BIGG Workshop

Nov 28-29, 2006

Sophia-Antipolis, France

Self-preservation in Autonomic Networking: Blending and Balancing Selfishness and Cooperation

Prof. Ioannis Stavrakakis

DEPT. OF INFORMATICS & TELECOMMUNICATIONS

http://www.di.uoa.gr/~ioannis/

ΕΘΝΙΚΟ & ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

NATIONAL & KAPODISTRIAN UNIVERSITY OF ATHENS

Funded in part by EU project CASCADAS

Networking'05 / IEEE TPDS Infocom'06 / Infocom'07 / …

…… a means of
reducing management complexity

…… a means of reducing management complexity

A MYTH ?!

Ioannis Stavrakakis
DEPT OF INFORMATICS
& TELECOMMUNICATIONS

Back to … the Greek origins:

Autonomic (Communication) Elements ←→ own laws

Back to … the Greek origins:

Autonomic (Communication) Elements ←→ own laws

- ACEs manage themselves

  ACEs undertake some of the (higher, system-wide) management burden and help improve performance, scalability, etc

Back to … the Greek origins:

Autonomic (Communication) Elements ←→ own laws

- ACEs manage themselves

  ACEs undertake some of the (higher, system-wide) management burden and help improve performance, scalability, etc

- Their interactions can also create anarchy (complexity increase)

  Overall, system-wide (management) complexity increases

Back to … the Greek origins:

Autonomic (Communication) Elements ←→ own laws

- ACEs manage themselves
  ACEs undertake some of the (higher, system-wide) management burden and help improve performance, scalability, etc

- Their interactions can also create anarchy (complexity increase)
  Overall, system-wide (management) complexity increases

