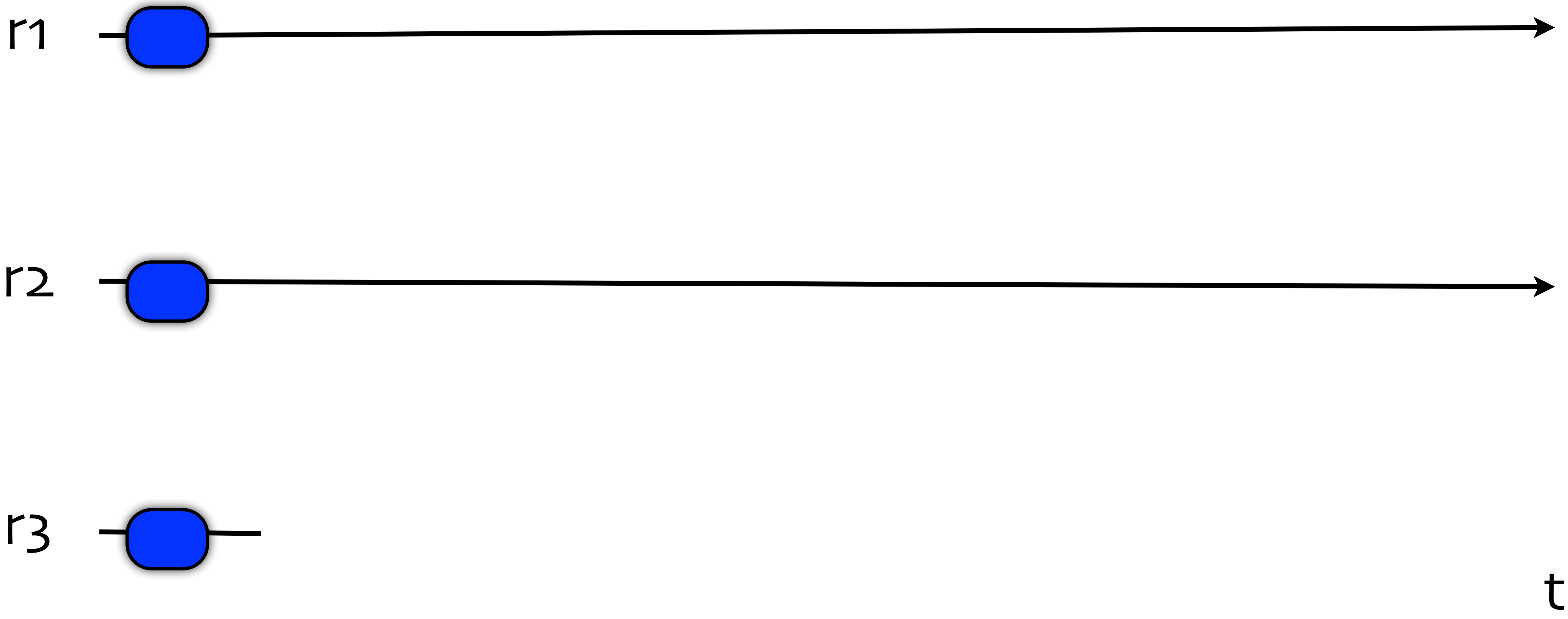
- Full Transfer
- Missed Updates or Delta Transfer
    - Parallel Recovery
    - Convergence Phases

