Distributing streaming media content using cooperative networking

Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou
Microsoft Research

Kunwadee Sripanidkulchaip
Carnegie Mellon University

NOSSDAV'02, May 12-14, 2002, Miami, Florida, USA.

Content distributions of today

(a) infrastructure-based content distribution

- Akamai

(b)peer-to-peer content distribution

- BitTorrent

Paper Focus - Coop-Net

- Cooperative Networking
  - combines aspects of infrastructure-based and peer-to-peer content distribution
- Distributing streaming media content
  - live and on-demand
- Evolution, not revolution
  - complement rather than replace the traditional client-server framework.

Key Concept of CoopNet

- Addresses overload problem by having clients cooperate with each other to distribute content, thereby alleviating the load on the server

Why not pure p2p model

- Access of resourceful servers that hosts content and (directly) serves clients.
    - *presence of a central server simplifies a lot*

- CoopNet is only invoked when the server is unable to handle the load imposed by clients

Must have

- Mechanism that is robust against interruptions caused by the frequent joining and leaving of individual peers.
    - Paper focus on the distruption and packet loss caused by node arrivals and departures

Main weapon

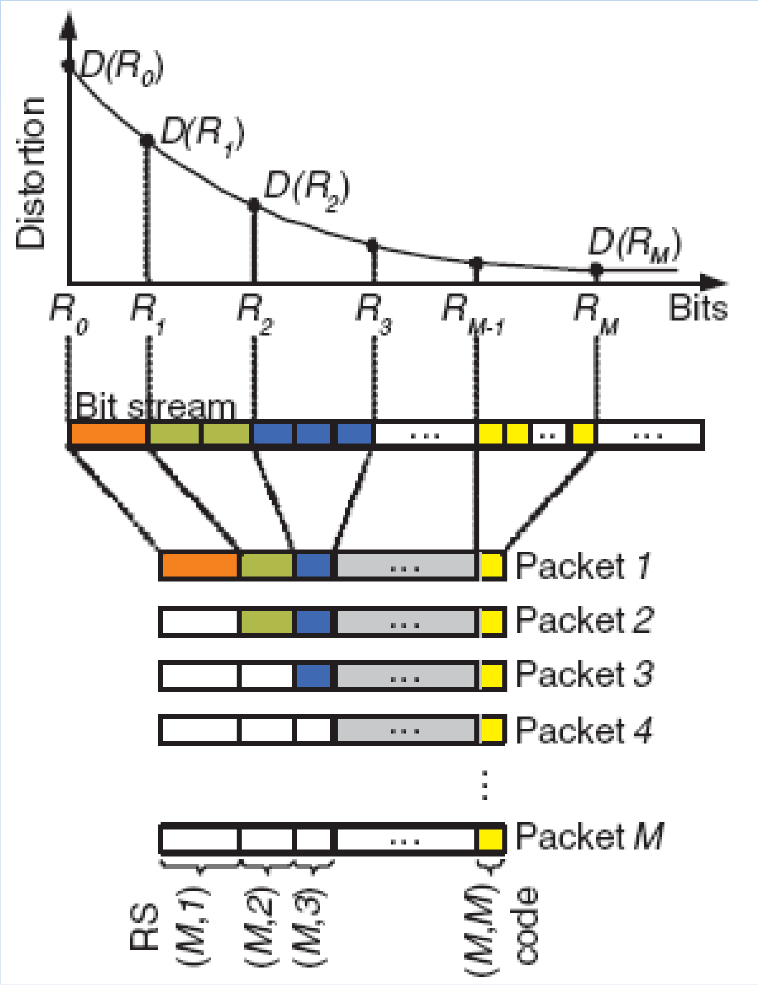
- CoopNet employs multiple description coding (MDC)
  - *The streaming media content, whether live or on-demand, is divided into multiple sub-streams*

Multiple Description Coding (MDC)

- M > 1 separate streams
- every subset of descriptions must be decodable

Multiple Description Coding (MDC)

- GoF(Group of Frames) 1 sec– Bit Streams
- R_m – m Received bits
- $D(R_m)$ – Distortion



- M packets are equally important; only the number of re-eived packets determines the reconstruction quality of the GOF.

