Tradeoffs in Cloud and Peer-assisted Content Delivery Systems

Niklas Carlsson — **Linköping University**

György Dan — KTH Royal Institute of Technology

Derek Eager — University of Saskatchewan

Anirban Mahanti — NICTA

September 5, 2012

LiU expanding reality

Motivation

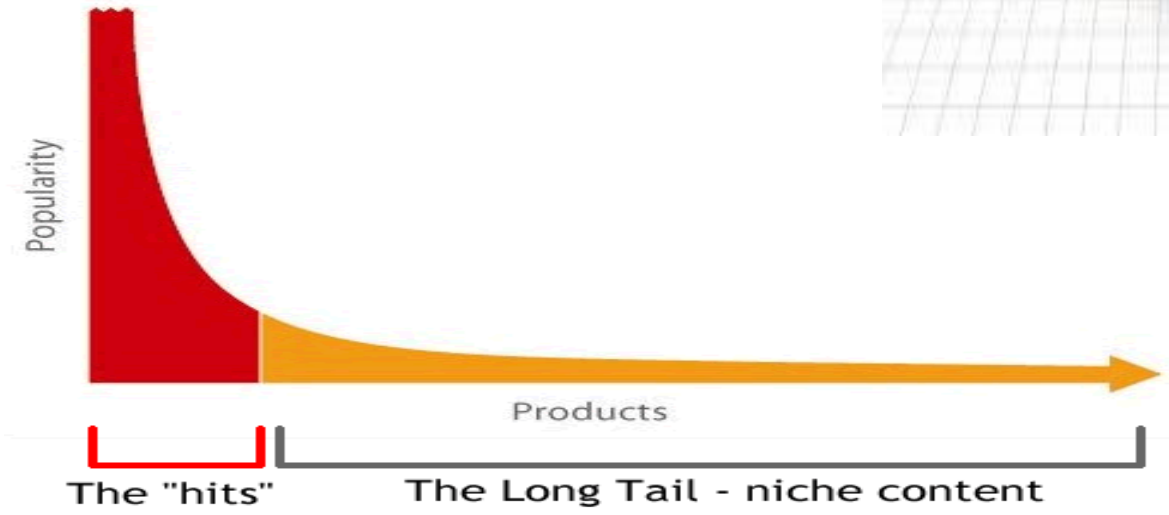
- Content provider wanting to minimize its delivery cost
  - Catalogue of many contents
  - Different popularity
  - Average service guarantees

Motivation

- Cost-efficient solution must scale with regards to both:
  - Request rate
  - Number of availible contents

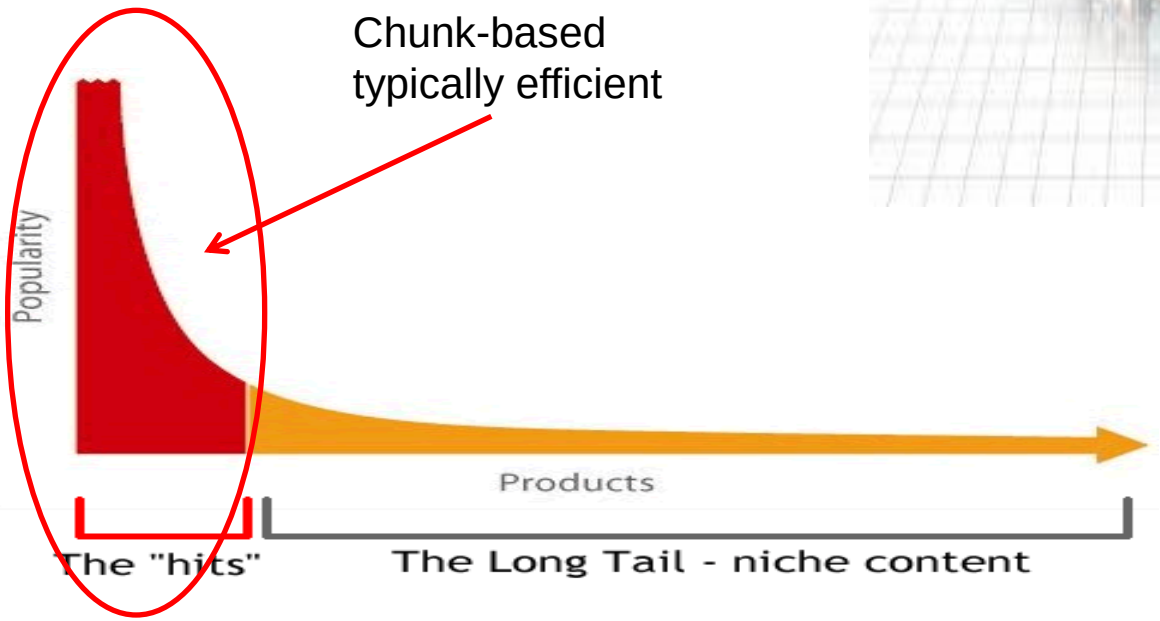
Motivation

- Cost-efficient solution must scale with regards to both:
  - Request rate
  - Number of availible contents



Motivation

- Cost-efficient solution must scale with regards to both:
  - Request rate
  - Number of availible contents

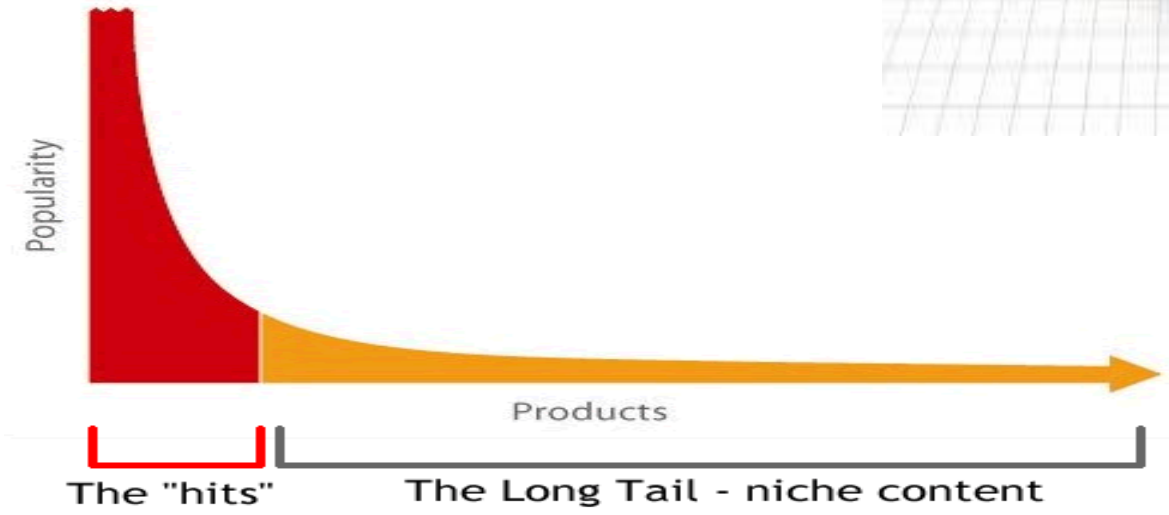


Motivation

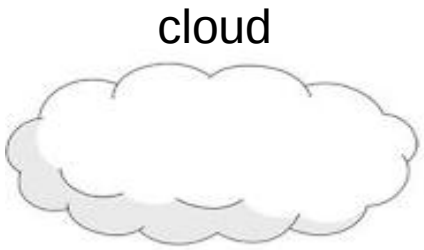
- Cost-efficient solution must scale with regards to both:
  - Request rate
  - Number of availible contents

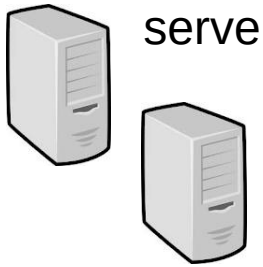
Motivation

- Cost-efficient solution must scale with regards to both:
  - Request rate
  - Number of availible contents



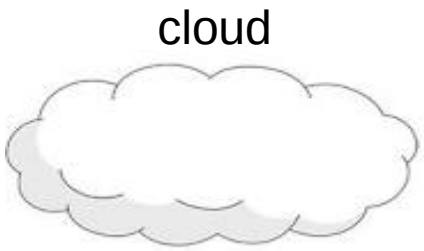
Cost model

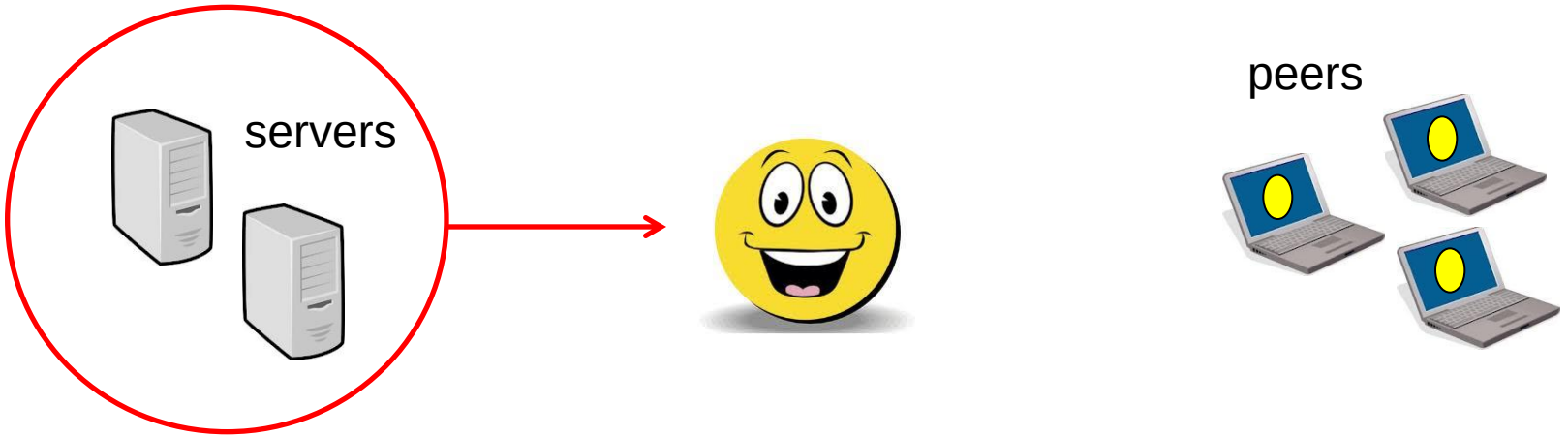




- Client can download from either
  - Origin servers (all contents)
  - Cloud storage/servers (subset of contents)
  - Other clients (peers)

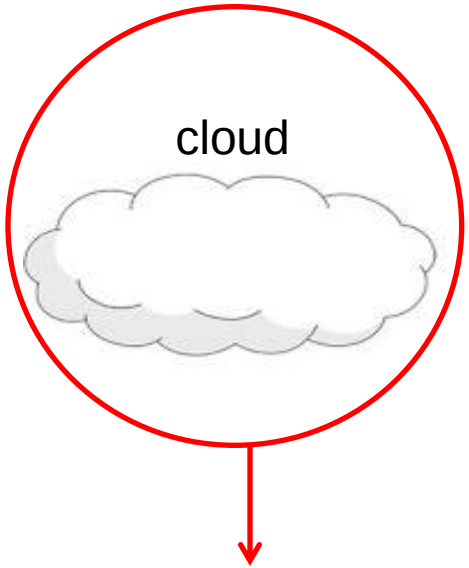
Cost model

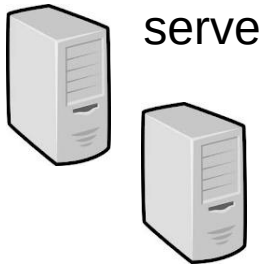




- Client can download from either
  - Origin servers (all contents)
  - Cloud storage/servers (subset of contents)
  - Other clients (peers)

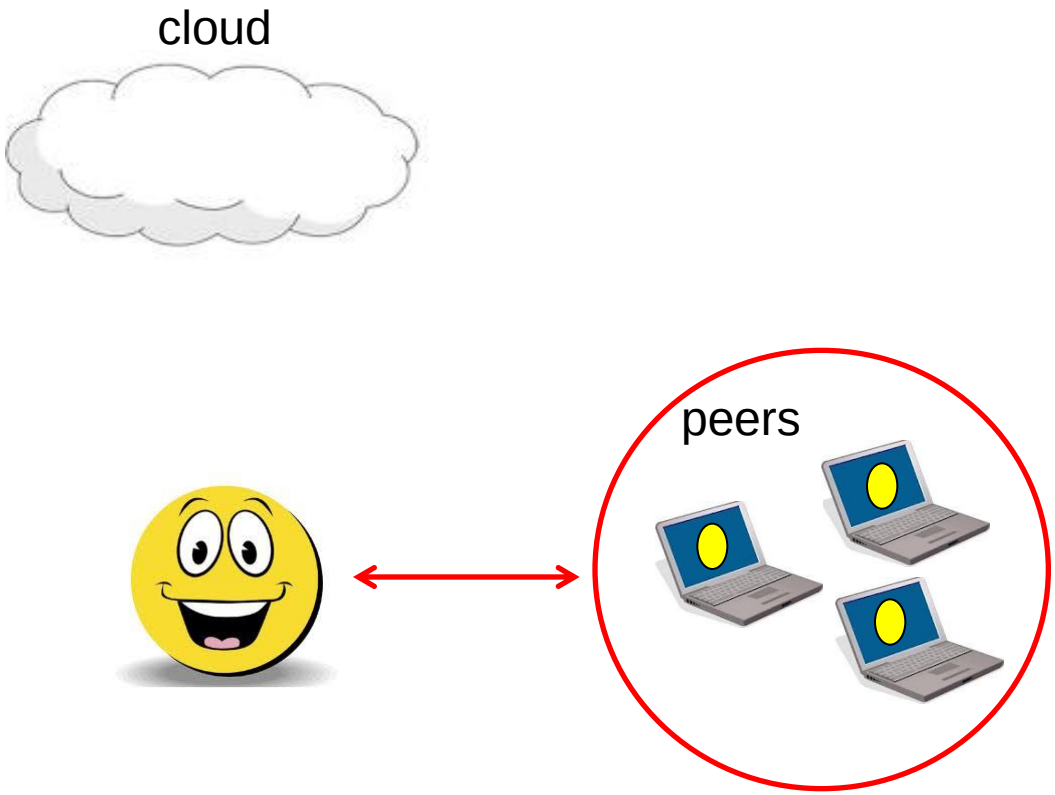
Cost model





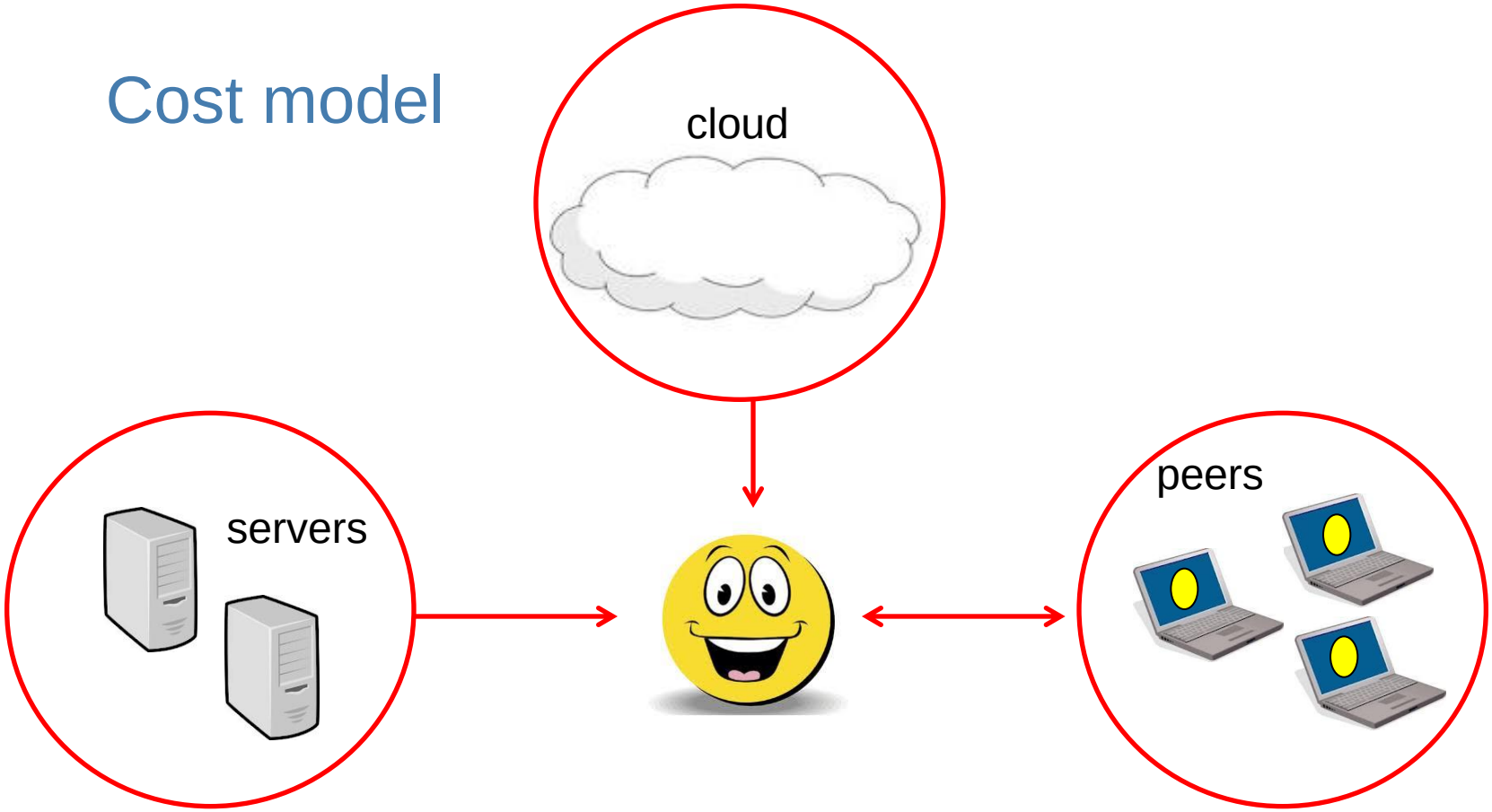
- Client can download from either
  - Origin servers (all contents)
  - Cloud storage/servers (subset of contents)
  - Other clients (peers)

Cost model



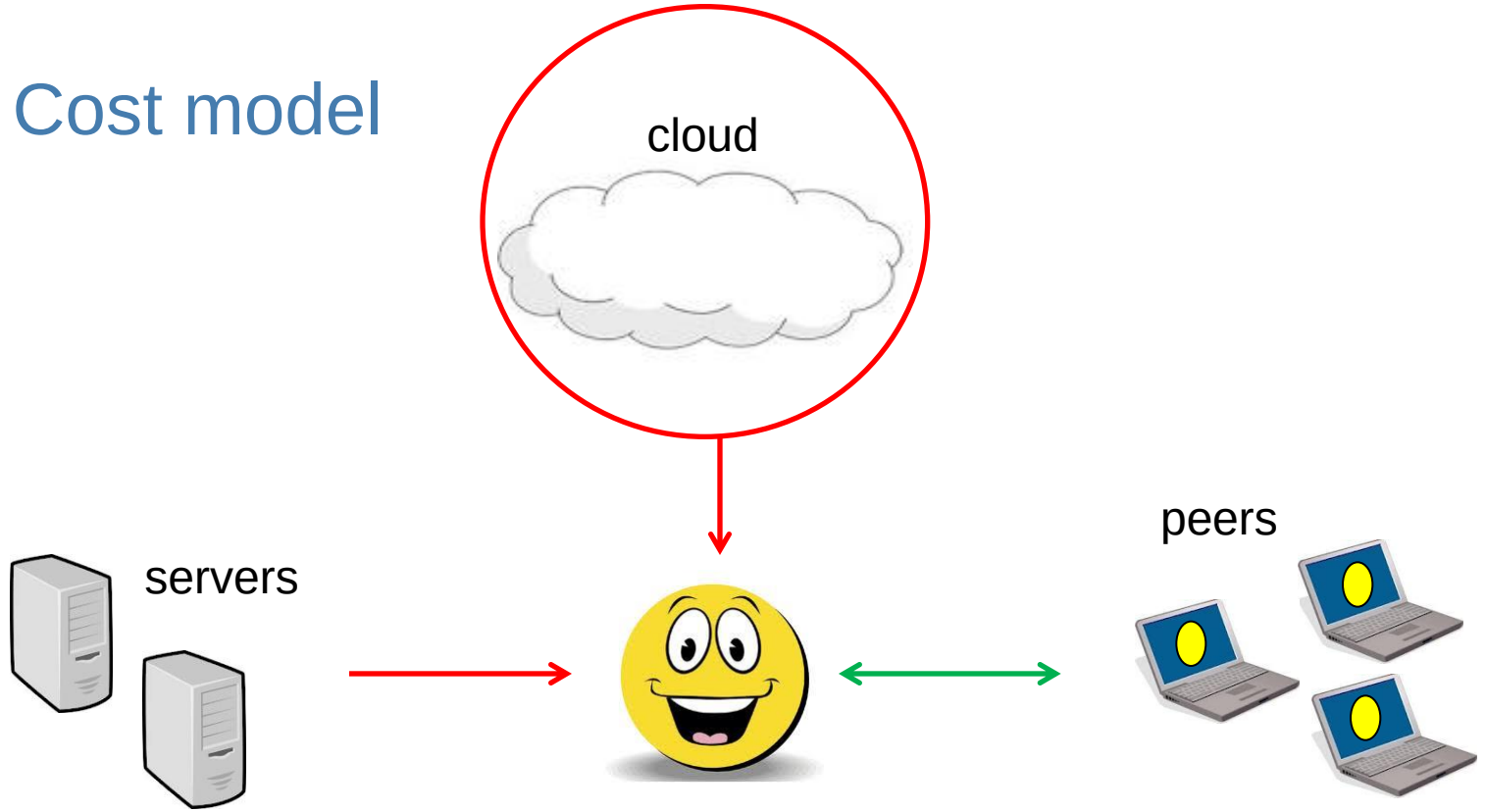
- Client can download from either
  - Origin servers (all contents)
  - Cloud storage/servers (subset of contents)
  - Other clients (peers)