Full Transfer

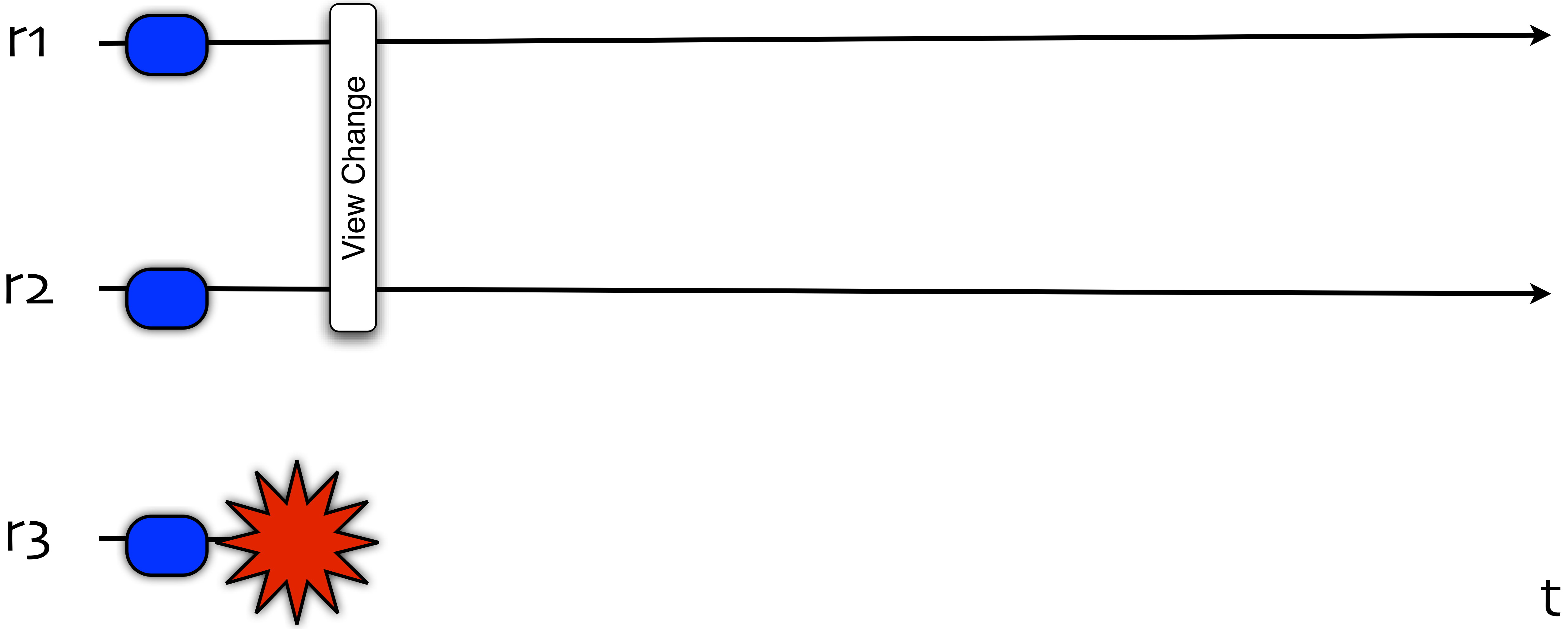
Full Transfer



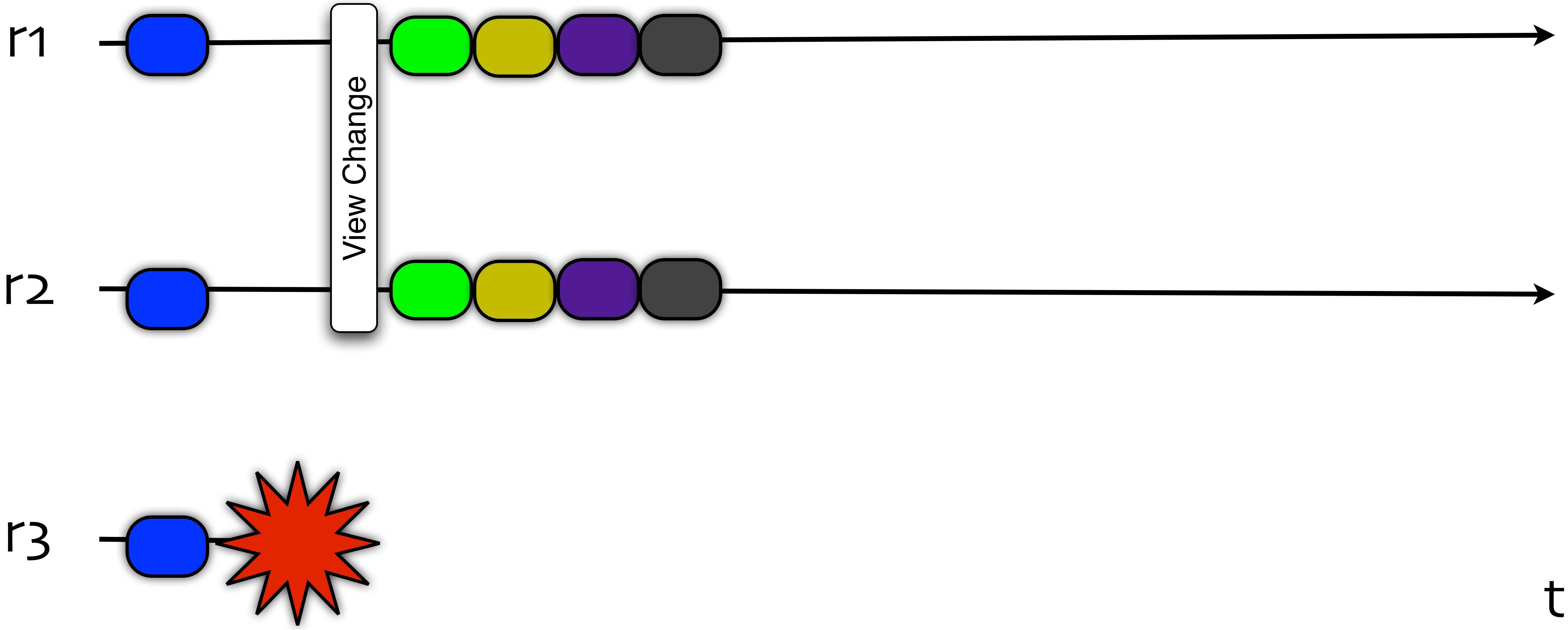
Full Transfer



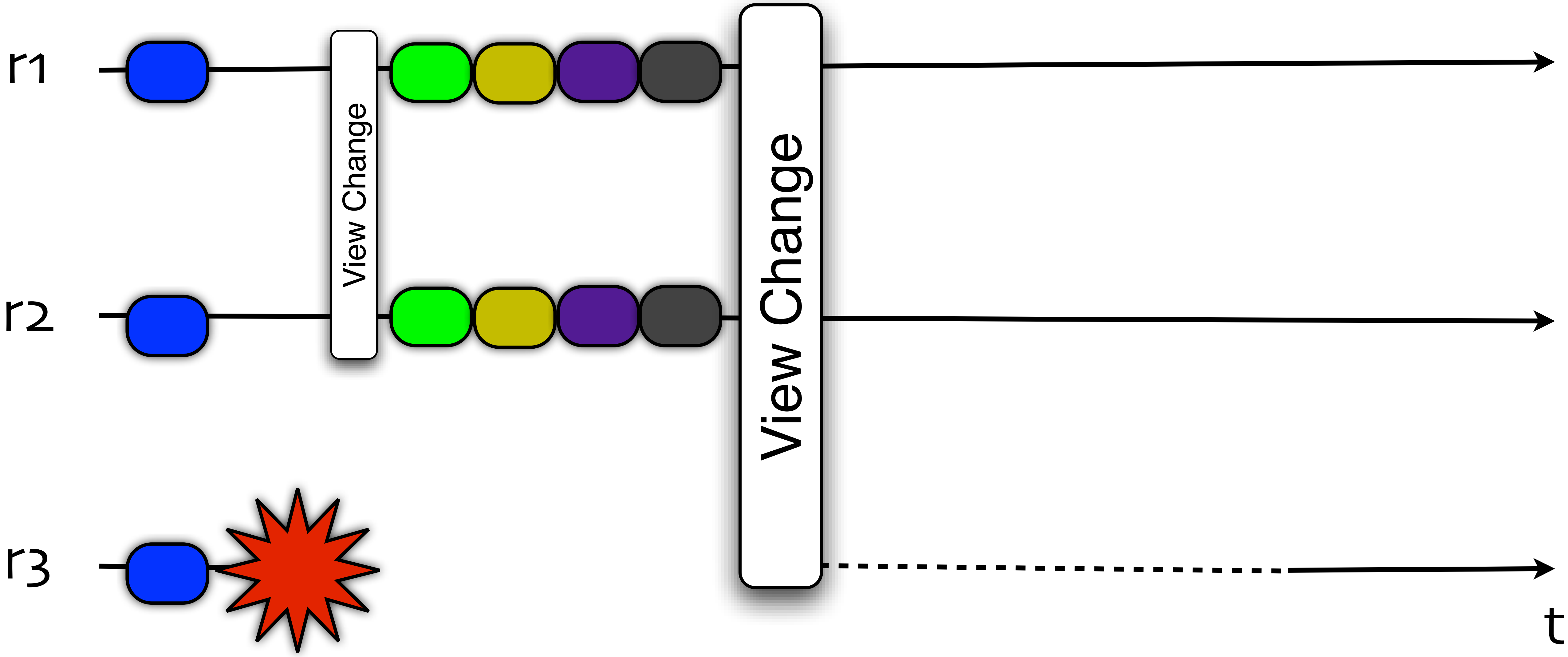
Full Transfer



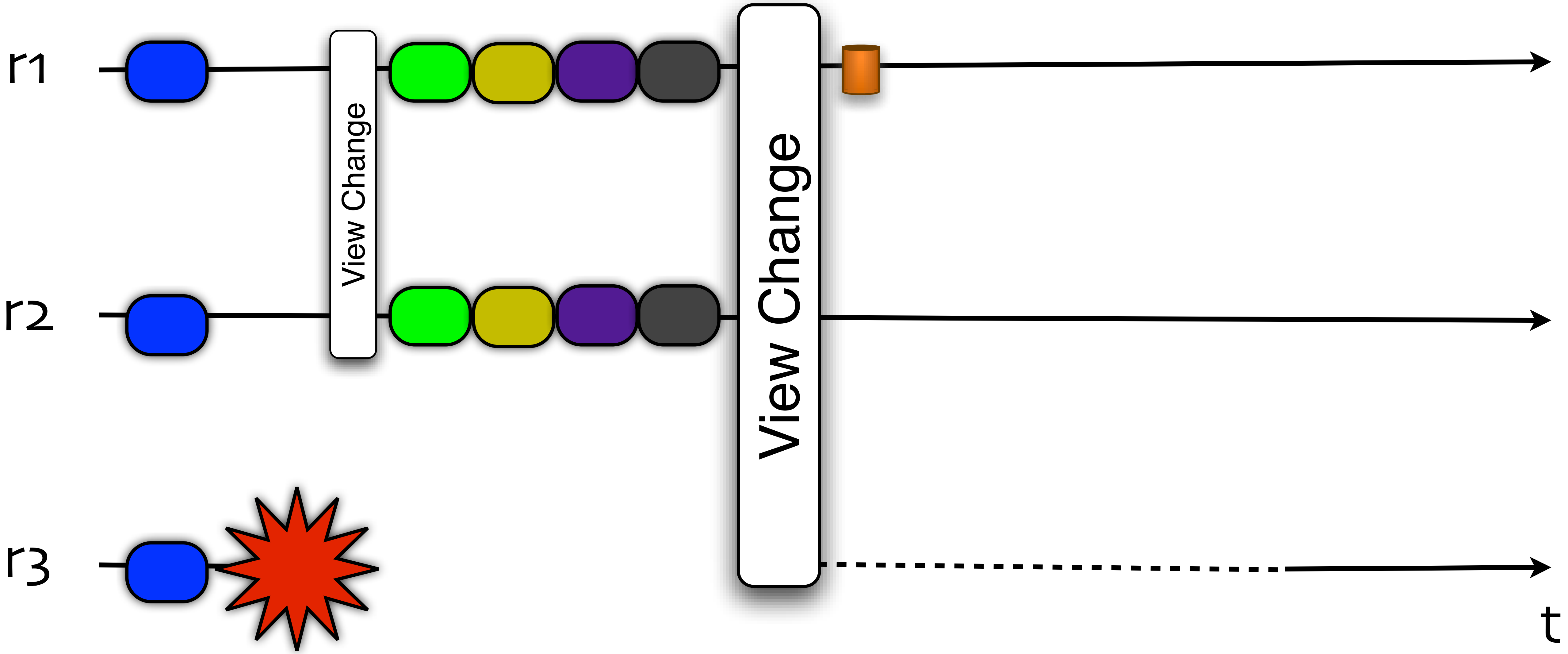
Full Transfer



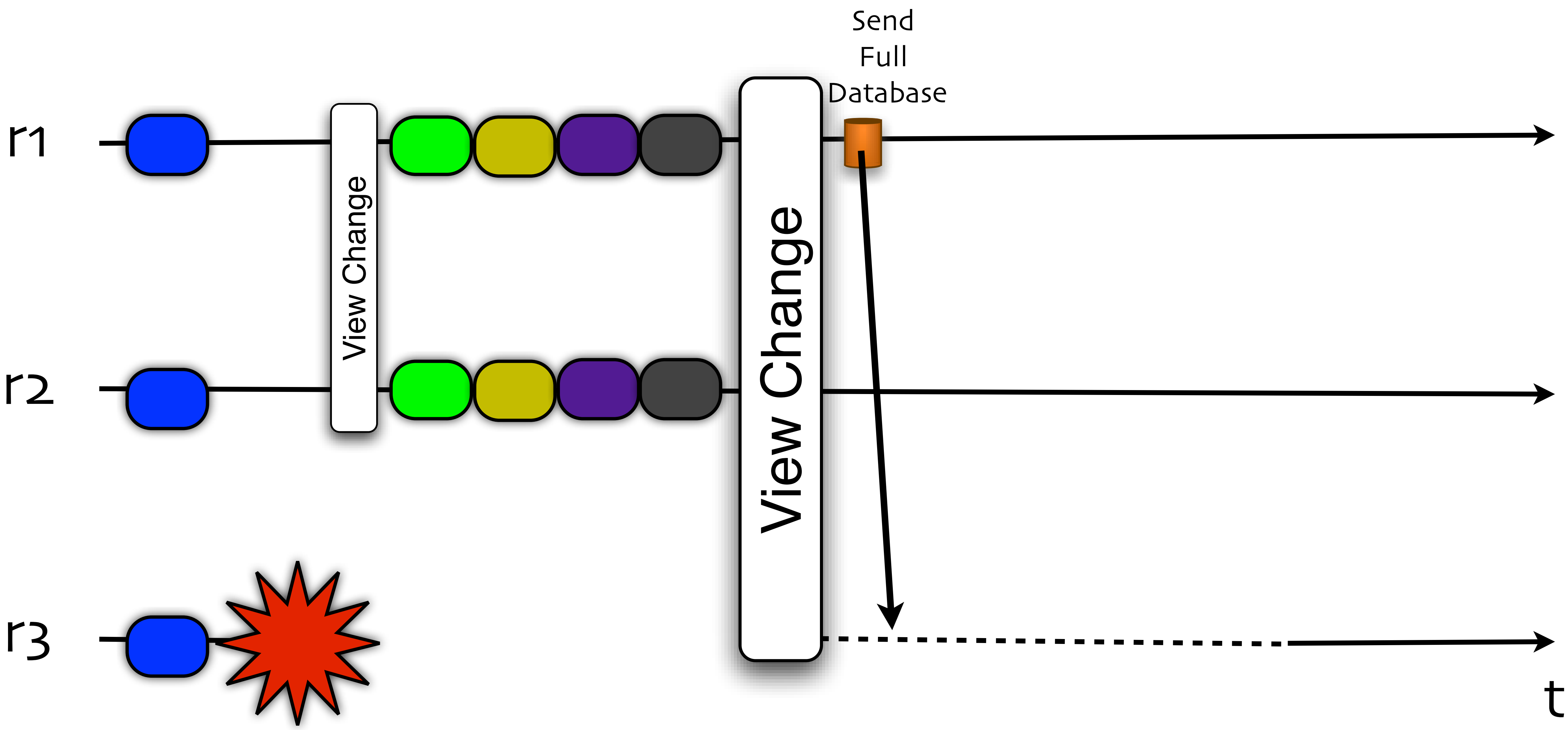
Full Transfer



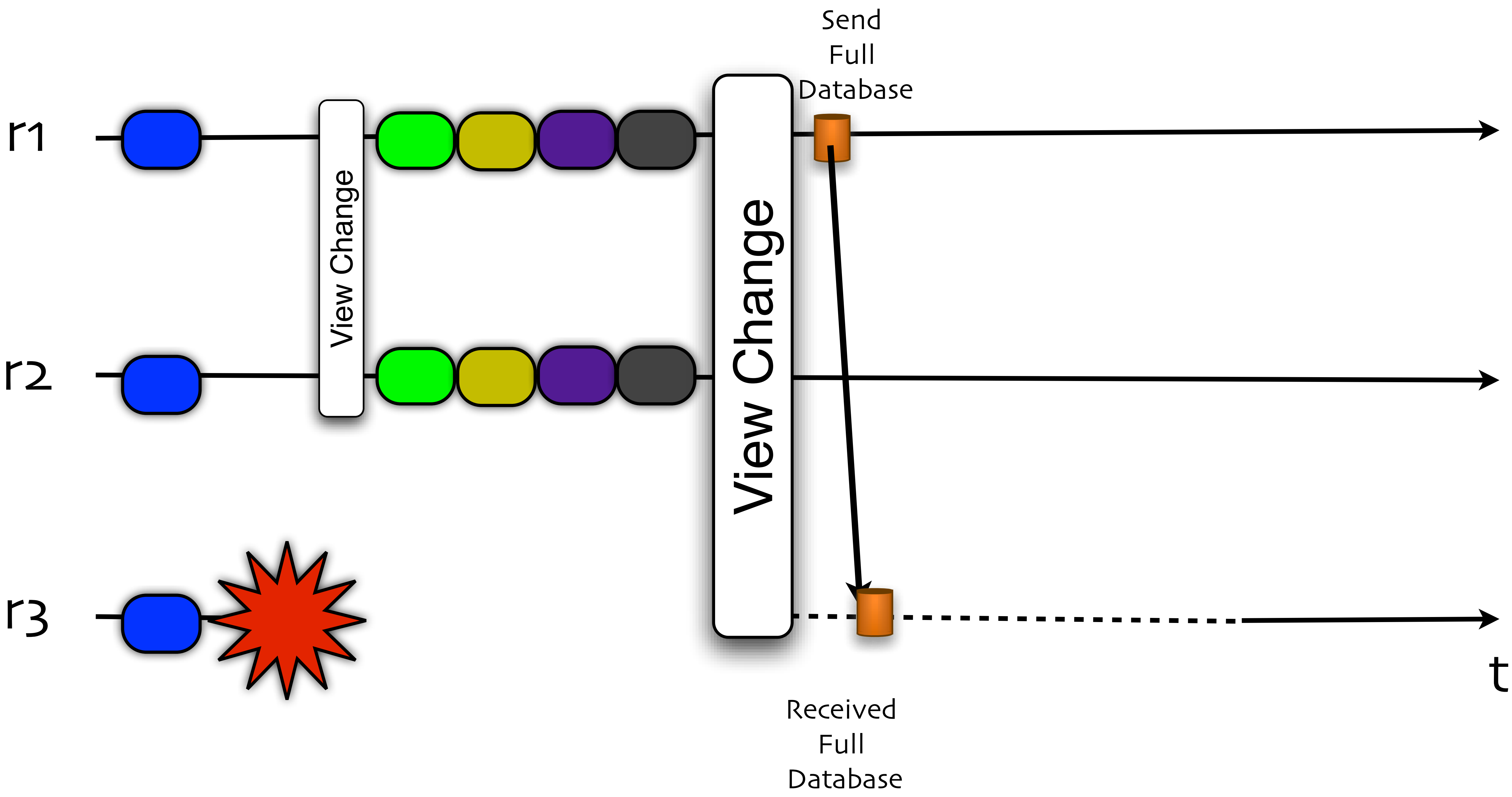
Full Transfer



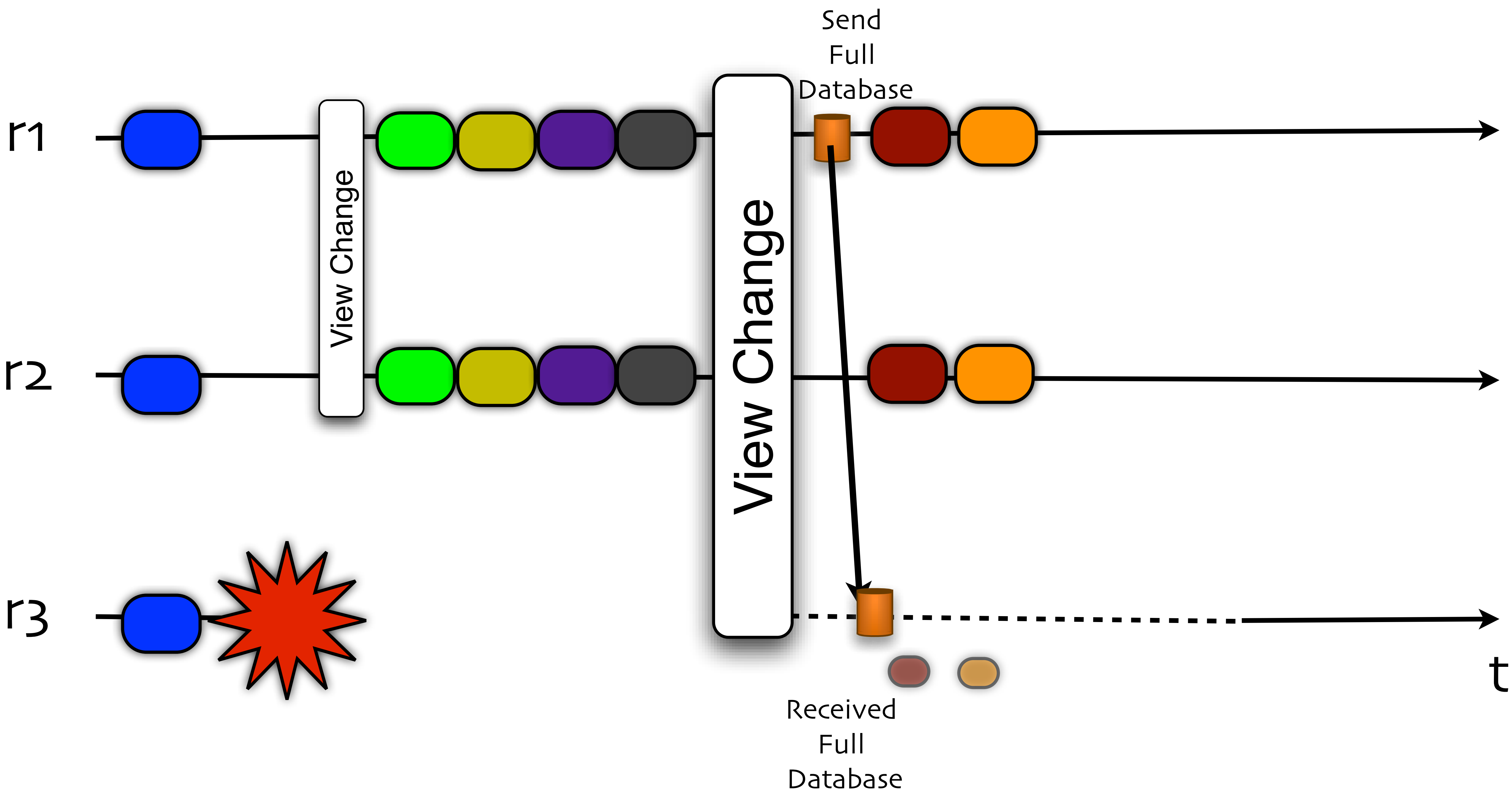
Full Transfer



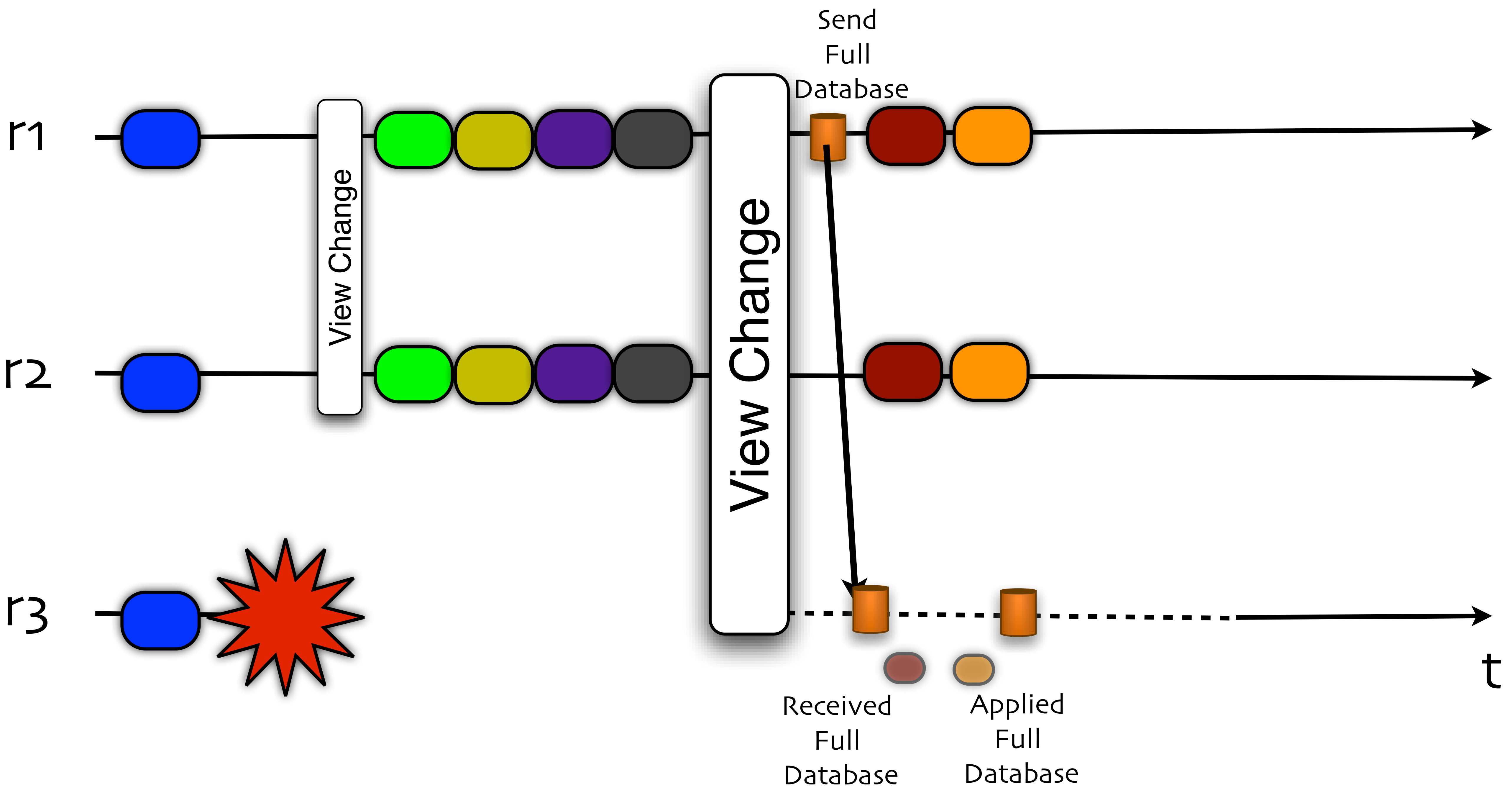