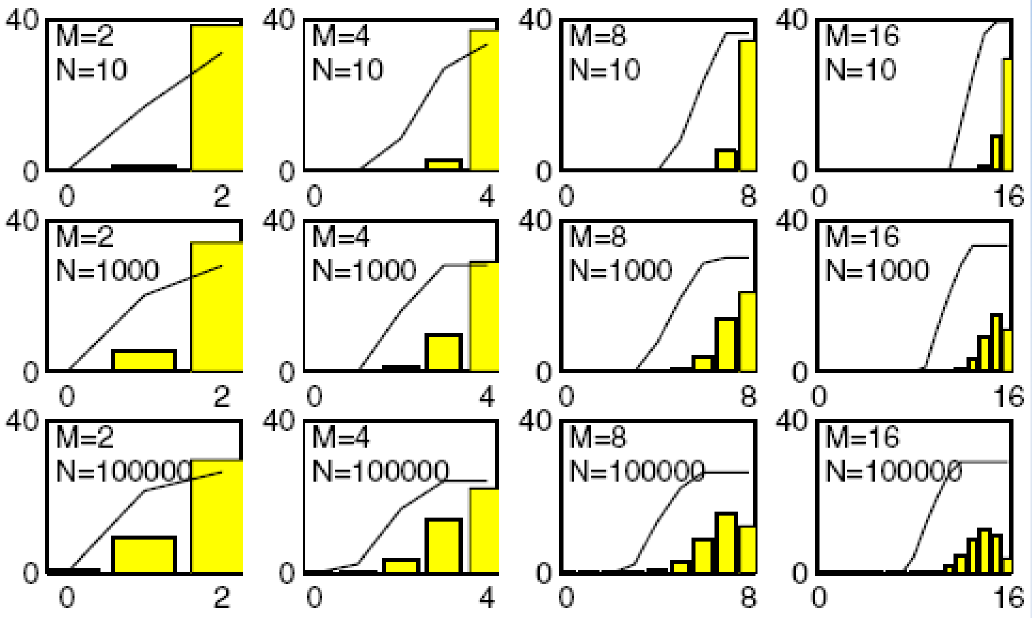
CoopNet Analysis: Quality During Multiple Failures

- AV signal into M descriptions(GoF)
  - *M different distribution trees*
  - *each rooted at the server(central point)*

- N destination hosts
  - *N destination hosts receive all M descriptions*
  - *a host n will receive the mth description if none of its ancestors in the mth tree fail.*
    - *Deeper trees means problem*

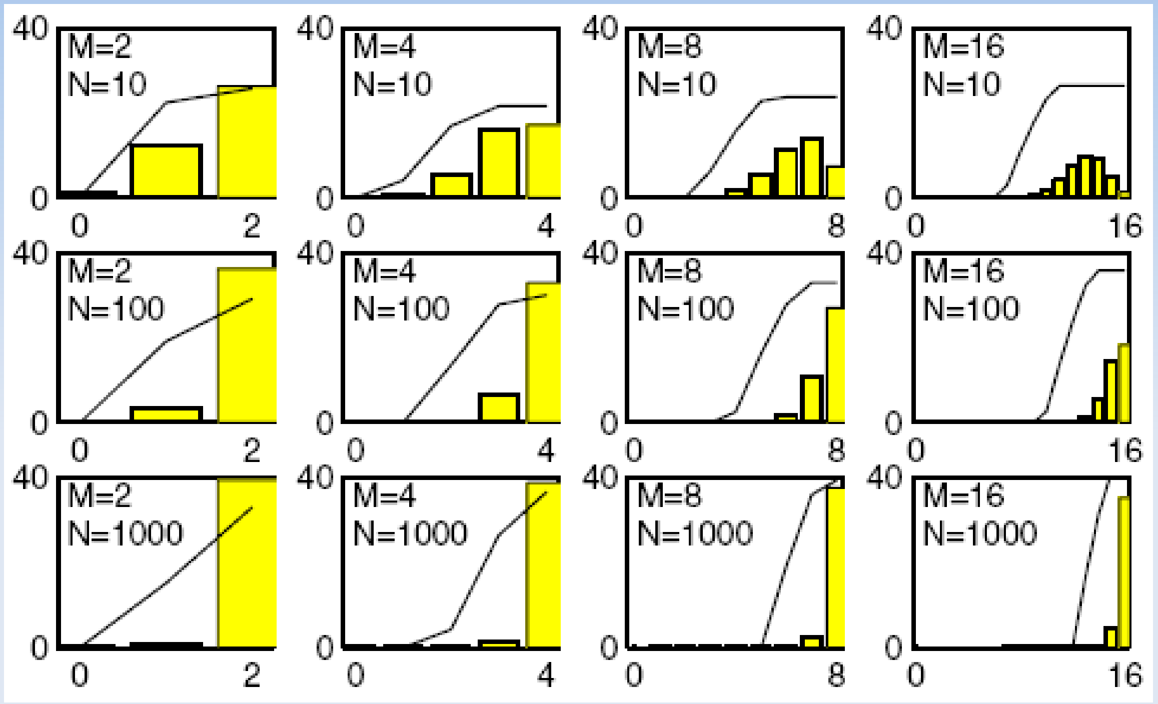
CoopNet Analysis: Quality During Multiple Failures