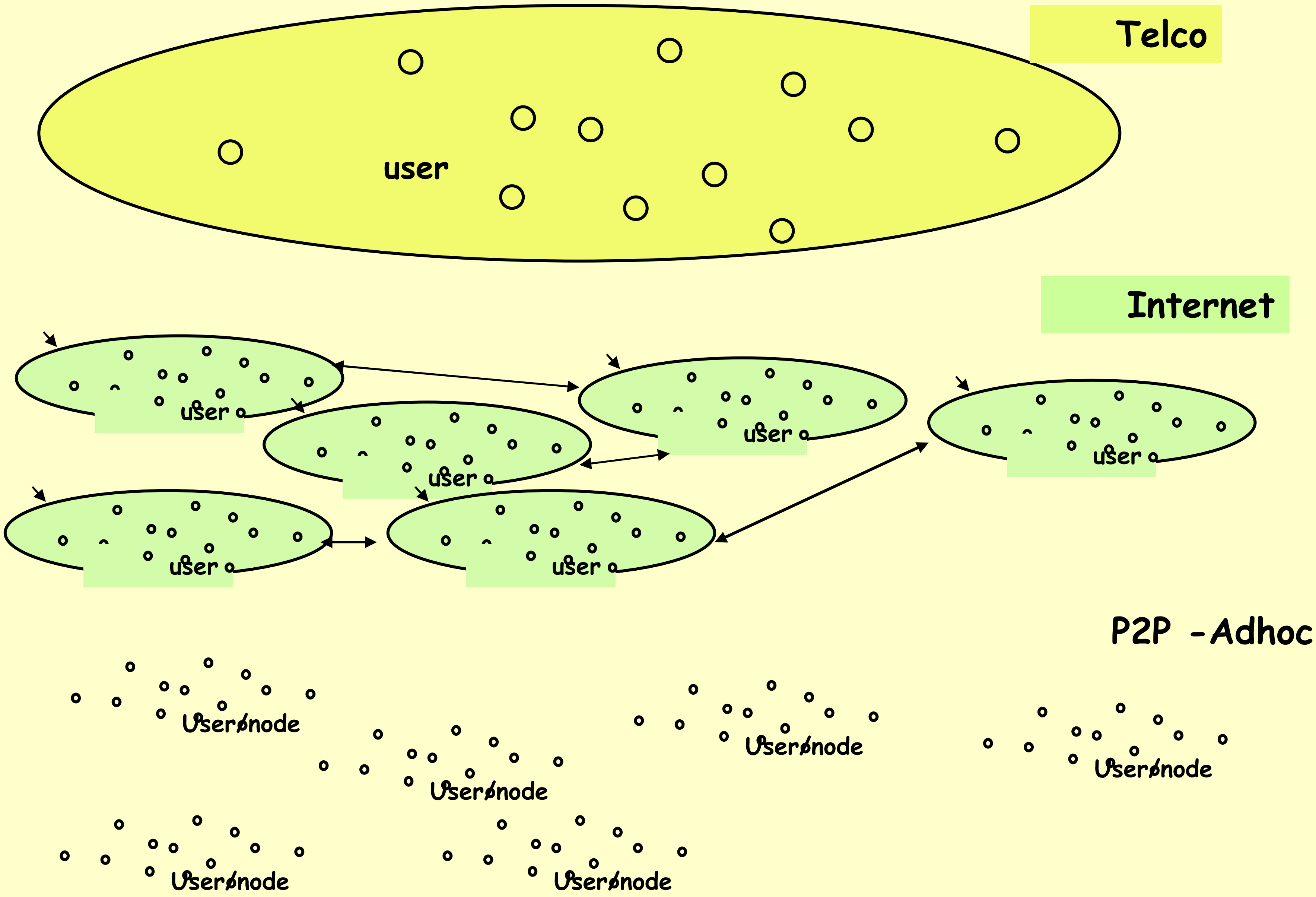
↓

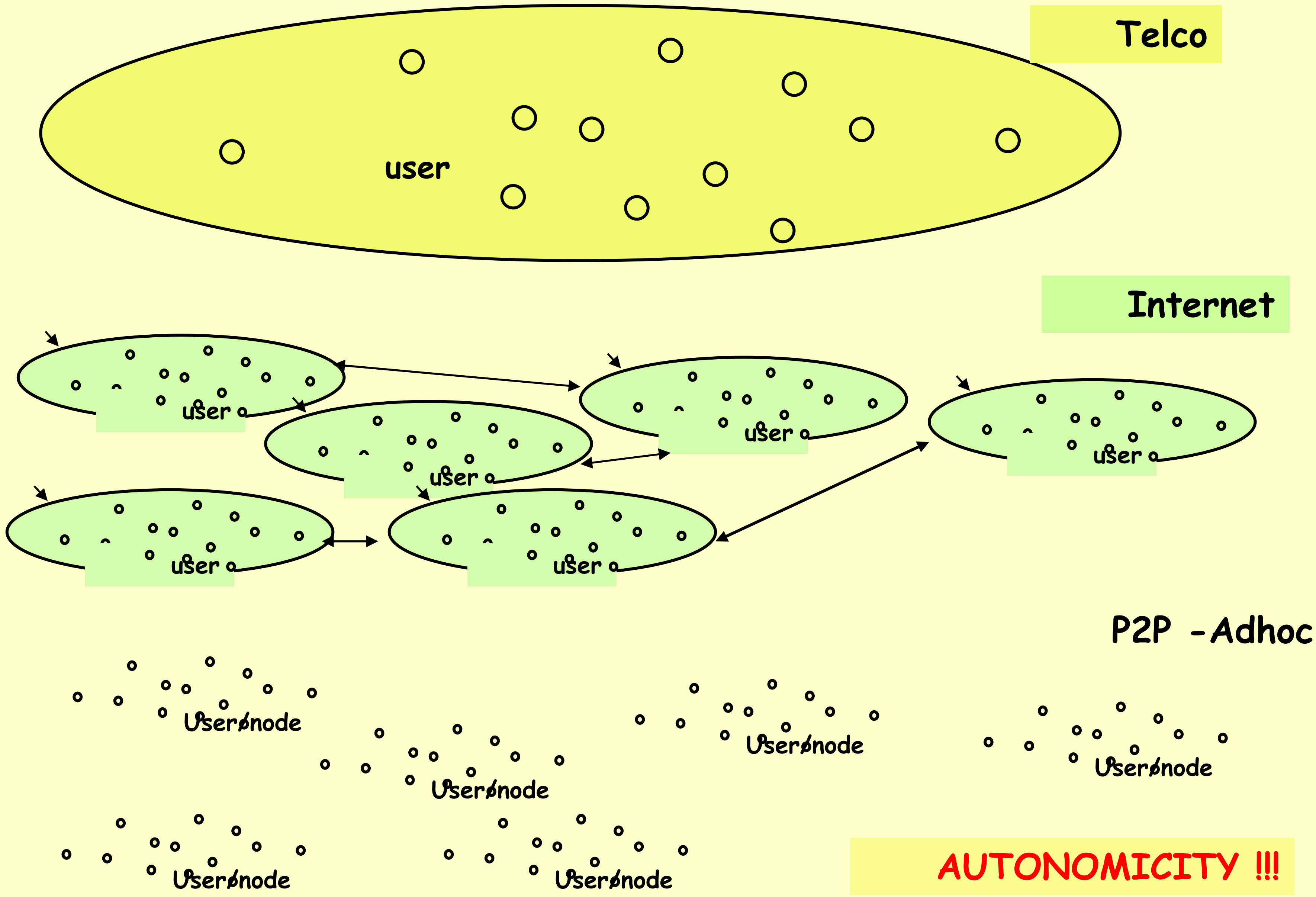
The complexity of managing an ACE-based environment can be reduced provided that ACEs' interactions are well designed