Full Transfer



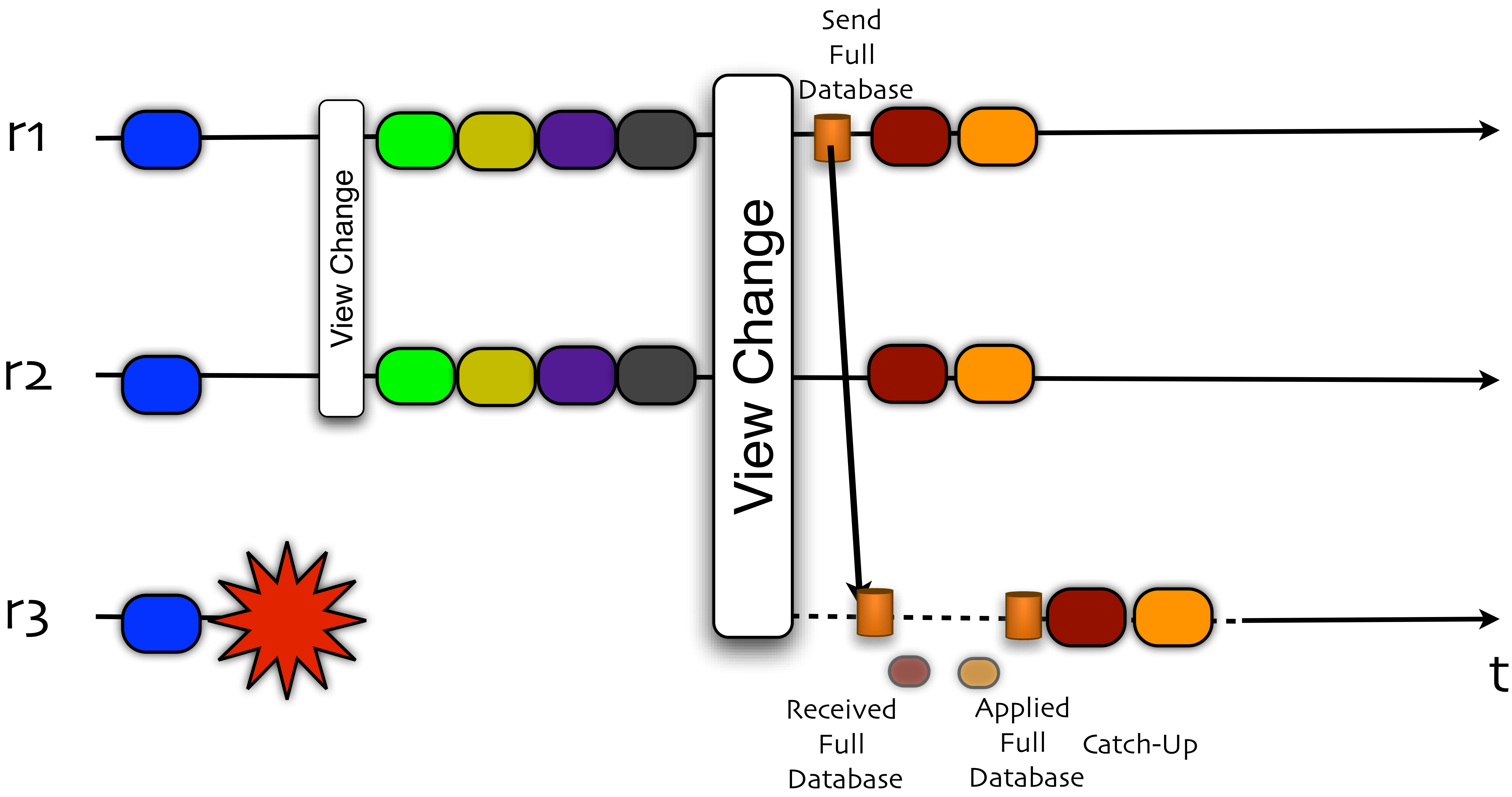
Full Transfer



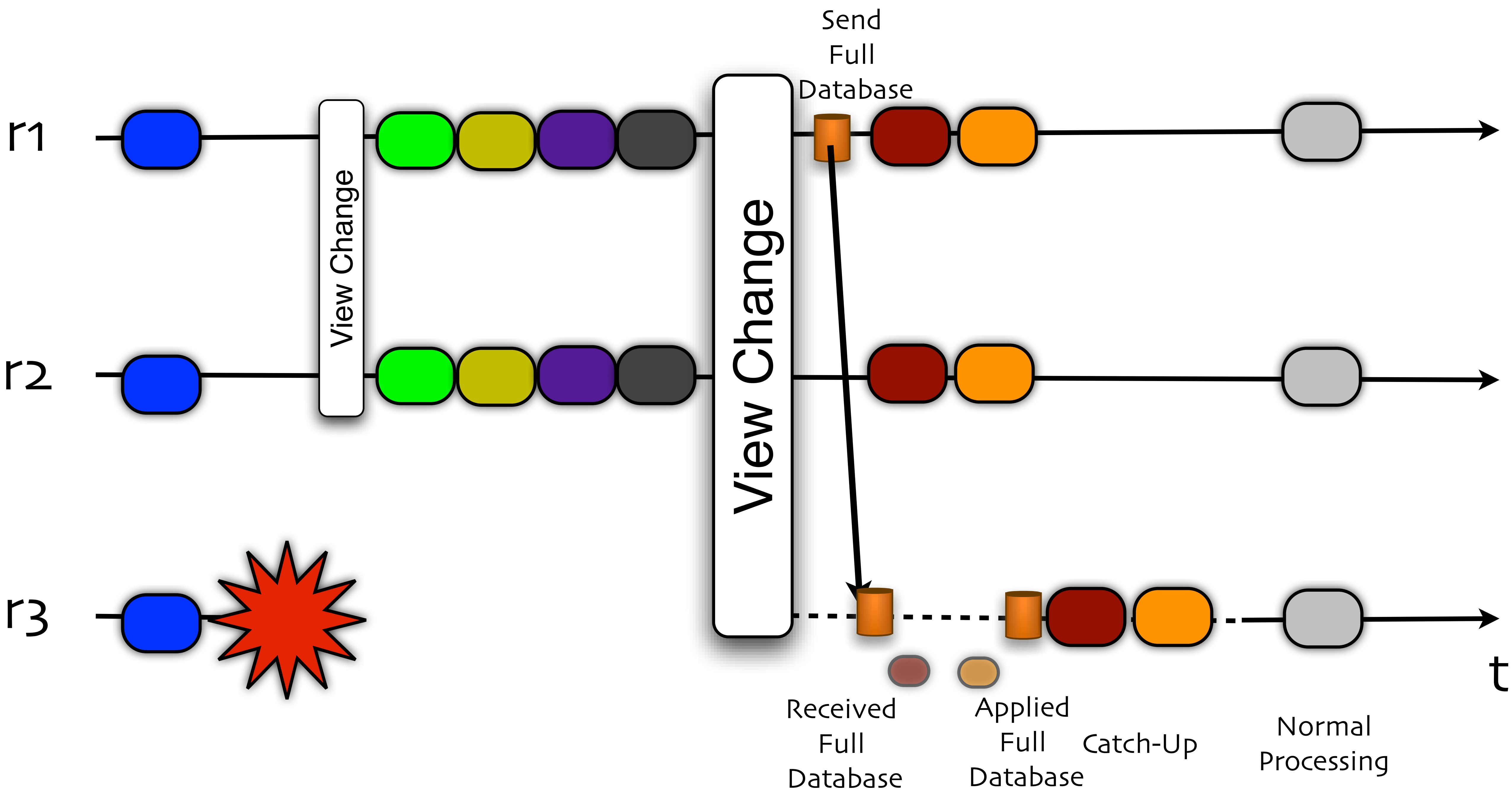
Full Transfer



Full Transfer

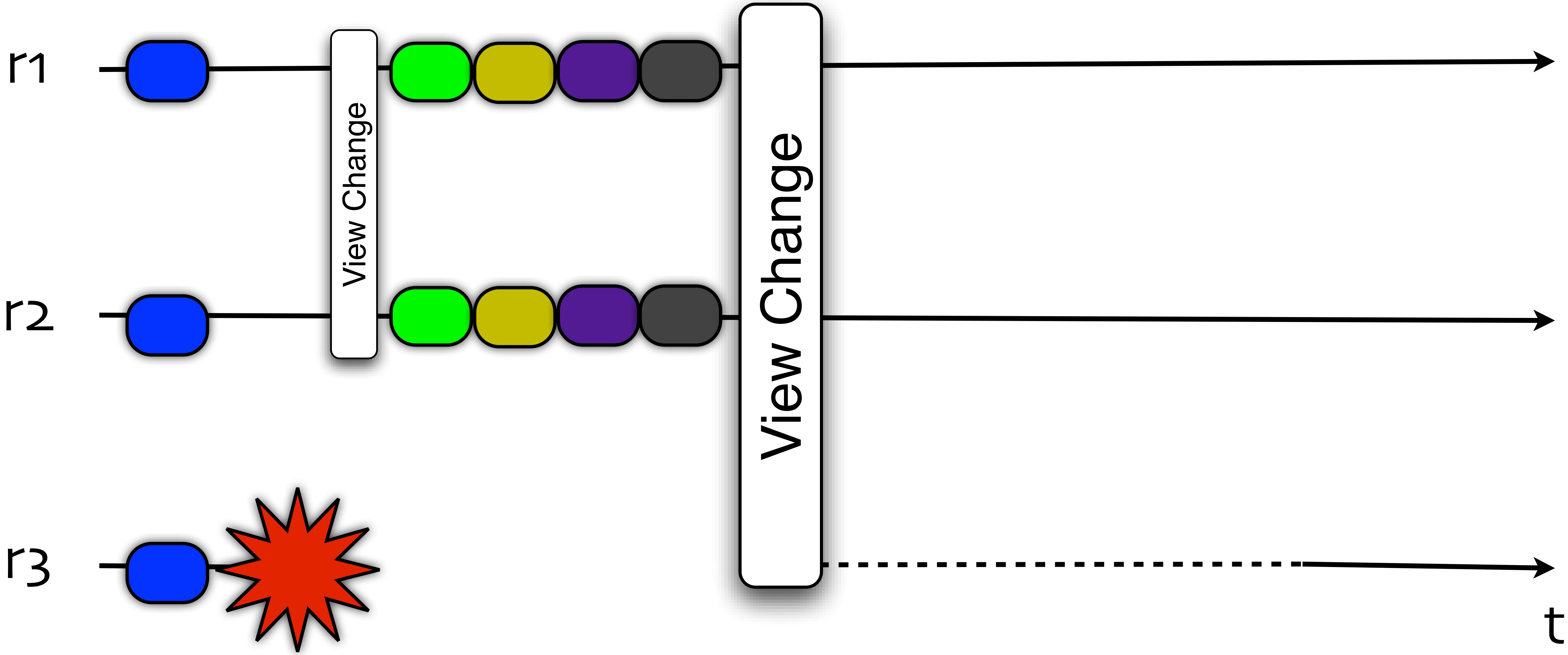


Full Transfer

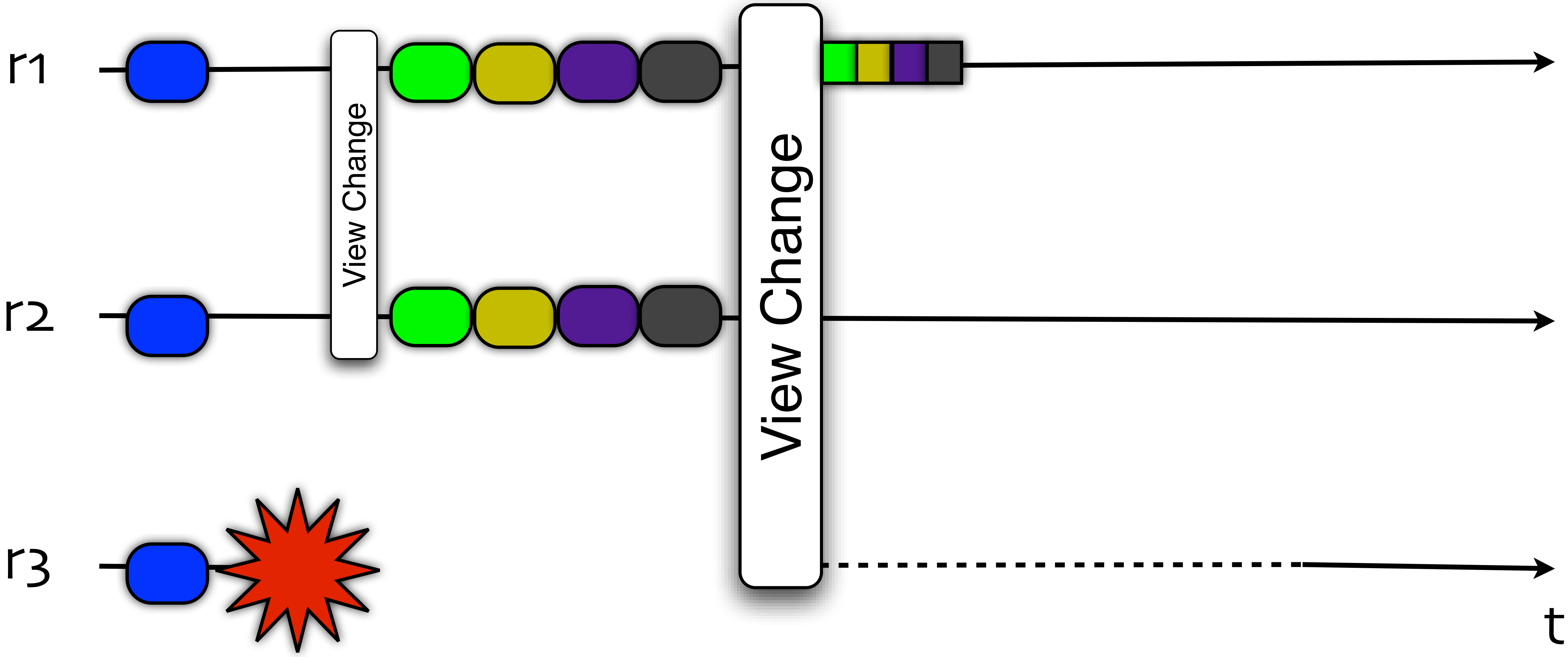


Missed Updates

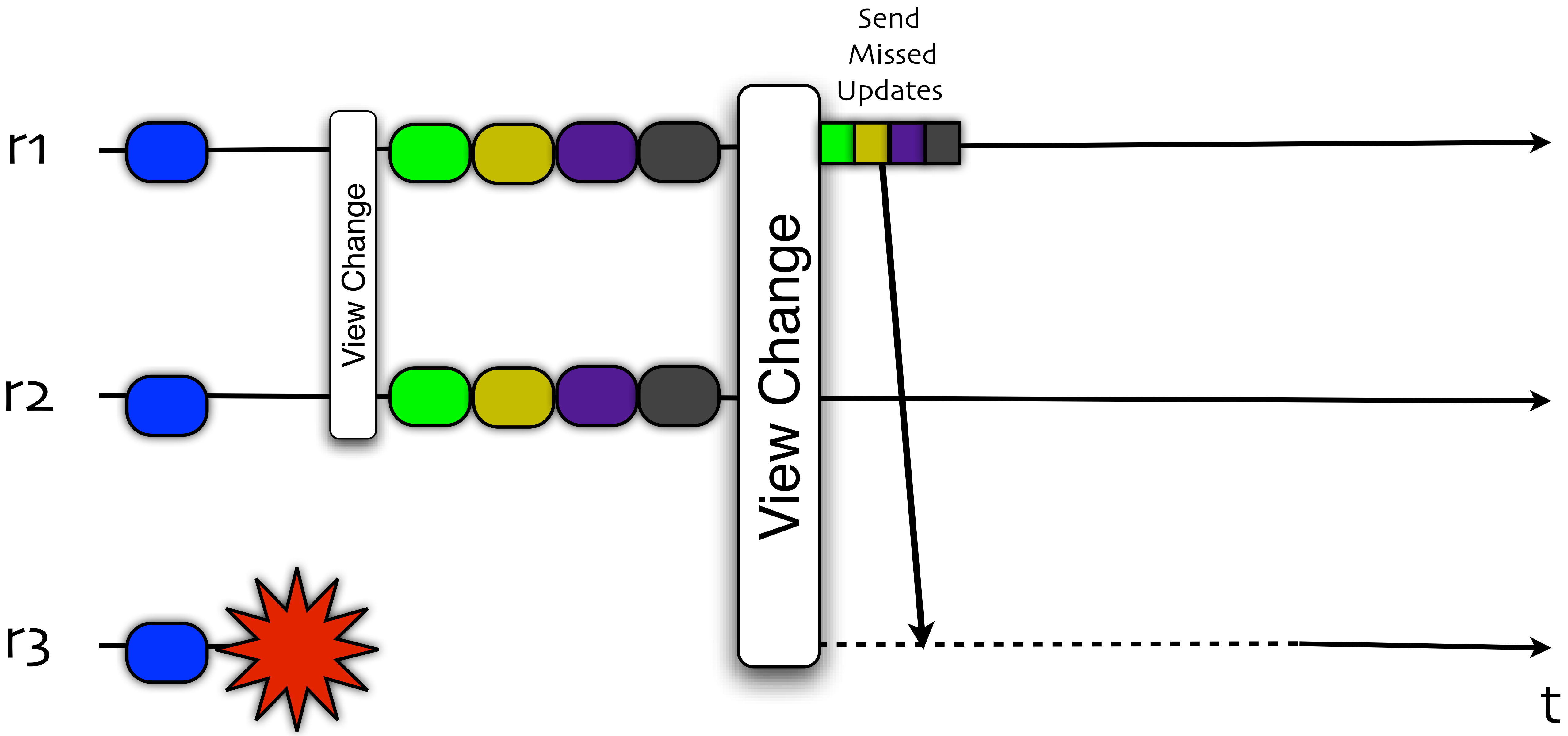
Missed Updates



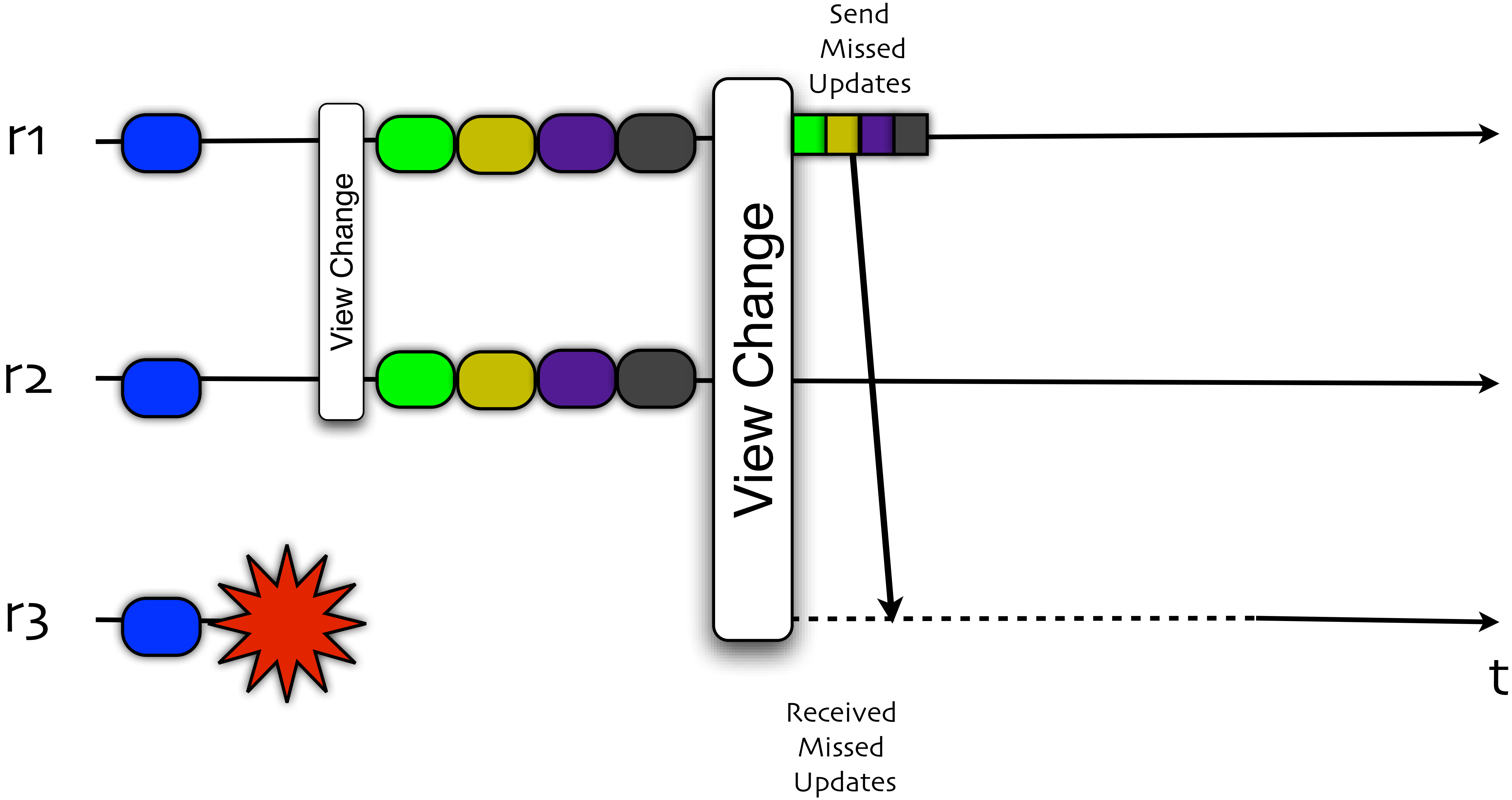
Missed Updates



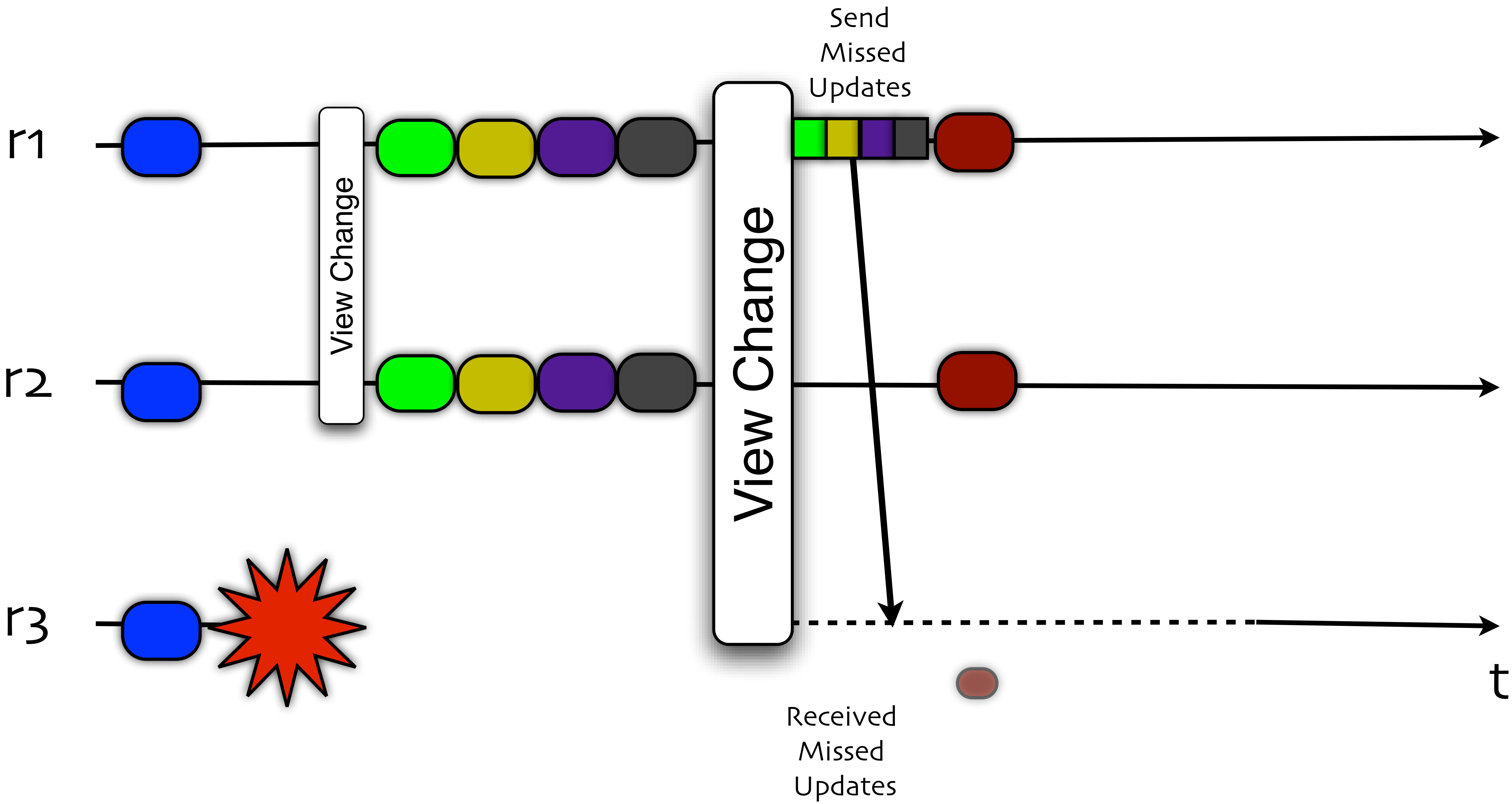
Missed Updates



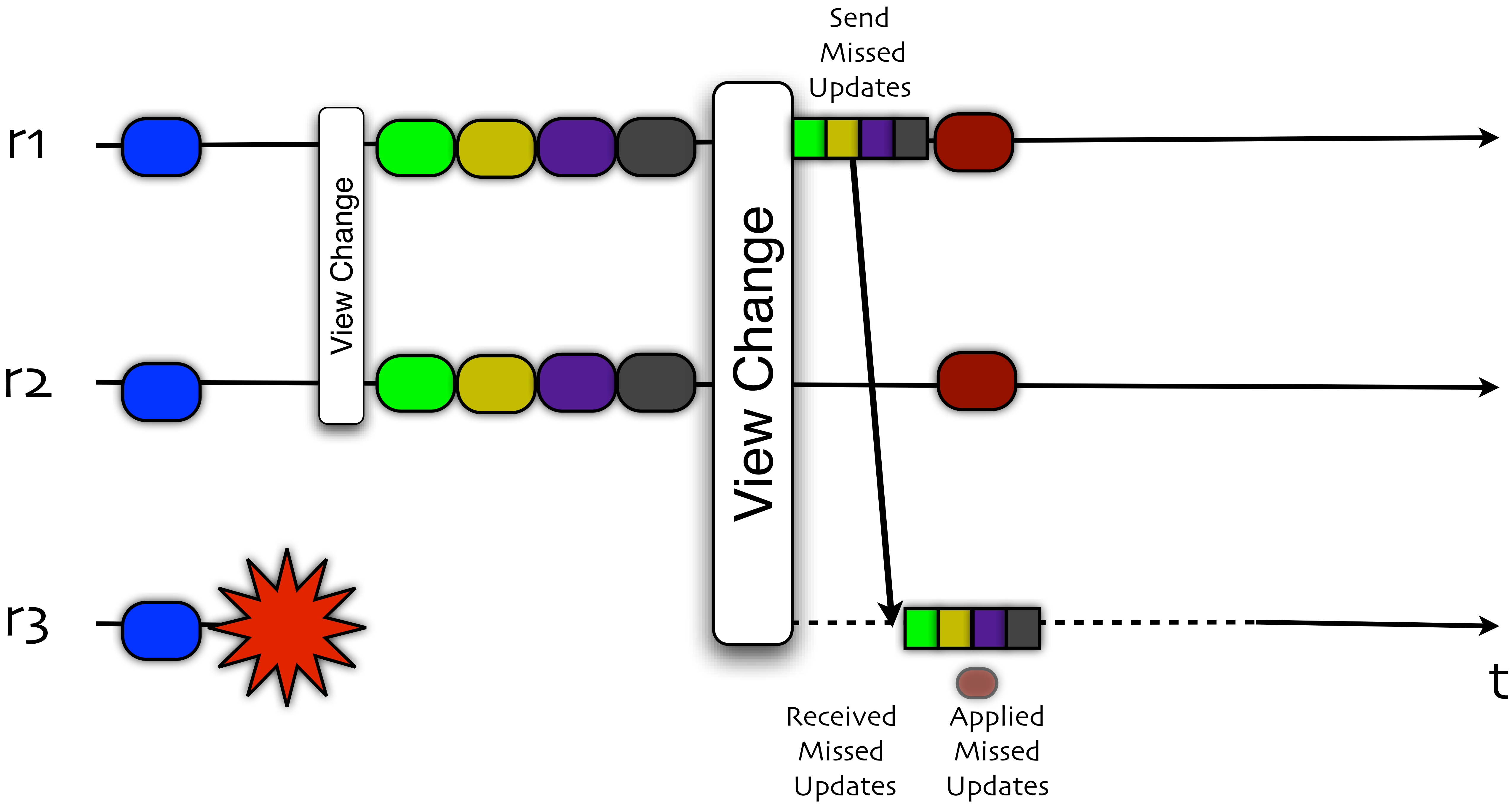