Cost model



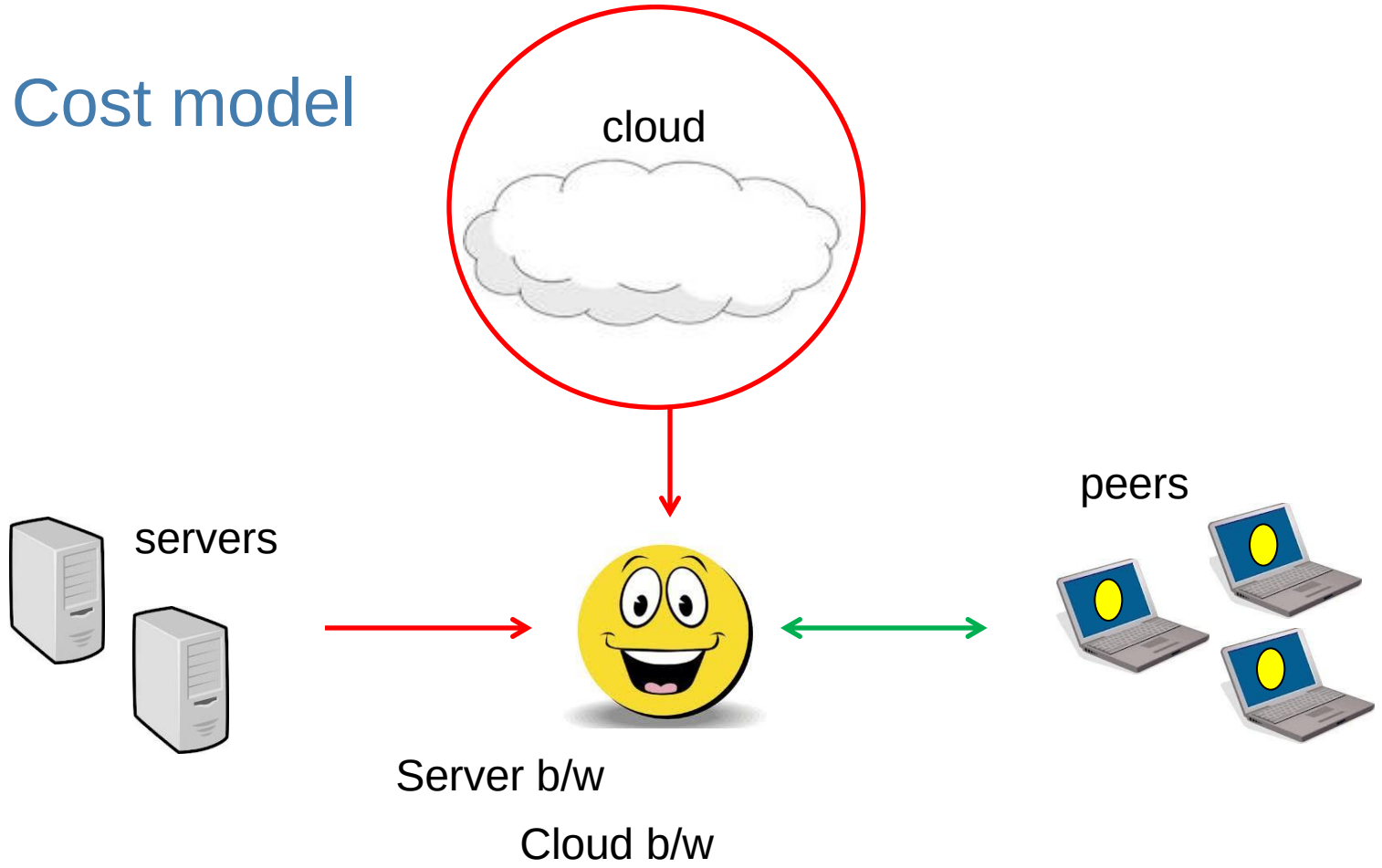
- Client can download from either
  - Origin servers (all contents)
  - Cloud storage/servers (some contents)
  - Other clients (peers)

Cost model



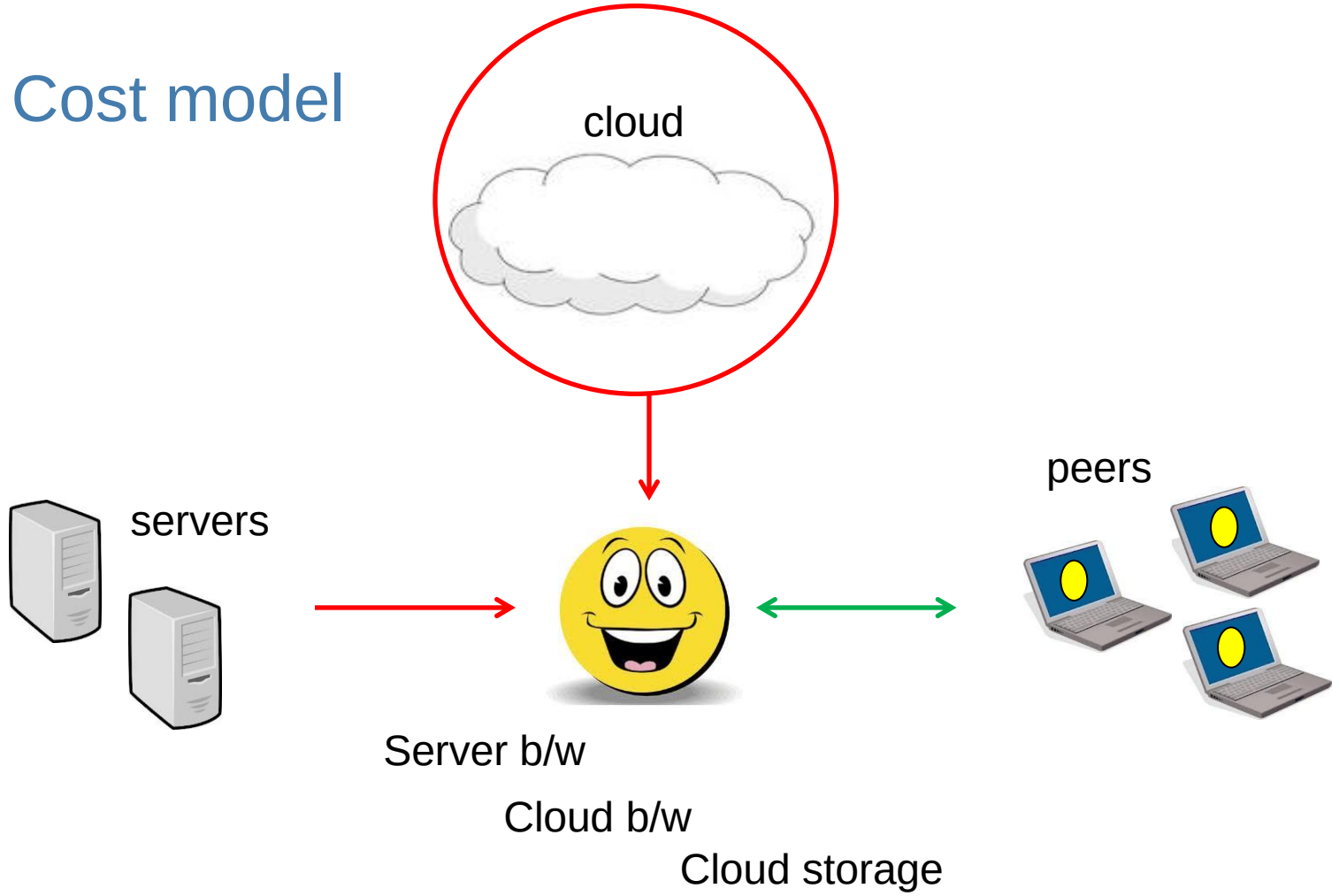
- Client can download from either
  - Origin servers (all contents)
  - Cloud storage/servers (subset of contents)
  - Other clients (peers)

Cost model

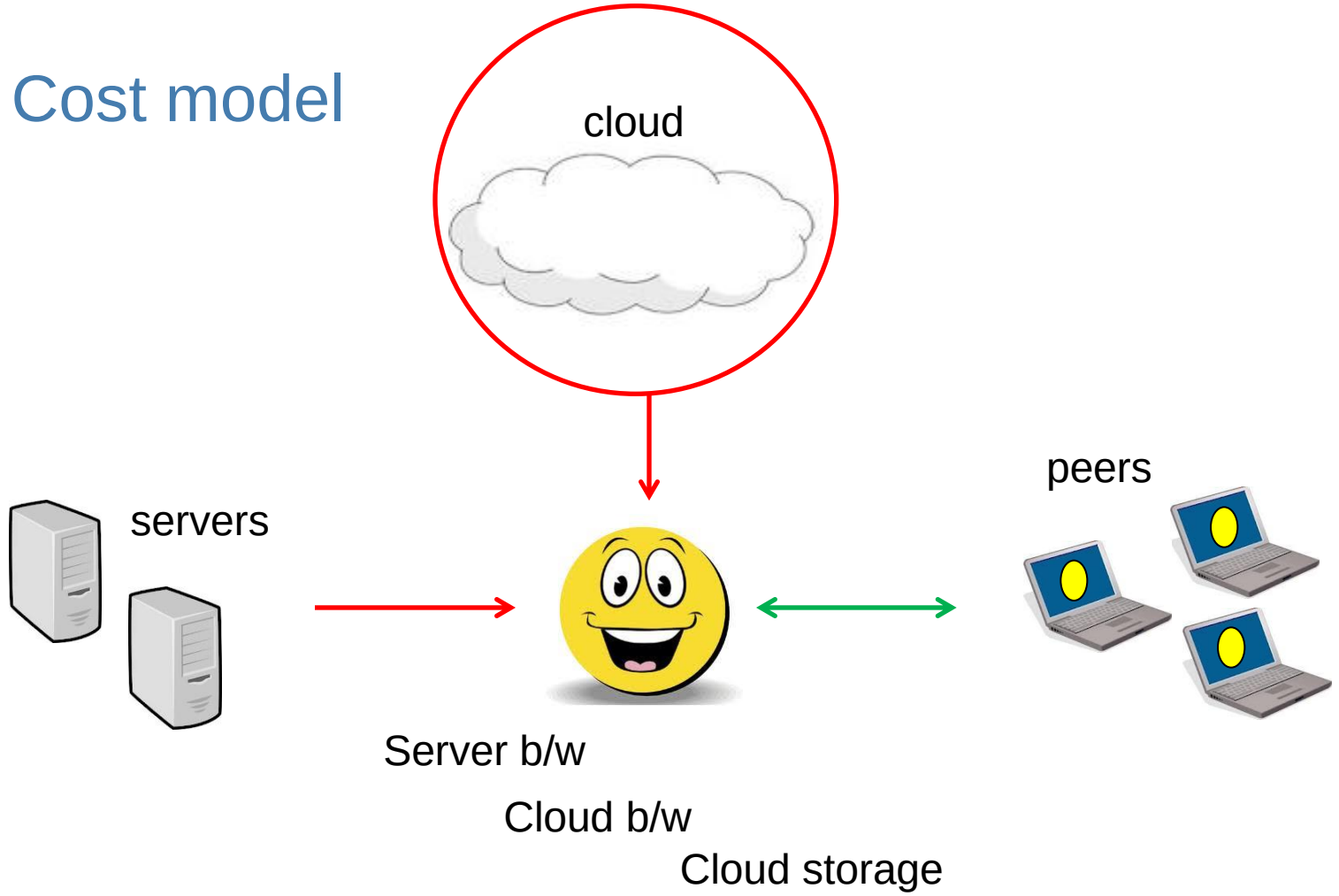


- Simple cost model
  - Three (3) basic cost components

Cost model

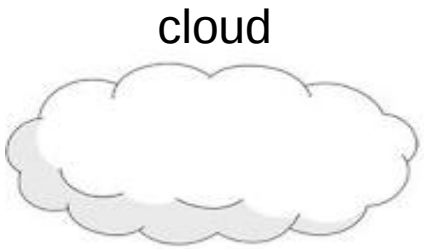


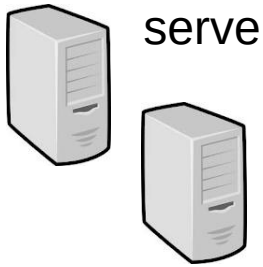
Cost model



$$\sum_{i \in N/M} B_i^s + c \sum_{i \in M} B_i^c + C\,|M|$$

Cost model





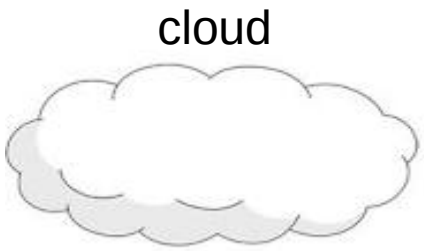
Server b/w

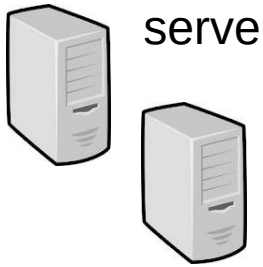
Cloud b/w

Cloud storage

$$\sum_{i \in N/M} B_i^s + c \sum_{i \in M} B_i^c + C\,|M|$$

Cost model





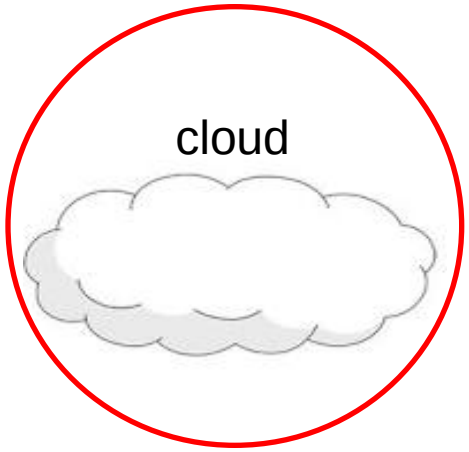
Server b/w

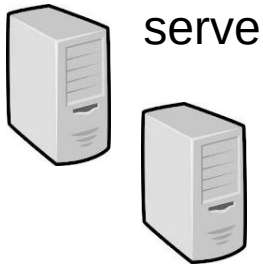
Cloud b/w

Cloud storage

$$\sum_{i \in N/M} B_i^s + c \sum_{i \in M} B_i^c + C\,|M|$$

Cost model





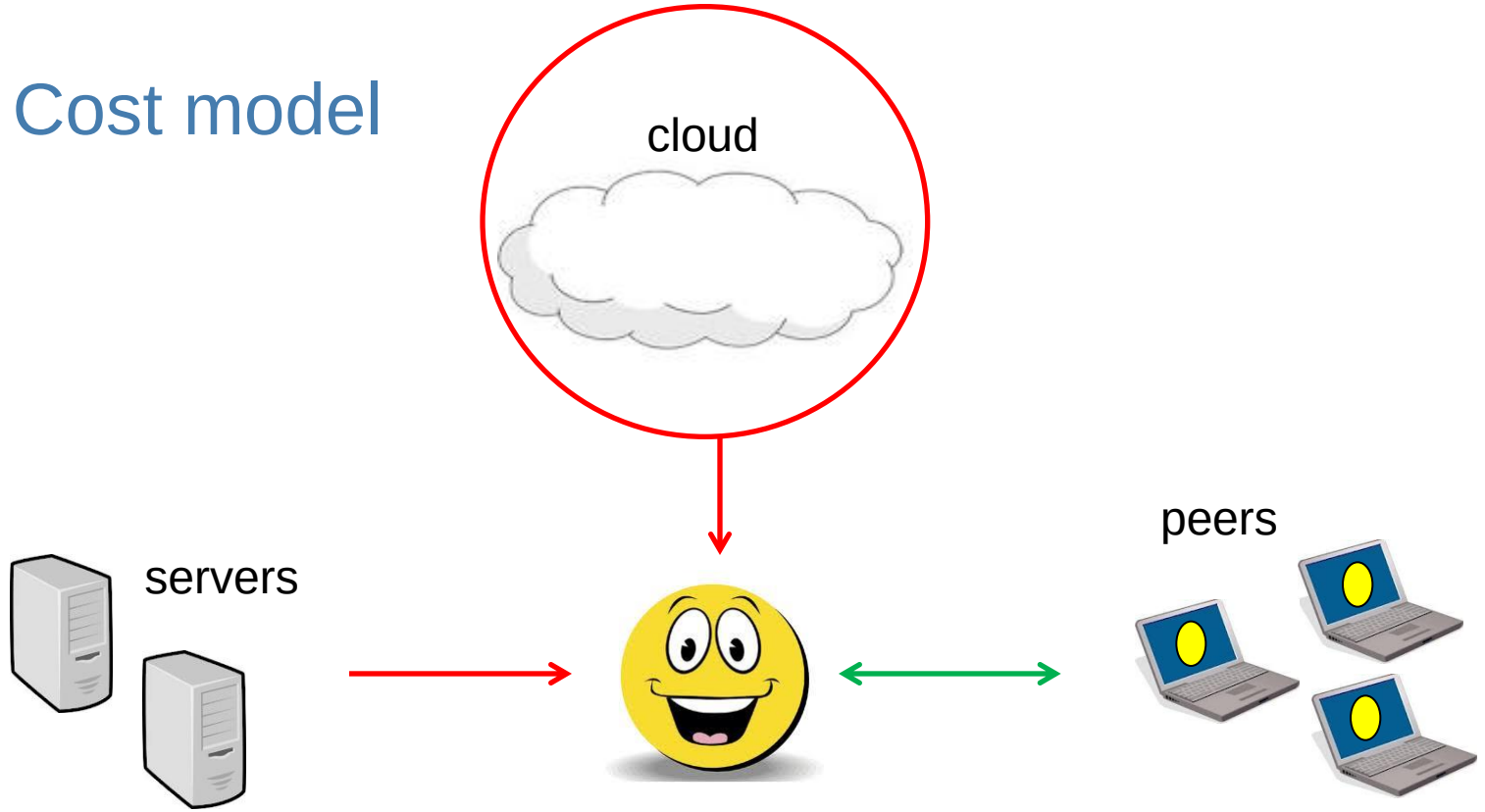
Server b/w

Cloud b/w

Cloud storage

$$\sum_{i \in N/M} B_i^s + c \sum_{i \in M} B_i^c + C\,|M|$$

Cost model



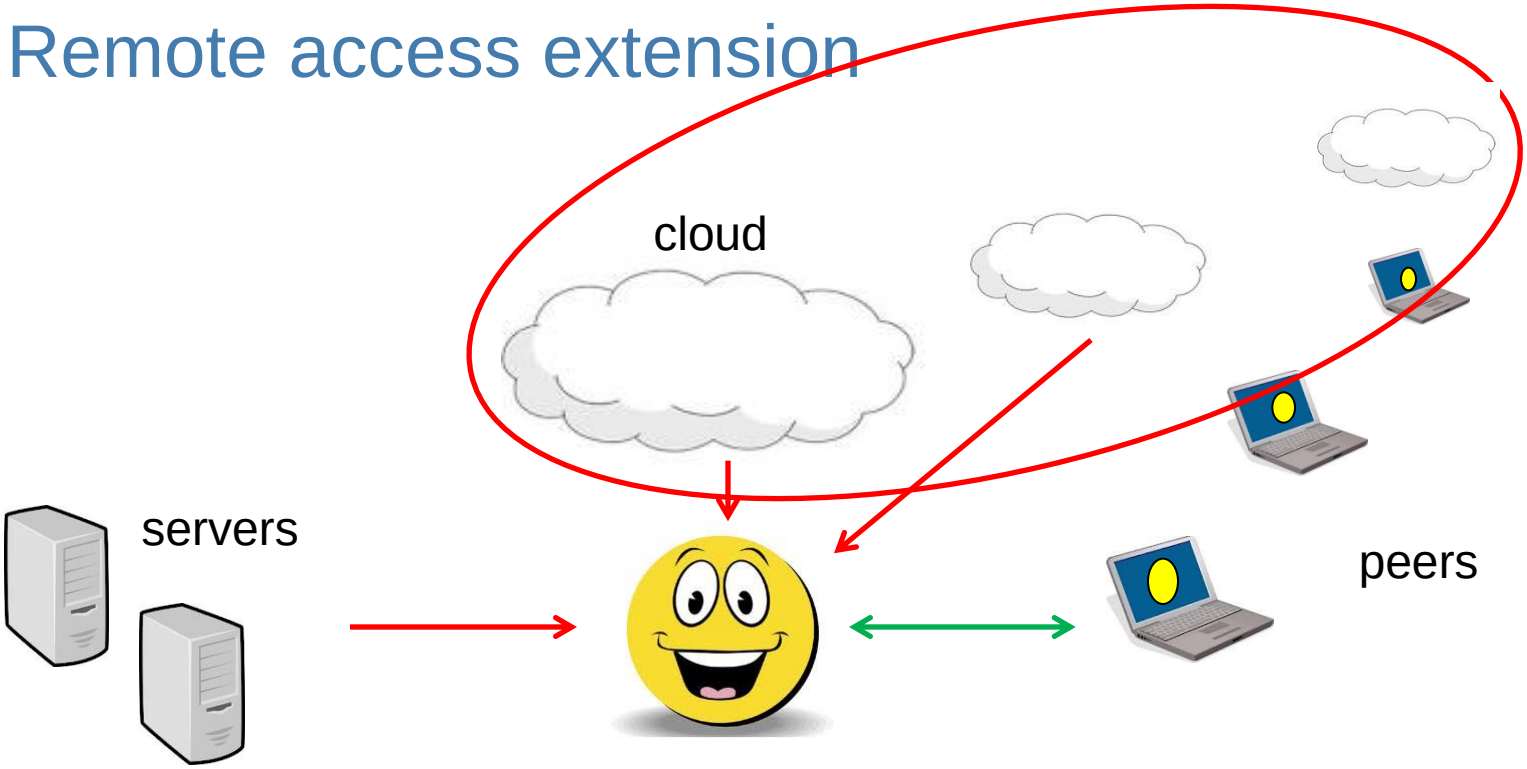
Server b/w

Cloud b/w

Cloud storage

$$\sum_{i \in N/M} B_i^s + c \sum_{i \in M} B_i^c + C\,|M|$$

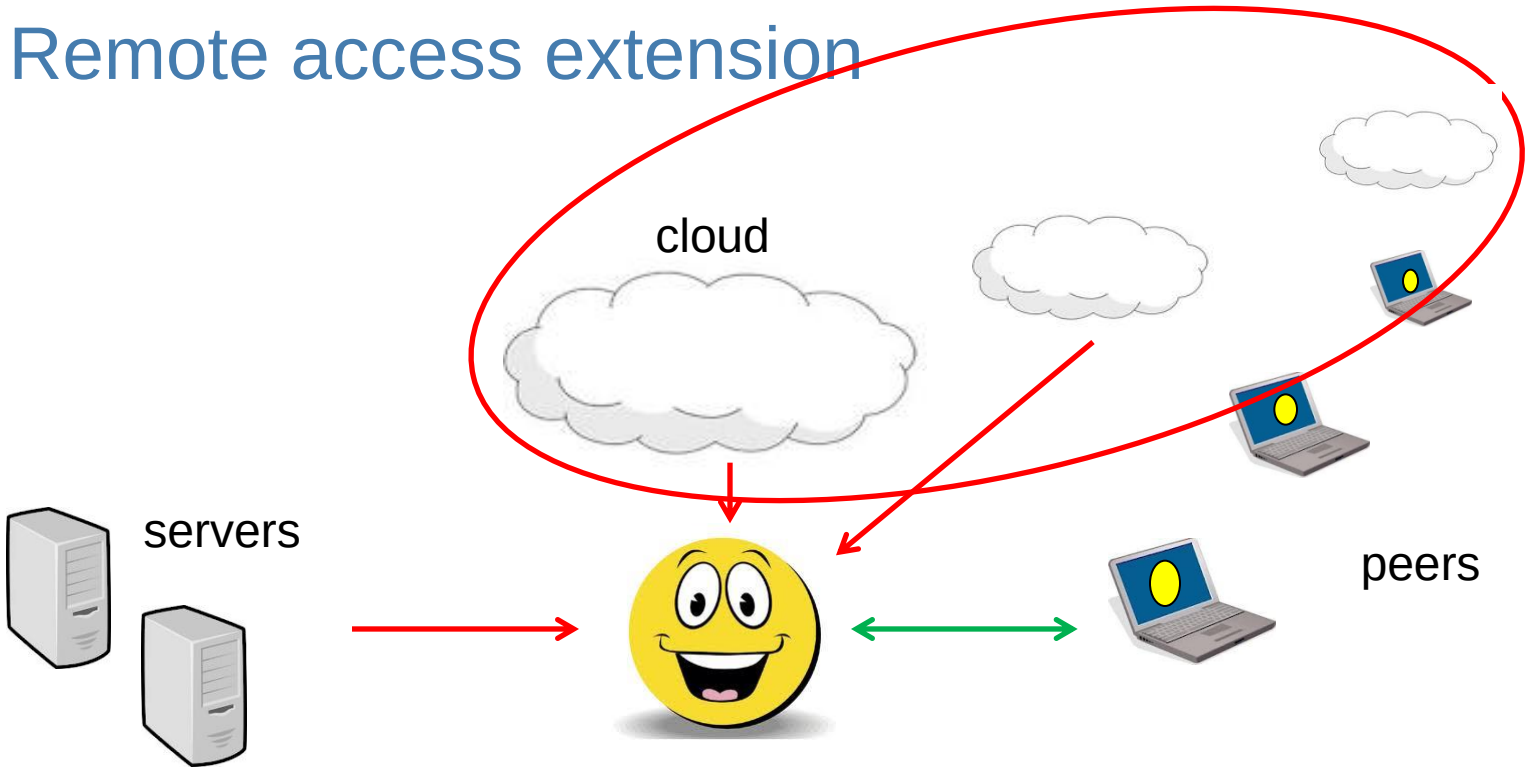
Remote access extension



- Extensions
  - Full version (in paper)

$$\sum_{i \in N} B_i^s + c \sum_{i \in M} \sum_{j \in P_i} B_{i,j}^c (1 + q_{ij} f_{ij}) + C \sum_{i \in N} | P_i |$$