(science of interactions)

AUTONOMICITY : The emerging networking perspective

… back to … the history of networking


Telco
user
Internet
user
user
user
user
user
user
P2P -Adhoc
User node
User node
User node
User node
User node
User node



Telco
user
Internet
user
user
user
user
user
user
P2P -Adhoc
User node
User node
User node
User node
User node
User node
AUTONOMICITY !!!

AUTONOMICITY : The emerging networking perspective

- A fact of life due to decentralization of resources, ownership and management
- Most networks are formed (at least in the periphery) through Small Contributions of Large Populations of ACEs

ACEs ←→ own laws (behavior) + independently owned

→ ACEs are selfish (own utility maximization)

→ ACEs should cooperate to form a network or enhance service

Blending and Balancing

Selfishness and Cooperation

is key to

(user and network)

self-preservation

(soft-security management)

in

Autonomic Networking

Self Preservation through Game theory (GT) and Mechanism Design (MD)

ACEs (with potentially conflicting interests) aim to maximize their utility

A macroscopic approach

Certain level of stationarity assumed (off-line design)

If ACEs are cheating, detection and reaction mechanisms are needed (see also next)

Self Preservation through an "engineering" approach

A microscopic approach to self-preservation, aiming at designing the detailed workings of the various elements that make up complex systems

Design mechanisms that are fundamentally open to cooperation, interaction, and sharing of resources with other such entities and mechanisms, provided that these do not harm its local utility (or more accurately the utility of some rational owner(s) that control this entity).

Self-preservation is just an additional functional layer whose purpose is to remain quiet (inactive) as long as the interactions between other entities are beneficial and intervene by modulating the local behavior only when these interactions lead to *mistreatment*, defined here as the utility level below which an ACE is considered to be under-performing.

Self Preservation through an "engineering" approach

- Applied to "open systems" in which group settings (e.g., number of nodes, distances, demand patterns) change dynamically ("off-line" optimizations are not possible).

- Rational entities should adjust their scheme dynamically so as to avoid or respond to mistreatment if and when it emerges.