Missed Updates



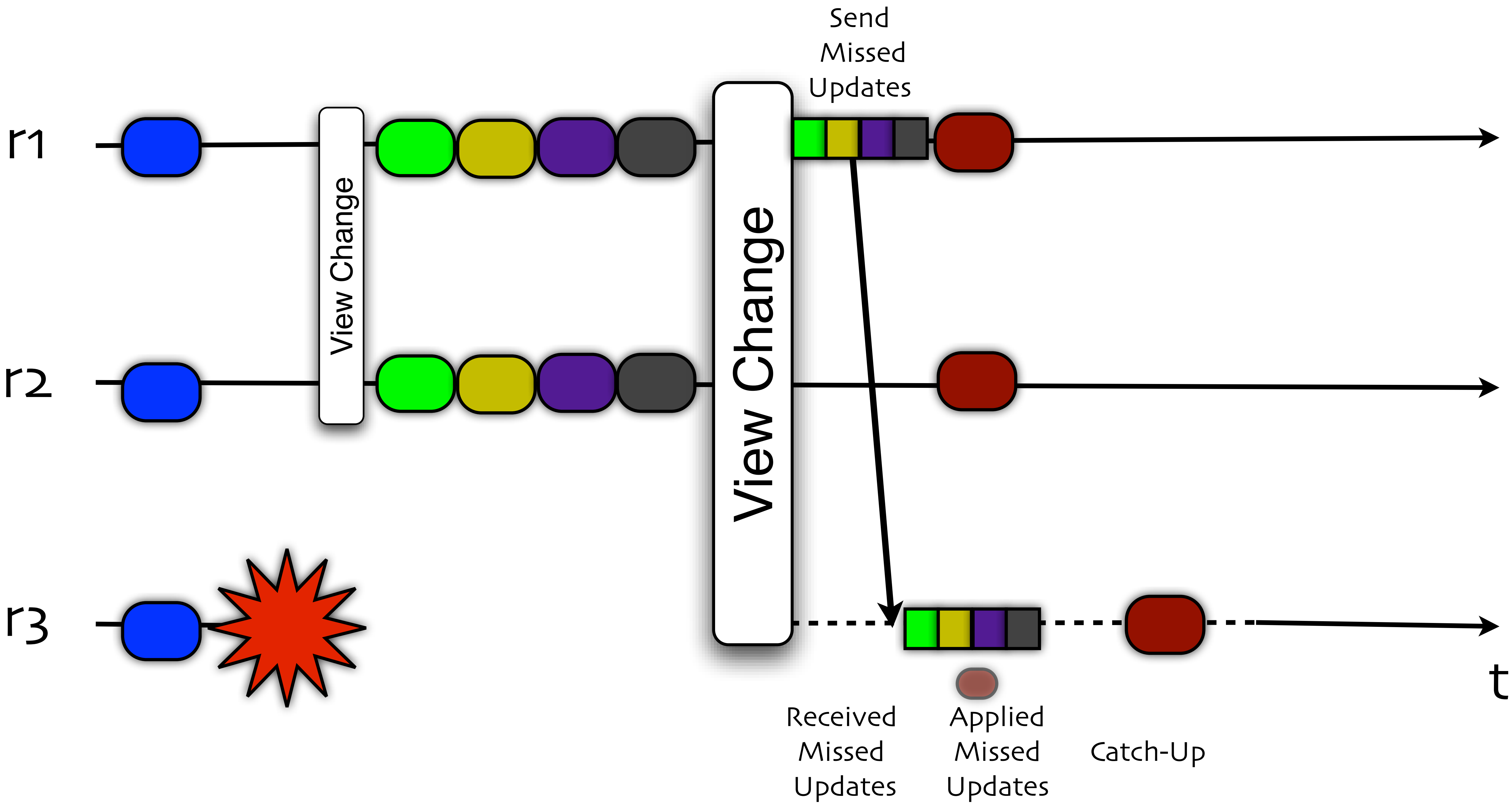
Missed Updates



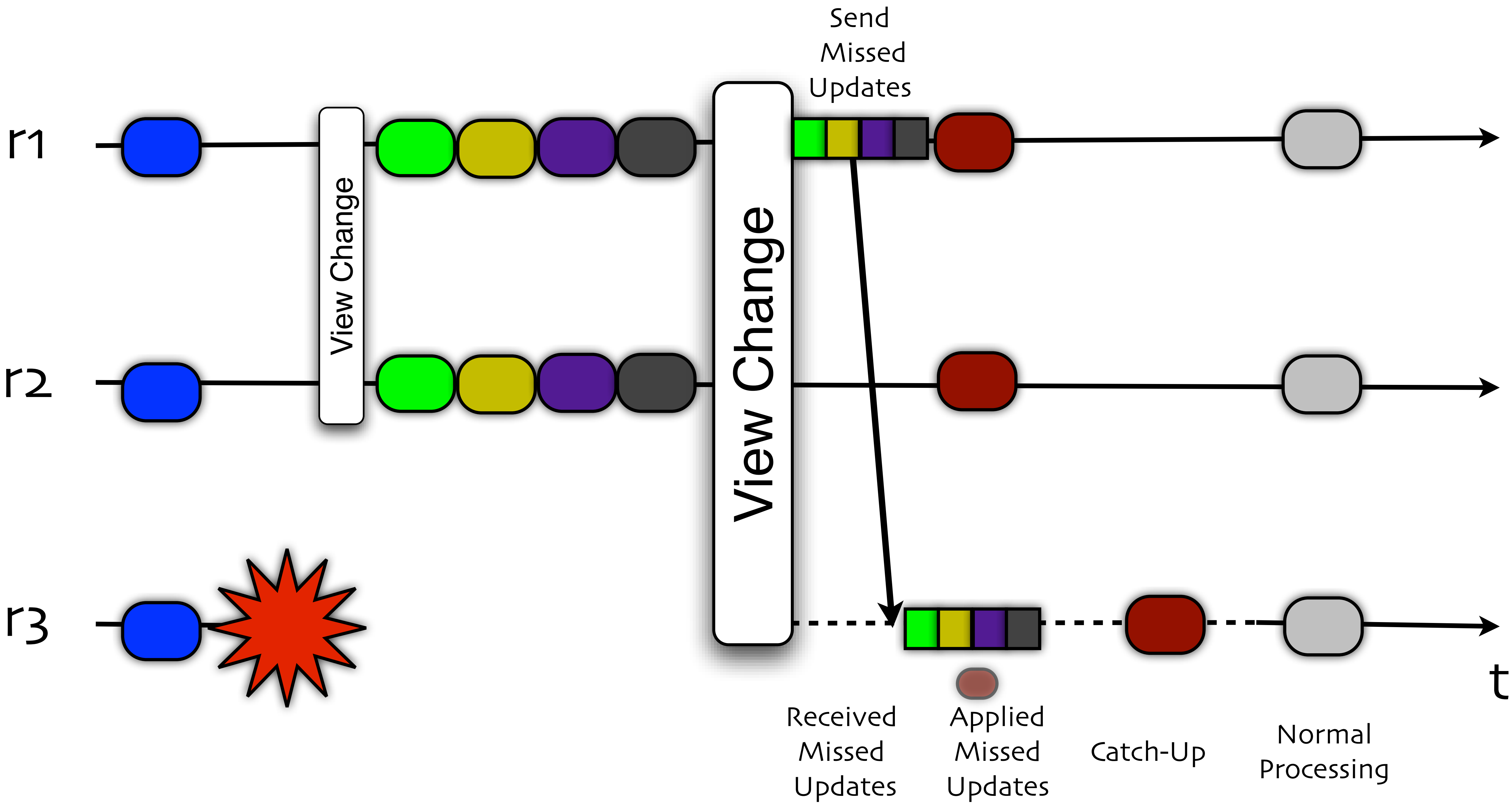
Missed Updates



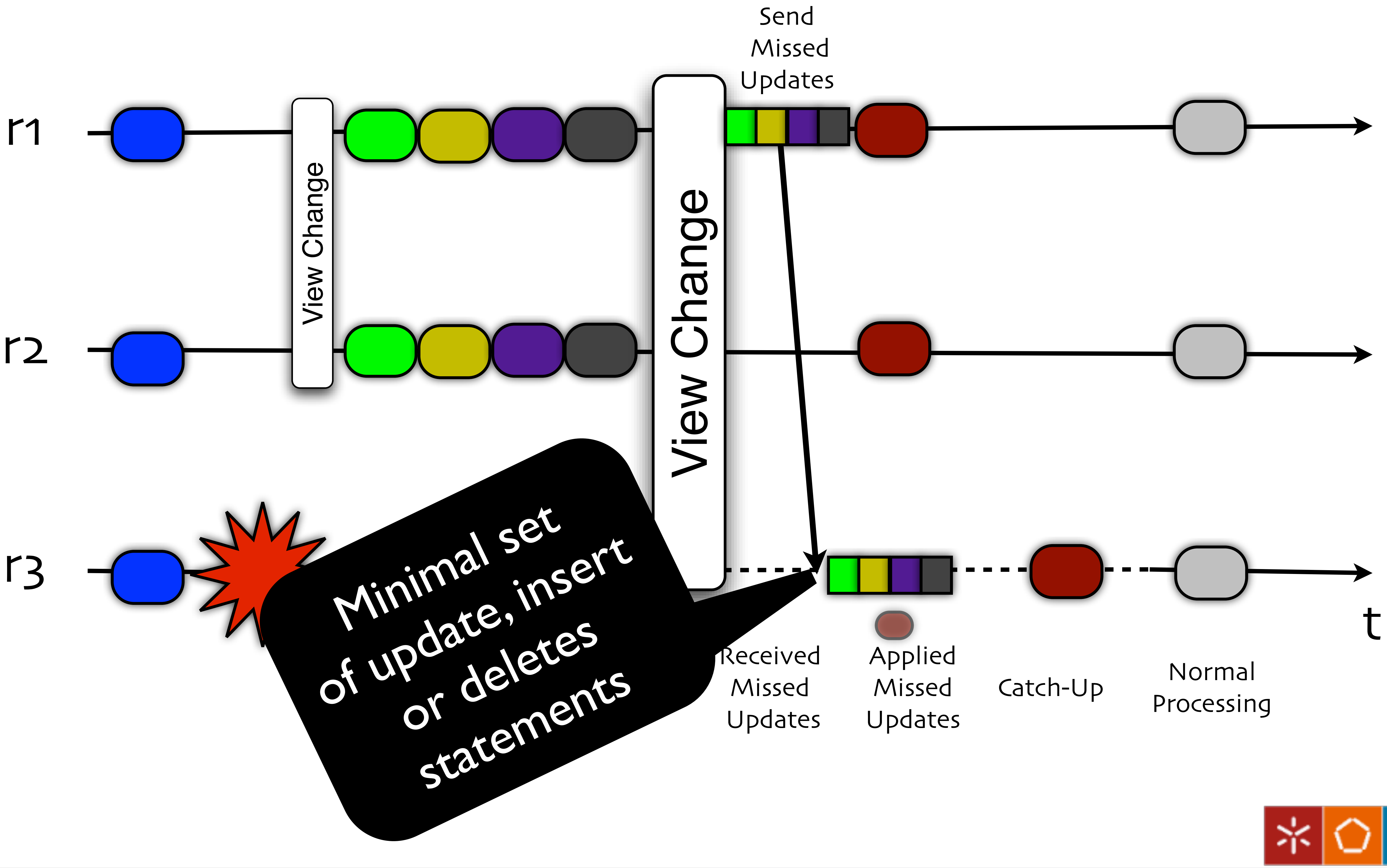
Missed Updates



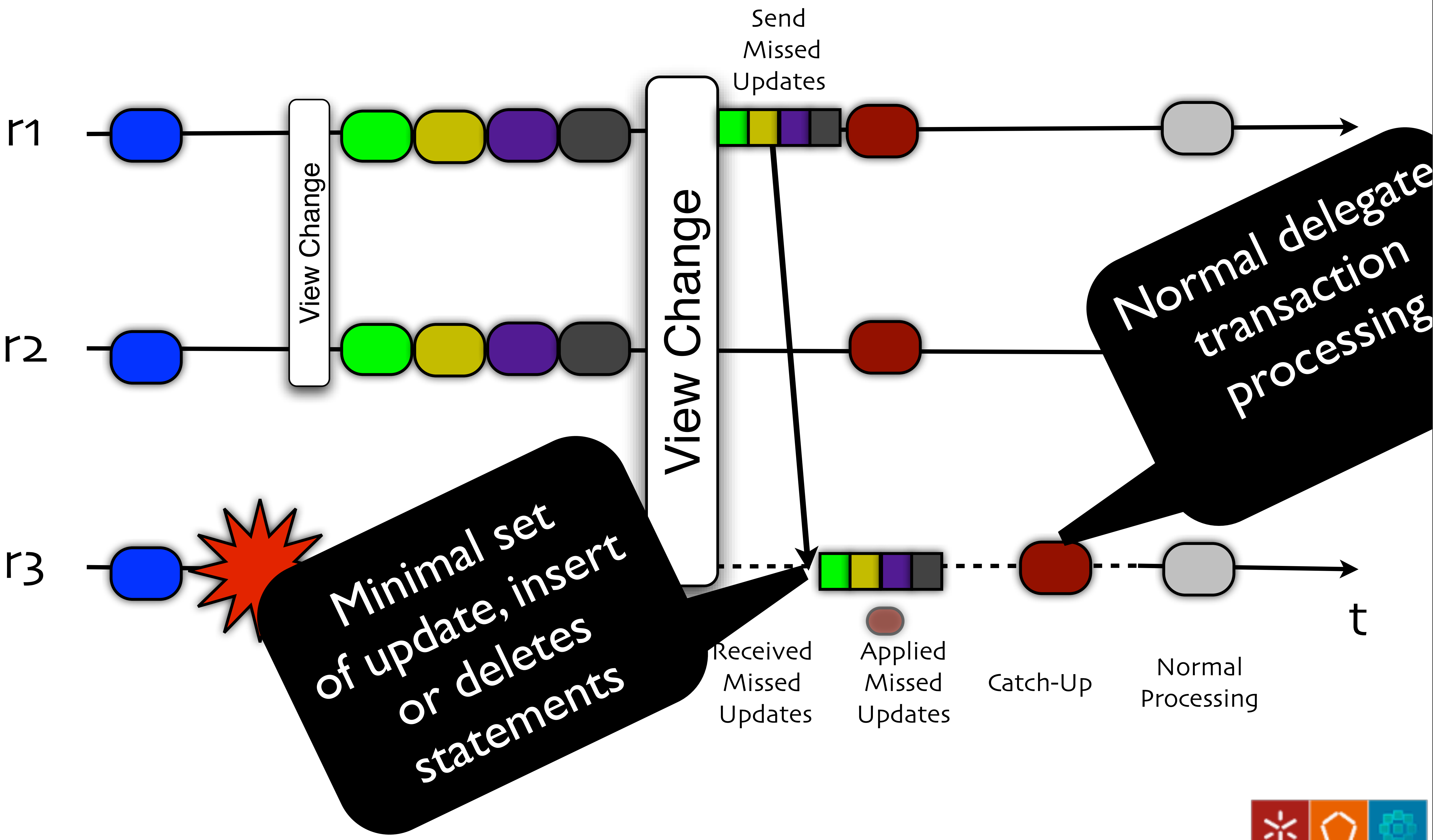
Missed Updates



Missed Updates



Missed Updates

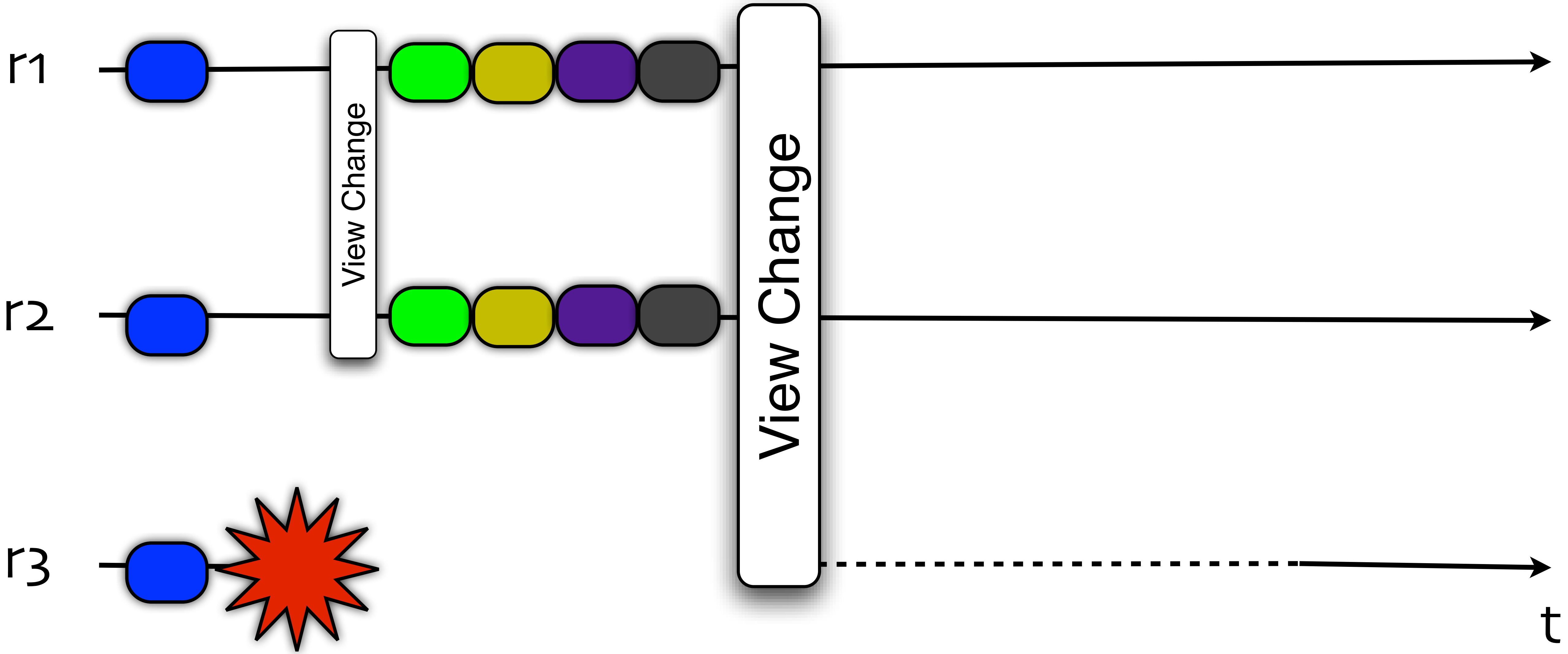


Normal delegate transaction processing

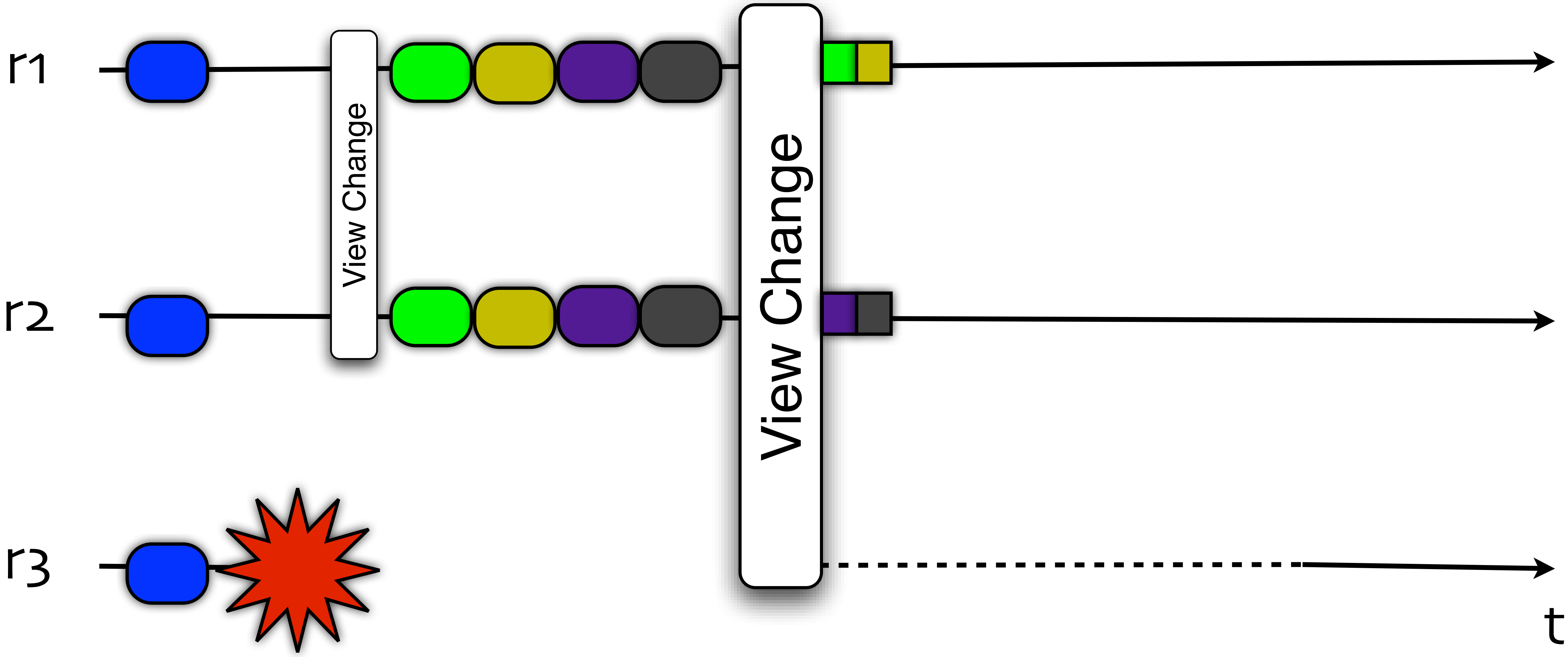
Minimal set of update, insert or deletes statements