Remote access extension



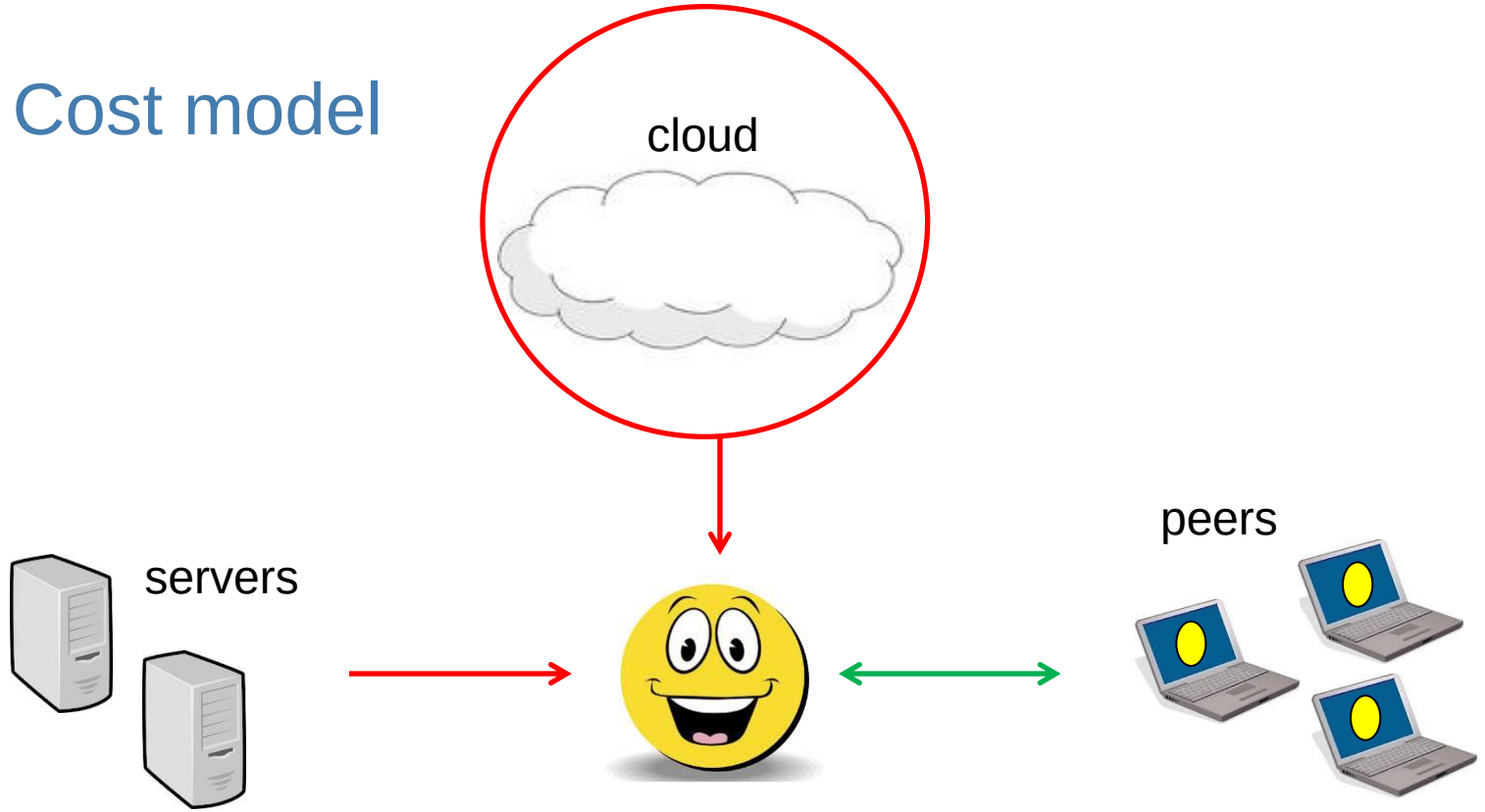
- Extensions
  - Full version (in paper)

Different remote access cost

Locations of each content

$$\sum_{i \in N} B_i^s + c \sum_{i \in M} \sum_{j \in P_i} B_{i,j}^c (1 + q_{ij} f_{ij}) + C \sum_{i \in N} | P_i |$$

Cost model



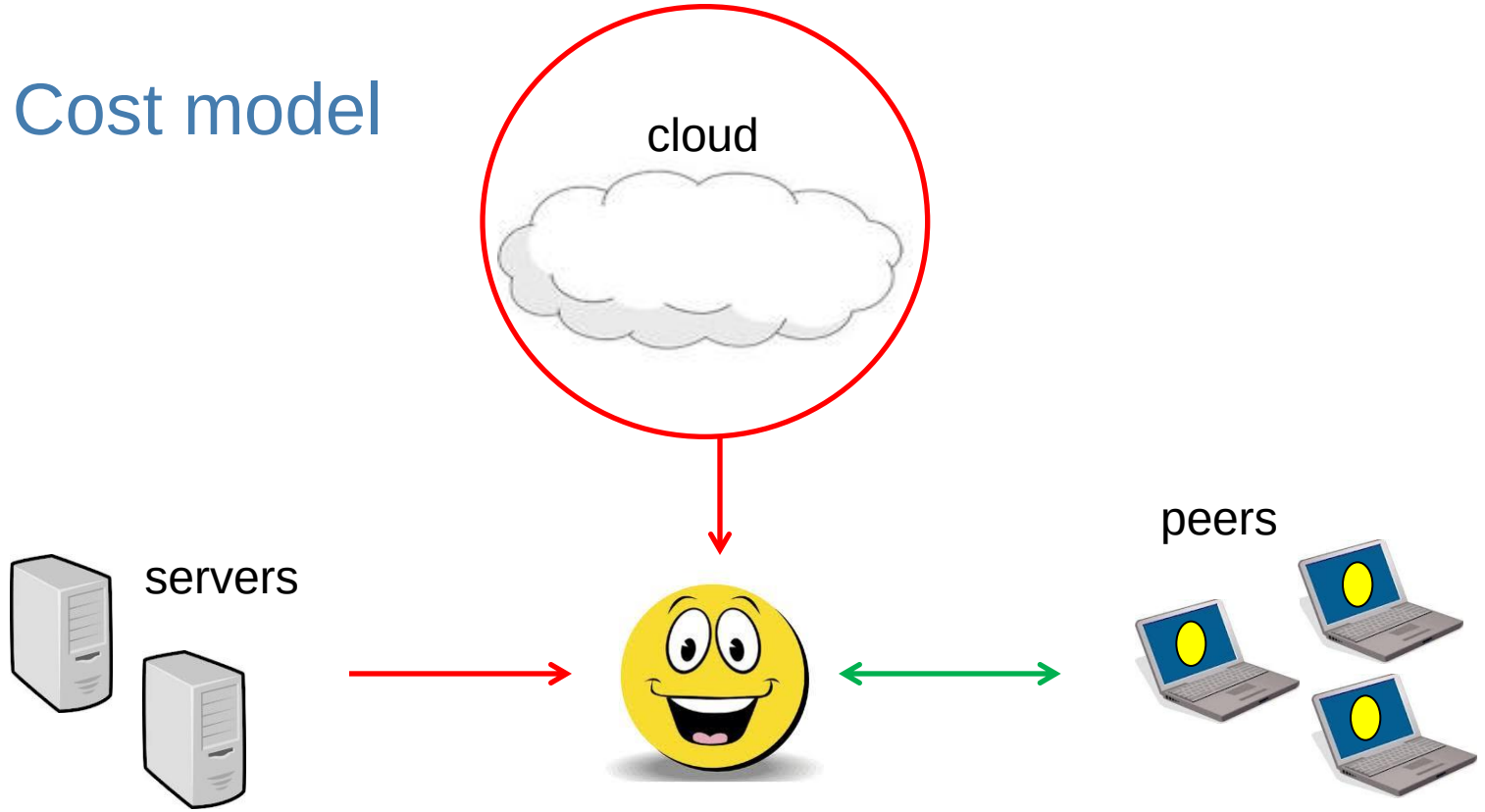
Server b/w

Cloud b/w

Cloud storage

$$\sum_{i \in N/M} B_i^s + c \sum_{i \in M} B_i^c + C\,|M|$$

Cost model



Server b/w

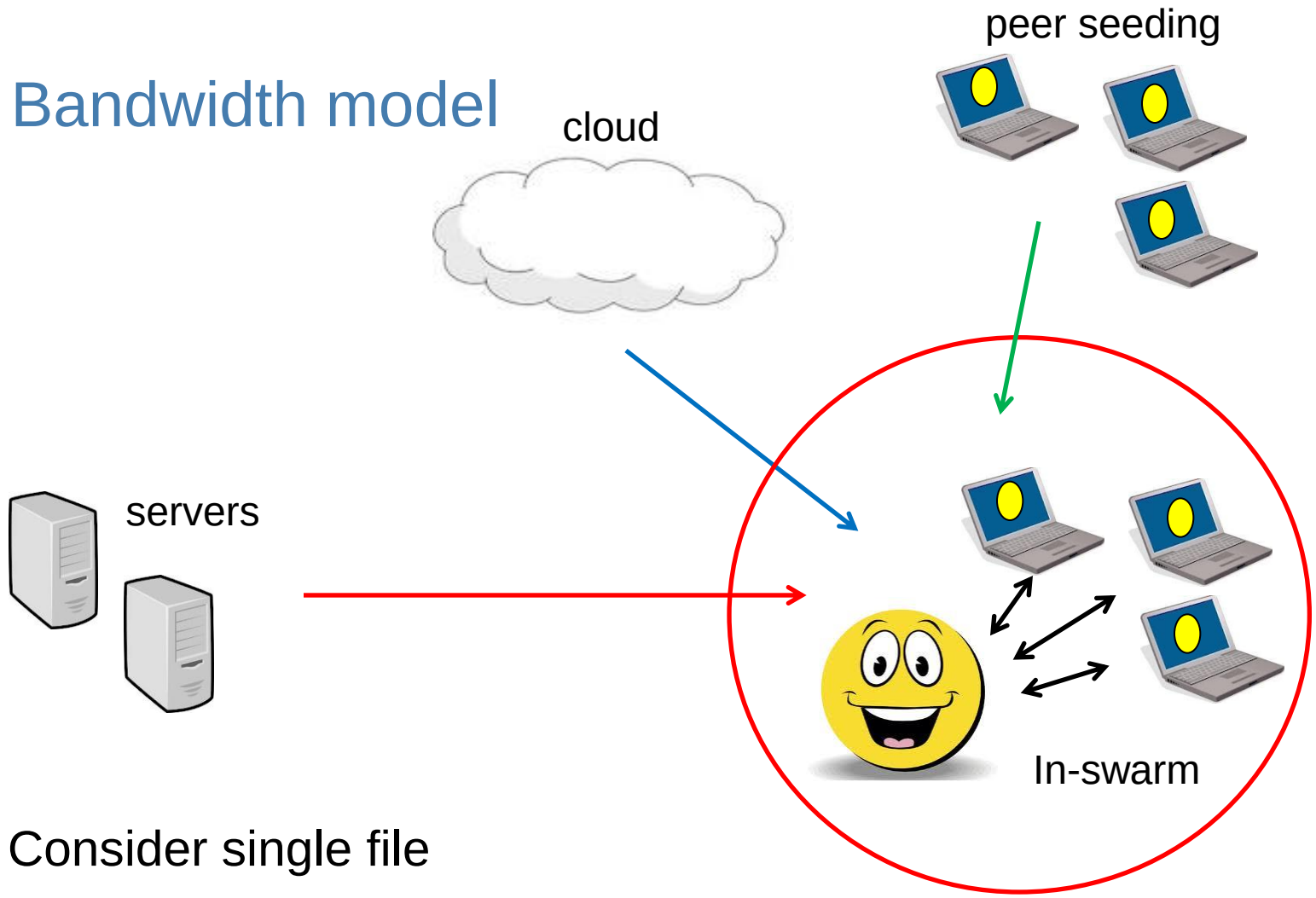
Cloud b/w

Cloud storage

Minimize

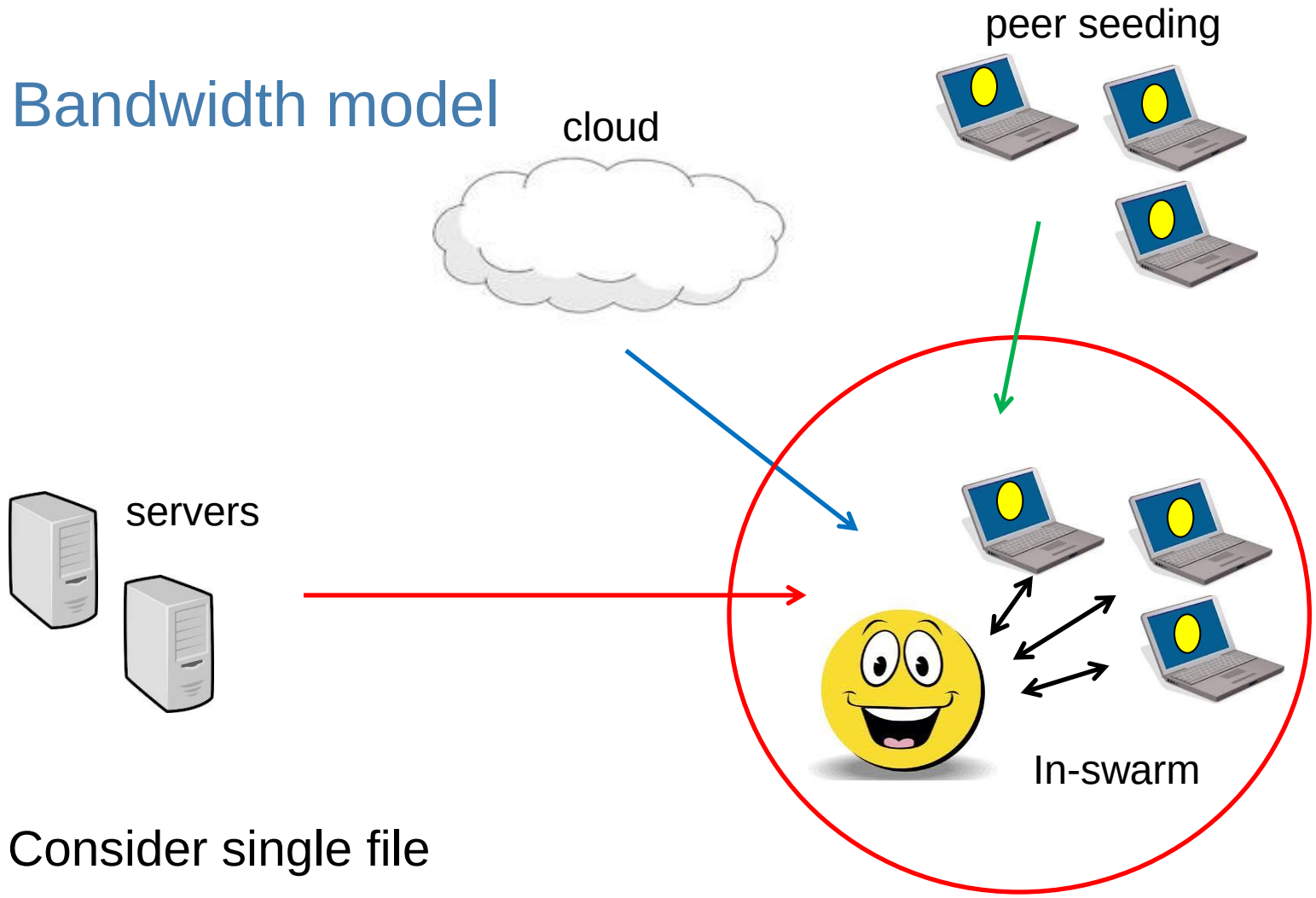
$$\sum_{i \in N/M} B_i^s + c \sum_{i \in M} B_i^c + C\,|M|$$

Bandwidth model



- Consider single file

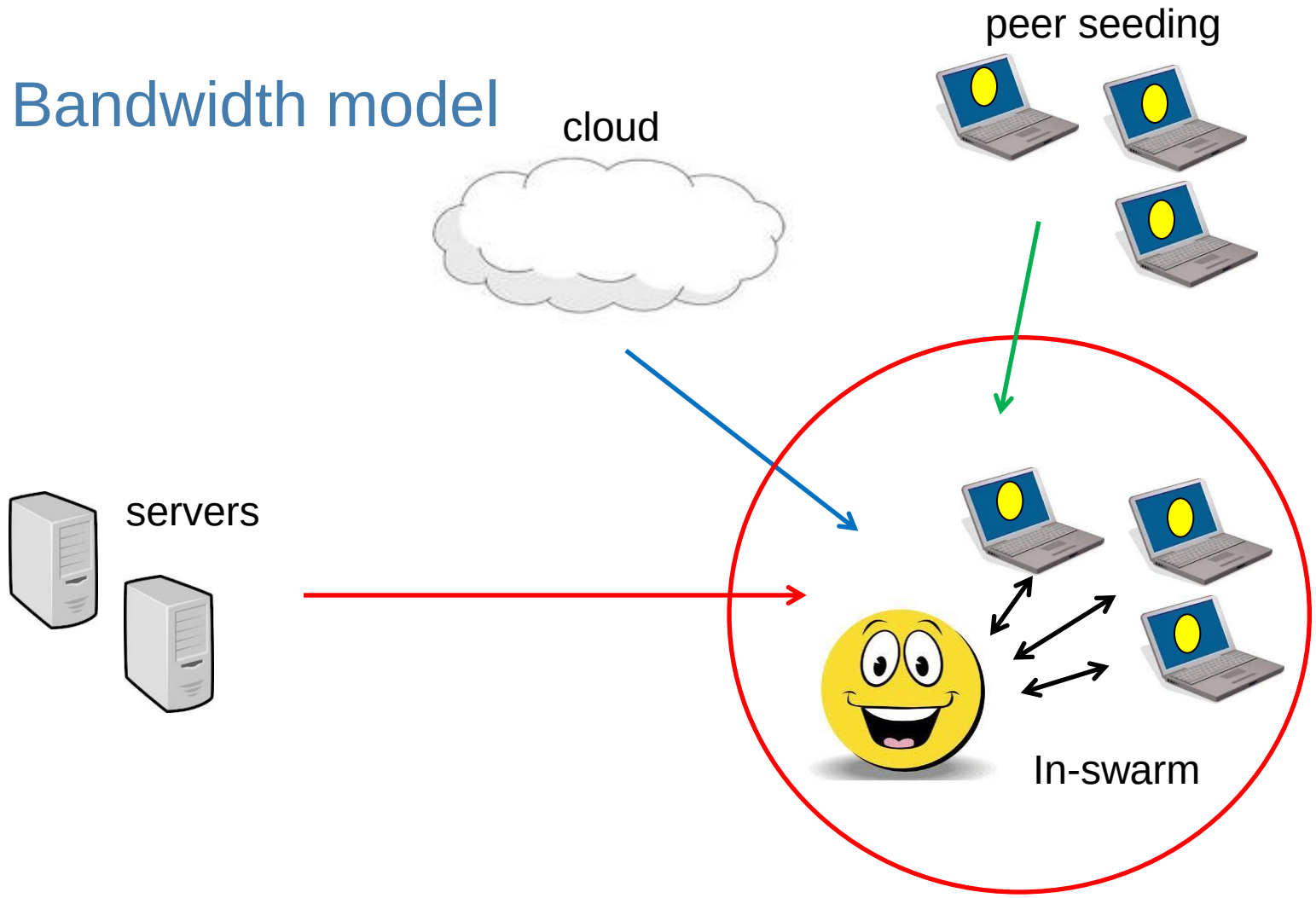
Bandwidth model



- Consider single file

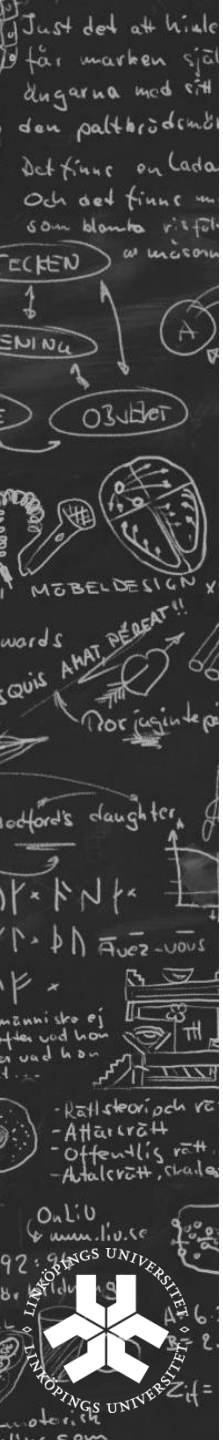
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