- Function of the number of description received
  - Line (Quality)
  - Box (prob. distribution)
- N hosts
- M segments per GoF



CoopNet Analysis: Quality During Multiple Failures

- as M increases, for fixed N, the distribution again becomes Gaussian
- hosts that receive 100% decreases. However of hosts receive fewer than 50% decreases
  - *resulting in an increase in quality on average.*



Tree Management

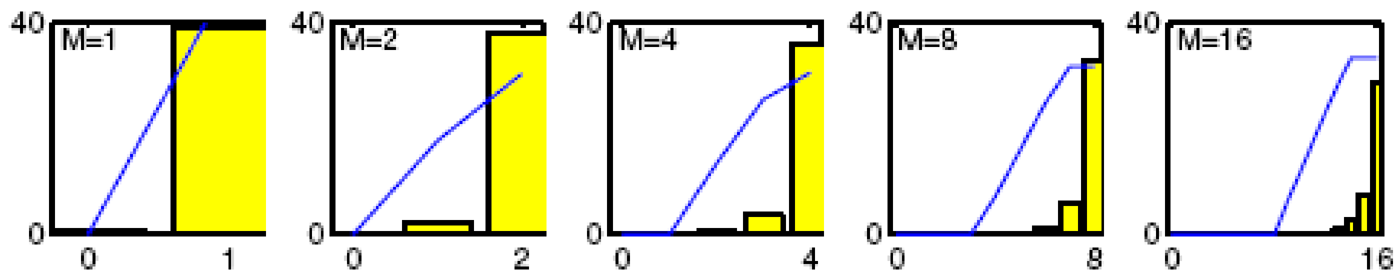
- Short and wide tree:
  - short to minimize the latency
  - wide as much as its bandwidth will allow
- Eficiency versus tree diversity(conflict):
  - Eficiency by reflect the underlying network topology
  - Diversity, generated by random, makes more hobust
- Quick join and leave
- Scalability

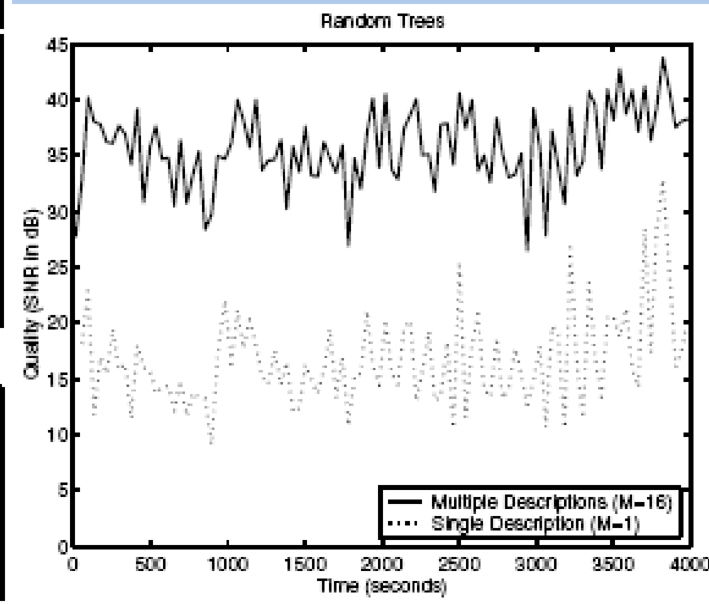
Recap

- server is not overloaded since the burden of distributing content is shared by all peers
- Centralization makes things simpler and faster
  - *server has full knowledge of the topology*
- *Most departures are graceful*

Effectiveness of MDC

M	100%	[87.5,100)	[75,87.5)	[50,75)	[25,50)	0
1	98.1	0	0	0	0	1.90
2	94.80	0	0	5.05	0	0.16
4	89.54	0	9.24	1.13	0.09	0.005
8	82.07	14.02	3.19	0.70	0.016	0
16	71.26	25.11	3.26	0.37	0.002	0