Parallel Recovery

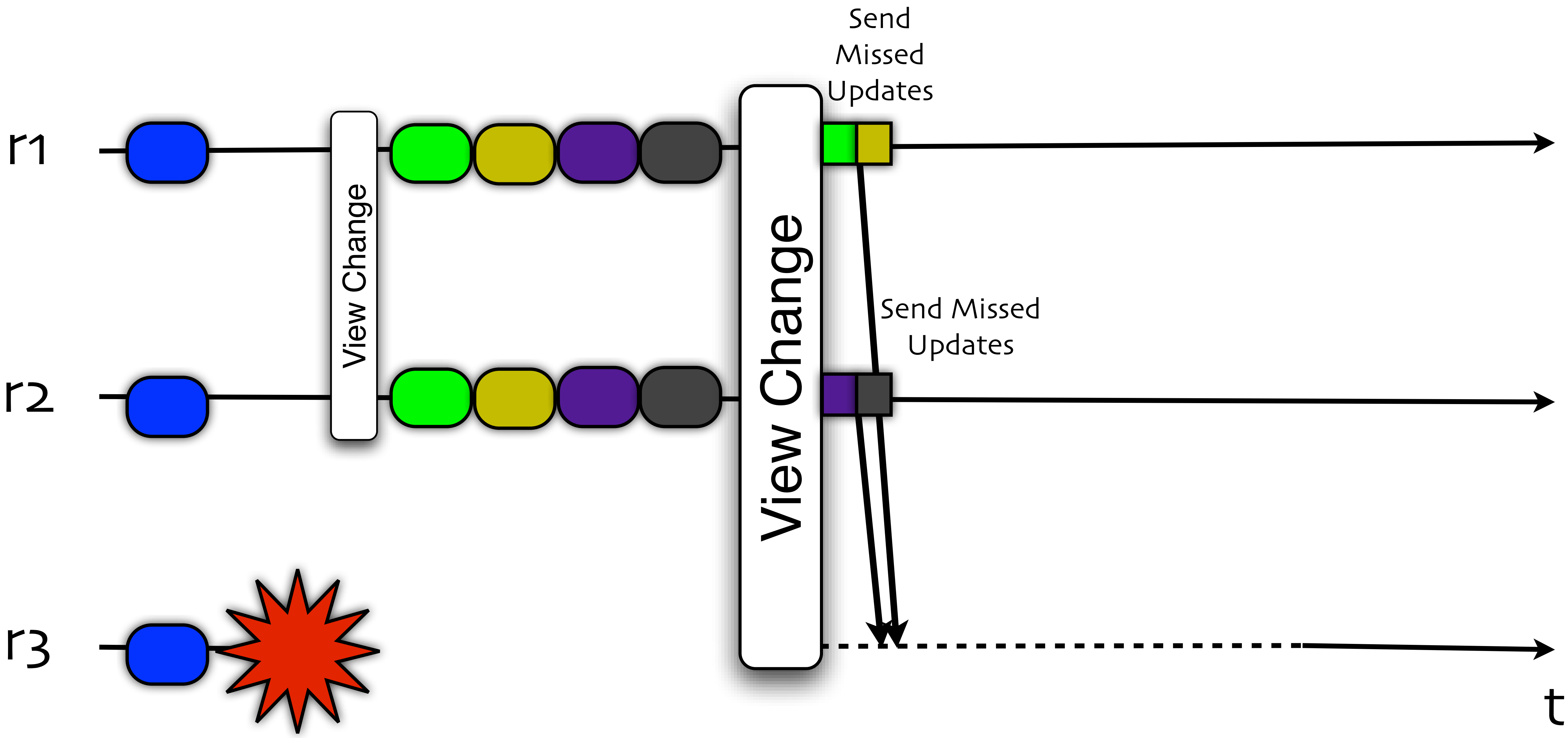
Parallel Recovery



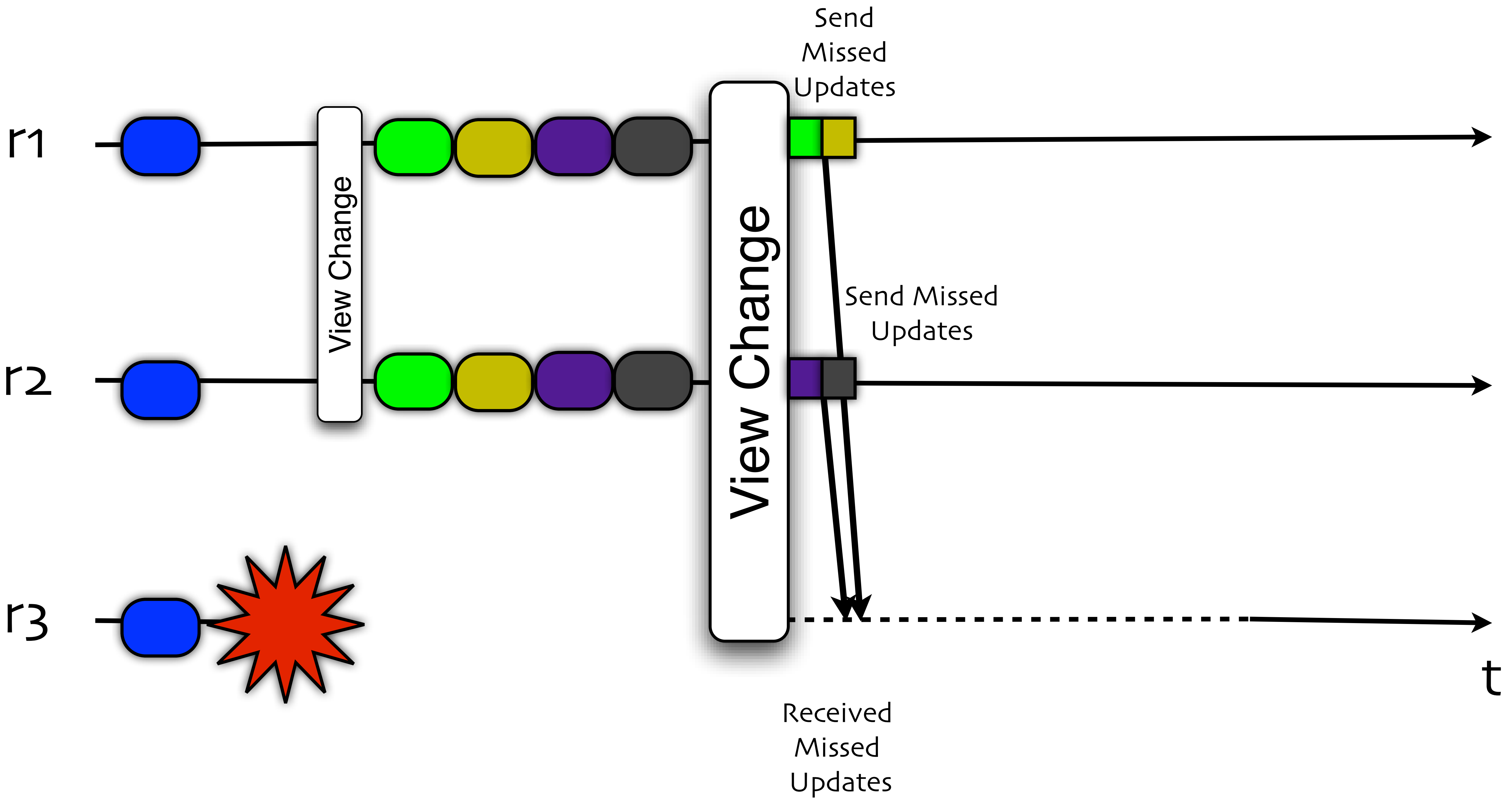
Parallel Recovery



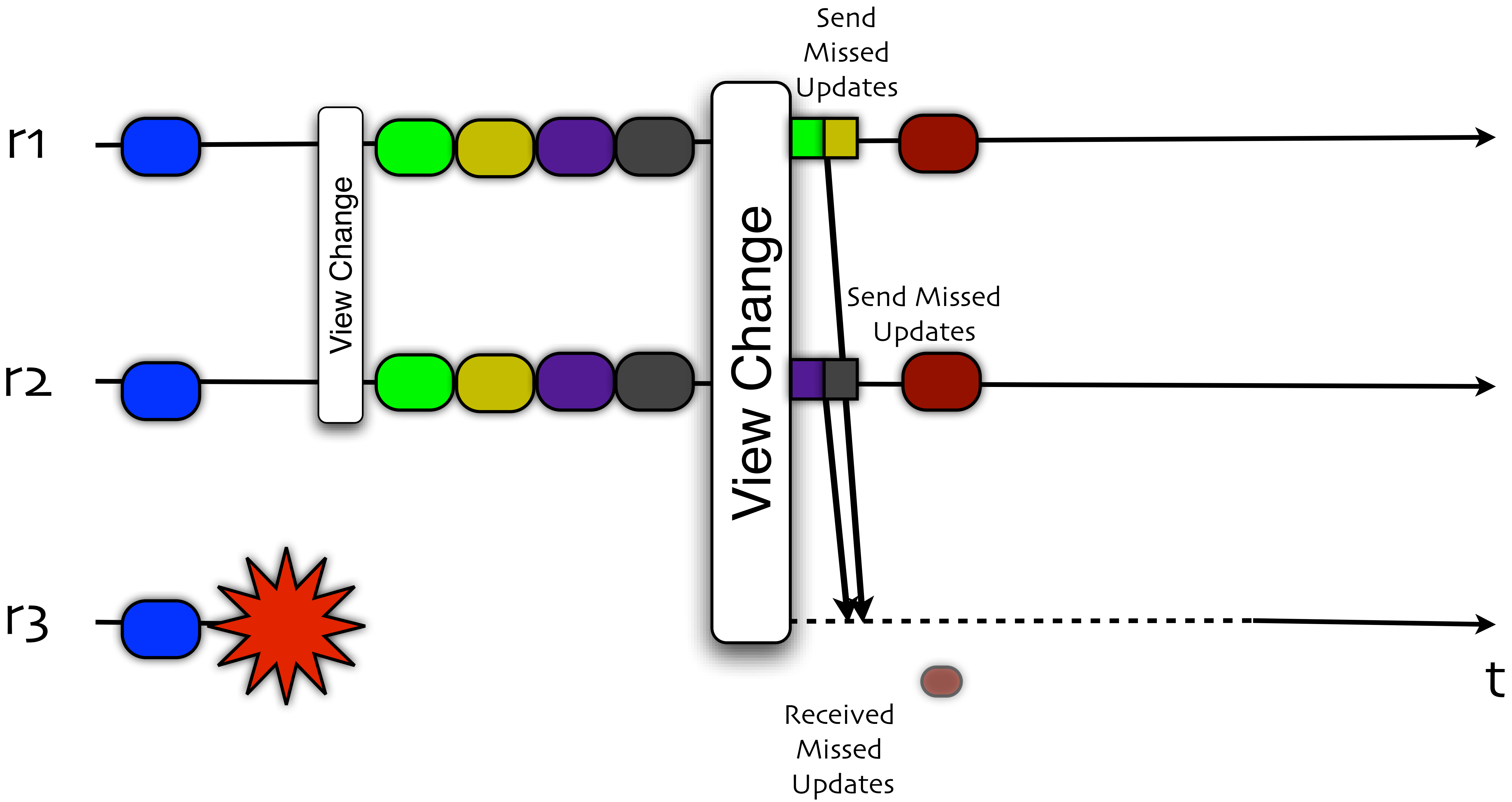
Parallel Recovery



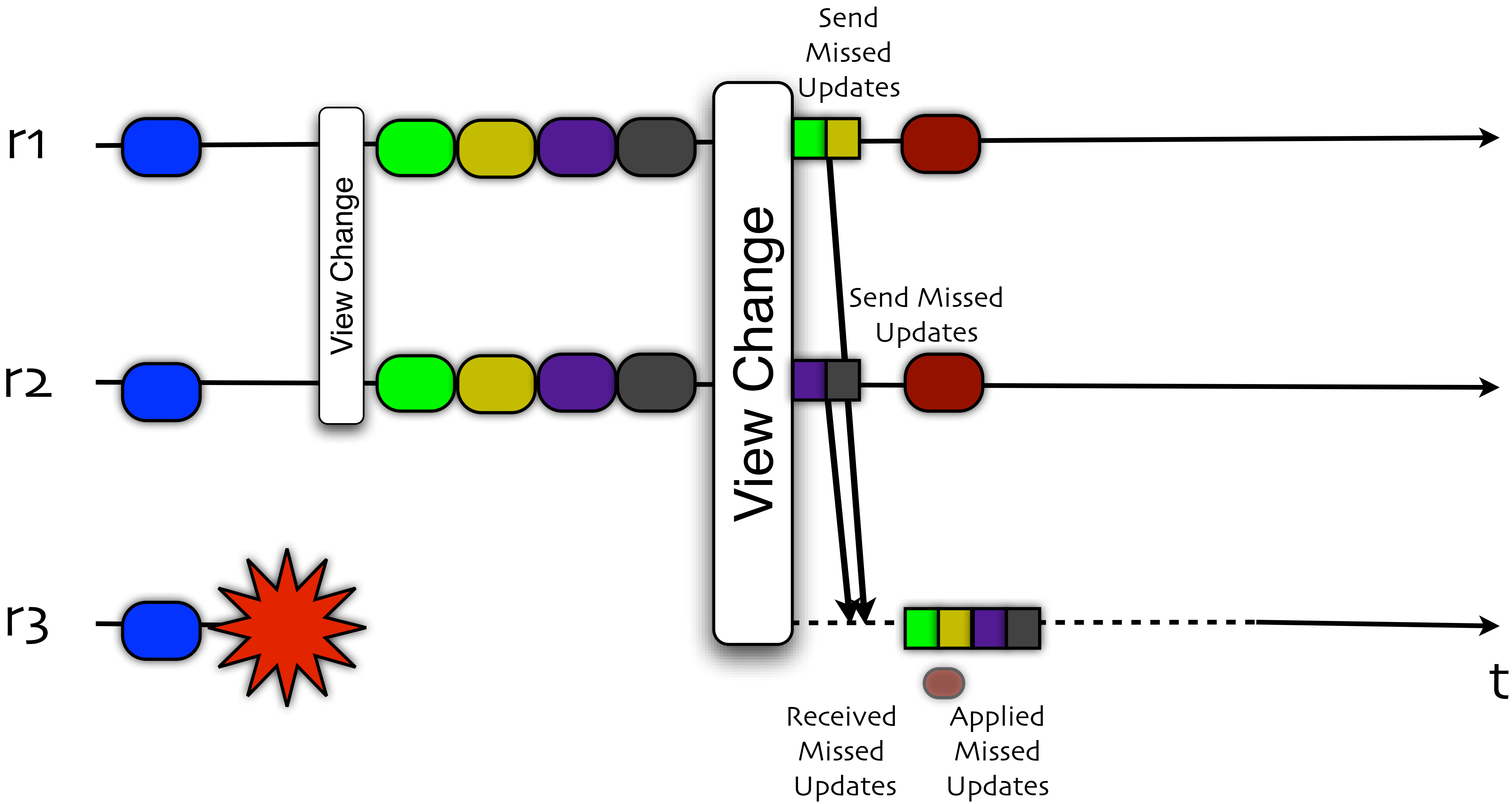
Parallel Recovery



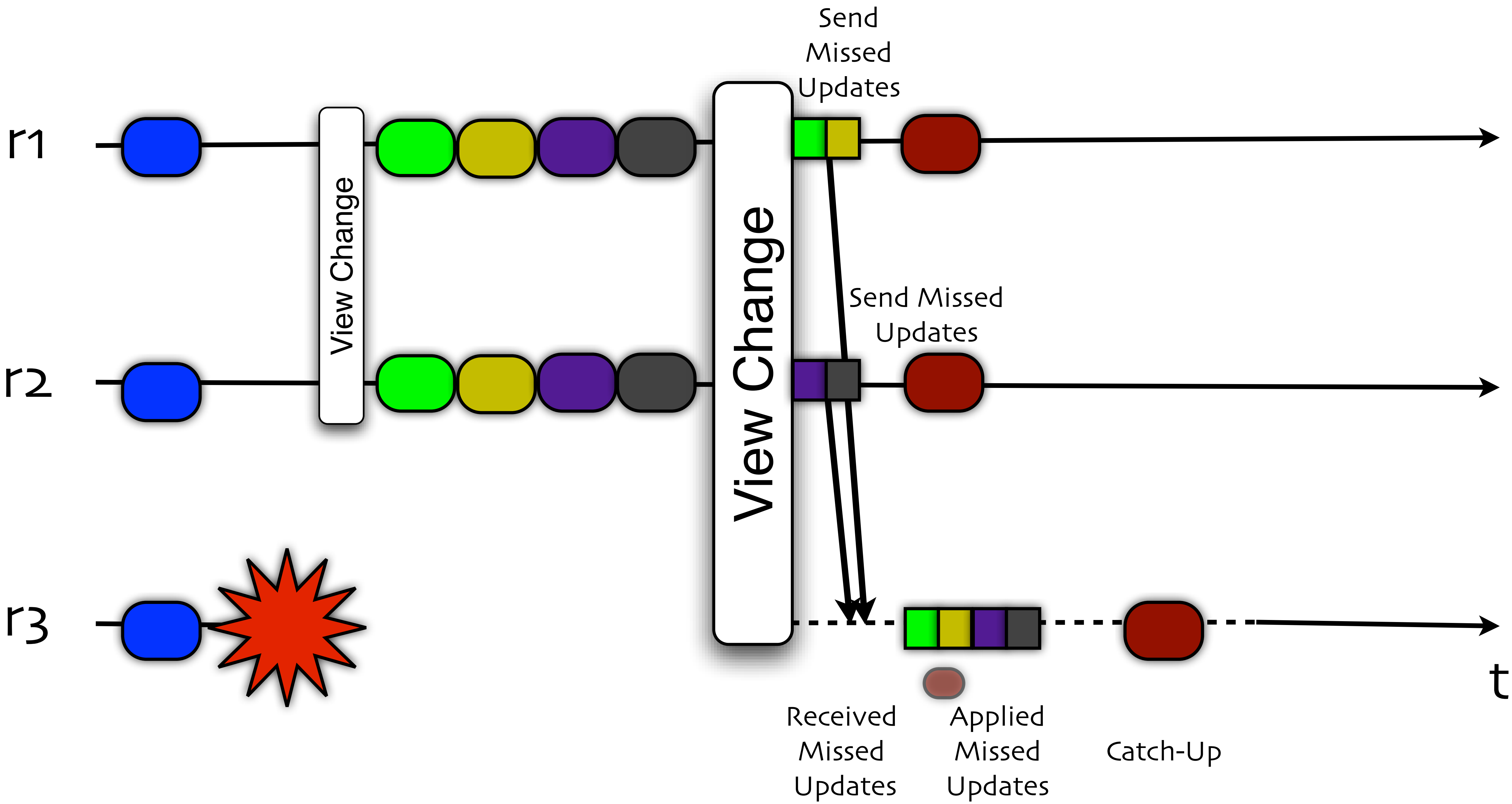
Parallel Recovery



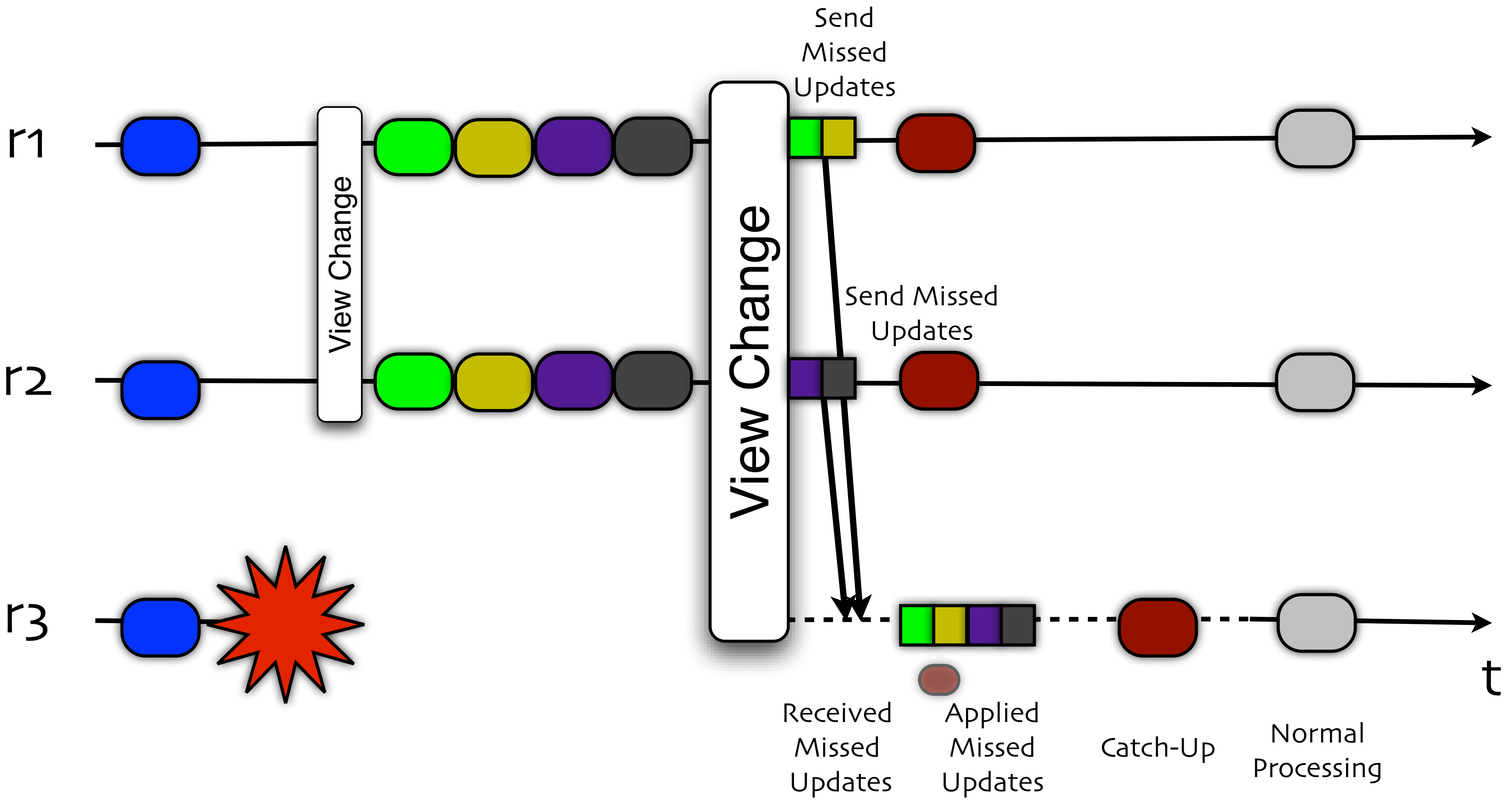
Parallel Recovery



Parallel Recovery

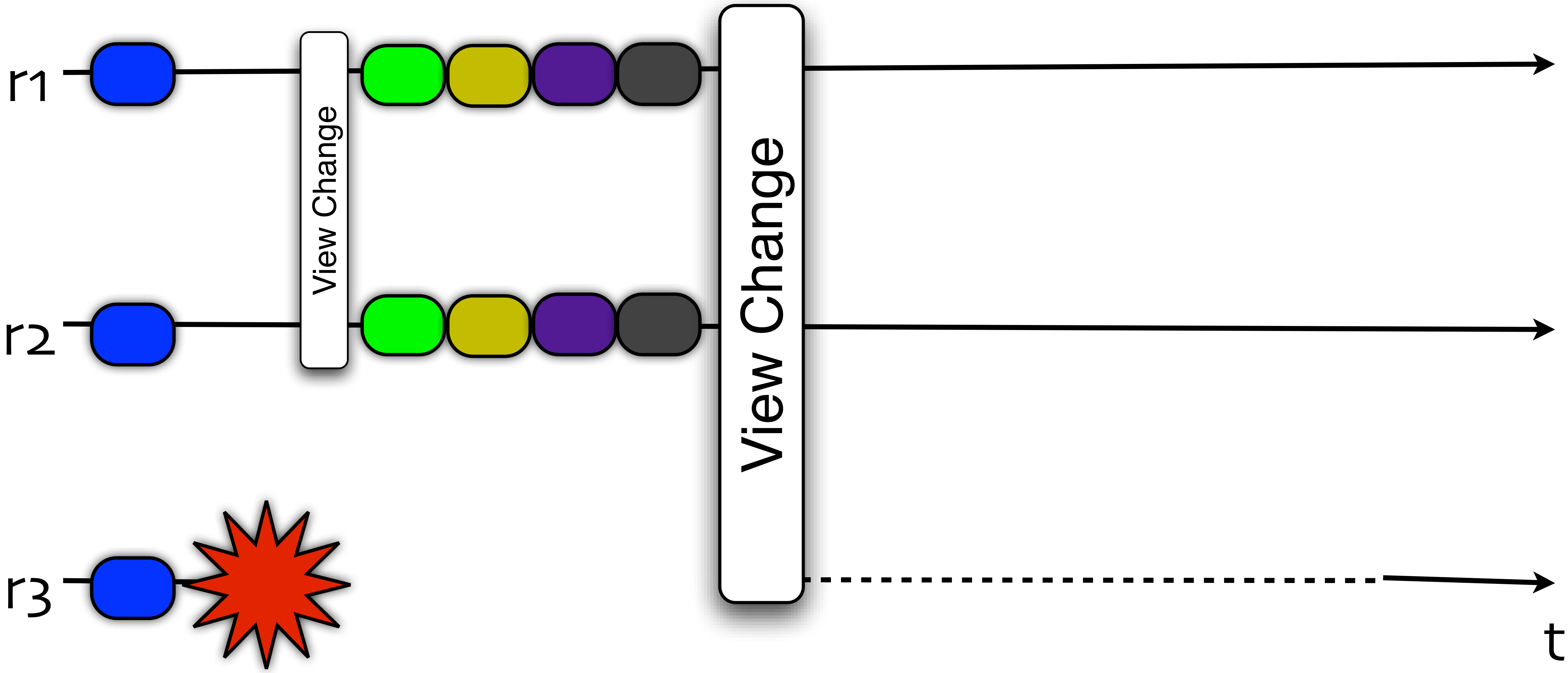


Parallel Recovery

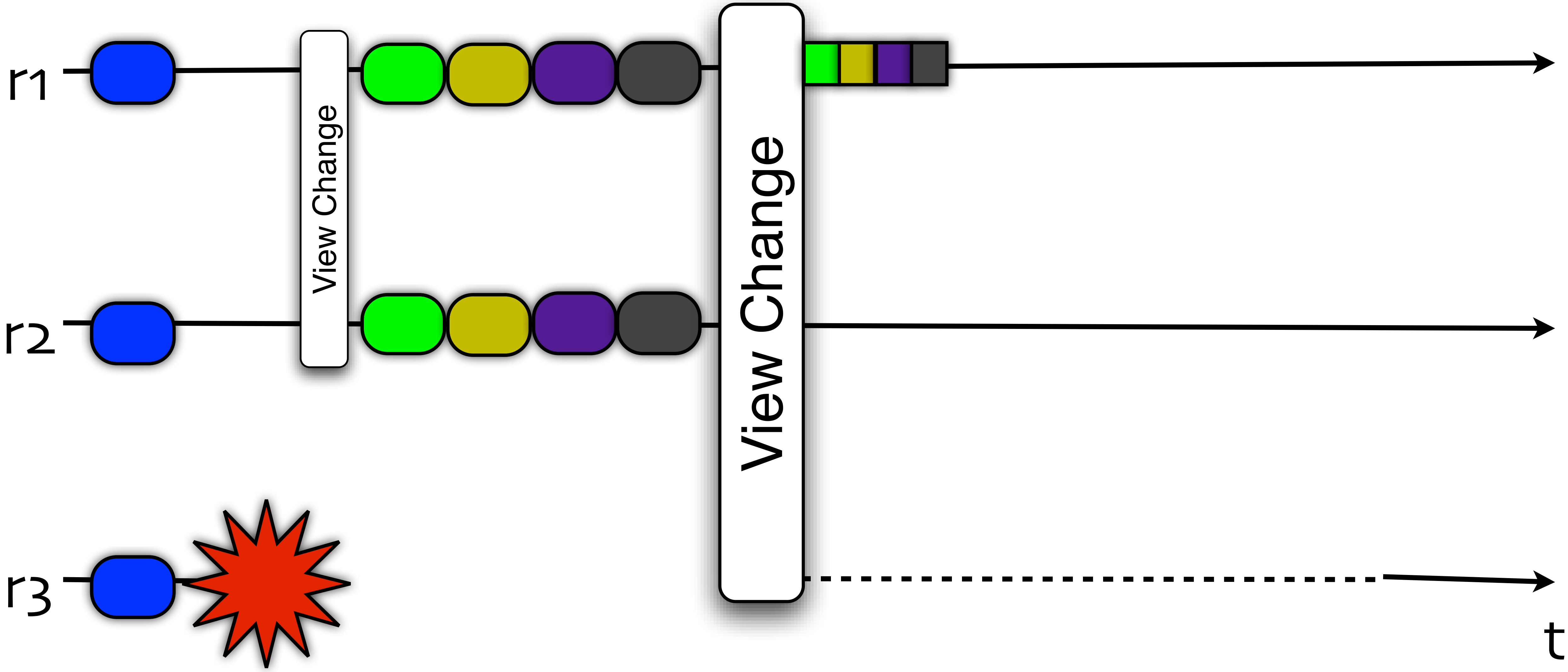


Convergence Phases

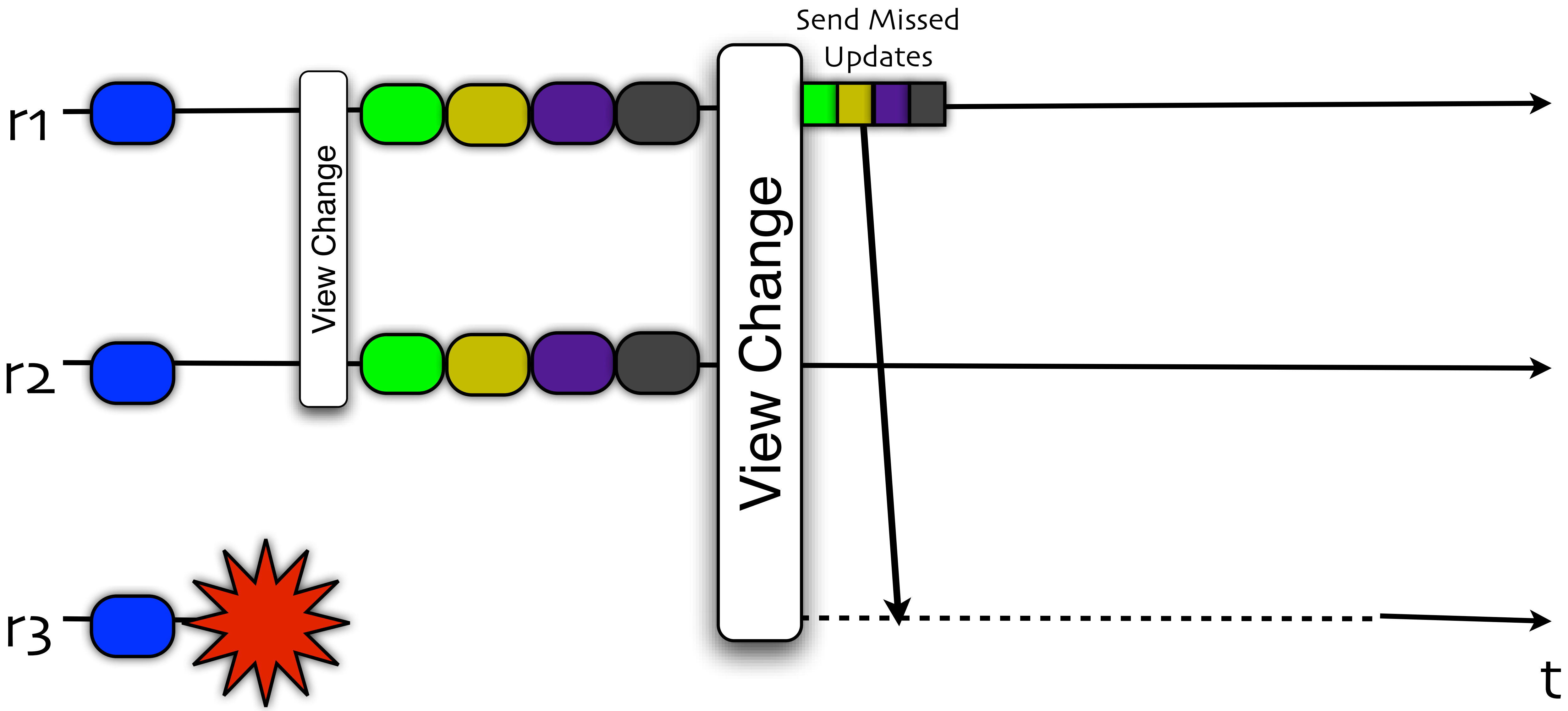
Convergence Phases



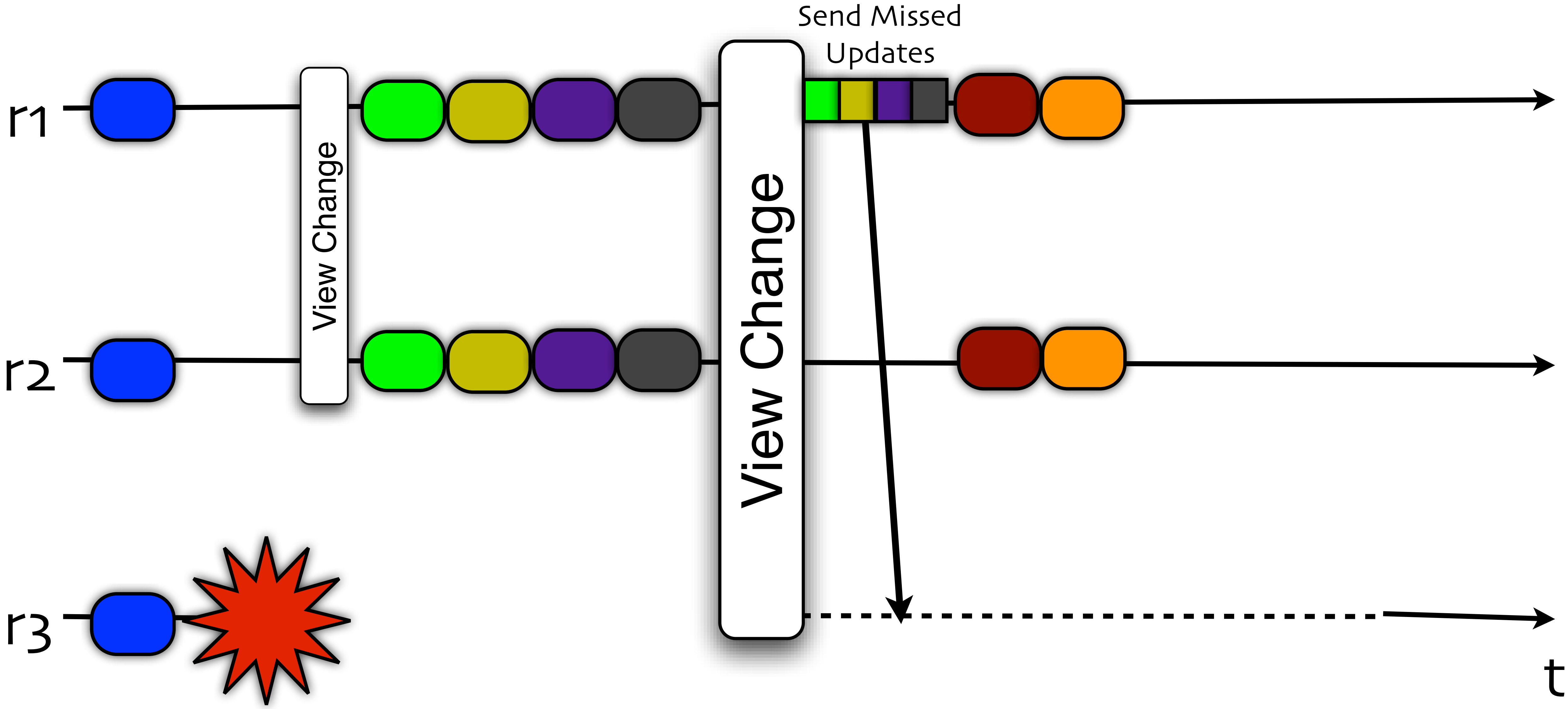
Convergence Phases



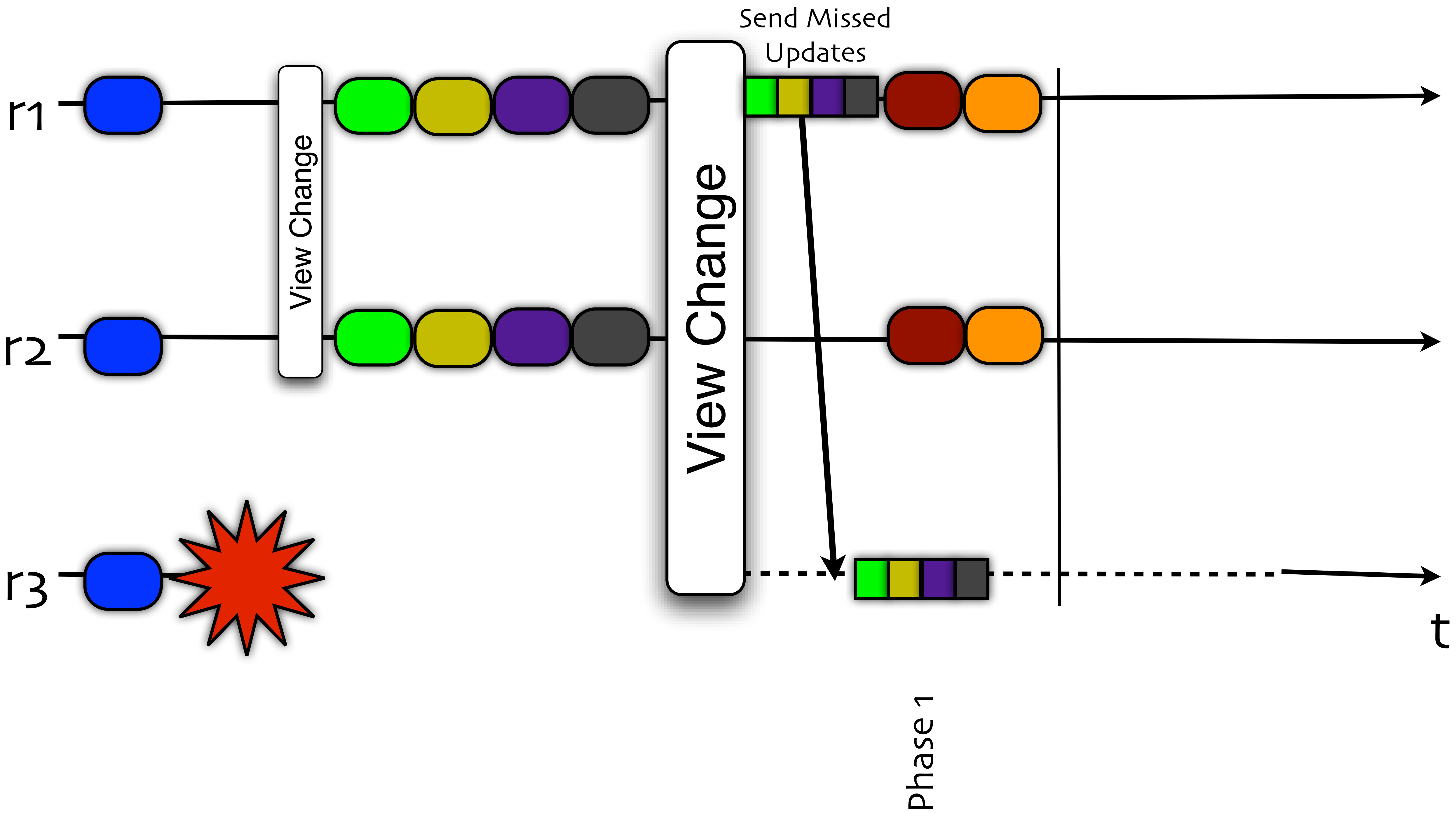
Convergence Phases



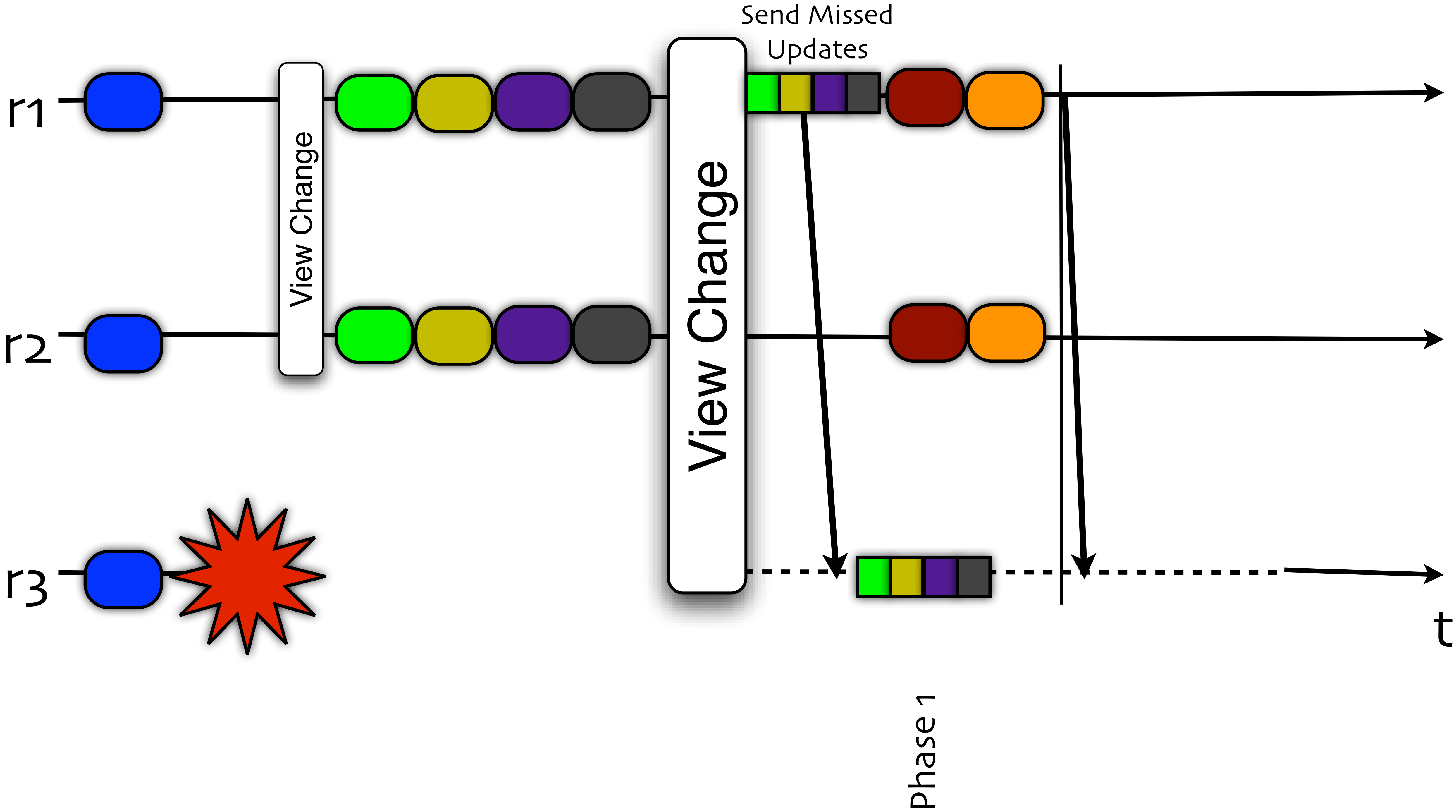
Convergence Phases



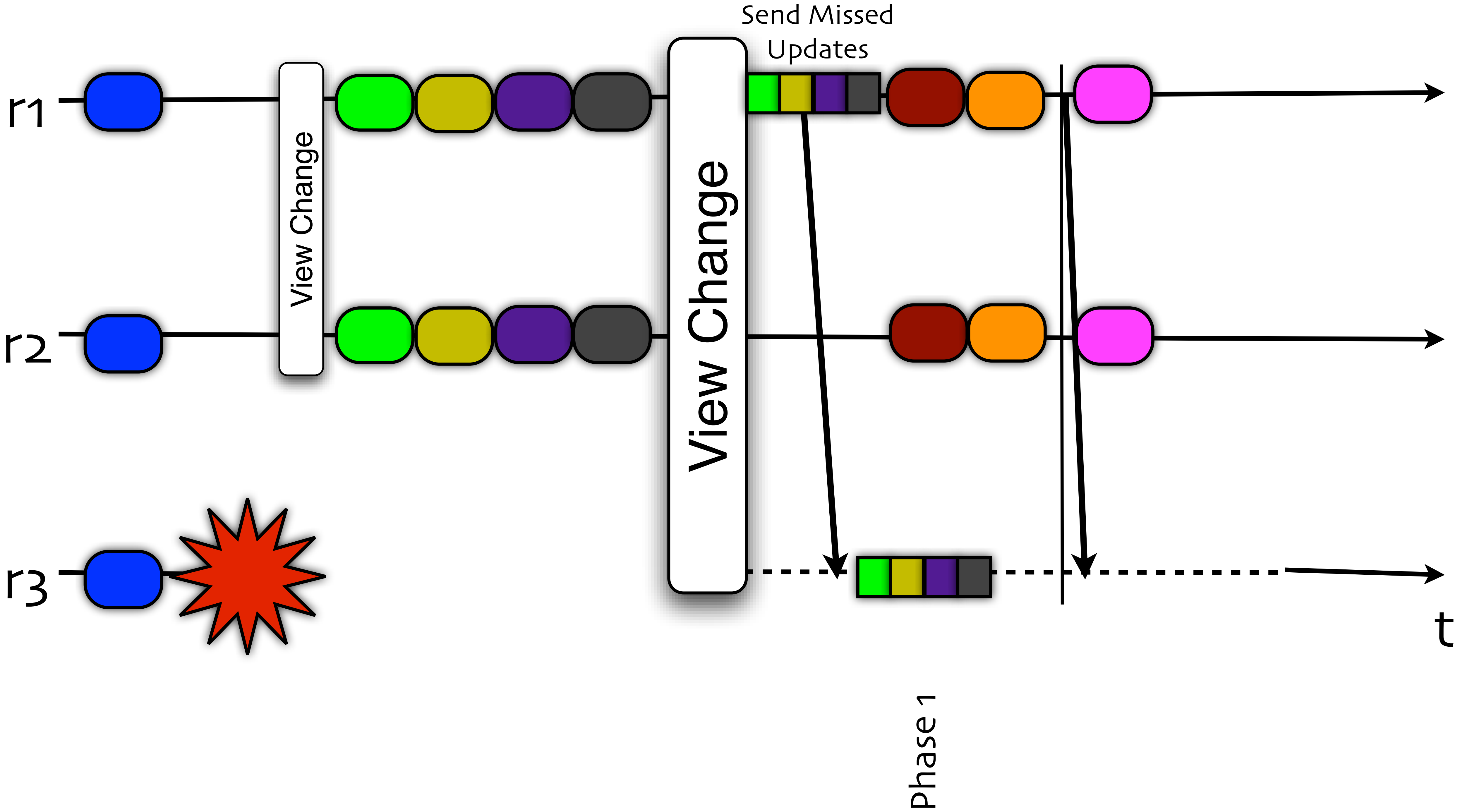
Convergence Phases



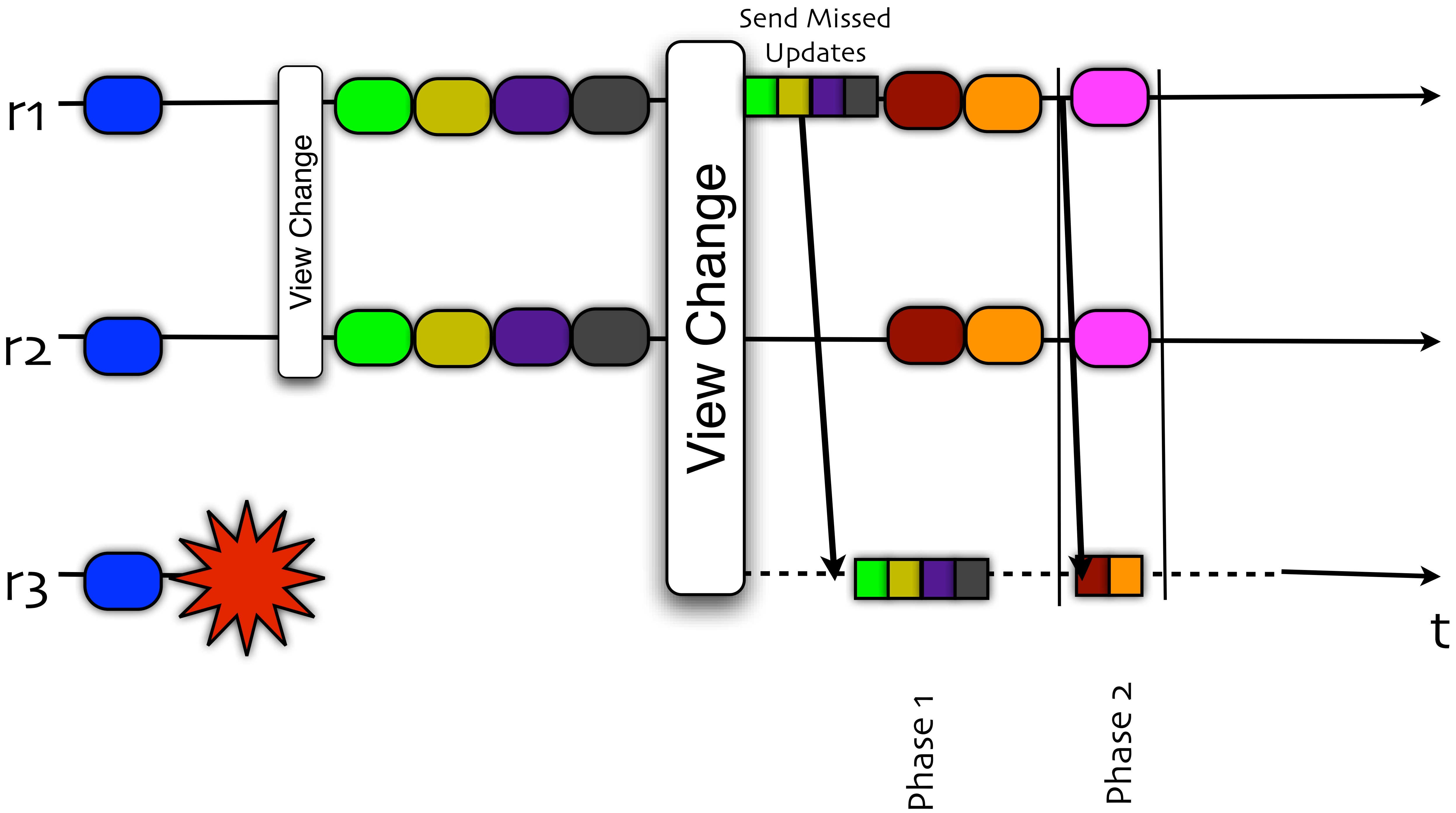
Convergence Phases



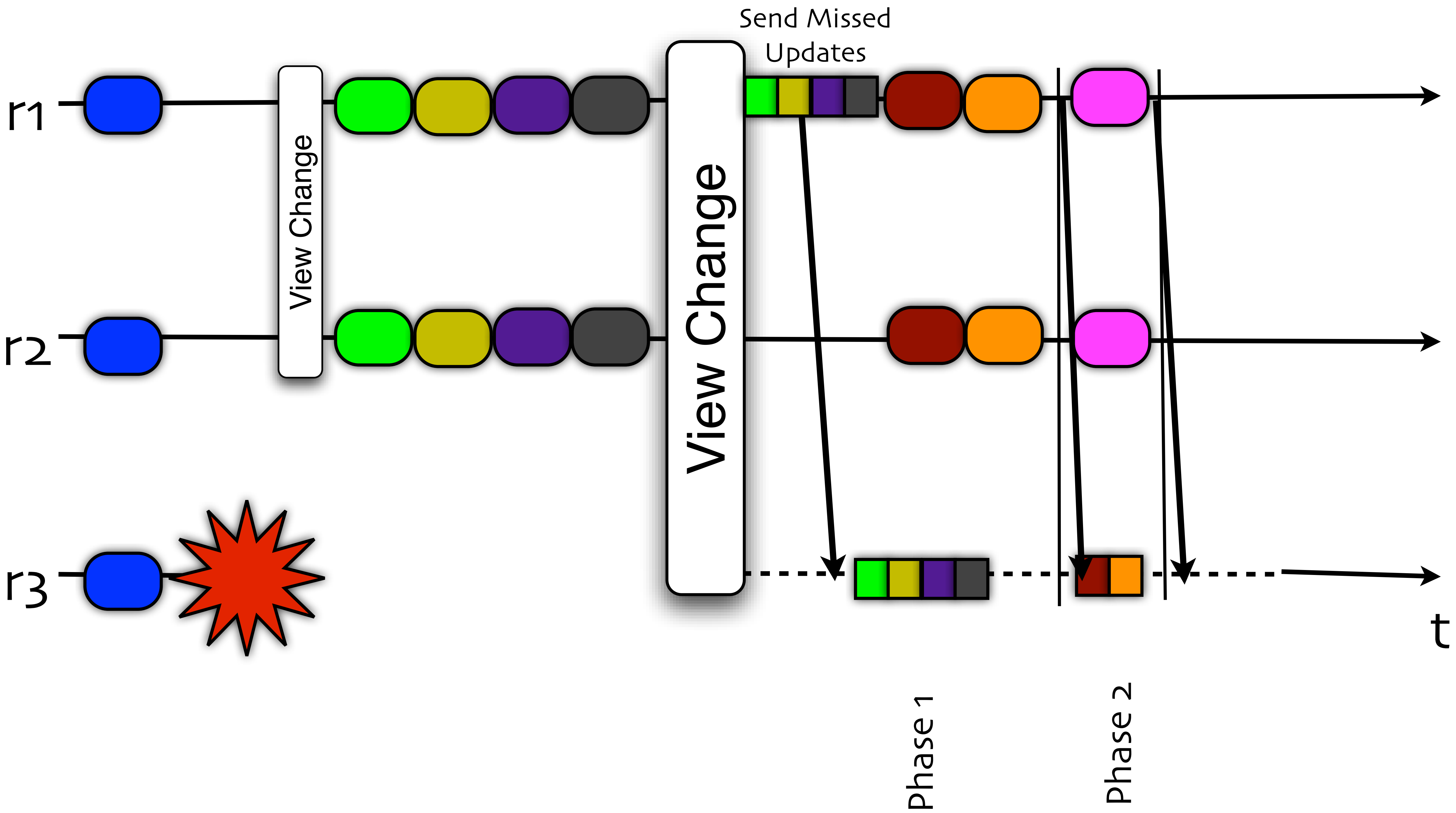