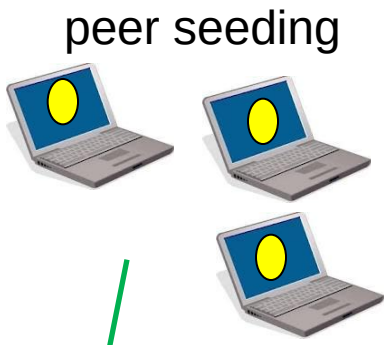
Bandwidth model

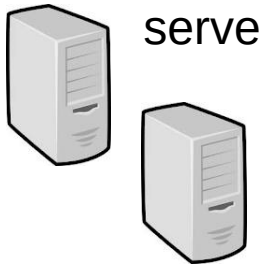


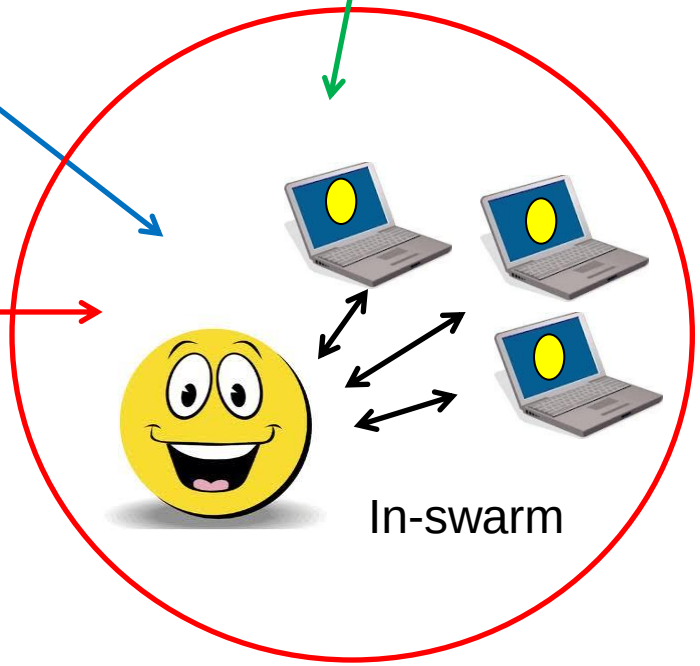
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

Bandwidth model





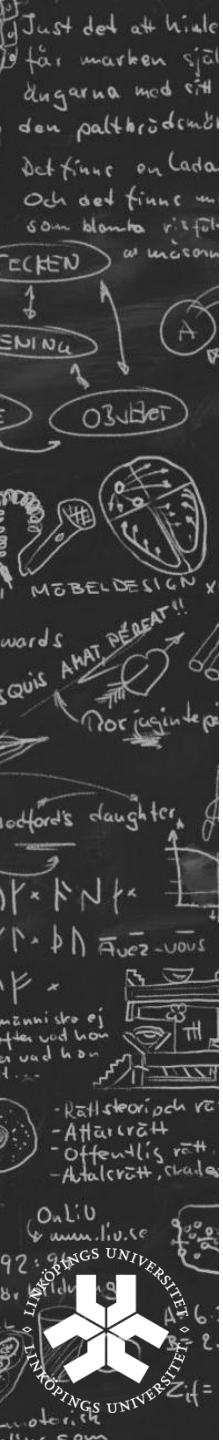


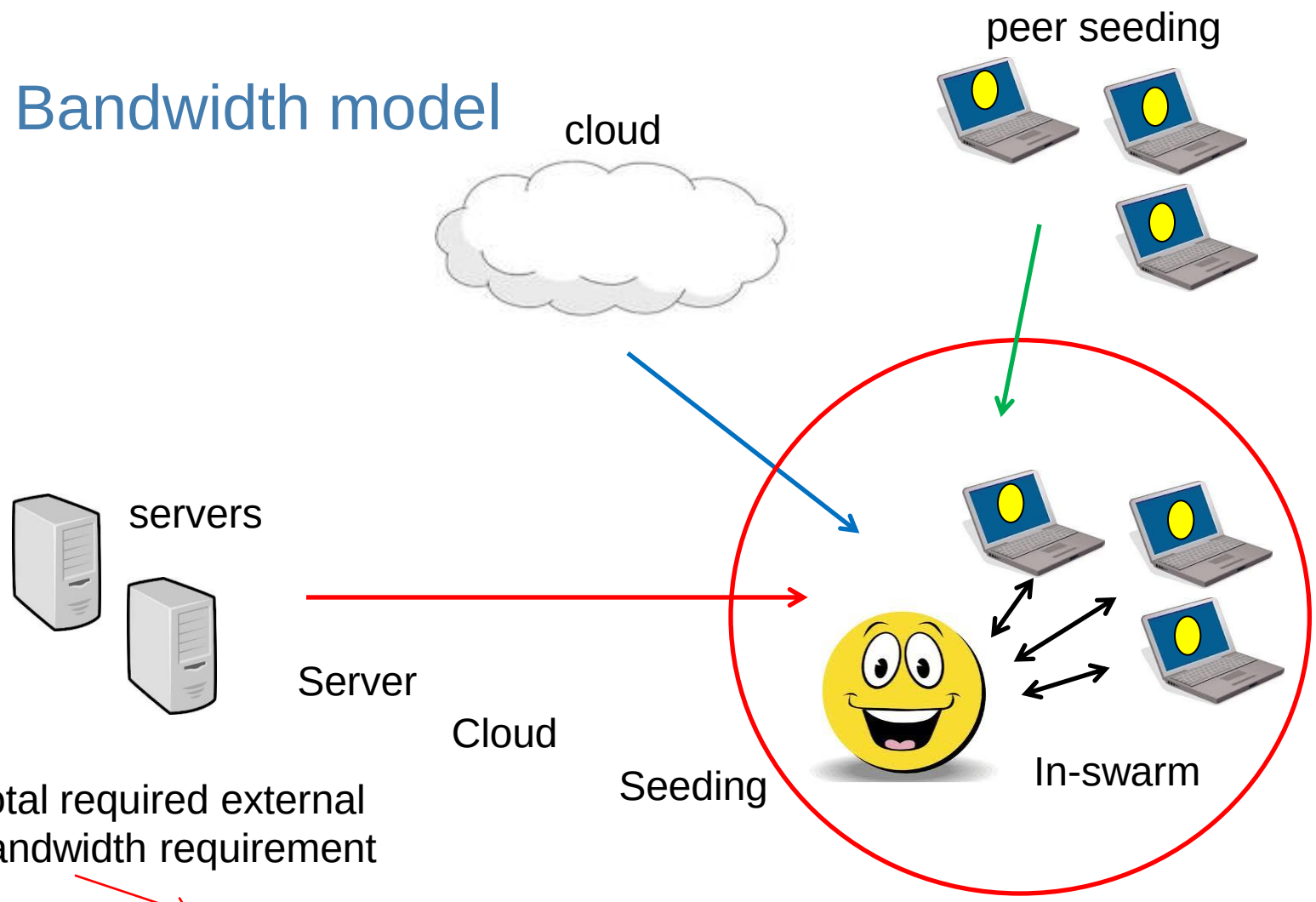


Total required external bandwidth requirement

$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

Bandwidth model

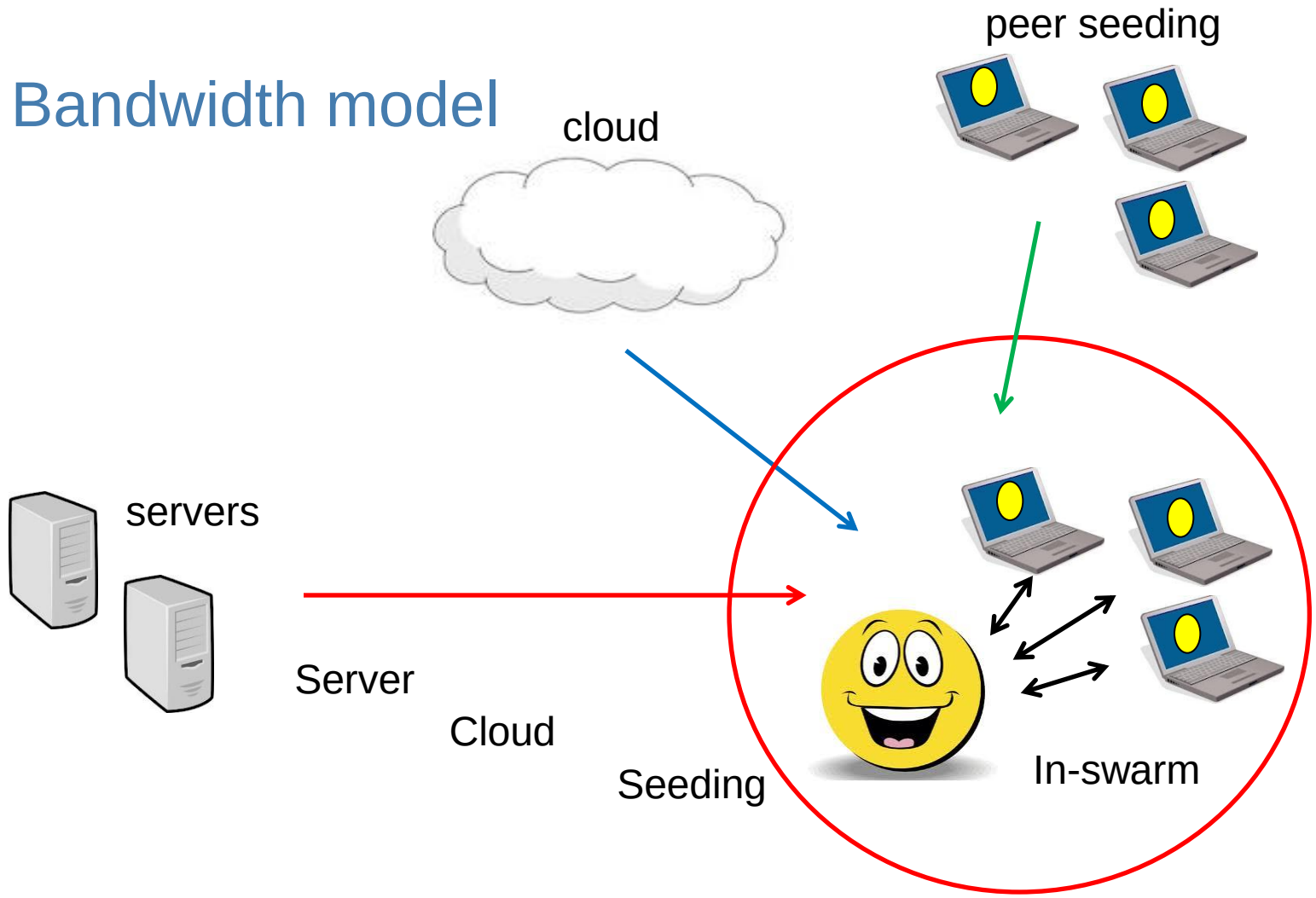




Total required external bandwidth requirement

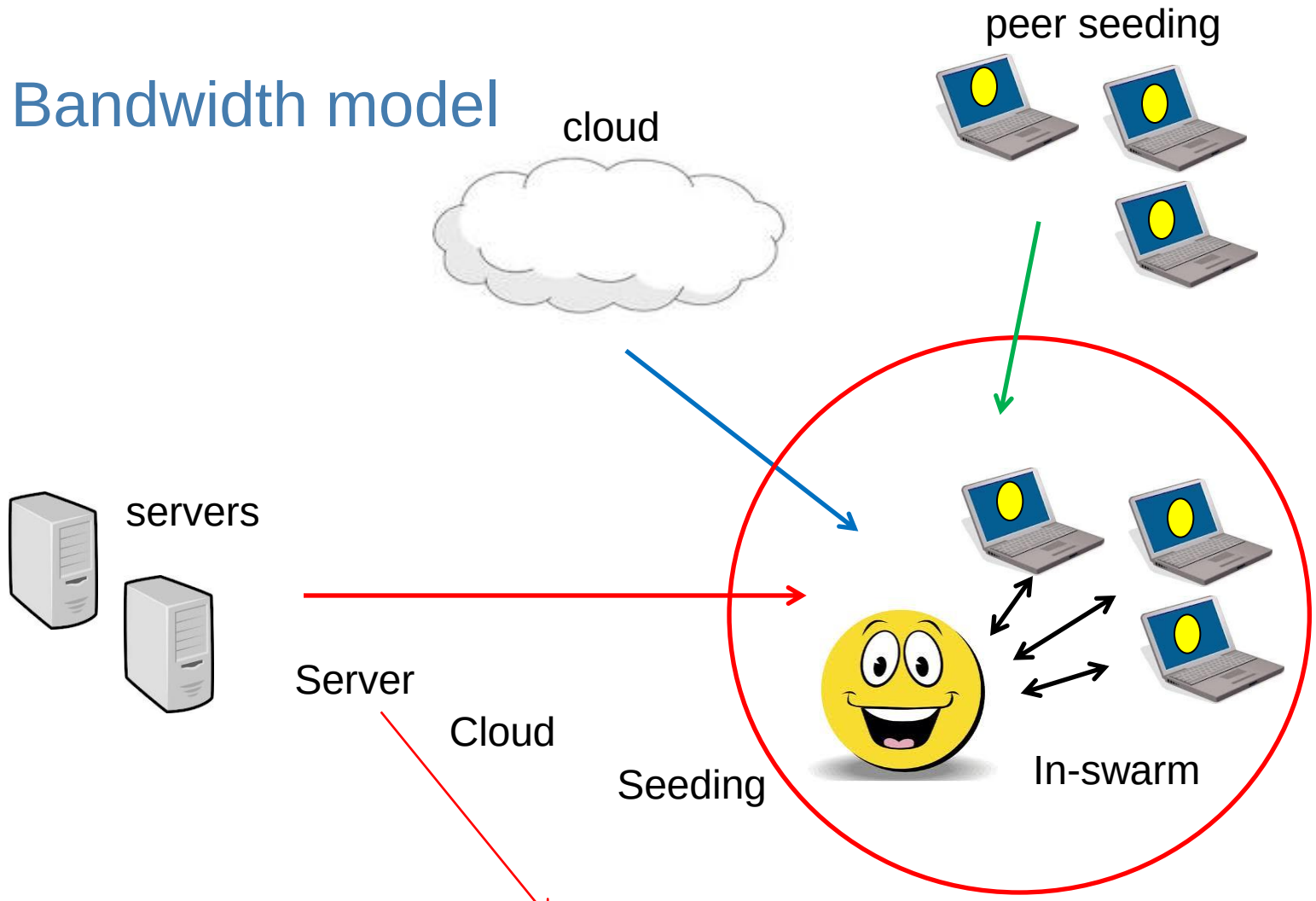
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

Bandwidth model



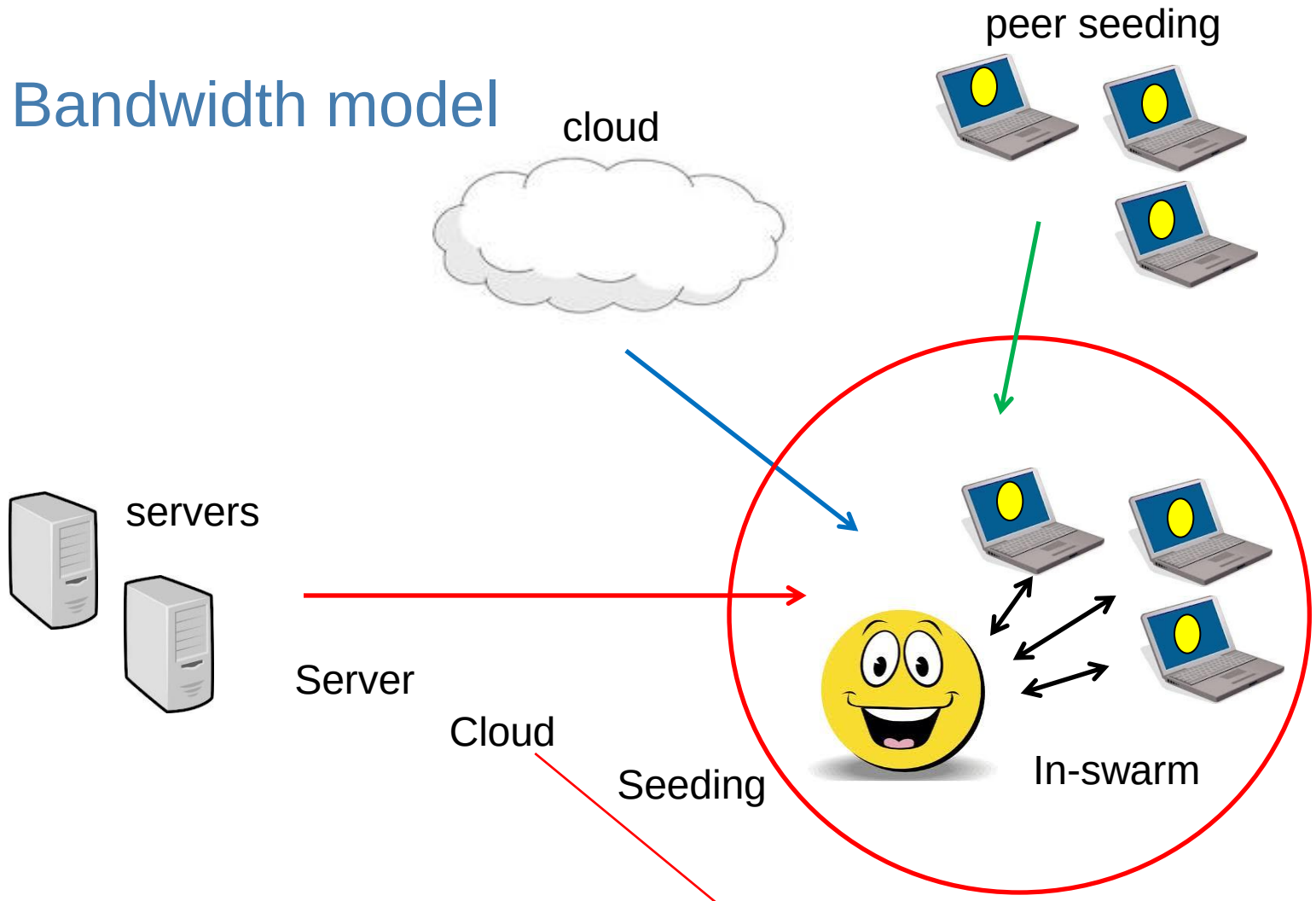
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

Bandwidth model



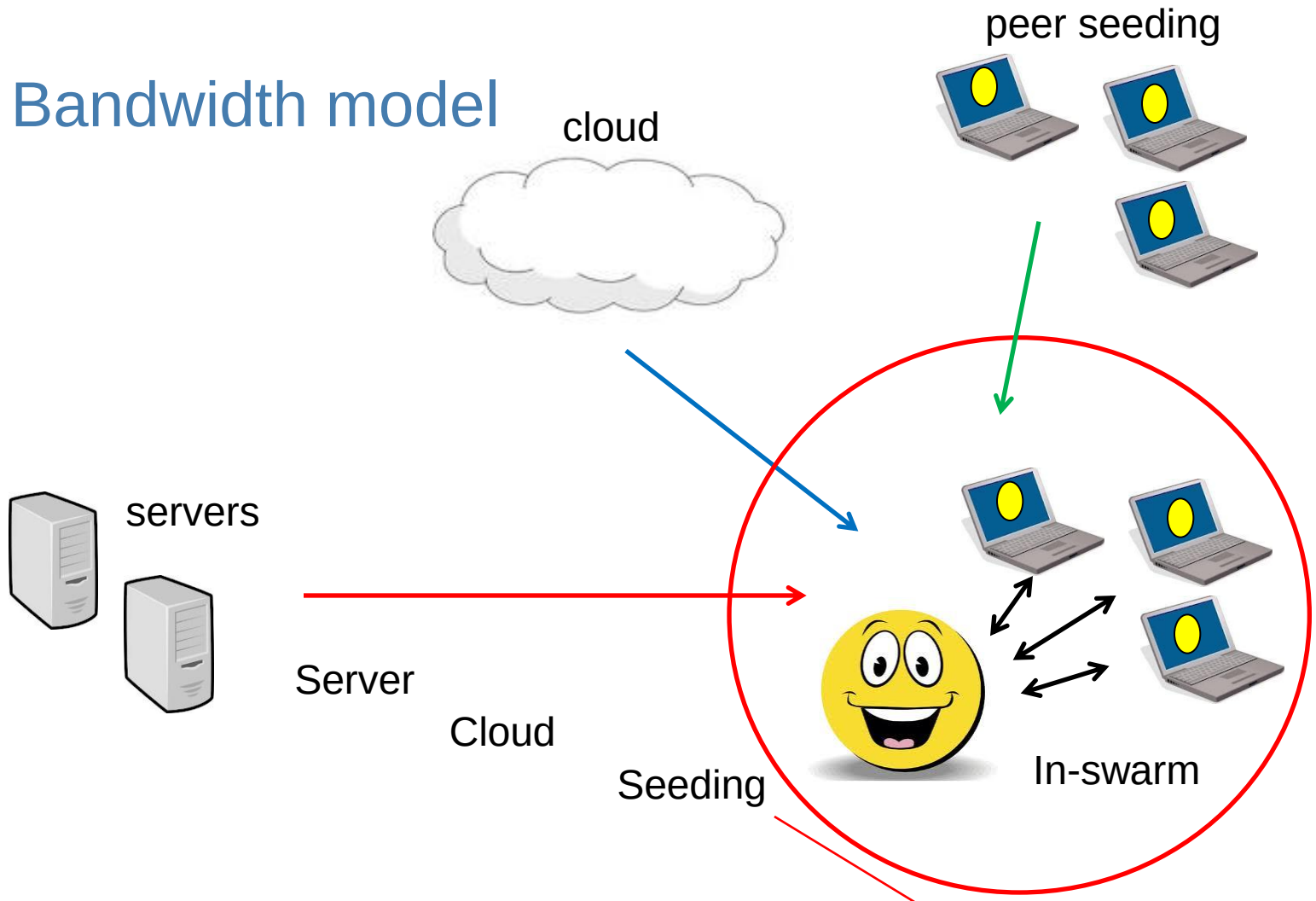
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

Bandwidth model



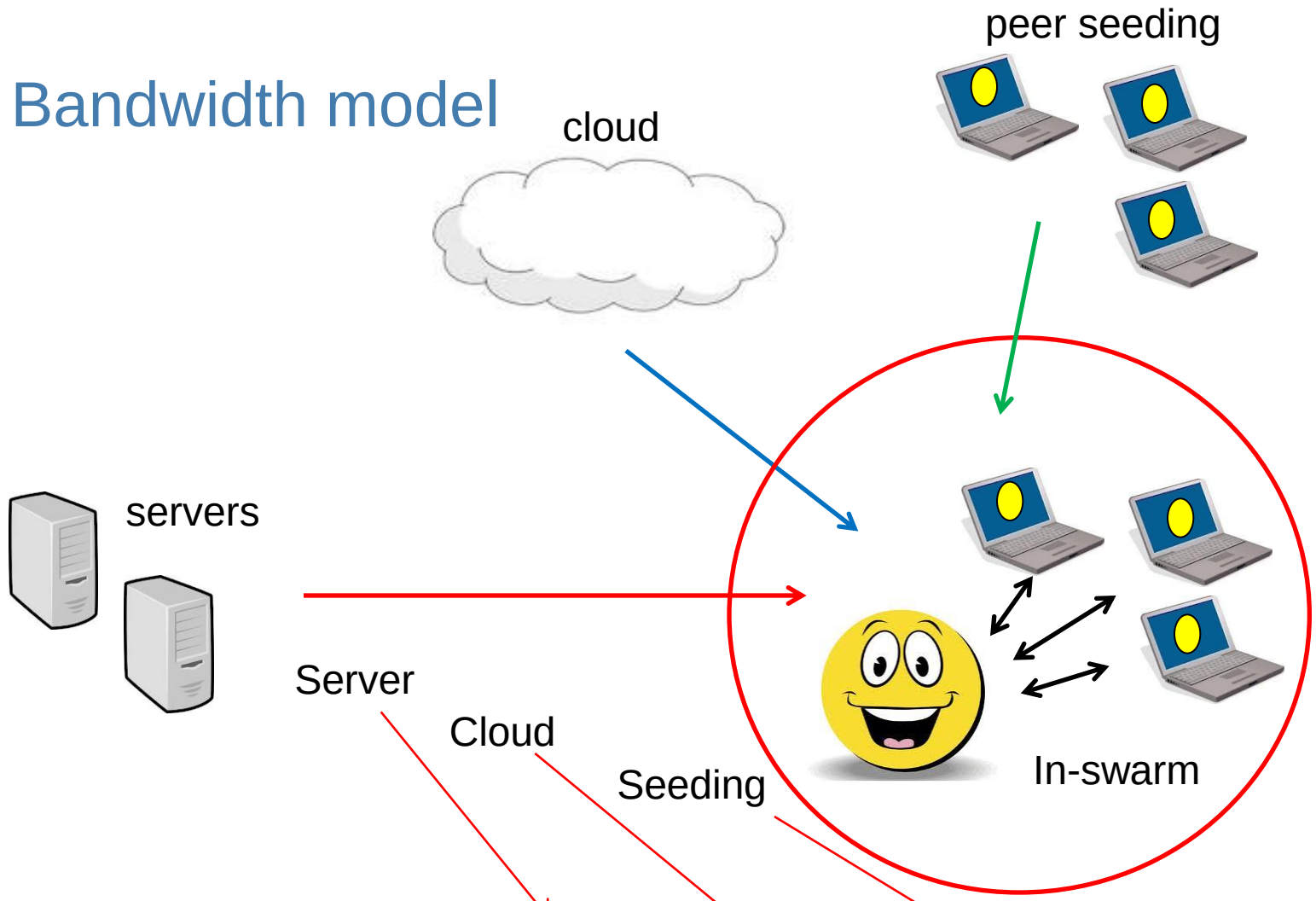
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

