Fair loss links

- **Fair loss** message sent infinitely often from pi to pj and neither pj nor pi crashes => the message is delivered infinitely many times
- **Finite duplication** message m sent a finite number of times from pi to pj => m is delivered a finite number of times by pj
- **No creation** no message is delivered unless it was sent

Stubborn links

- **Stubborn delivery** correct pi sends message m to correct pj => pj delivers m an infinite number of times
- **No creation** no message is delivered unless it was sent

Reliable (perfect) links

- **Validity** pi, pj correct => every message sent from pi to pj is eventually delivered by pj
- **No creation** no message is delivered unless it was sent
- **No duplication** no message is delivered by a process more then once

Perfect failure detector

- **Strong completeness** eventually every process that crashes is permanently suspected by every correct process
- **Strong accuracy** no process is suspected before it crashes

Eventually perfect failure detector

- **Strong completeness** eventually every process that crashes is permanently suspected by every correct process
- **Eventual accuracy** eventually, no process is suspected before it crashes

Best effort broadcast

- **Validity** pi, pj correct => every message broadcast by pi is eventually delivered by pj
- **No creation** no message is delivered unless it was broadcast.
- **No duplication** no message is delivered more than once.

Reliable broadcast

- **Validity** pi, pj correct => every message broadcast by pi is eventually delivered by pj
- **No creation** no message is delivered unless it was broadcast
- **No duplication** no message is delivered more than once
- **Agreement** for every message m, a correct process pi delivers m => every correct process delivers m

Uniform reliable broadcast

- **Validity** pi, pj correct => every message broadcast by pi is eventually delivered by pj
- **No creation** no message is delivered unless it was broadcast
- **No duplication** no message is delivered more than once
- **Uniform agreement** for every message m, a process delivers m => every correct process delivers m

Reliable causal broadcast

- **Validity** pi, pj correct => every message broadcast by pi is eventually delivered by pj
- **No creation** no message is delivered unless it was sent
- **No duplication** no message is delivered more than once
- **Agreement** for every message m, a correct process pi delivers m => every correct process delivers m
- **Causal order** if any process pi delivers m2 => pi must have delivered all m1 such that m1->m2

Reliable FIFO links

- **Validity** pi, pj correct => every message sent by pi to pj is eventually delivered by pj
- **No creation** no message is delivered unless it was sent
- **No duplication** no message is delivered more than once
- **FIFO** messages are delivered in the same order as they were sent

Stoppable stubborn links

- **Stubborn delivery** pi, pj correct and pi sends m to pj => pj delivers m an infinite number of times unless pi receives a stop event for m.
- **No creation** no message is delivered unless it was sent

Perfect stoppable links

- **Validity** pi, pj correct => every message sent from pi to pj is eventually delivered by pj unless pi receives a stop event for m
- **No duplication** no message is delivered more than once
- **No creation** no message is delivered unless it was sent

(Uniform) Total order broadcast

- **Validity** pi, pj correct => every message broadcast by pi is eventually delivered by pj
- **No duplication** no message is delivered more than once
- **No creation** no message is delivered unless it was broadcast
- **Agreement** for any message m, a correct process delivers m => every correct process delivers m
- **Total order** m1, m2 messages, correct pi delivers m1 without having delivered m2 => no correct process delivers m2 before m1

(Uniform) Total order broadcast

- **Validity** pi, pj correct => every message broadcast by pi is eventually delivered by pj
- **No duplication** no message is delivered more than once
- **No creation** no message is delivered unless it was broadcast
- **Uniform agreement** for any message m, a process delivers m => every correct process delivers m
- **Uniform total order** m1, m2 messages, pi delivers m1 without having delivered m2 => no process delivers m2 before m1

Consensus

- **Validity** every value decided is a value proposed
- **Agreement** no two processes decide differently
- **Termination** every correct process eventually decides
- **Integrity** no process decides more than once

(1, N) Regular register

- **Termination** a correct process invokes an operation => the process eventually receives the corresponding confirmation
- **Validity** A read returns the last value written, or the value concurrently written

(1, N) Atomic register

- **Termination** a correct process invokes an operation => the process eventually receives the corresponding confirmation
- **Validity** A read returns the last value written, or the value concurrently written
- **Ordering** a read returns v2 after a read that precedes it has returned v1 => v1 cannot be written after v2

Terminating reliable broadcast

- **Validity** pi is correct and broadcasts m => pi eventually delivers m
- **Integrity** pi delivers the message msg => either msg = phi or msg = m broadcast by src
- **Termination** every correct process eventually delivers exactly one message
- **Agreement** correct pi delivers m => every correct process delivers m

Uniform terminating reliable broadcast

- **Validity** pi is correct and broadcasts m => pi eventually delivers m
- **Integrity** pi delivers the message msg => either msg = phi or msg = m broadcast by src
- **Termination** every correct process eventually delivers exactly one message
- **Agreement** pi delivers m => every correct process delivers m

Group membership

- **Local monotonicity** process p installs view (j, M) after (k, N) => j > k and |M| < |N|
- **Agreement** no two processes install views (j, M), (j, M') such that M != M'
- **Completeness** p crashes => there exists an integer j such that every correct process eventually installs view (j, M) such that p is not in M
- **Accuracy** p installs view (j, M) and pj is not in M => pj crashed

View synchrony

- GM **Local monotonicity** process p installs view (j, M) after (k, N) => j > k and |M| < |N|
- GM **Agreement** no two process install views (j, M) and (k, N) such that M != N
- GM **Completeness** p crashes => there exists an integer j such that every correct process eventually installs view (j, M) such that p is not in M
- GM **Accuracy** p installs view (j, M) and pj is not in M => pj crashed
- RB **Validity** pi, pj correct => every message broadcast by pi is eventually delivered by pj
- RB **No creation** no message is delivered unless it was broadcast
- RB **No duplication** no message is delivered more than once

Non blocking atomic commit

- **Agreement** no two processes decide differently
- **Termination** every correct process eventually decides
- **Commit-Validity** 1 can be decided <=> every correct process proposes 1
- **Abort-Validity** 0 can only be decided if some process crashes or proposes 0