Convergence Phases



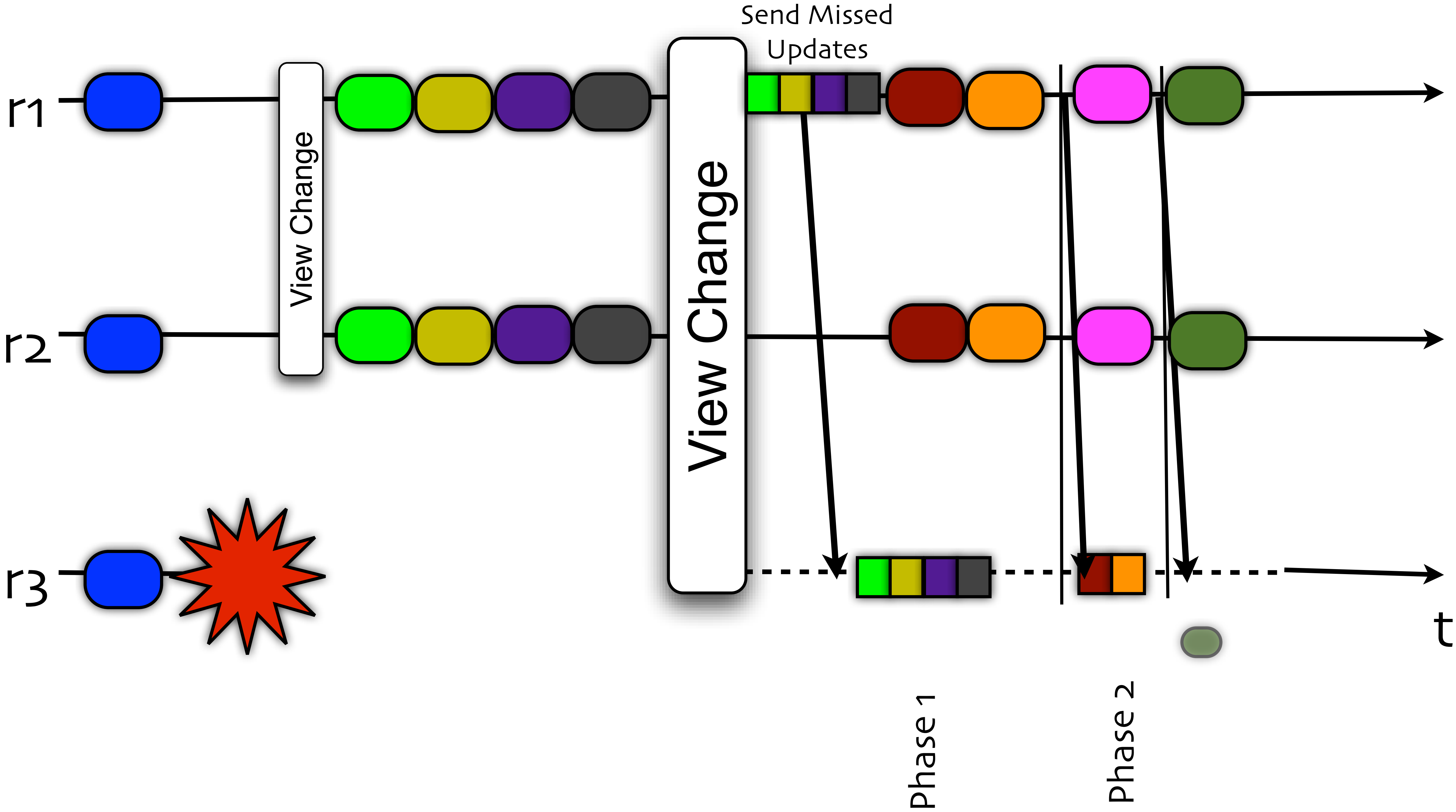
Convergence Phases



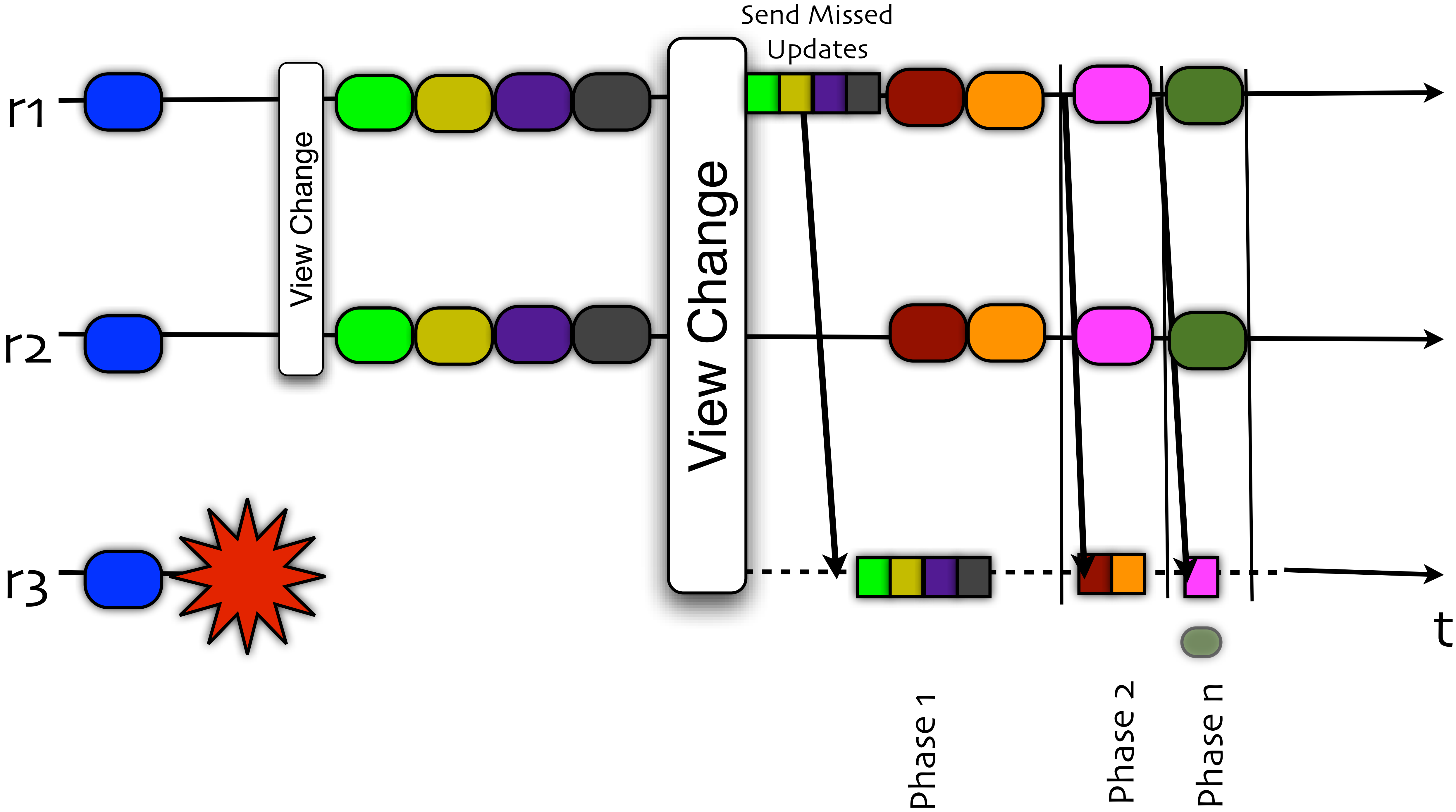
Convergence Phases



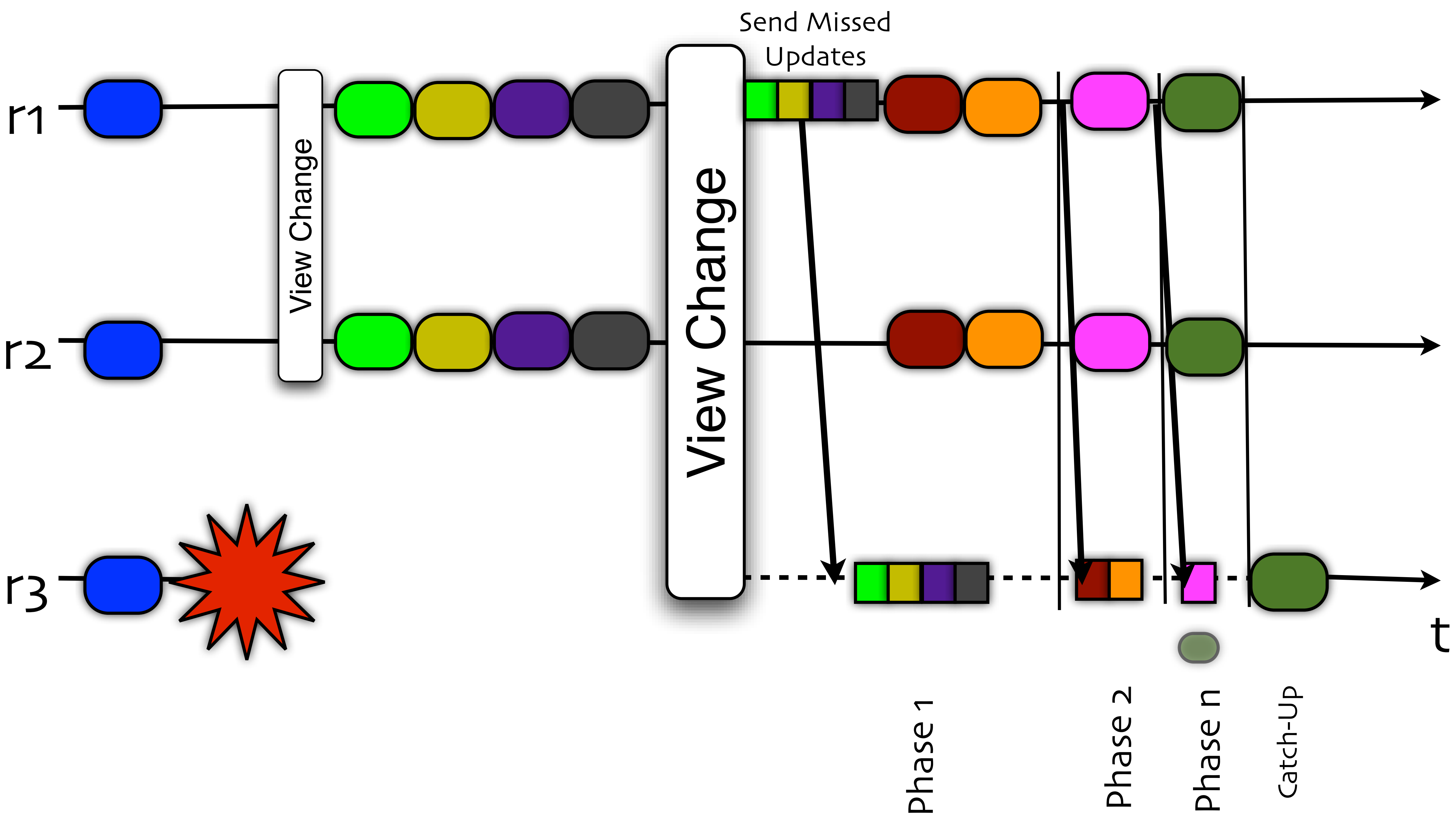
Convergence Phases



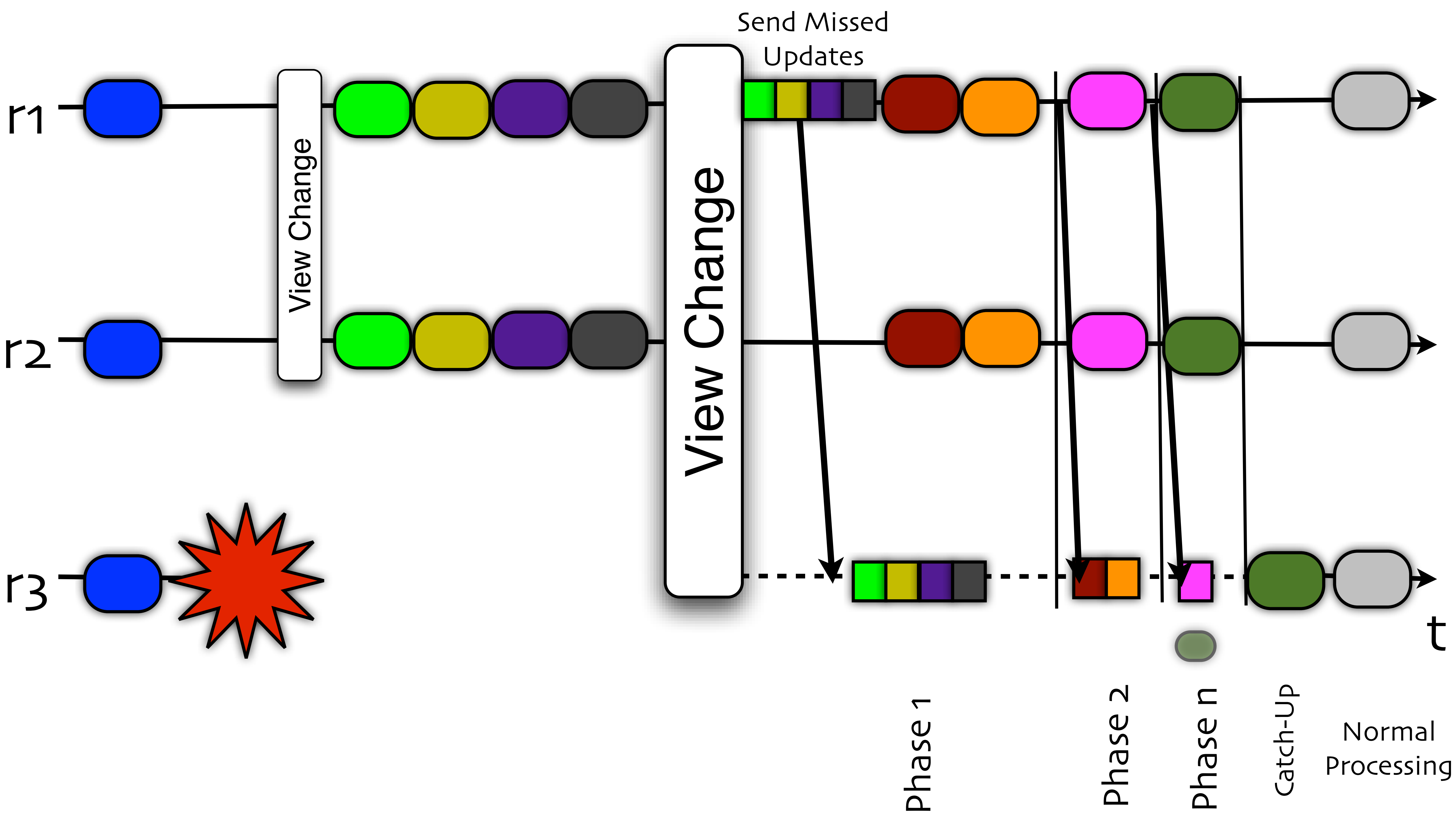
Convergence Phases



Convergence Phases



Convergence Phases



Experimental Setting

- 4 commodity servers on par with the systems used in recent related work
  - Intel Core 2 Duo at 2.13GHz, 1GB RAM and dedicated SATA HD.
- Replicas ran an instance of PostgreSQL 8.1 and a Java Virtual Machine (1.5.0) for the Replication service.
- No failures during the recovery process
- At most one replica was recovering during evaluation
- Flow Control on the incoming rate of update transactions during recovery
- Average of three independent samples

Experiments’ Workload

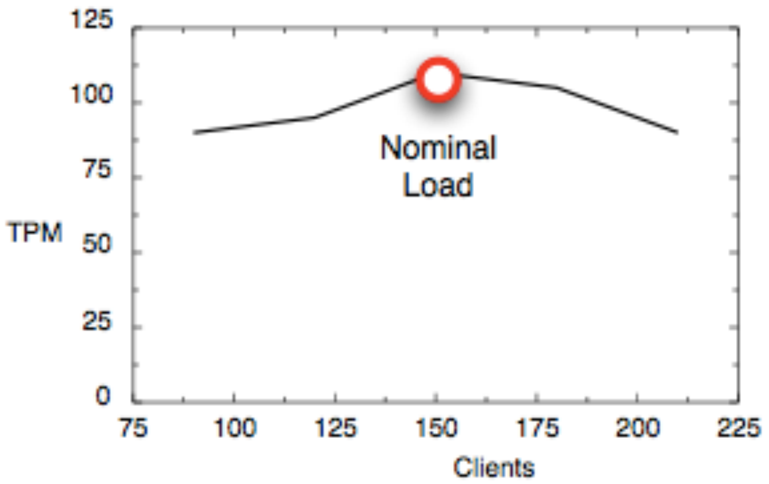
Experiments' Workload

- Two standard benchmarks