Bandwidth model



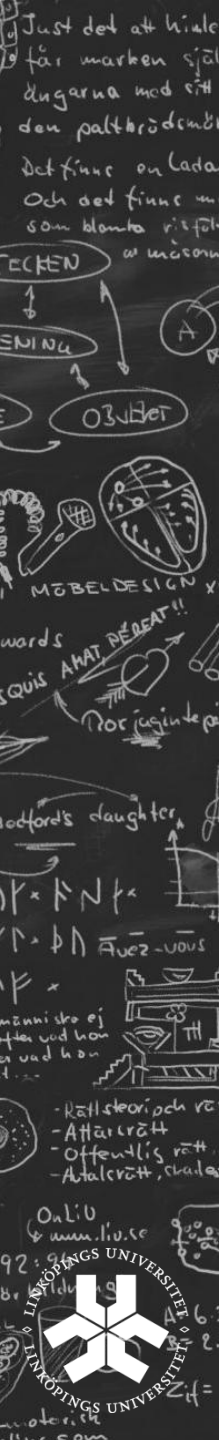
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

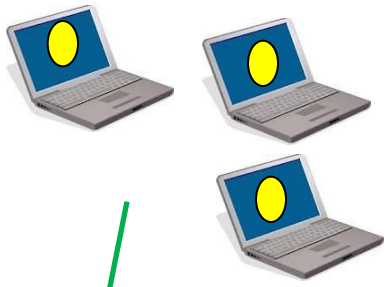
Bandwidth model

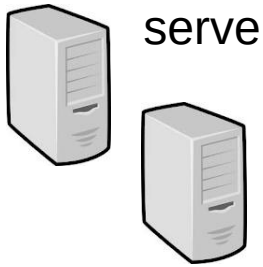


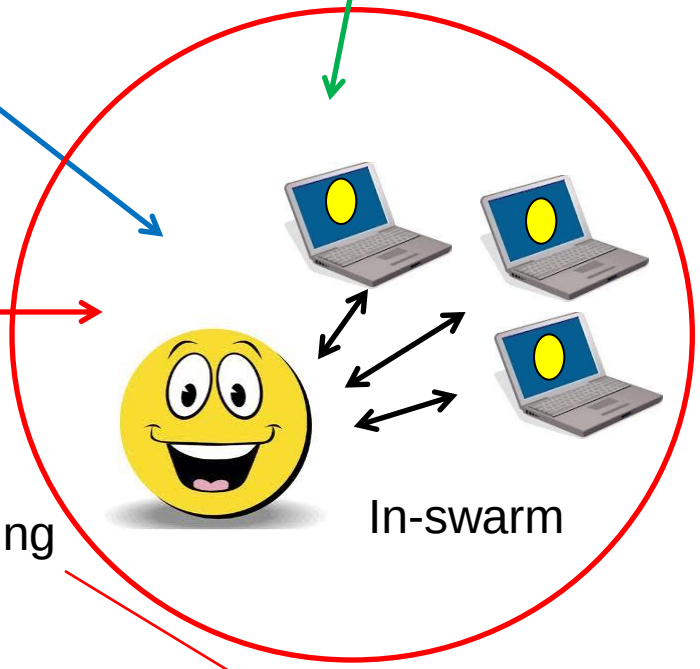
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

Bandwidth model









Server

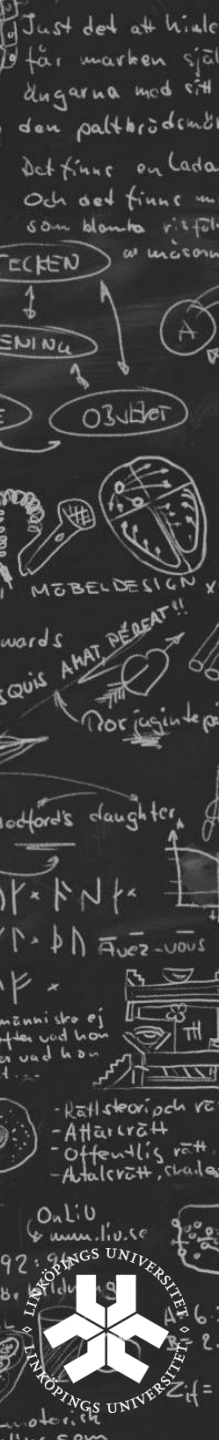
Cloud

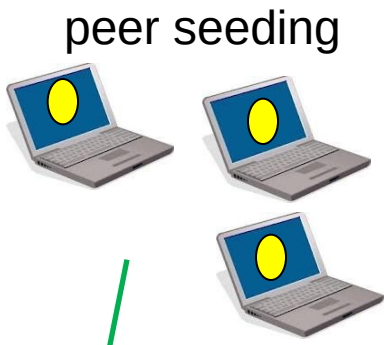
Seeding

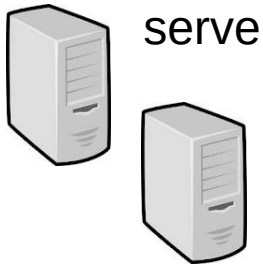
Total required external bandwidth requirement

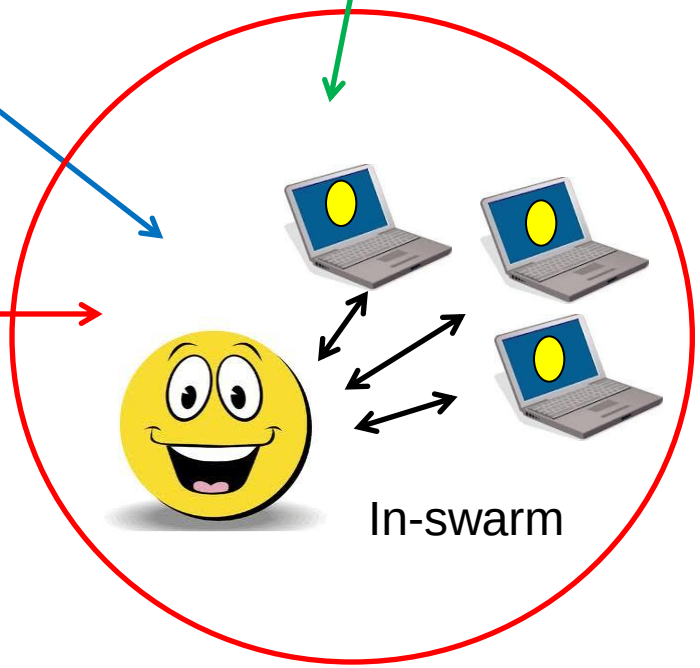
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

External bandwidth requirement





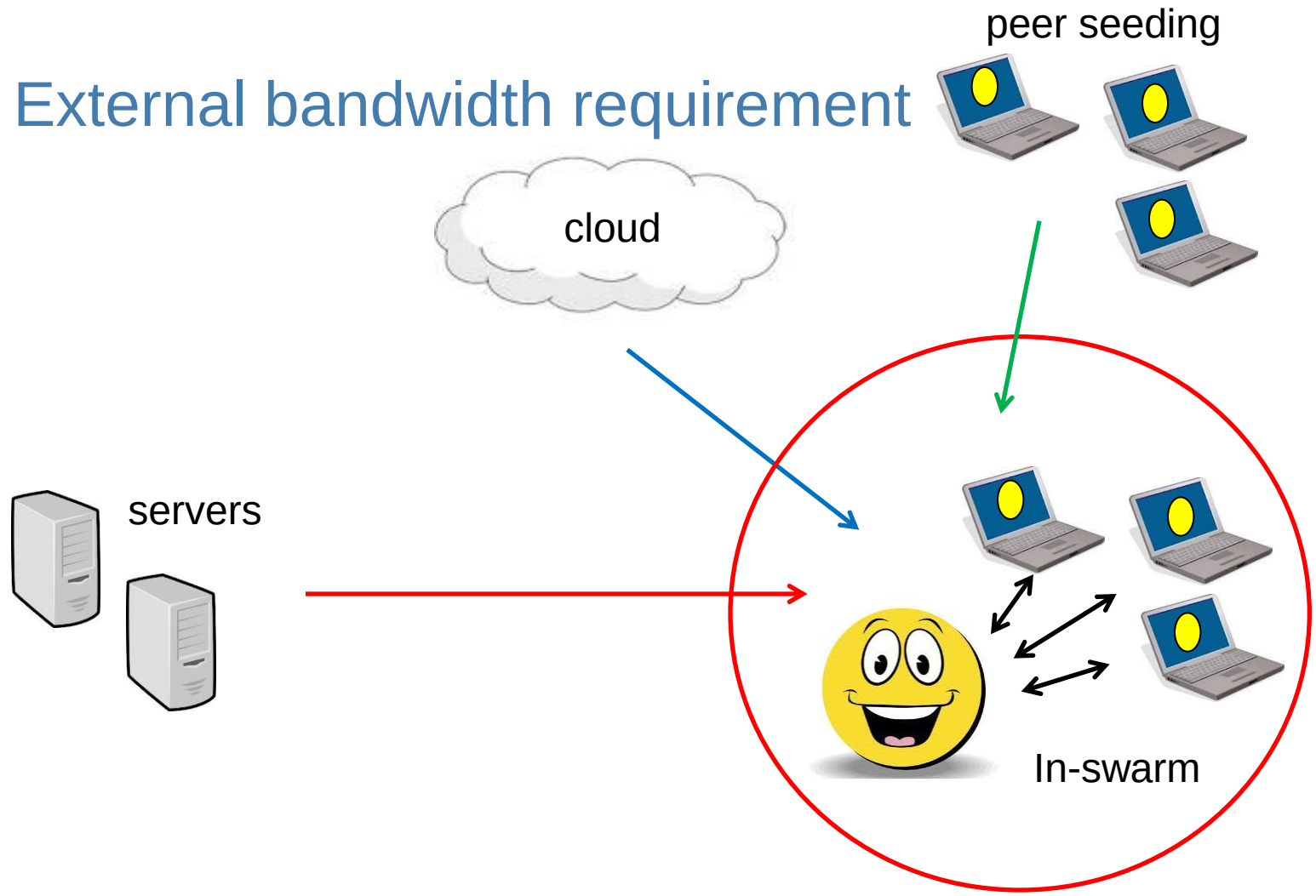




Total required external bandwidth requirement

$$B(\lambda_i + \phi_i)$$

External bandwidth requirement



$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set
- Assume (for simplicity)
  - Poisson arrivals, piece fractions, and independent pieces on each peer

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set
- Assume (for simplicity)
  - Poisson arrivals, piece fractions, and independent pieces on each peer

Departure rate

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set
- Assume (for simplicity)
  - Poisson arrivals, piece fractions, and independent pieces on each peer

Leave k peers behind

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set
- Assume (for simplicity)
  - Poisson arrivals, piece fractions, and independent pieces on each peer

Peer k missing 1/(k+1)
Peer k-1 missing 2/(k+1)
…
Peer 1 missing k/(k+1)

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set
- Assume (for simplicity)
  - Poisson arrivals, piece fractions, and independent pieces on each peer

Peer k missing 1/(k+1)
Peer k-1 missing 2/(k+1)
…
Peer 1 missing k/(k+1)

Departure rate

Leave k peers behind

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

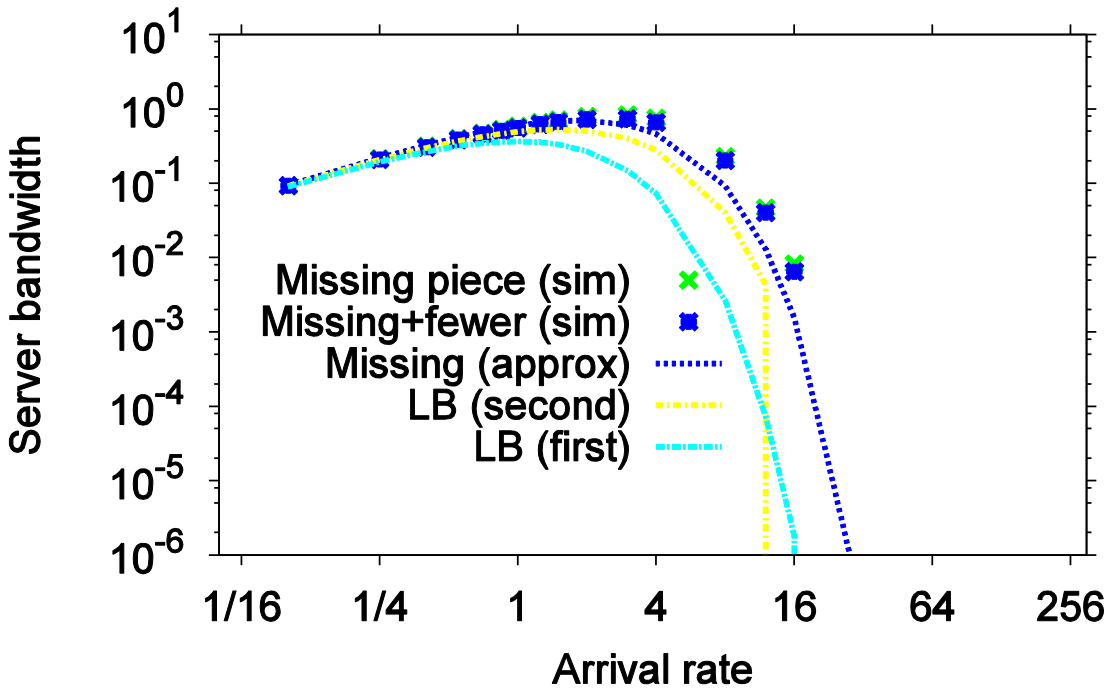
External bandwidth requirement

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set
- Assume (for simplicity)
  - Poisson arrivals, piece fractions, and independent pieces on each peer

$$B(\lambda_i) \approx \lambda \sum_{k}^{\infty} \frac{(\lambda L/U)^k}{k!} e^{-\lambda L/U} \frac{k!}{(k+1)^k} L$$

External bandwidth requirement

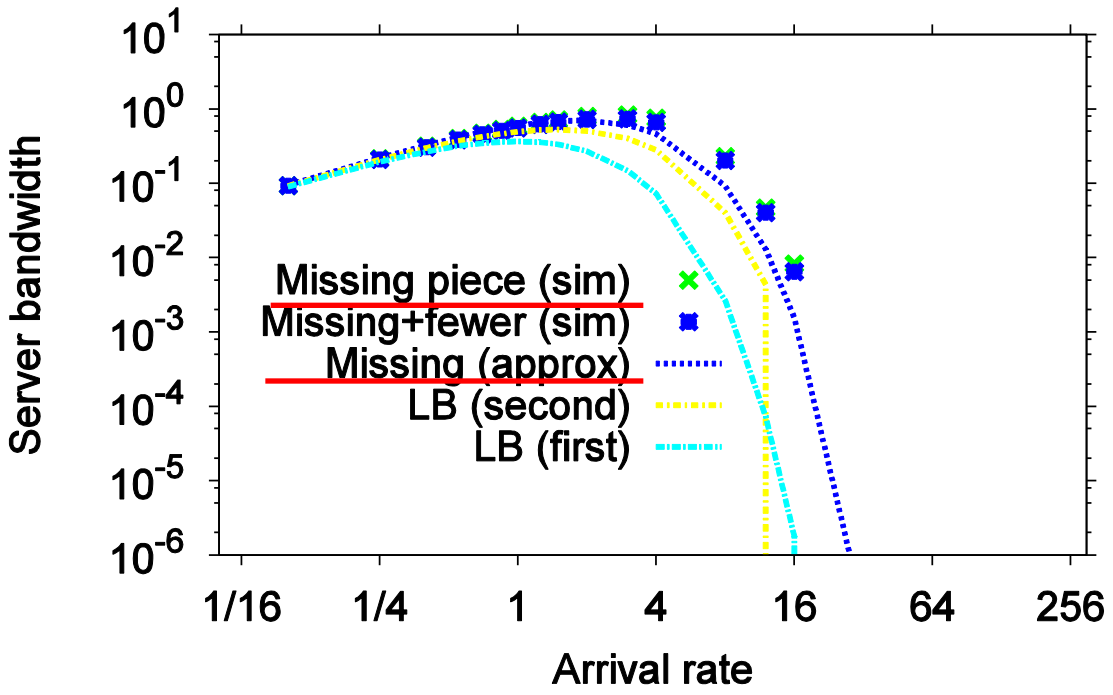
- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set



- Approximation fairly accurate
- Very small benefit prioritizing young peers (with few pieces)
- Self sustainability

External bandwidth requirement

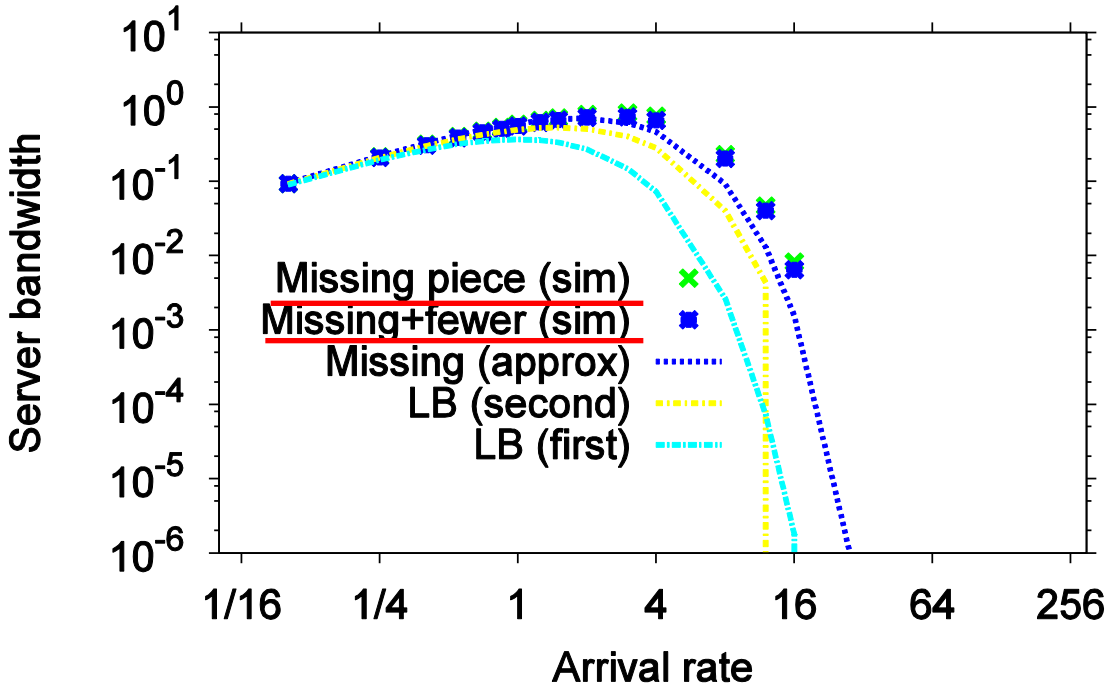
- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set



- Approximation fairly accurate
- Very small benefit prioritizing young peers (with few pieces)
- Self sustainability

External bandwidth requirement

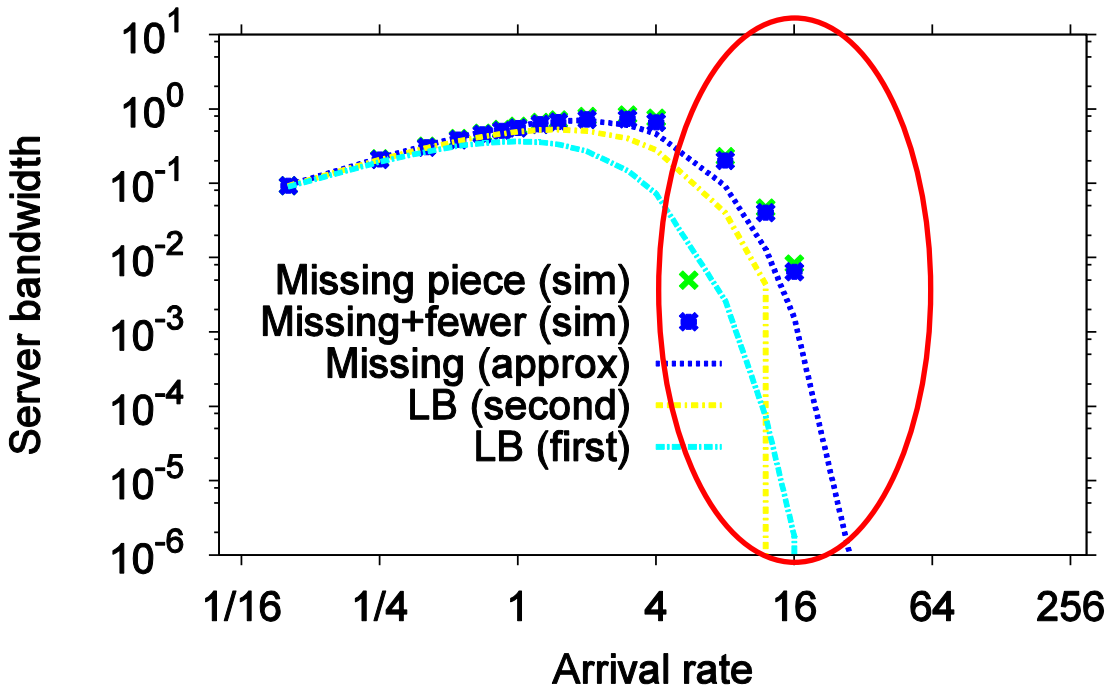
- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set



- Approximation fairly accurate
- Very small benefit prioritizing young peers (with few pieces)
- Self sustainability

