Cooperative Backup in Dynamic Systems

M.-O. Killijian

ReSIST: Resilience for Survivability in IST
First Open Workshop, BUTE, 21-22 March 2007

Cooperative Backup for Dynamic Systems

- Dynamic Systems in a Ubiquitous World
  - Nomadic devices
  - Mostly disconnected operations
  - Opportunistic wireless communication with similar devices
  - Peer-to-peer model of interactions
  - Embedded data generation
- Secure Cooperative Backup for Nomadic Devices
  - Leverage encounters for storing data
  - Even when no infrastructure is available

Cooperative Backup for Dynamic Systems

- Backup = protection of critical private data against
  - Permanent and transient faults affecting a data owner
  - Theft or loss of a data owner

Cooperative Backup for Dynamic Systems

- Backup = protection of critical private data against
  - Permanent and transient faults affecting a data owner
  - Theft or loss of a data owner
- New threats on backups
  - Malicious (and accidental) faults
  - Confidentiality, integrity and availability
- New threats on service
  - Selfish denial of service (refusal to cooperate)
    - Free-riding : consumption without contribution
    - “Tragedy of the commons” (Hardin 1968)
    - Attacks must be made unprofitable
  - Malicious denial of service (sabotage)
    - Attacks must be made ineffective or too costly

Cooperative Backup for Dynamic Systems

- Challenges
  - No prior organization
  - Ephemeral interactions
  - Limited energy, computation and storage
  - Only intermittent access to a fixed infrastructure
- Usual criteria for classic functionalities
  - User transparency
  - Usability
  - etc.

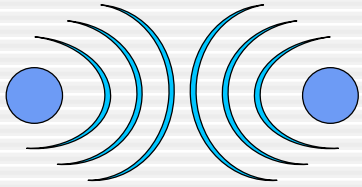
Overview

- Motivations
- Data Availability: Data scattering
  Data encoding and redundancy control [Courtès et al. 07]
  - (n,k) codes
  - Evaluation using GSPN and Markov chains
- Service Availability: Cooperation Incentives
  Crypto-challenges that can be delegated [Oualha et al. 07]
  - Probabilistic cooperation checking
  - Evaluation using game theory

Scattering and Redundancy

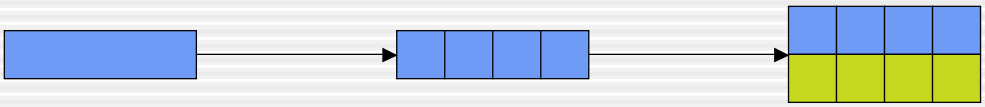
- Opportunistic communication to peers and to infrastructure
- Ephemeral encounters
  - Duration/frequency ?
  - Amount of data ?
  - Reliability of contributors ?
  - ➔ Scattering of fragments
- Untrusted and unreliable contributors
  - Ability to get fragments back ?
  - ➔ Replicate fragments
- Limited storage resources
  - ➔ Trade-off between redundancy and resource use
  - ➔ Optimization of gained availability vs resources