  - TPC-C: write intensive workload, IO stressing (15 warehouses, 150 clients, 2.2GB database)
  - TPC-W, read intensive workload, CPU stressing (Shopping Mix, 400 clients, 10000 items, 2.4GB database)

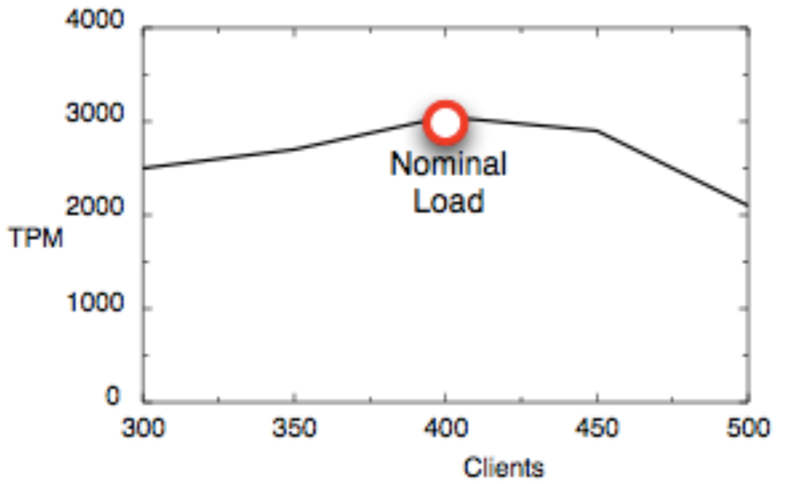
Experiments' Workload

- Two standard benchmarks
  - TPC-C: write intensive workload, IO stressing (15 warehouses, 150 clients, 2.2GB database)
  - TPC-W, read intensive workload, CPU stressing (Shopping Mix, 400 clients, 10000 items, 2.4GB database)
- Evaluation of recovery with all replicas close to their nominal capacity, maximum load that does not saturate machines.

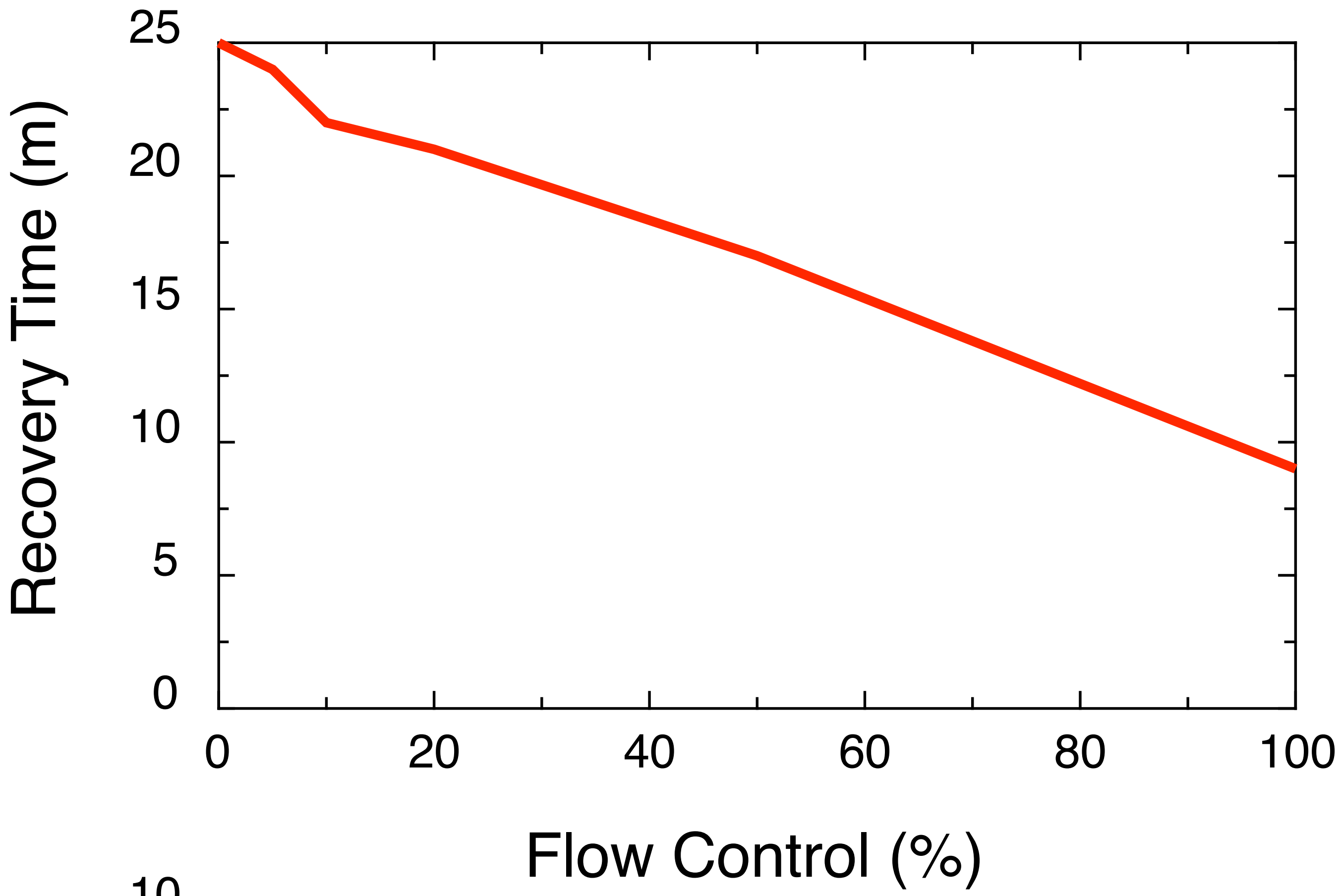
TPC-C



TPC-W

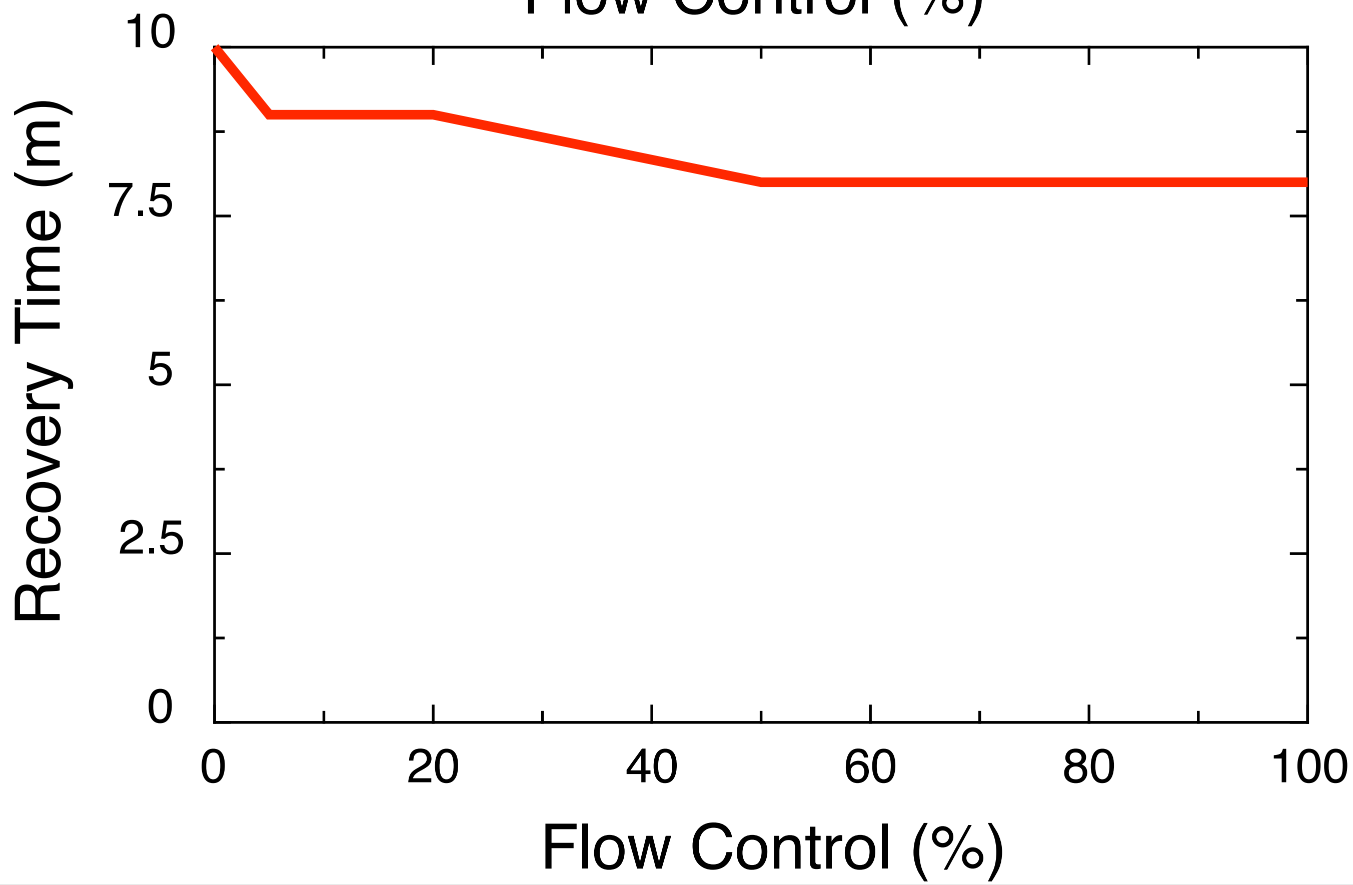


Recovery Time: Full Transfer



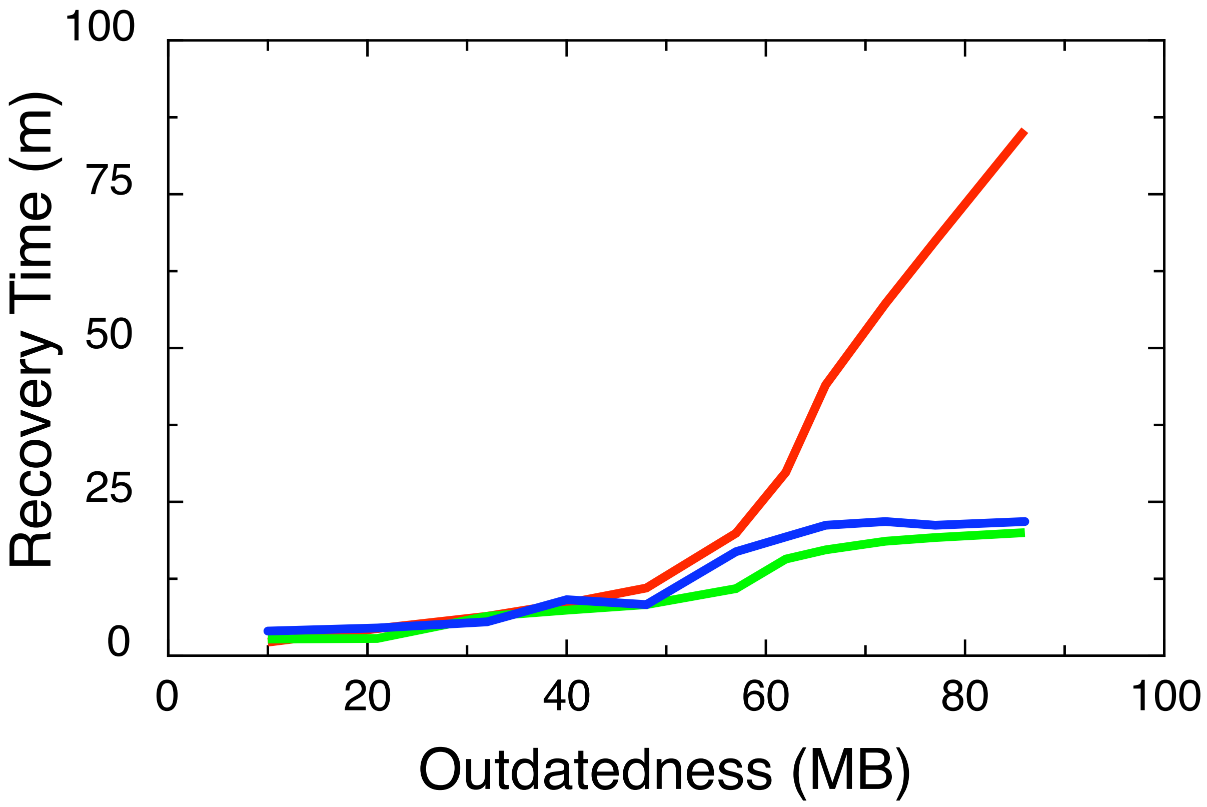
TPC-C 150 clients
2.2GB database
throughput 110 tpm

Specific database
dump and restore tools



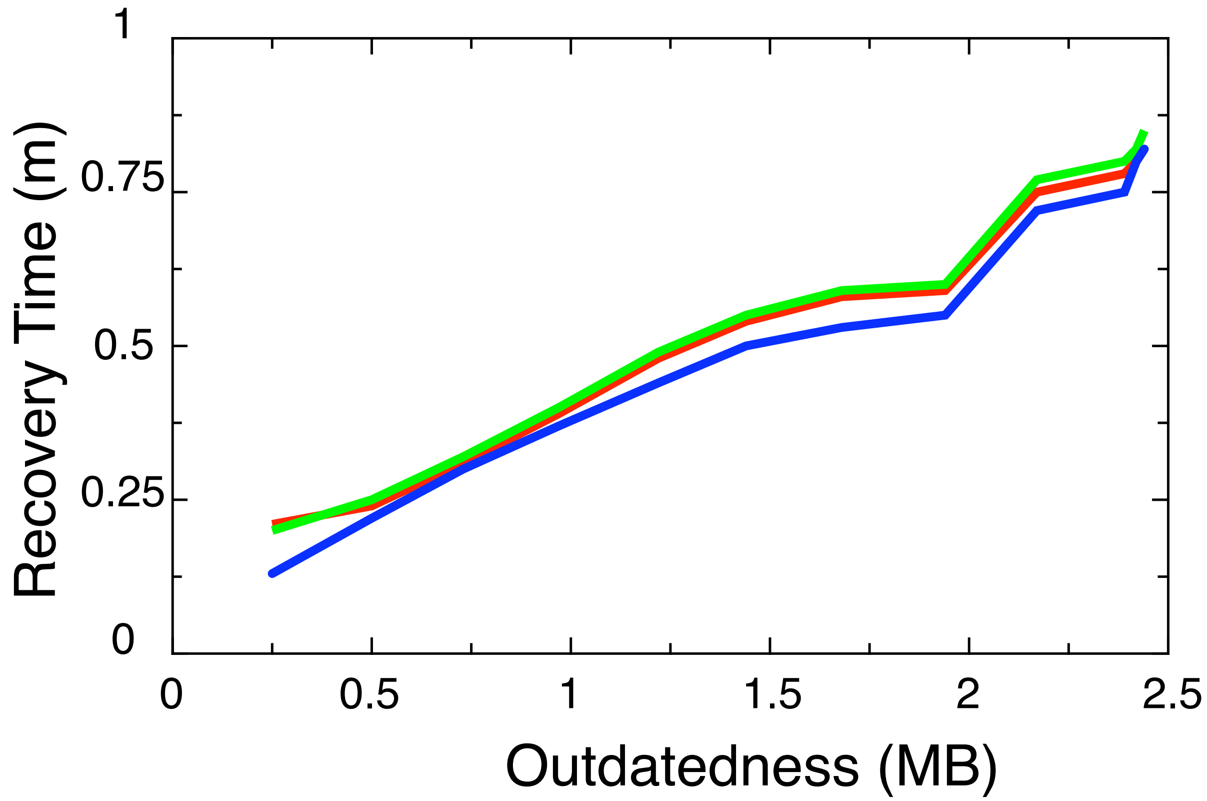
TPC-W 400 clients
2.4GB database
throughput 3000 tpm