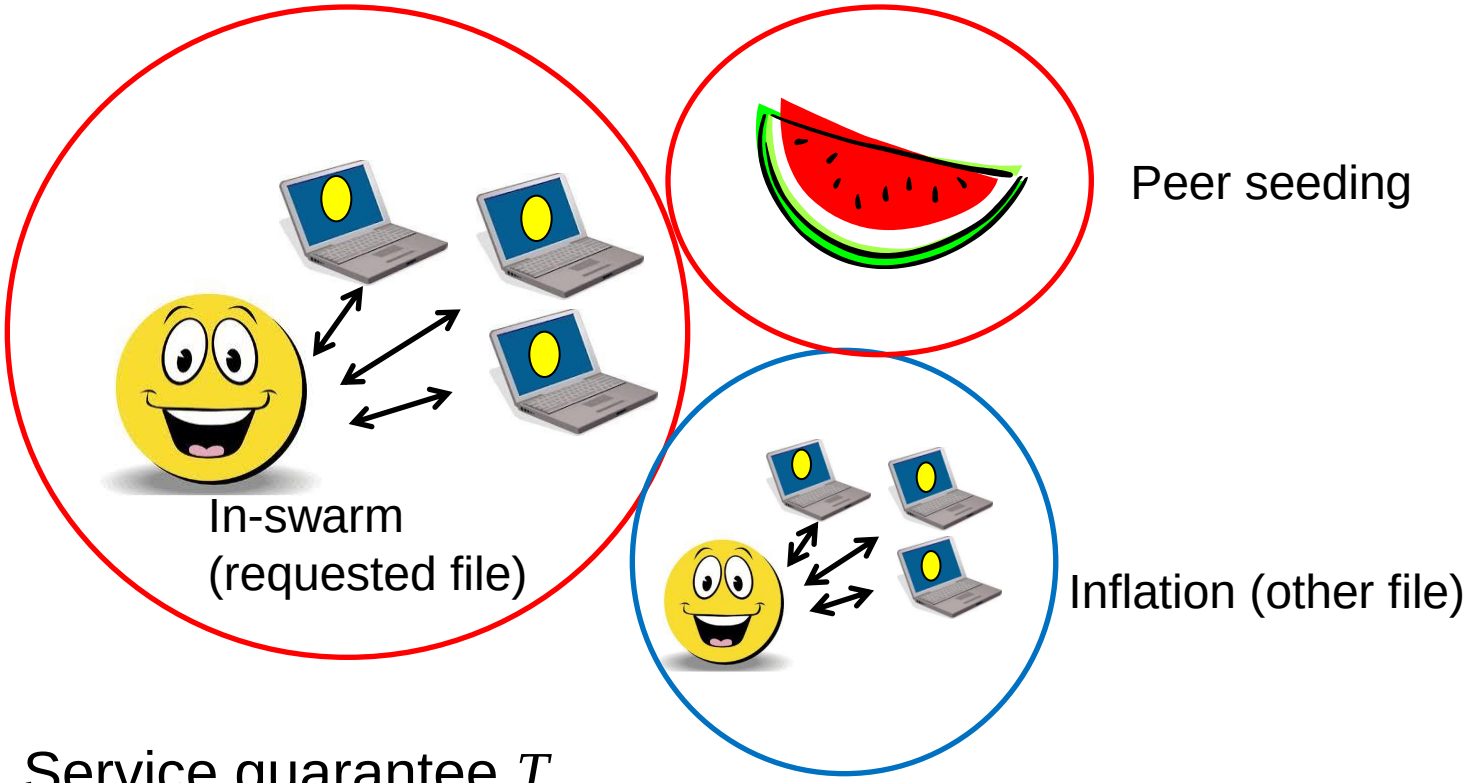
Self sustainability

- Consider **missing piece** policy
  - Server upload only one piece at a time whenever there is at least one piece missing among peer set



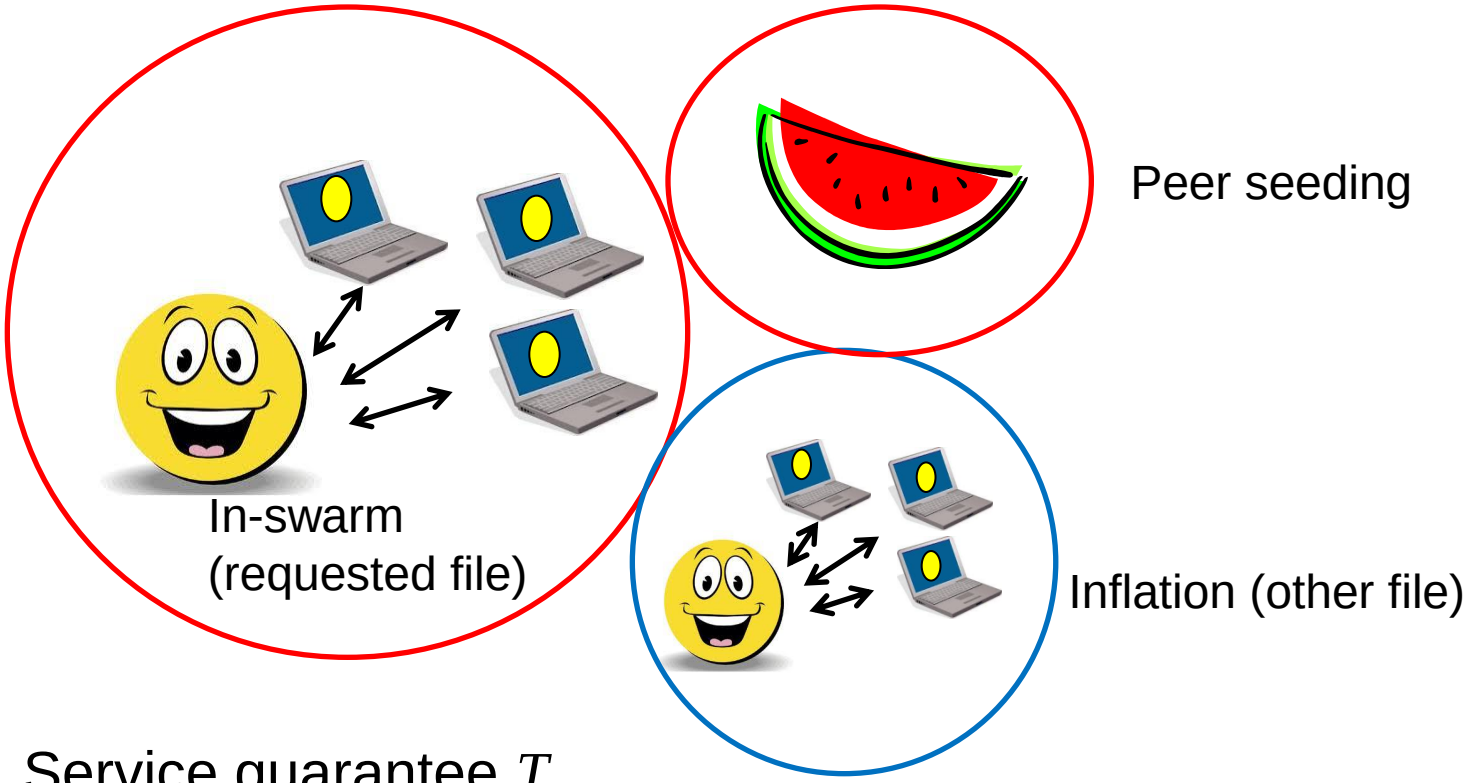
- Approximation fairly accurate
- Very small benefit prioritizing young peers (with few pieces)
- Self sustainability

Peer upload bandwidth



- Service guarantee T
- Help out, but **only** during download/service

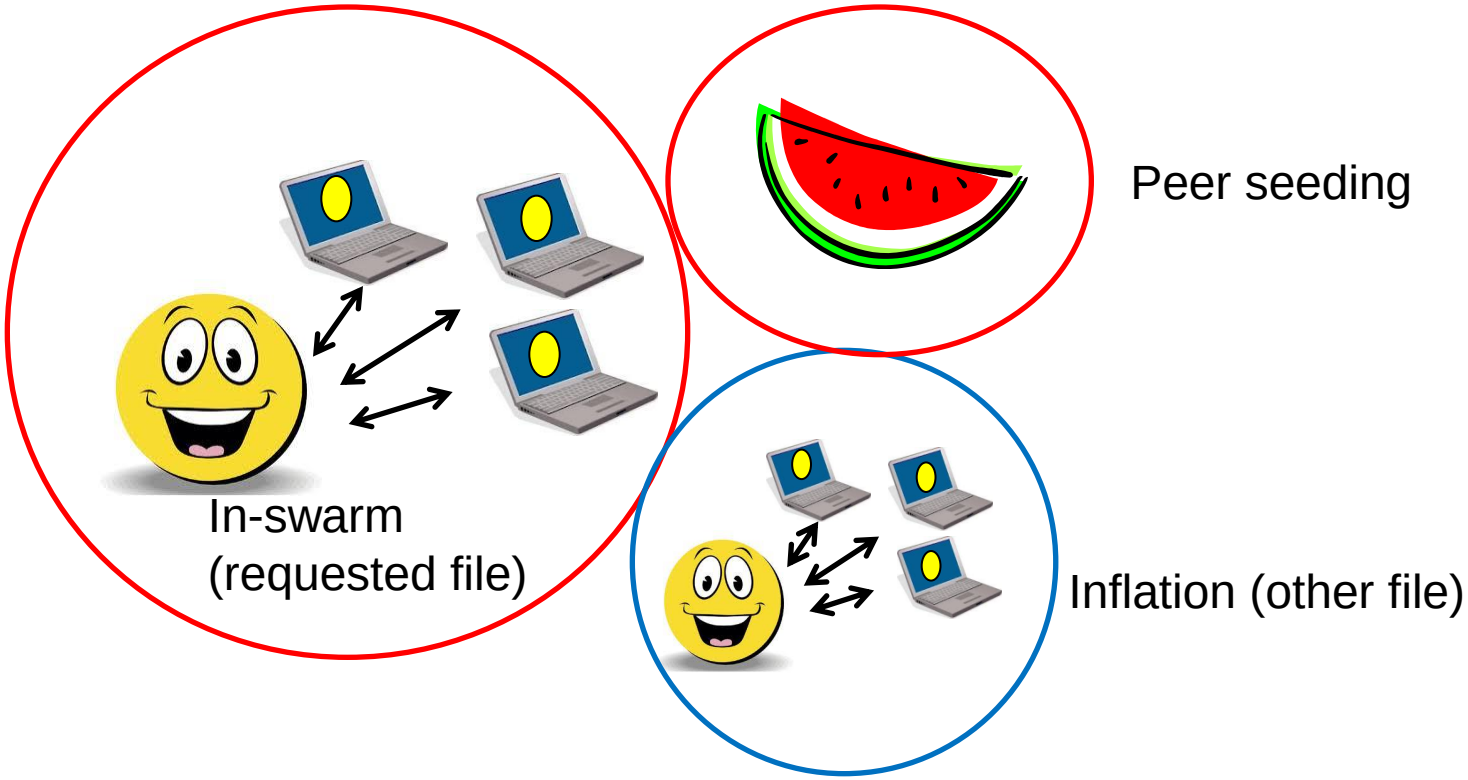
Peer upload bandwidth



- Service guarantee T
- Help out, but **only** during download/service

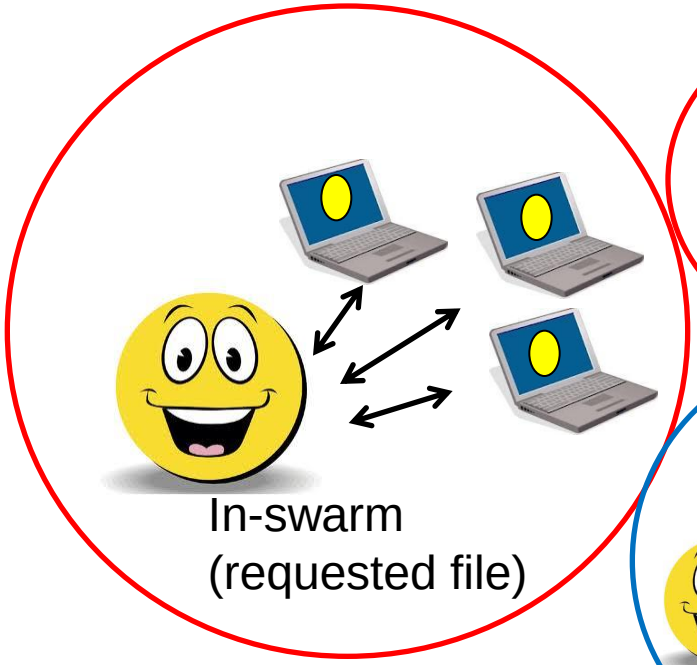
$$\sum_{i \in N} \phi_i L + \sum_{i \in N} B_i^p \leq \sum_{i \in N} \lambda_i (UT - L)$$

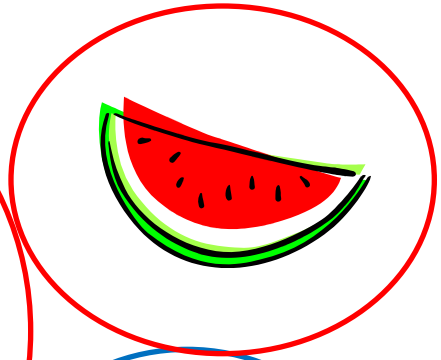
Peer upload bandwidth



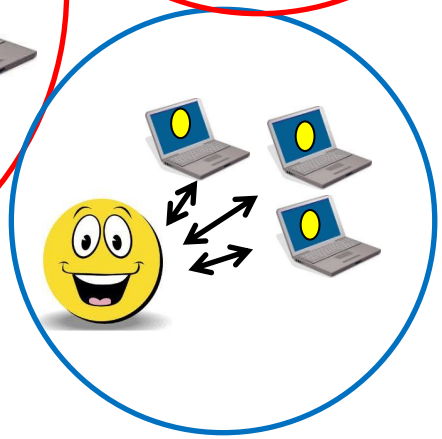
$$\sum_{i \in N} \phi_i L + \sum_{i \in N} B_i^p \leq \sum_{i \in N} \lambda_i (UT - L)$$

Peer upload bandwidth





Peer seeding

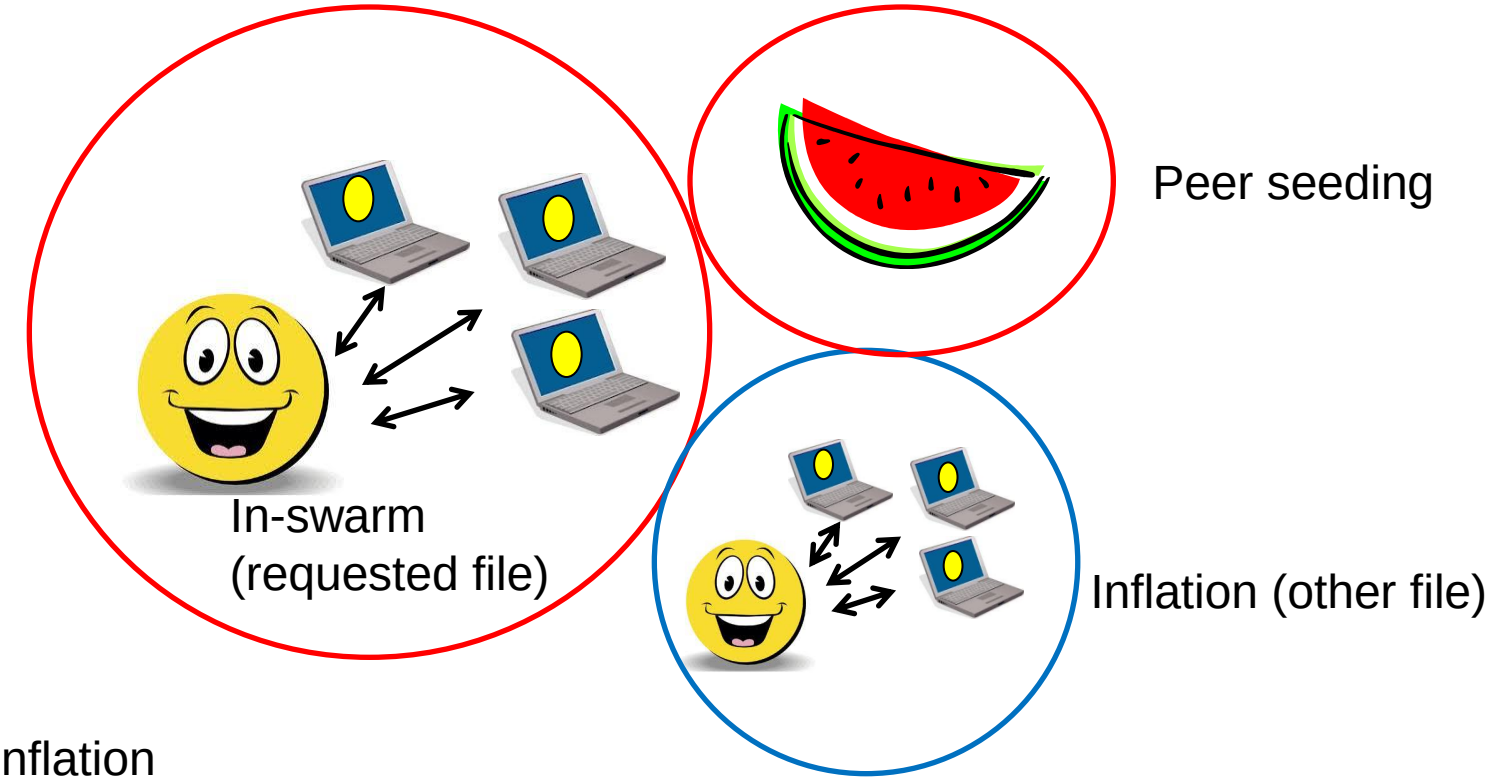


Inflation (other file)

Spare upload bandwidth

$$\sum_{i \in N} \phi_i L + \sum_{i \in N} B_i^p \leq \sum_{i \in N} \lambda_i (UT - L)$$

Peer upload bandwidth

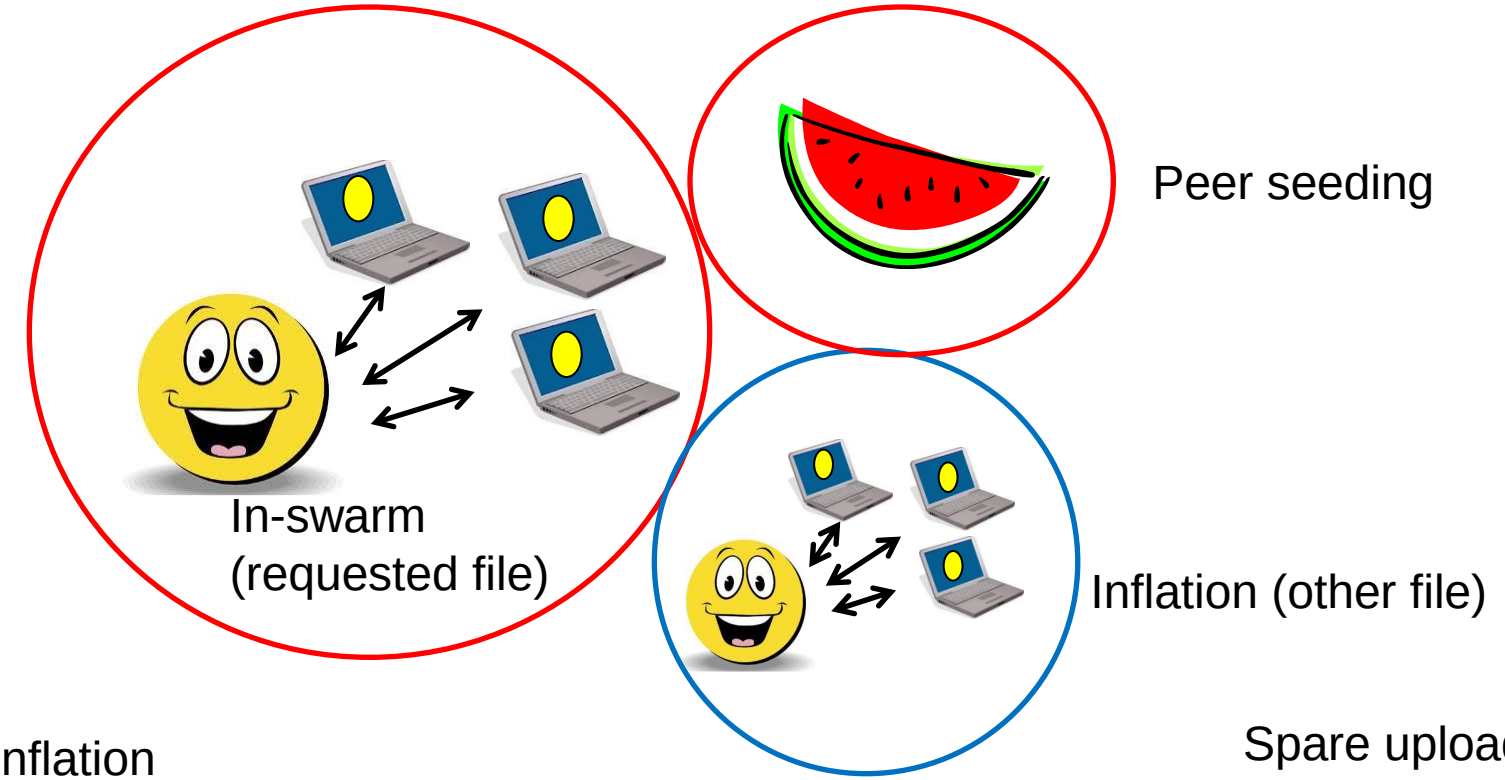


Inflation contributions

Peer seeding

$$\sum_{i \in N} \phi_i L + \sum_{i \in N} B_i^p \leq \sum_{i \in N} \lambda_i (UT - L)$$