Self Preservation through an "engineering" approach

Detection Mechanism: *How can a node realize that it is being mistreated?*

- "reference point" for a mistreatment test (threshold)
  - Assuming *__a priori knowledge__* (of key parameters and certain stationarity level): compute greedy local cost thresholds (used in a game-theoretic framework).

  - Assuming *__NO priori knowledge:__* estimate and update thresholds in an on-line manner.

    A generic promising approach for this is *__emulation.__* A node follows some specific behavior and at the same time emulates other alternative behaviors (some more, some less cooperative) and switches between them when their relative performances with respect to the local utility changes.

Self Preservation through an "engineering" approach

- Mitigation Mechanism

  to react to mistreatment (e.g., object admission control mechanism used by the nodes to decide whether to cache or not an incoming object)

- Control Scheme

  A programmatic scheme for controlling the parameter(s) that affect the mitigation mechanism. (e.g., set the prob of admitting an object)

Example

Self-Preservation of independently owned storage resources

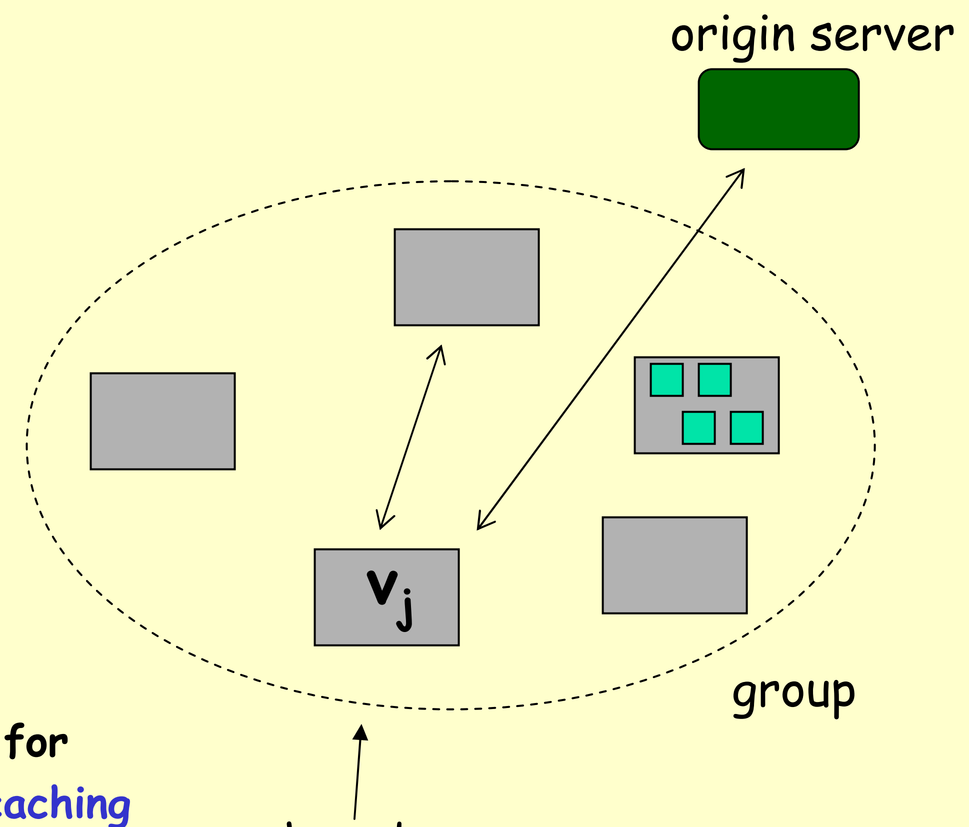
A distributed replication/caching group

Applications

Content distribution

Shared memory

Network file systems



a collection of

- **n** nodes
- **N** objects

nodes storage used for either replication or caching

replication → **permanent copies**

caching → **temporary copies**

each node

- local client population
- generates requests (preference profile over the N objects)
- requests serviced by the "closest copy"
- fixed amount of storage capacity