Recovery Time: Convergence Phases



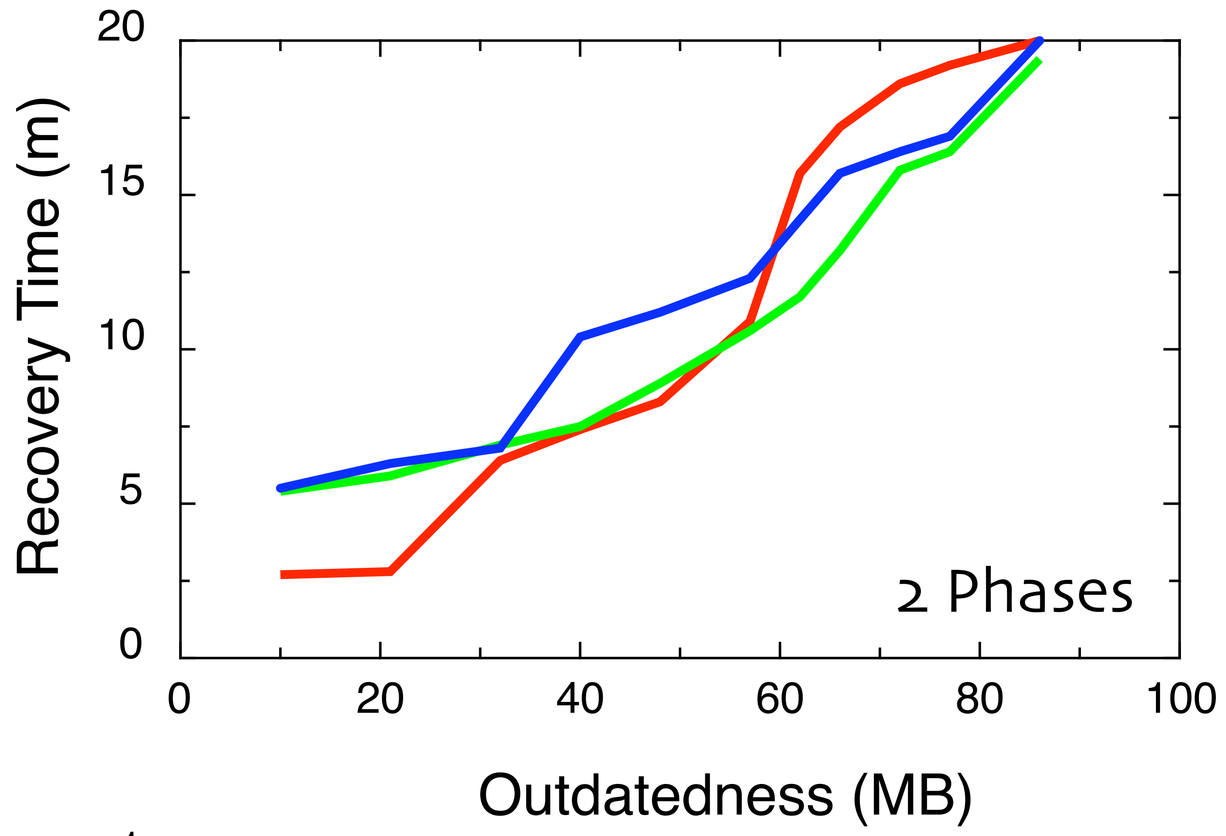
TPC-C 150 clients
2.2GB database
throughput 110 tpm

🟥 1 phase
🟩 2 phases
🟦 5 phases



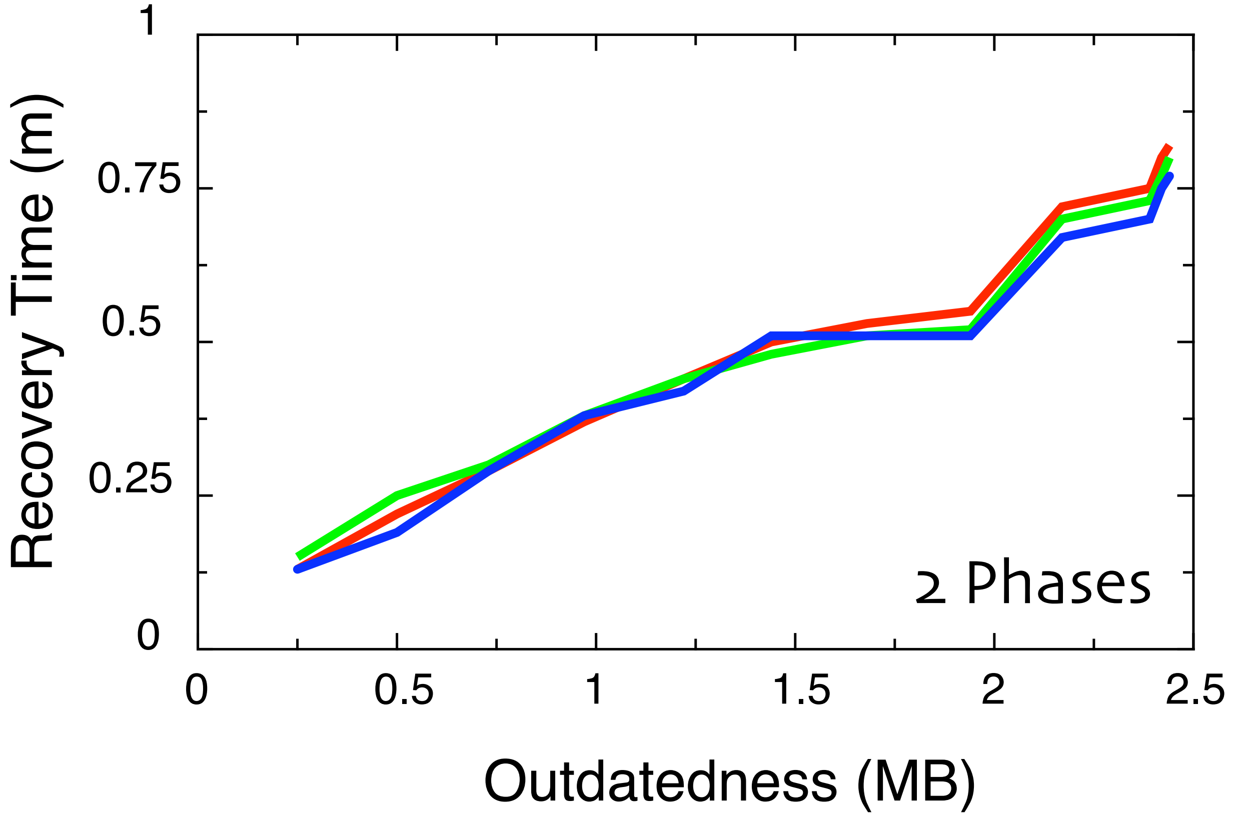
TPC-W 400 clients
2.4GB database
throughput 3000 tpm

Recovery Time: State Donors



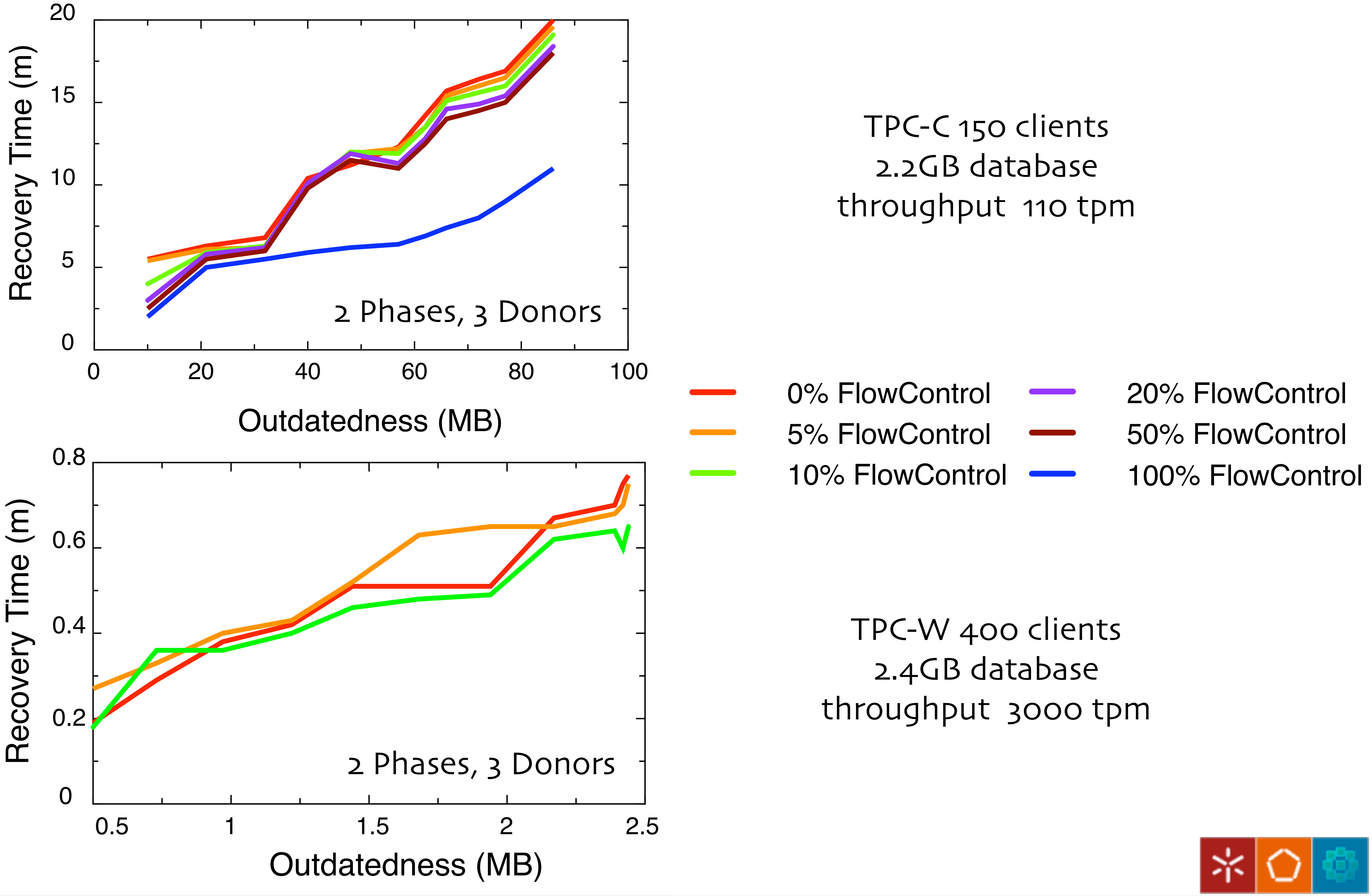
TPC-C 150 clients
2.2GB database
throughput 110 tpm

🟥 1 donor
🟩 2 donors
🟦 3 donors



TPC-W 400 clients
2.4GB database
throughput 3000 tpm

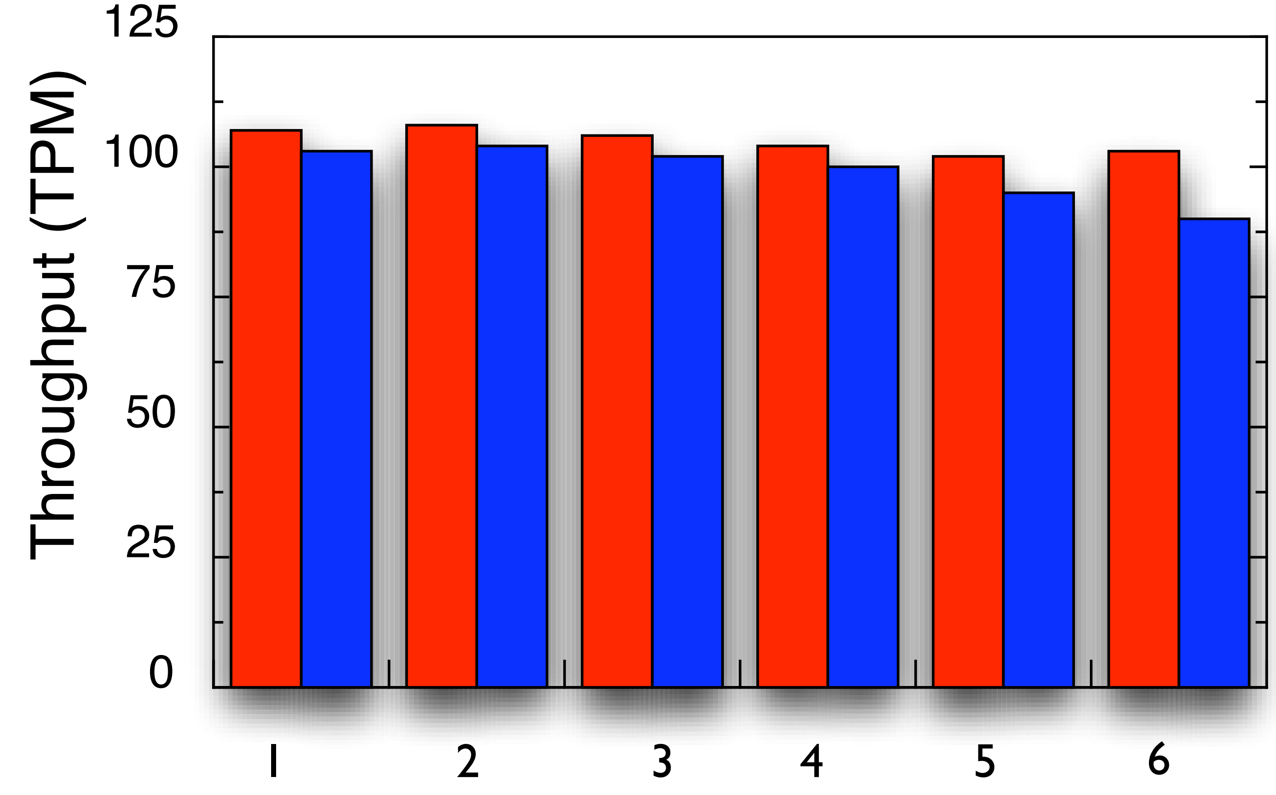
Recovery Time: Flow Control



TPC-C 150 clients
2.2GB database
throughput 110 tpm

TPC-W 400 clients
2.4GB database
throughput 3000 tpm

Throughput



TPC-W, 400 clients, 2.4GB database

Before 🟥 and After 🟦 recovery

1 -> 1 state donor, 2 convergence phases and no flow control

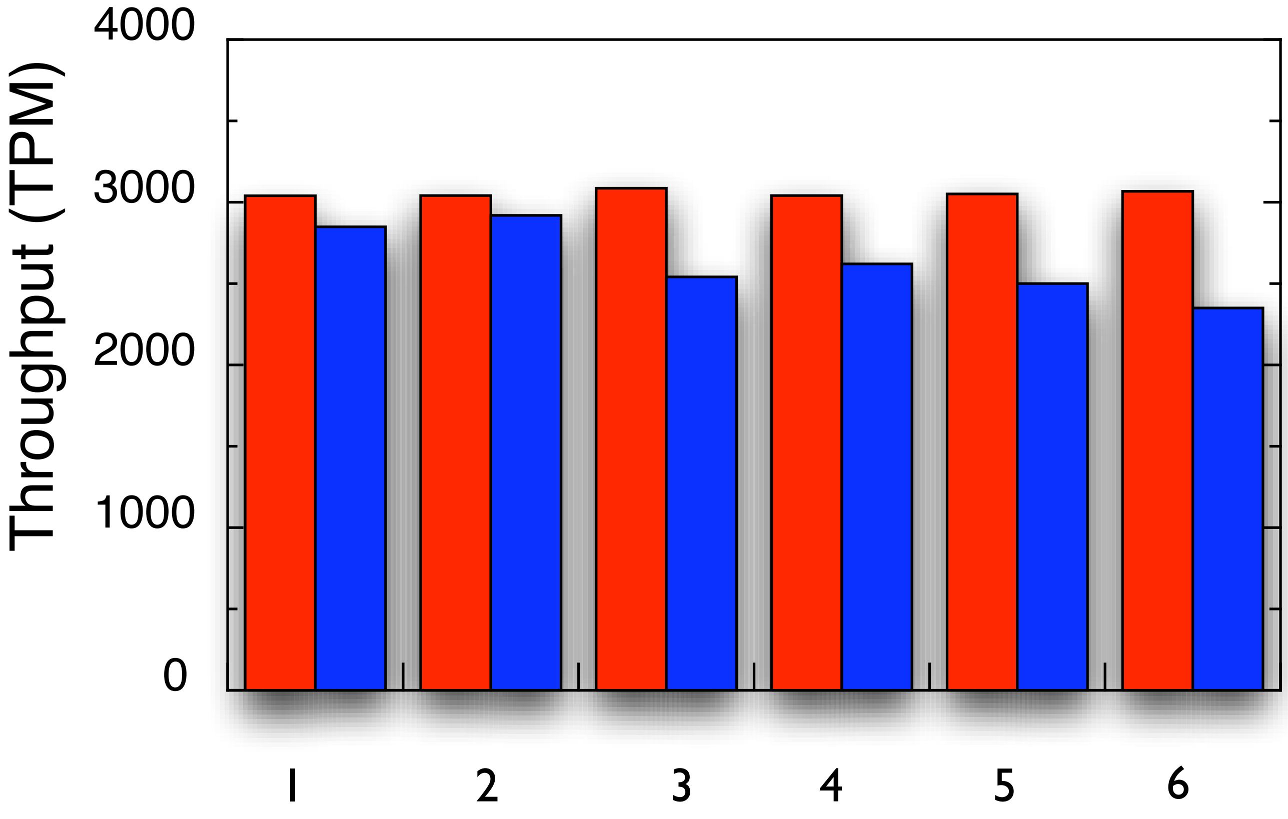
2 -> 1 state donor, 5 convergence phases and no flow control

3 -> 2 state donor, 2 convergence phases and no flow control

4 -> 3 state donor, 2 convergence phases and no flow control

5 -> 3 state donor, 2 convergence phases and 5% flow control

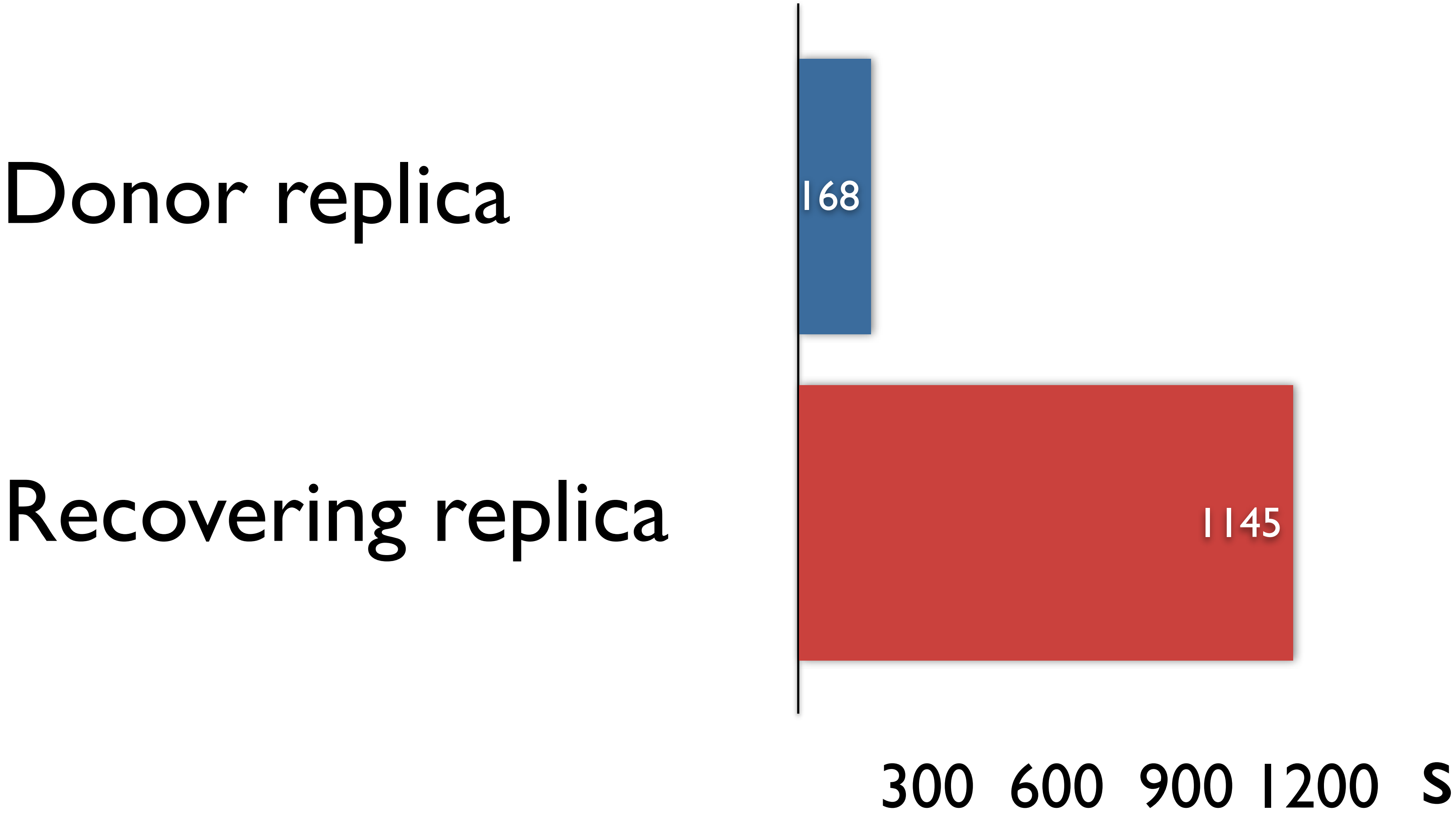
6 -> 3 state donor, 2 convergence phases and 10% flow control



TPC-C, 150 clients, 2.2GB database

Donation vs. Recovery Time

- Recovery is about 7x longer than donation
- For a TPC-C workload, a 86MB recovery log and a single donor:



Conclusion

- Database online recovery protocol combining several previously proposed optimization techniques

Conclusion

- Database online recovery protocol combining several previously proposed optimization techniques
- The results of our tests do not reveal any relevant effect of the optimizations on the recovery time or on the overall cluster performance either.

Conclusion

- Database online recovery protocol combining several previously proposed optimization techniques
- The results of our tests do not reveal any relevant effect of the optimizations on the recovery time or on the overall cluster performance either.
- The capacity of the recovering replica to apply the received state turns out to be the salient limiting factor

Conclusion

- Database online recovery protocol combining several previously proposed optimization techniques
- The results of our tests do not reveal any relevant effect of the optimizations on the recovery time or on the overall cluster performance either.
- The capacity of the recovering replica to apply the received state turns out to be the salient limiting factor
- Most research has been targeted at optimizing the operations that are not, by a large margin, limiting factors in overall performance