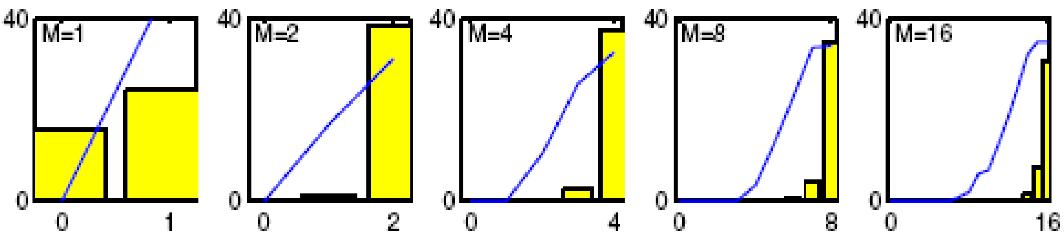


- 2 descriptions -> 94.80% of clients receive 100% of the descriptions
- 8 -> 96% (82.07% + 14.02%) of clients receive more than 87.5% of the descriptions

Impact of Repair Time: Rebuild tree after departure

M	100%	[87.5,100)	[75,87.5)	[50,75)	[25,50)	0
1	98.34	0	0	0	0	1.66
2	96.5	0	0	3.42	0	0.08
4	93.3	0	6.31	0.36	0.03	0
8	87.14	11.34	1.29	0.20	0.02	0
16	77.26	21.62	0.99	0.11	0.01	0

Table 2: Evolving Tree Experiment: probability distribution of descriptions received vs. number of distribution trees



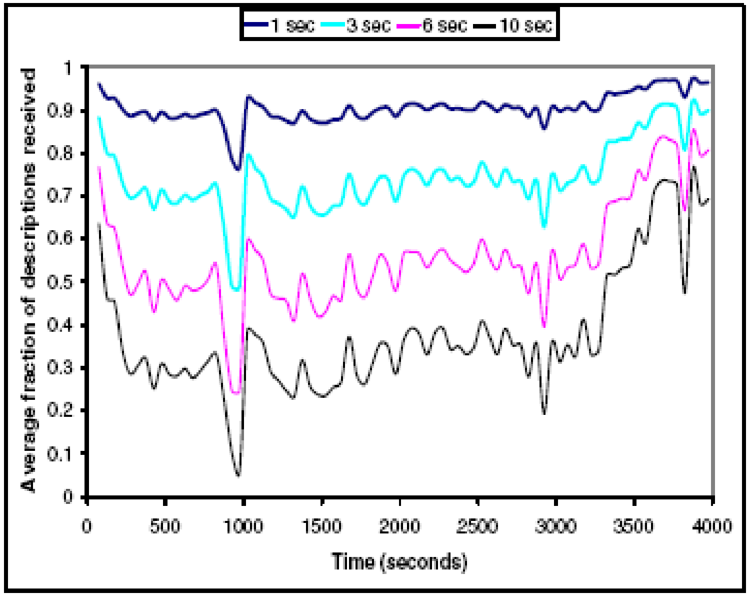
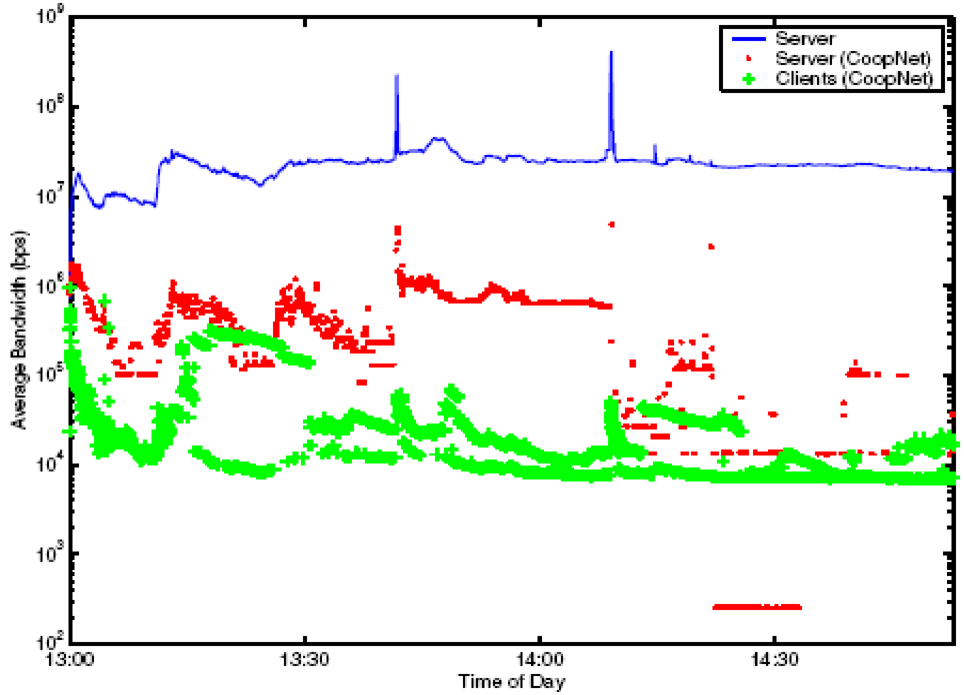
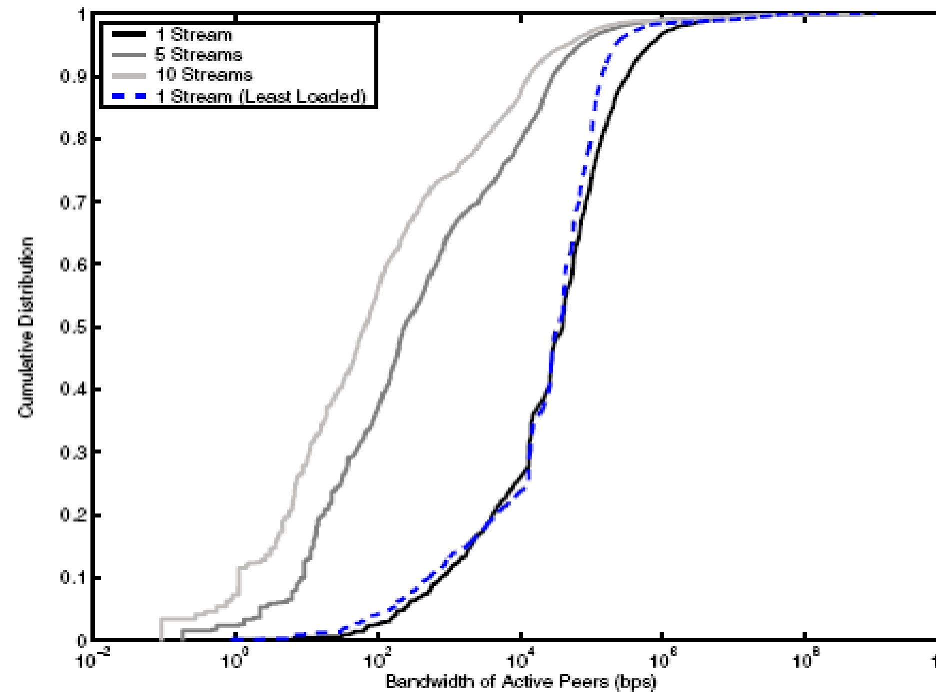


