



# Towards a Peer-Assisted Content Delivery Architecture

Bogdan Florescu      bogdan.florescu@cti.pub.ro  
Mugurel Ionuț Andreica      mugurel.andreica@cs.pub.ro



# Contents

- Introduction
- Hybrid architecture
- Servers
- Peers
- Peer stimulation policies
- Network simulator
- Related work
- Conclusions



# Introduction

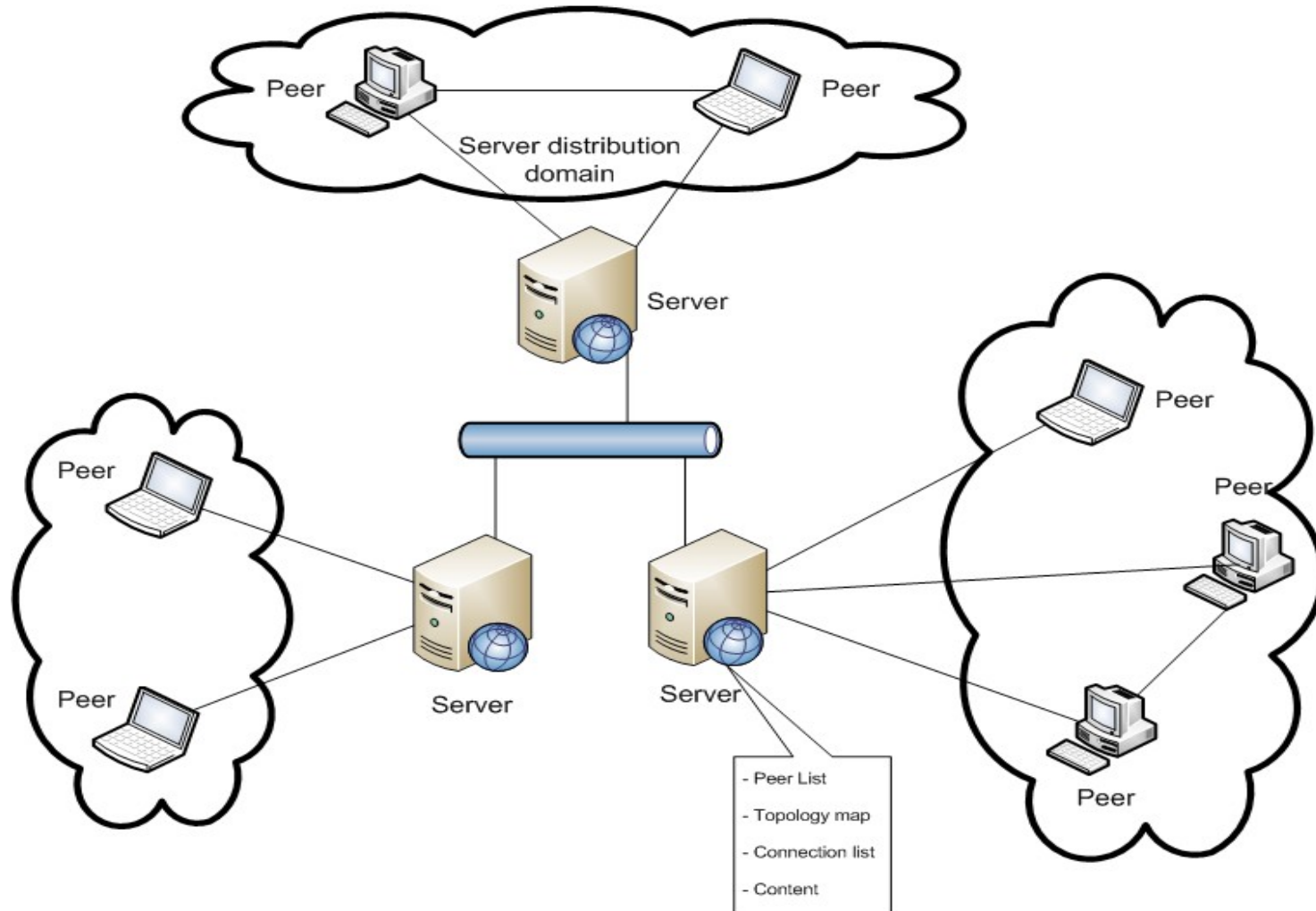
- Traditional CDNs
  - Client – server architecture
  - Dedicated storage capacity
  - High-bandwidth connection
  - Easy data backup and security control
  - High deployment and maintenance costs
- Peer-to-peer systems
  - Great scalability
  - Leverage clients' resources
  - Security issues
  - No guarantee for content availability