➔ Modeling and evaluation of scattering policies



Examples

Classic redundancy



➔ 1 fault / size = x2

(n,k) codes

➔ 4 faults / size = x2

(n,k) codes

(n,k) codes

Input: k fragments

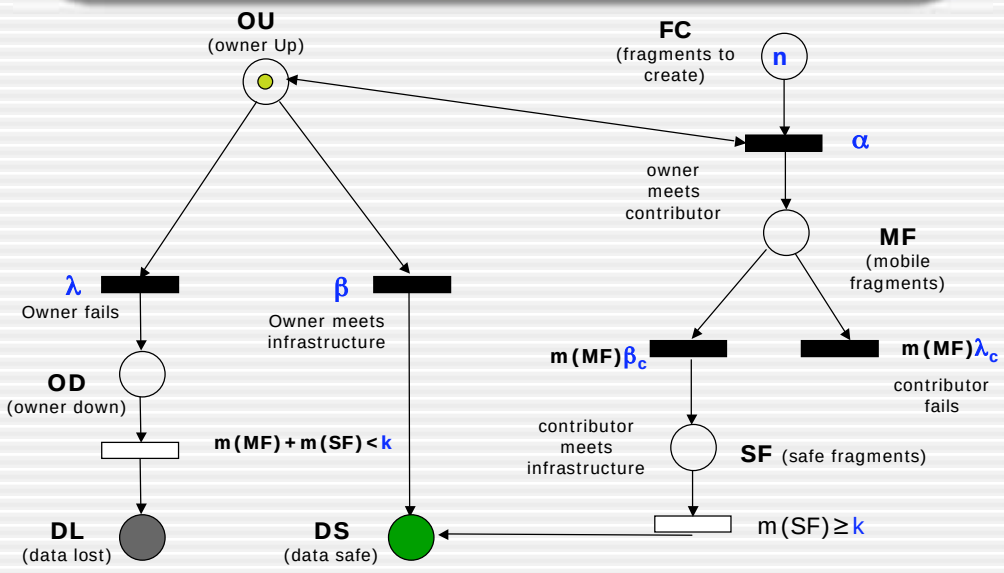
Output: n fragments

➔ n-k faults / size = n/k

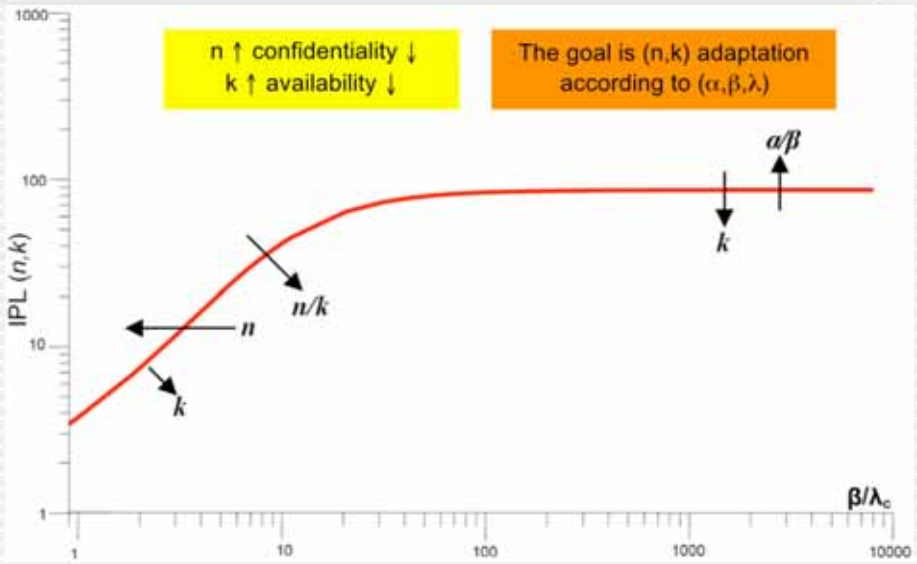
Which code to choose?
How to choose N and K?

→ Modeling and Evaluation

GSPN model for (n,k) erasure codes



Sensitivity analyses: summary



Service Availability

- Resource sharing
  - "Tragedy of the Commons" [Hardin68]
  - Free-riding (consumption without contribution)
- *Cooperation incentives*
  - *Money* (e.g., Buttyan's nuglets, claims, etc.)
    - Trade money for service
  - *Reputation*
    - Detect misbehavers, give them bad reputation
    - Don't cooperate with devices with bad reputation

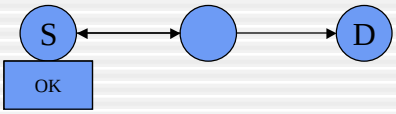
Buttyan's nuglets

- Each node maintains a counter (nuglet)
  - Decreased when sending its own packet
  - Increased when forwarding a packet
  - The counter must remain positive

- The policy must be enforced
  - Use of tamperproof hardware
    - SIMcards, JavaCards, etc.
    - TPM

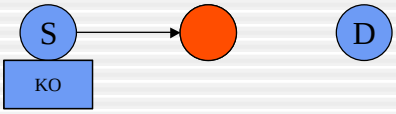
Marti's Watchdogs

- Each node possesses a watchdog
  - When a node sends a packet, the watchdog verifies that the neighbors forward it



Marti's Watchdogs

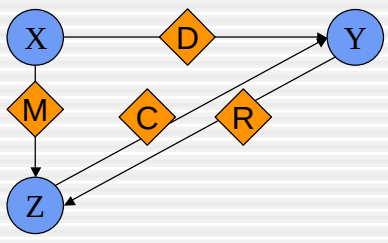
- Each node possesses a watchdog
  - When a node sends a packet, the watchdog verifies that the neighbors forward it

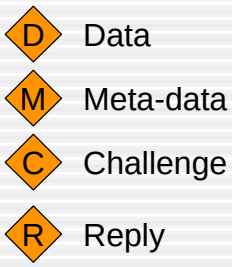


- Misbehaving nodes are detected:
  - Bad reputation
  - No cooperation

Reputation Establishment

- Reputation has to be based on cooperation observation
  - Does a contributor contributes ?
- Cooperative backup: does a contributor stores the data ?
  - Test it with challenges
  - Long-term and disconnected service
  - Challenges have to be delegated





Reputation Establishment

- Crypto-challenges that can be delegated [Oualha et al. 07]
  - Probabilistic verification
    - D — Data = Signed data
    - M — Meta-data = Public Key + # blocks
    - C — Challenge = Random block id
    - R — Reply = Signature of chosen block
- Z verifies the challenge reply to
  - Establish Y reputation
  - Choose to cooperate with Y

Current and Future work

- More general evaluation assumptions
  - Trust and cooperation wrt participating nodes (malicious, selfish)
  - Other dissemination strategies
- Adaptable Scattering Strategy
  - Online evaluation of (α,β,λ)
  - According to the user preferred policy
  - Compute and apply the best strategy
- Cooperative geo-service providing
  - A service is associated to a path
  - Nodes in the vicinity of the path cooperate to provide the service
- Failure detectors targeting cooperation faults
  - DoS attacks, Sybille attacks, etc.