Figure 8: The average fraction of descriptions received for various repair times.

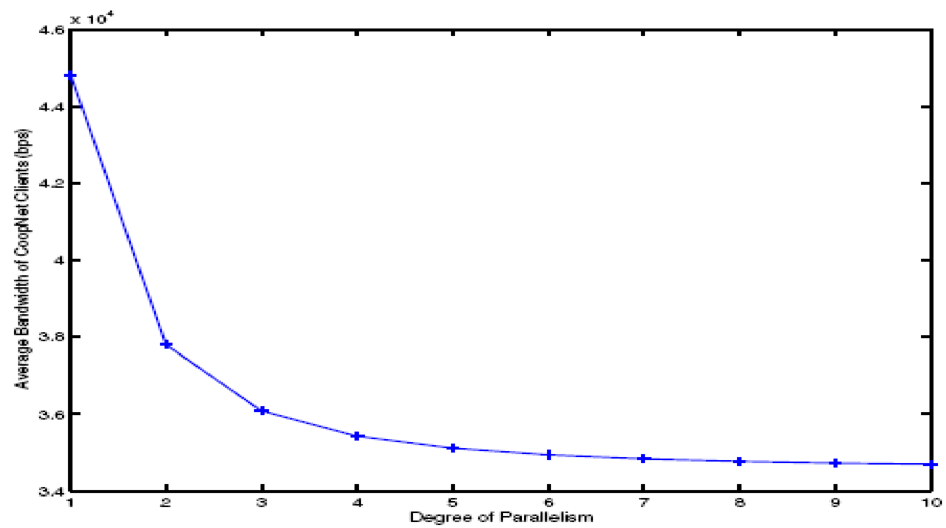
On-Demand Streaming



(a) Average bandwidth at server and cooperating peers.



(c) Distribution of bandwidth at active peers.



(b) Average bandwidth at peers when using distributed streaming.

Conclusão

- Solução tão boa quanto a qualidade da arvore e seu gerenciamento.
- Poderia aumentar muito a capacidade se utilizasse serviços tipo akamai em conjunto com o servidor central.
- Uso de PDN poderia facilitar a implementação.
  - No cliente por separar o código de dar suporte a CoopNet por uso de um protocolo de uso comum e geral, socks.
  - No servidor por facilitar o uso de redes como Akamai.