Peer upload bandwidth



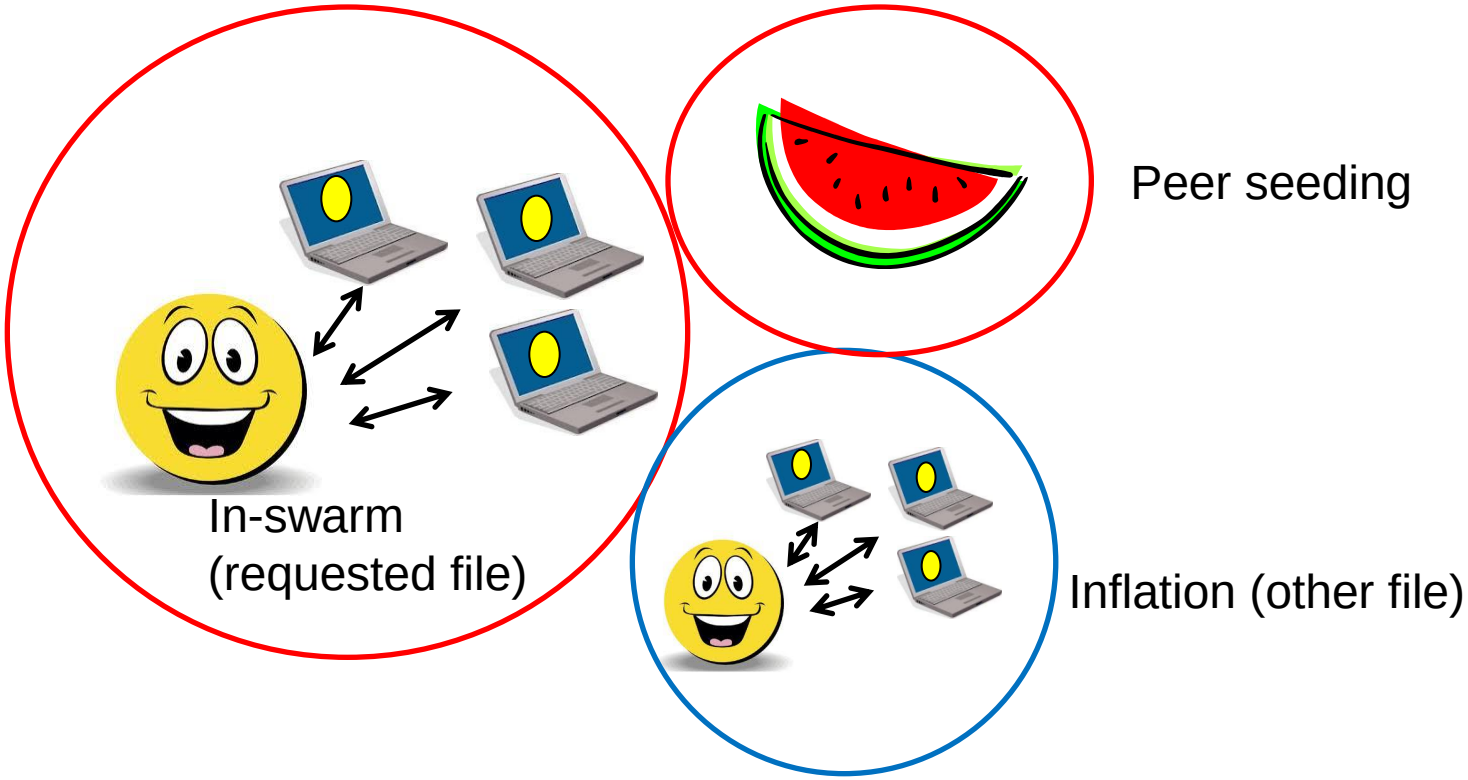
Inflation contributions

Peer seeding

Spare upload bandwidth

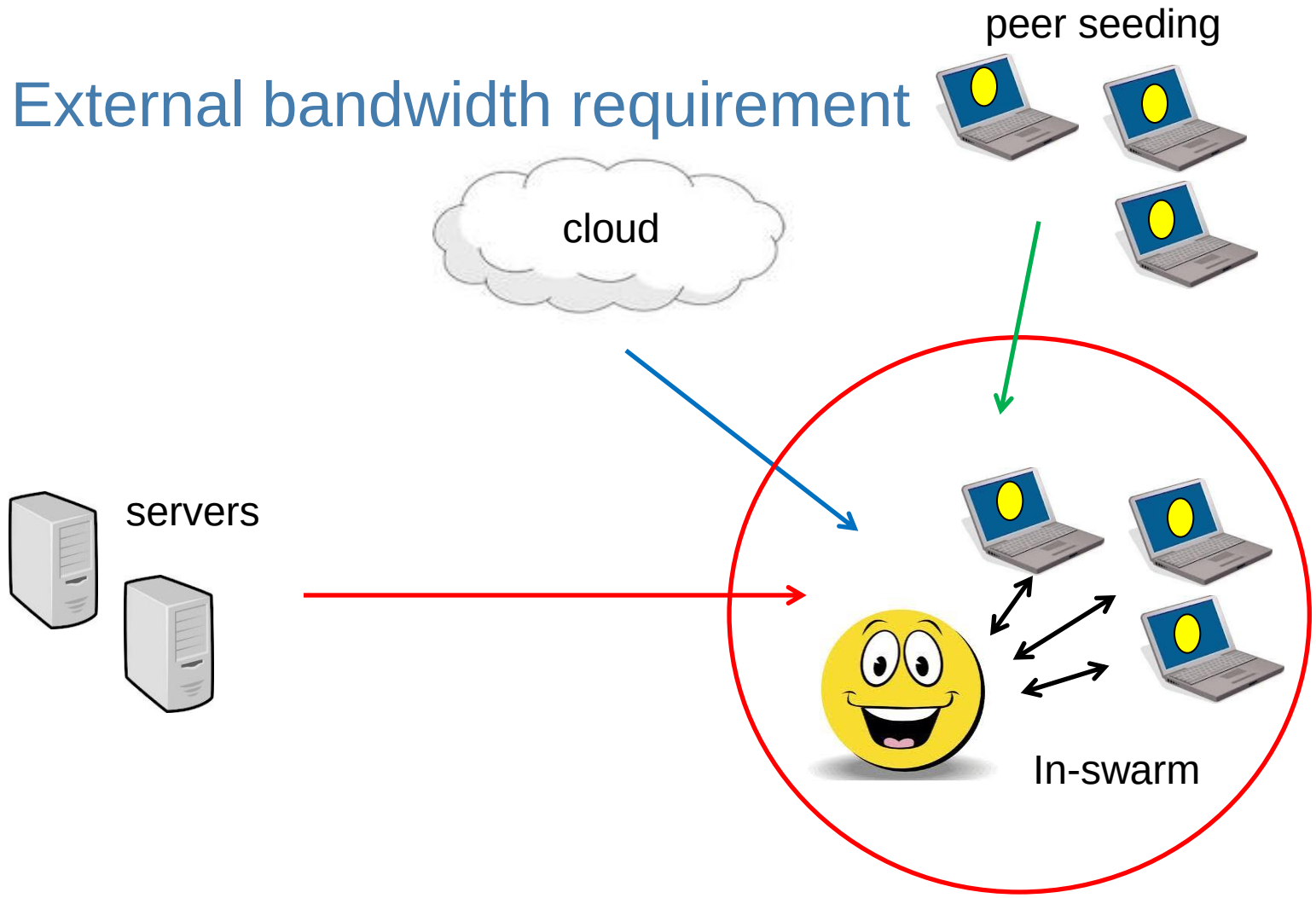
$$\sum_{i \in N} \phi_i L + \sum_{i \in N} B_i^p \leq \sum_{i \in N} \lambda_i (UT - L)$$

Peer upload bandwidth

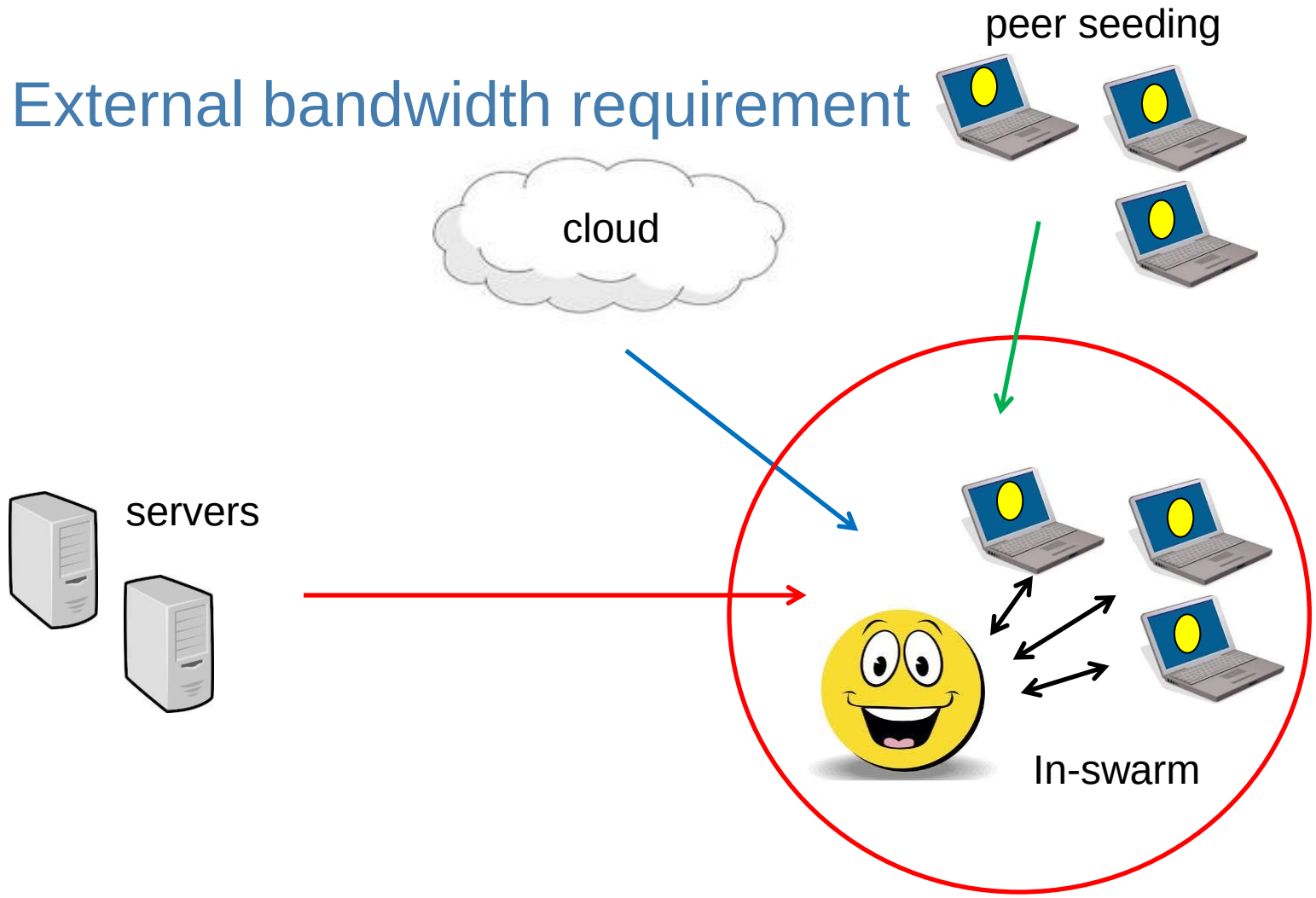


$$\sum_{i \in N} \phi_i L + \sum_{i \in N} B_i^p \leq \sum_{i \in N} \lambda_i (UT - L)$$

External bandwidth requirement

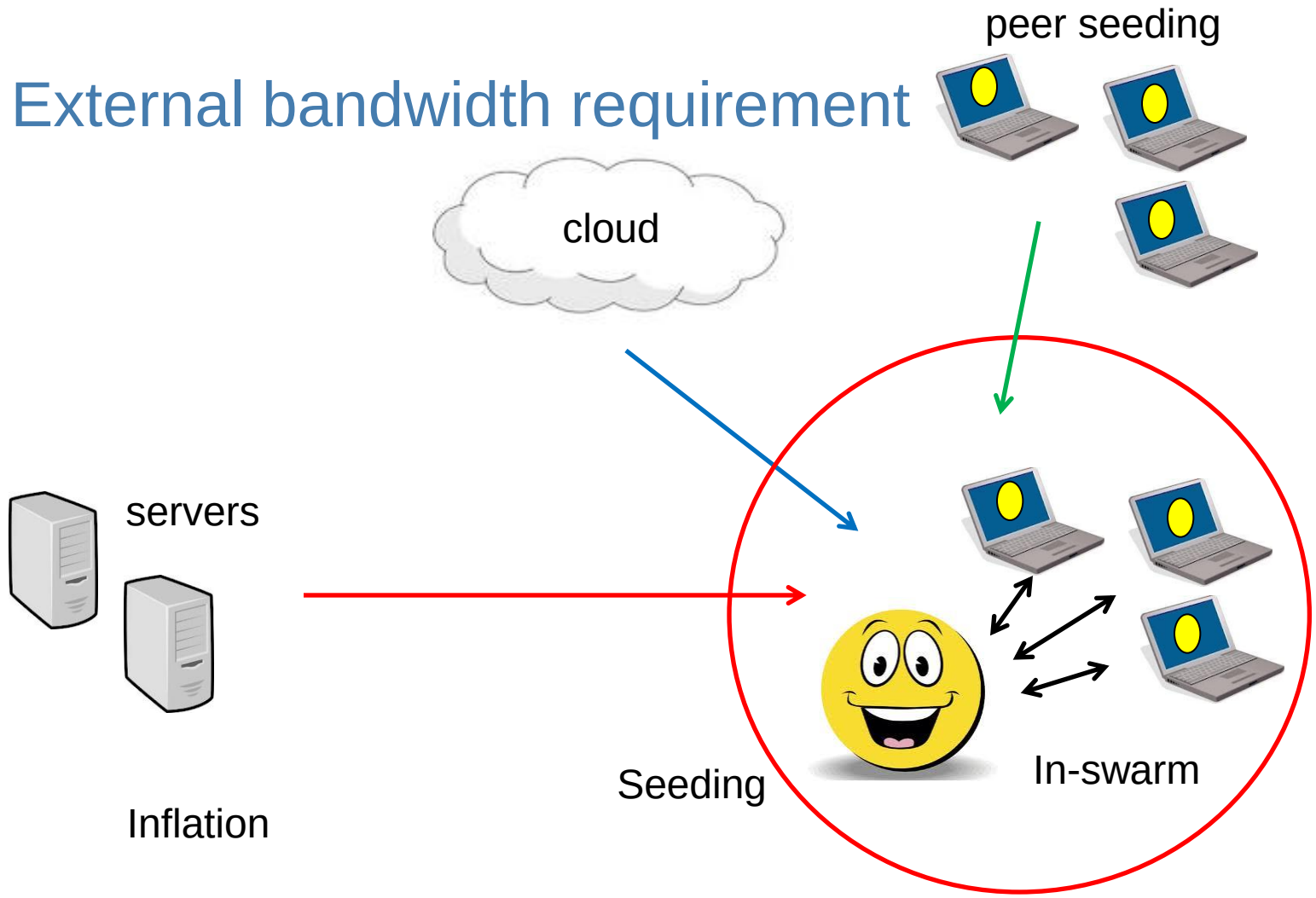


External bandwidth requirement



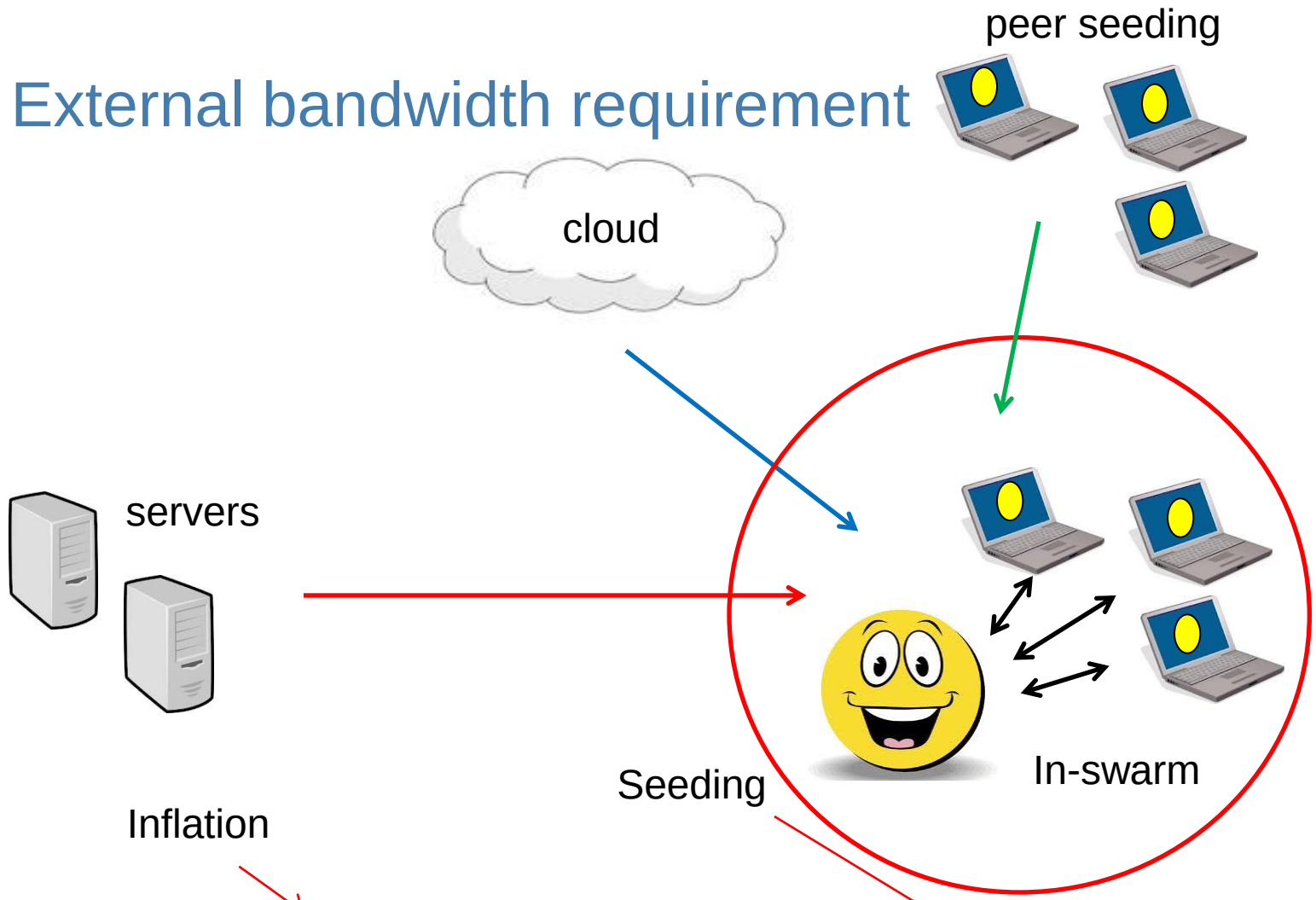
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

External bandwidth requirement



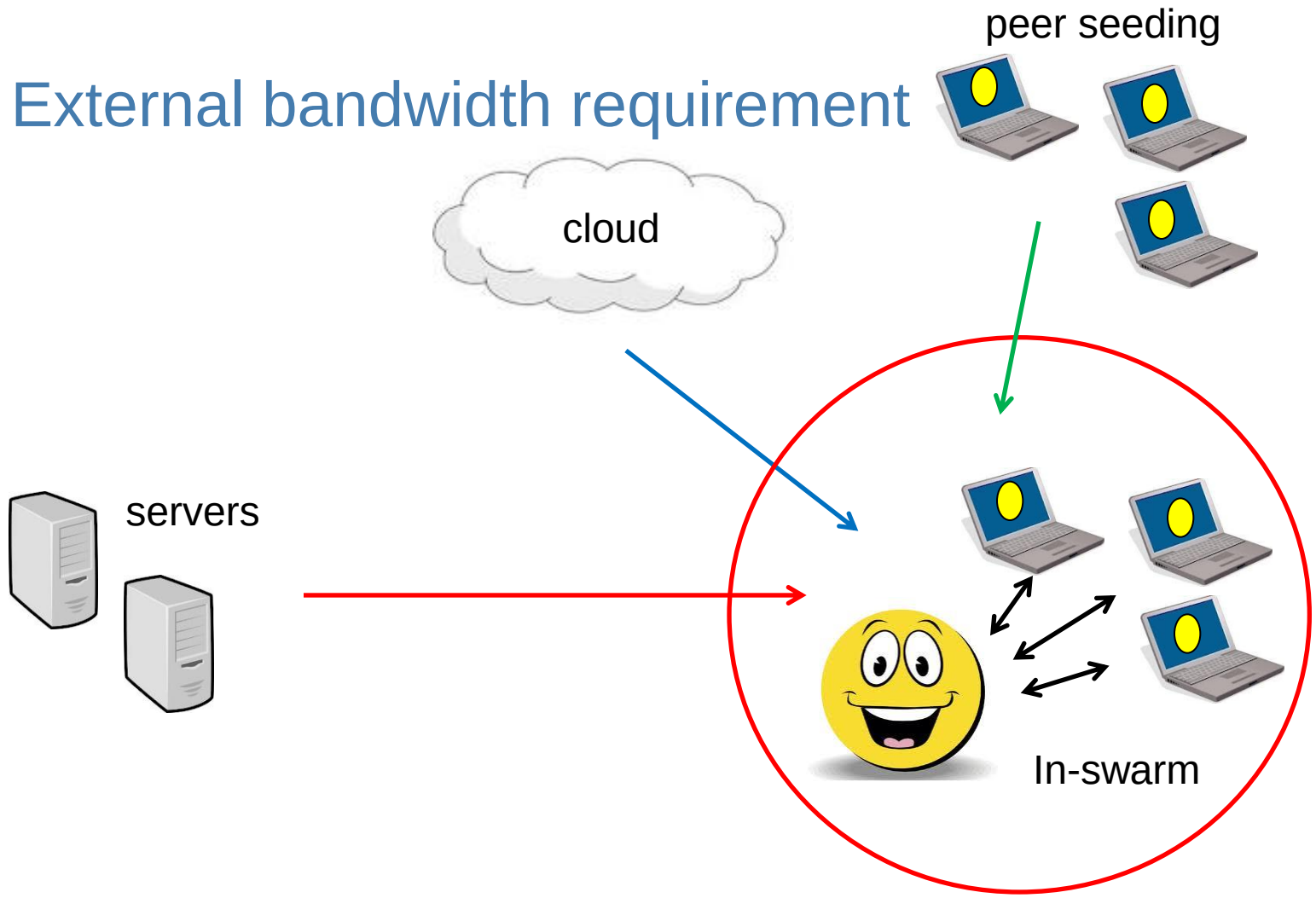
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

External bandwidth requirement



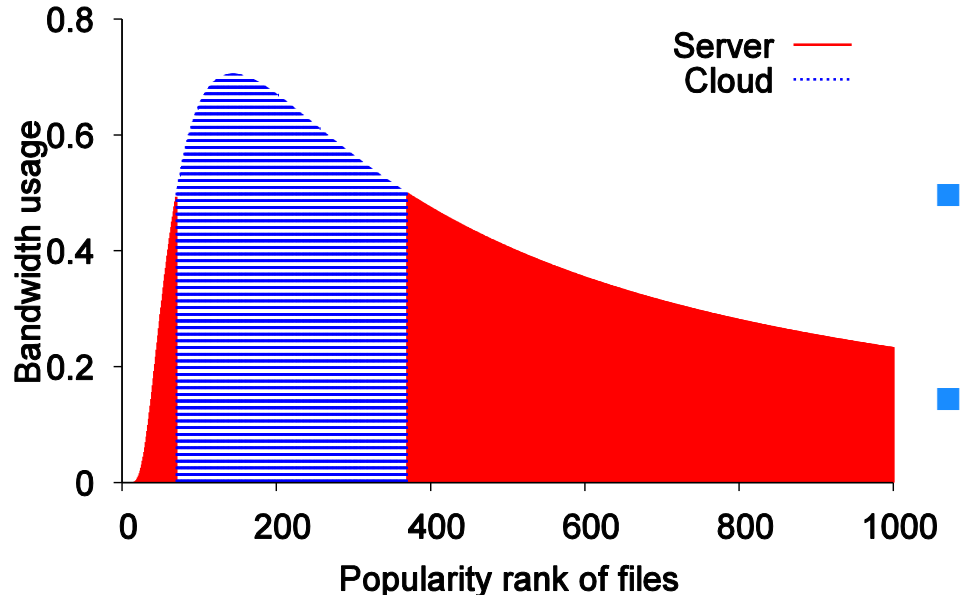
$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