Node selfishness brings a new perspective

the **traditional approach**:

- entire group under common control
- find replication/caching strategies to minimize the access cost of the <u>entire</u> group

but a **selfish node**:

- wants to minimize (or guarantee some level of) the access cost of local users <u>only</u>
- better model for applications with:
  - multiple/independent authorities
  - e.g., P2P, distributed web-caching

Therefore, **two new research questions**:

- "object replication under selfish nodes?" (off-line problem)
- "object caching under selfish nodes?" (on-line problem)

Example

Self Preservation of independently owned storage resources

*Nikolaos Laoutaris, Orestis Telelis, Vassilios Zissimopoulos, and Ioannis Stavrakakis. "**Distributed selfish replication**" IEEE Transactions on Parallel and Distributed Systems, 17(12):1401–1413, December 2006.*

*Nikolaos Laoutaris, Georgios Smaragdakis, Azer Bestavros, and Ioannis Stavrakakis. "**Mistreatment in distributed caching groups: Causes and Implications**". In Proceedings of IEEE INFOCOM'06, Barcelona, Spain, April 2006. (also in "**Distributed Selfish Caching**" accepted to IEEE TPDS)*

Self-Preservation under cooperation uncertainty

Although cooperation is "agreed" or expected, selfish behavior or other reasons may hinder it.

- ACEs may be unavailable
- ACEs may not cooperate as expected

Self-preservation in ACE-centric networking requires:

- Designing networking protocols that are effective or robust in the presence of (cooperation) uncertainties
- Employing trust/reputation/incentives mechanisms to enhance performance and robustness
- ……………………………………………

Designing networking protocols that are effective or robust in the presence of (cooperation) uncertainties

Example: P2P Streaming

Data-preserving vs Delay-preserving playout strategies

Delay-preserving strategies less sensitive to node churn

*C. Vassilakis, N. Laoutaris, I. Stavrakakis, "**On the Benefits of Synchronized Playout in Peer to Peer Streaming**", CoNEXT'06 poster presentation.*

Designing networking protocols that are effective or robust in the presence of (cooperation) uncertainties

Example: Routing in Opportunistic, Delay Tolerant Networks

Protocols under which critical control processes (e.g., the message spreading process) are controlled by inherently committed ACEs are more robust to uncooperativeness

When cooperation is not guaranteed (<100%), can outperform protocols that are better under fully cooperative environments

*A. Panagakis, A. Vaios, I. Stavrakakis, "**On the Effects of Cooperation in Delay Tolerant Networks**", IEEE COMSWARE'07, Jan. 8-12, 2007, Bangalore, India*

Designing networking protocols that are effective or robust in the presence of (cooperation) uncertainties

Example: Routing in Opportunistic, Delay Tolerant Networks

Protocols that are modulated by ACE reputation info are more robust to uncooperativeness

When cooperation is not guaranteed (<100%), engaging only "reputable" (above a threshold) ACEs, can outperform protocols that are blind to reputation and attempt to exploit all resources (not obvious)

*A. Panagakis, A. Vaios, I. Stavrakakis, "**On the Effects of Cooperation in Delay Tolerant Networks**", IEEE COMSWARE'07, Jan. 8-12, 2007, Bangalore, India*

Concluding:

Networking amounts to distributed resource usage and cooperation

Autonomicity amounts to self-managed/controlled and own-utility cautious entities

Autonomic Networking is very different than traditional networking

Concluding:

Autonomic Networking ⟷ **Self-preserving Networking**

Mistreatment-resilient Networking

- -satisfy own-utility concerns of ACEs
- -monitor and adapt rules of cooperation
- -protect against rational/irrational misbehaviors
- -design for uncertainly
- -incorporate reputation measures in protocol design
- - ………………….