# Hybrid Architecture

- Goal
  - Retain both models' benefits
  - Eliminate disadvantages
- Structure
  - Client – server core
  - Peer-to-peer additional layer
- Servers
  - Maintain content availability
- Peers
  - Take over server load

# Architecture Overview





# Peers

- Can connect to
  - CDN server
  - Other peers in server's list
- Enter / leave network at any time
  - No effect on remaining nodes communication
- Three states
  - Uploading – seeder
    - Lightweight distribution server
  - Downloading – leecher
    - Content consumer
  - Both



# CDN Servers

- Primary source for data
  - Ensure availability
- Maintain lists of connected peers
- Maintain topology map
- Maintain list of active connections
- Enforce policies
  - peer motivation and fair contribution
  - security



# Peer Stimulation Policies

- Role
  - Effectively use the P2P layer
- Bittorrent “ratio” + credit system
  - Ratio
    - $R = U / D$
    - Per file
    - Overall per client
  - Credits
    - for every ratio  $R \geq 1$ , the peer receives  $[R]$  credits
    - every download consumes 1 credit
    - provide
      - better download speed
      - higher priority for future data transfers
      - early access to specific content





## Peer Stimulation Policies (2)

- Alternative ratio
  - For clients with low-bandwidth connection
  - $R = T_u / T_d$ , where
    - $T_u$  = upload time
    - $T_d$  = download time
  - Server decides the appropriate method
- Dynamically limit transfer speeds for peers
  - ensure the download ends after  $R \geq 1$
  - computed considering various parameters
    - connection speed
    - file size being
    - peer past behavior etc.



# Simulator

- Network model
  - Undirected graph
    - N servers with different storage capacities
    - M data objects with different sizes
    - P clients
    - Full mesh connection graph
      - Between CDN servers
      - Between CDN server and its clients
    - Connection cost per unit of data
- Entities
  - EndPoint (server / client)
  - Connection
  - Request
  - SimulationManager



# Related work

- Pando Networks [1]
  - Commercial solution
  - Successfully deployed
- P2P Extension for CDNs [2]
  - Theoretical model
  - Security emphasis
- Media streaming in CDN-P2P [3,4]
  - Two different projects
  - Introduce contribution policies



# Conclusions

- No single model can ensure performance, scalability and reduced costs at the same time
- Our hybrid design combines peer-to-peer and client-server CDN architectures to achieve that goal
- Although relatively new, related work in the area prove model's viability

# Questions?





# Bibliography

- [1] Pando Networks (2011). <http://www.pandonetworks.com/>
- [2] Pakkala D. and J. Latvakoski (2005). Towards a Peer-to-Peer Extended Content Delivery Network. In: *Proceedings of the 14th IST Mobile & Wireless Communications Summit*
- [3] Xu, D., S. S. Kulkarniz, C. Rosenbergz, H.-K. Chaiz (2006). A CDN-P2P Hybrid Architecture for Cost-Effective Streaming Media Distribution. In: *Multimedia Systems*, Vol. 11 (4), pp. 383-399
- [4] Ha, D. (2008). A Novel Hybrid CDN-P2P Mechanism for Effective Real-Time Media Streaming. M.Sc. Thesis.