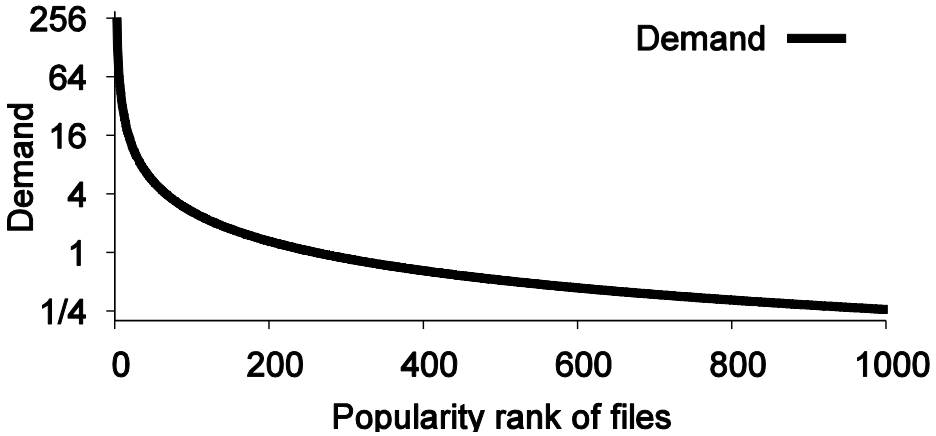
External bandwidth requirement



$$B(\lambda_i + \phi_i) \leq B_i^s + \sum_{j \in P_i} B_{ij}^c + B_i^p$$

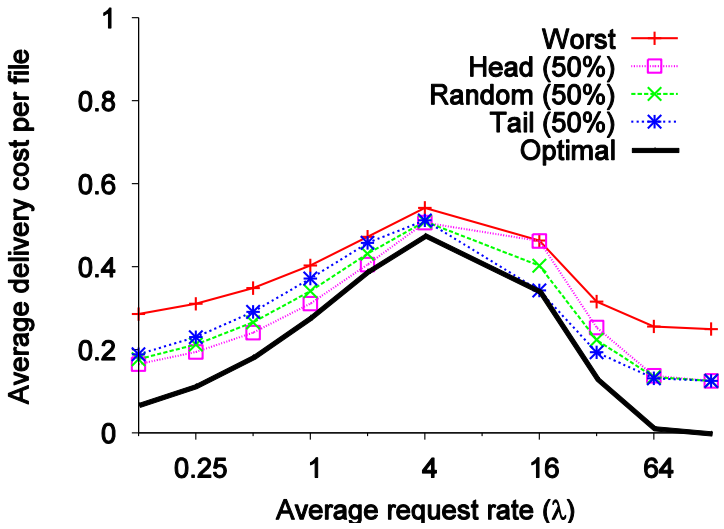
No seeding or bundling (UT=L)

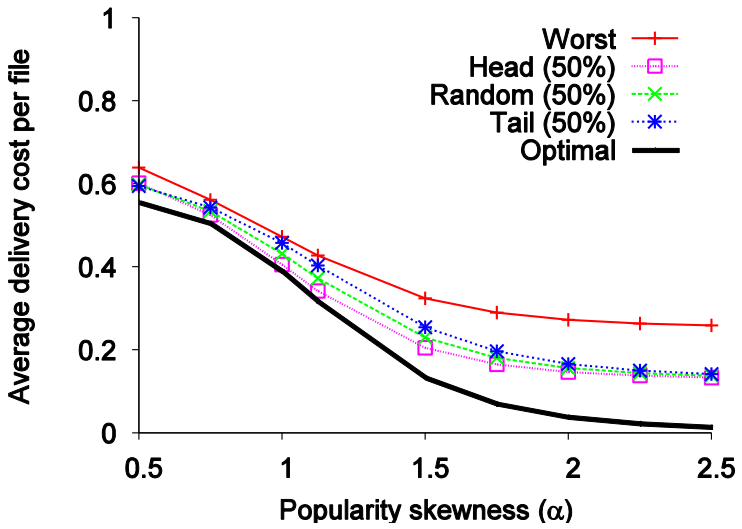




- Example allocation for optimal policy (when no seeding or bundling)
- Intermediately popular files pushed to the cloud

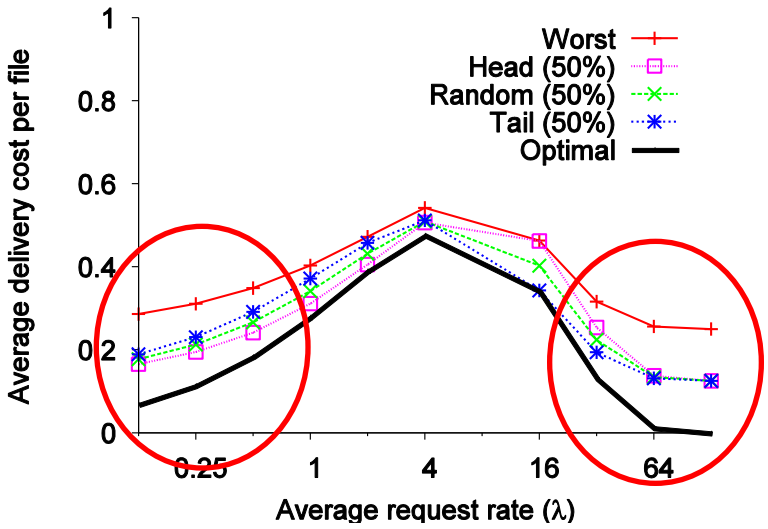
No seeding or bundling (UT=L)

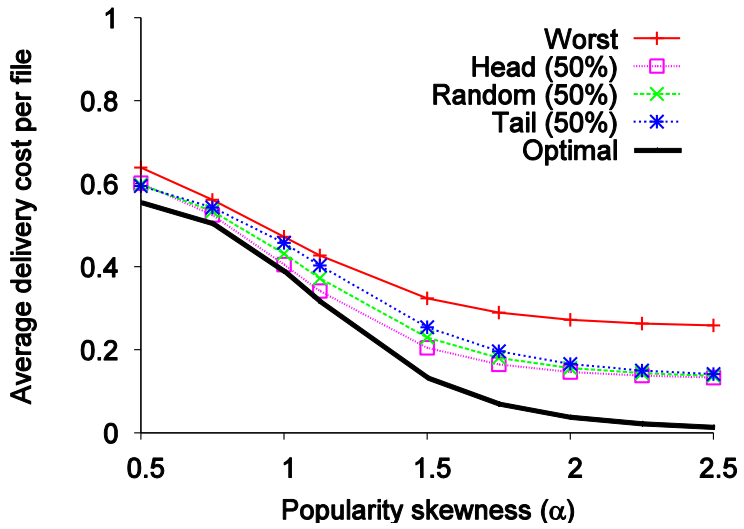




- Policy comparison (which files to push to cloud)
  - Optimal (intermediate) vs. baseline policies
- Big differences when either
  - High/low load
  - High popularity skew
- Catalogue size has little impact (not shown)

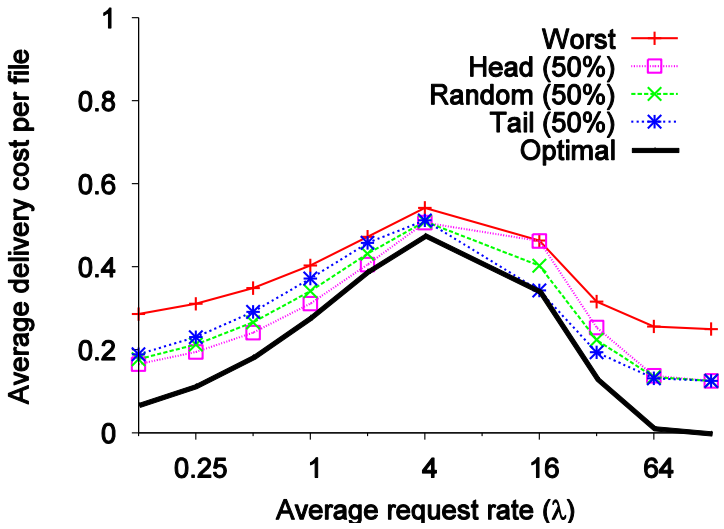
No seeding or bundling (UT=L)

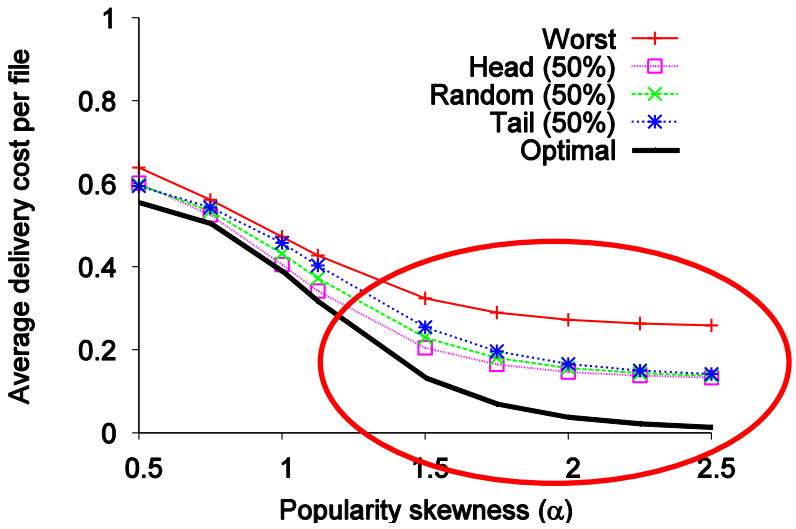




- Policy comparison (which files to push to cloud)
  - Optimal (intermediate) vs. baseline policies
- Big differences when either
  - High/low load
  - High popularity skew
- Catalogue size has little impact (not shown)

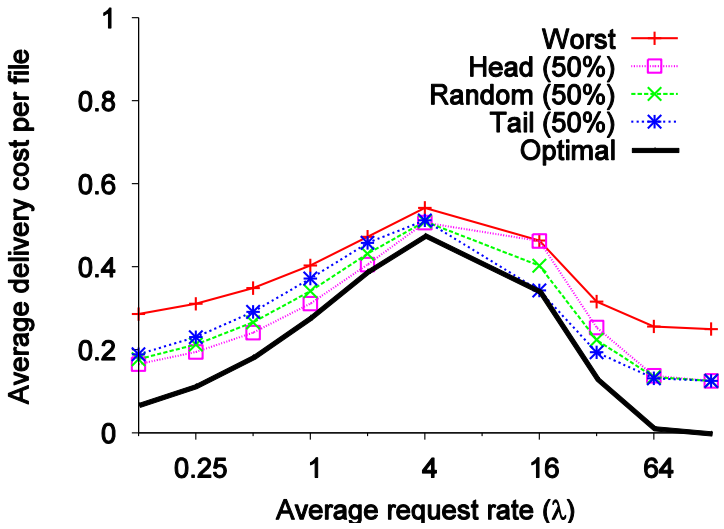
No seeding or bundling (UT=L)

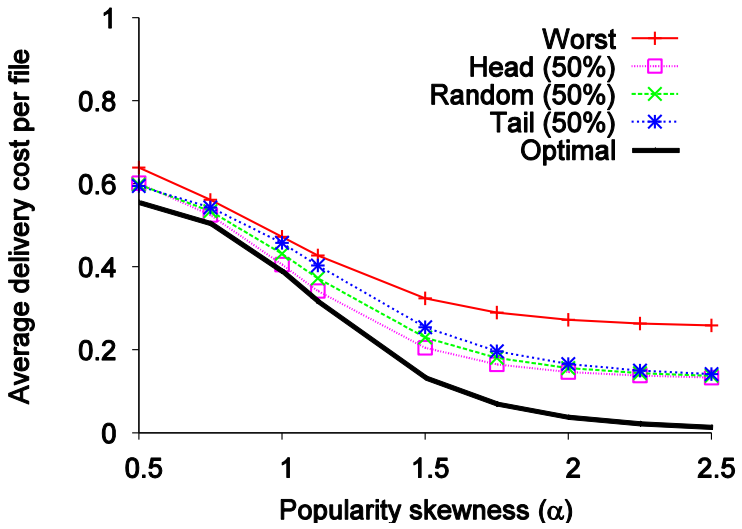




- Policy comparison (which files to push to cloud)
  - Optimal (intermediate) vs. baseline policies
- Big differences when either
  - High/low load
  - High popularity skew
- Catalogue size has little impact (not shown)

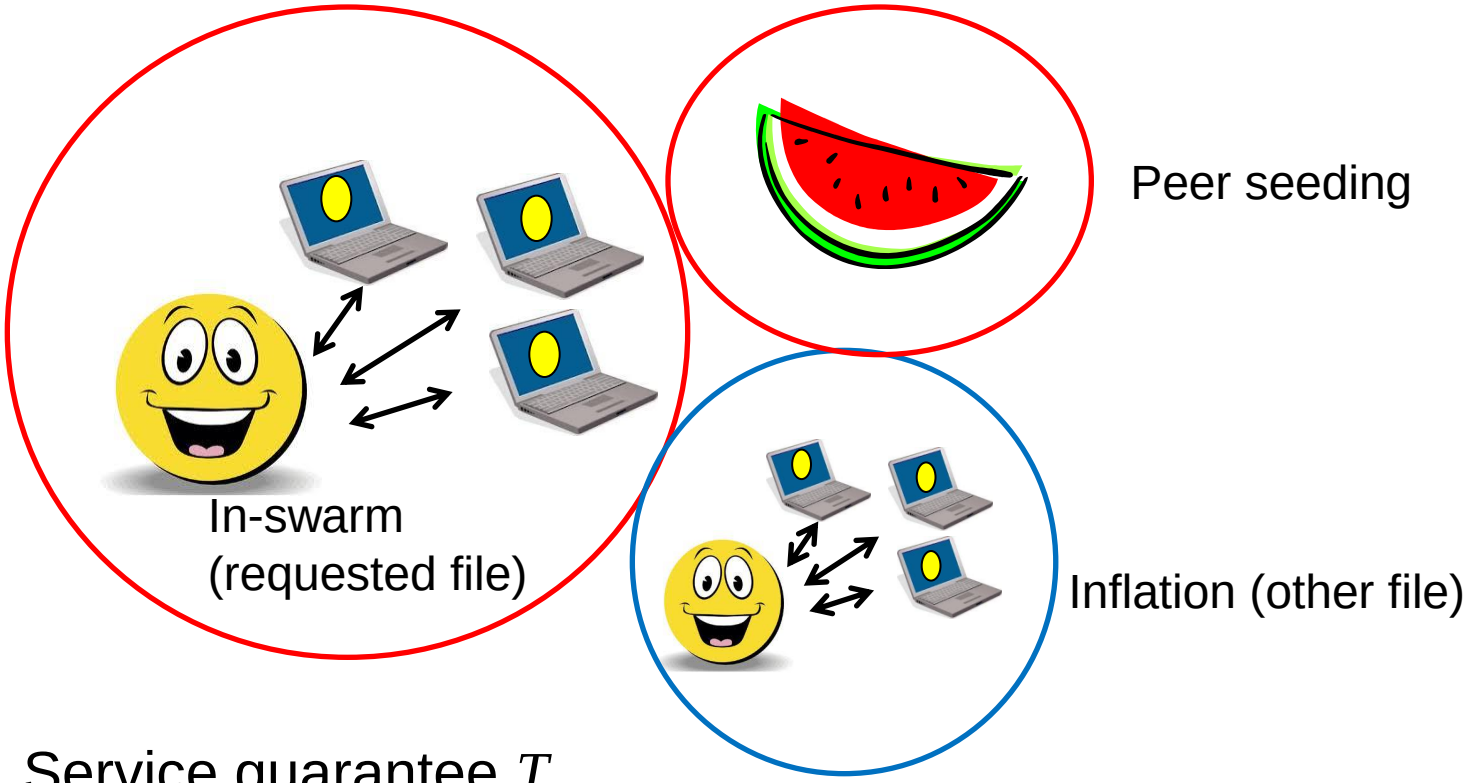
No seeding or bundling (UT=L)





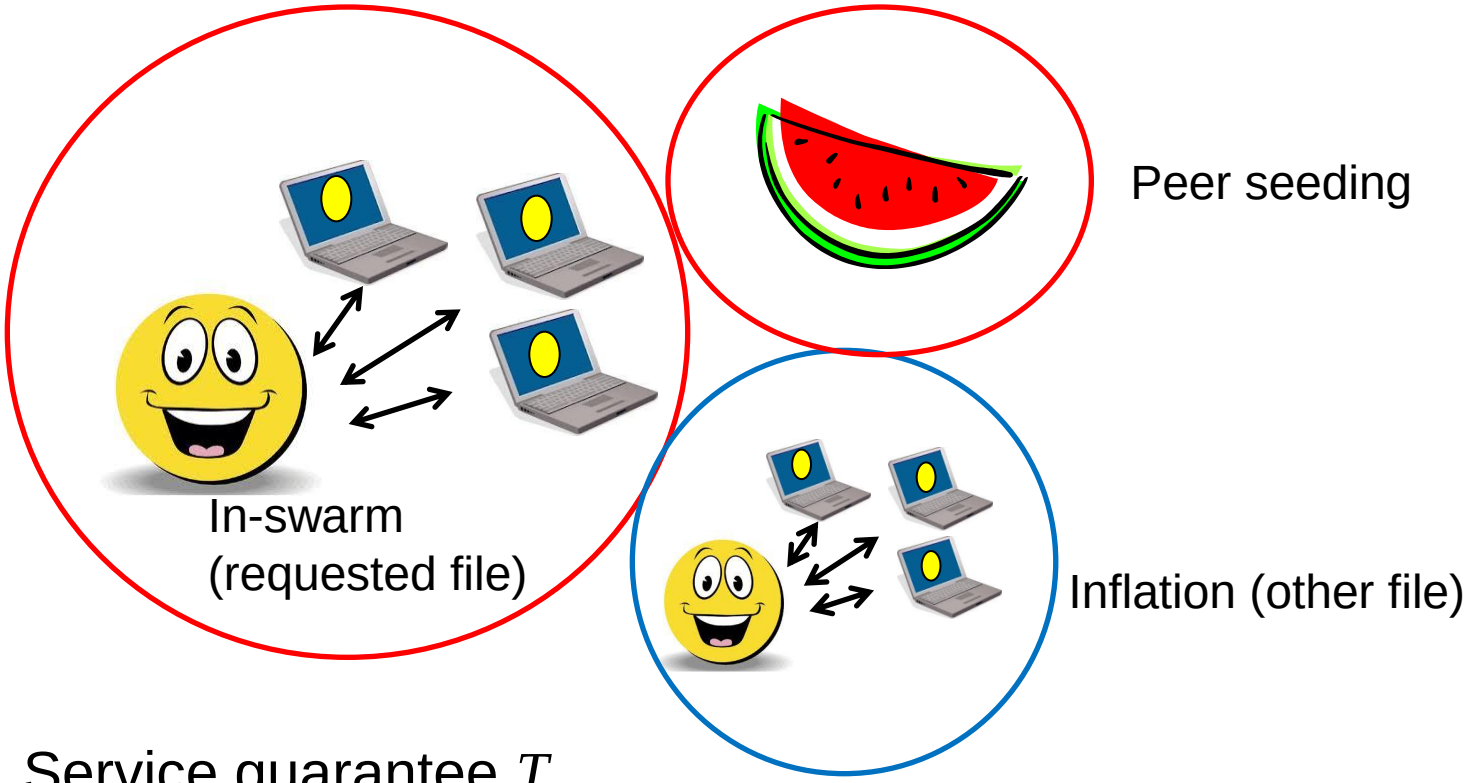
- Policy comparison (which files to push to cloud)
  - Optimal (intermediate) vs. baseline policies
- Big differences when either
  - High/low load
  - High popularity skew
- Catalogue size has little impact (not shown)

Peer upload bandwidth



- Service guarantee T
- Help out, but **only** during download/service

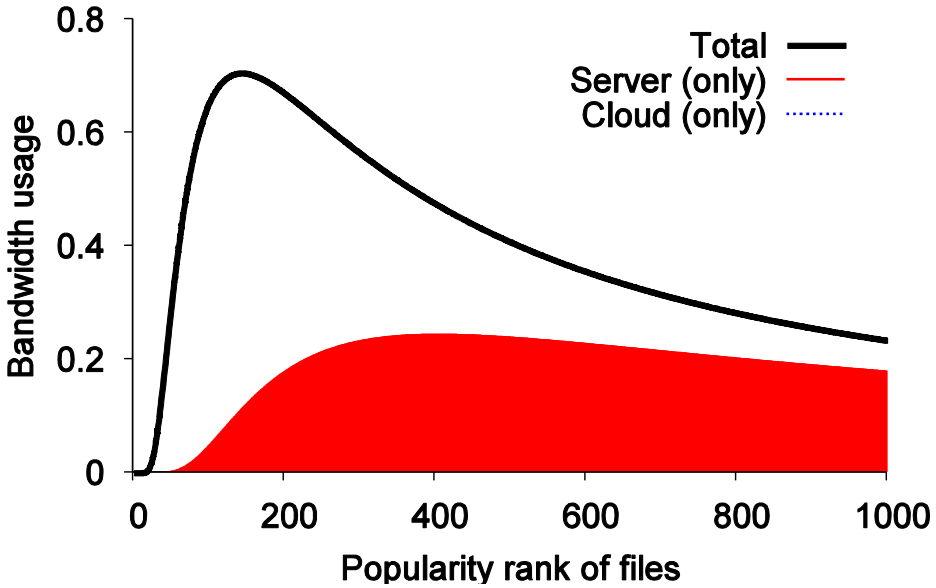
Peer upload bandwidth



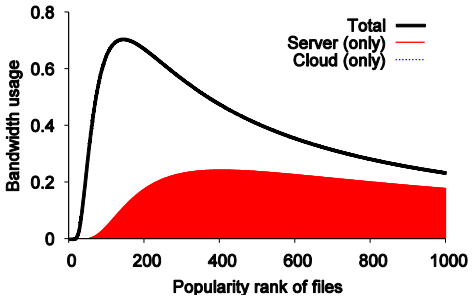
- Service guarantee T
- Help out, but **only** during download/service

How to best use this bandwidth?

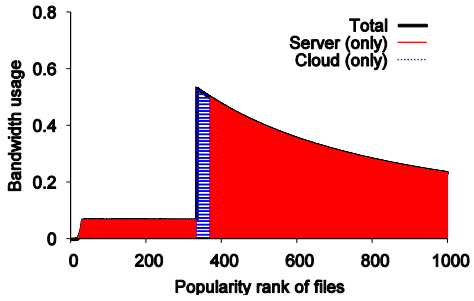
Seeding only (UT > L)



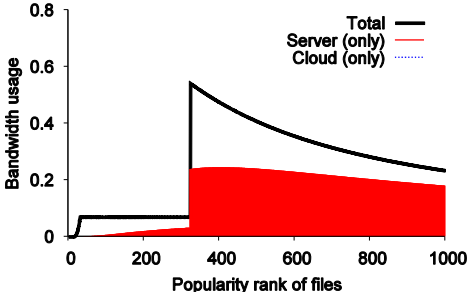
- Help less popular



Seeding (only)

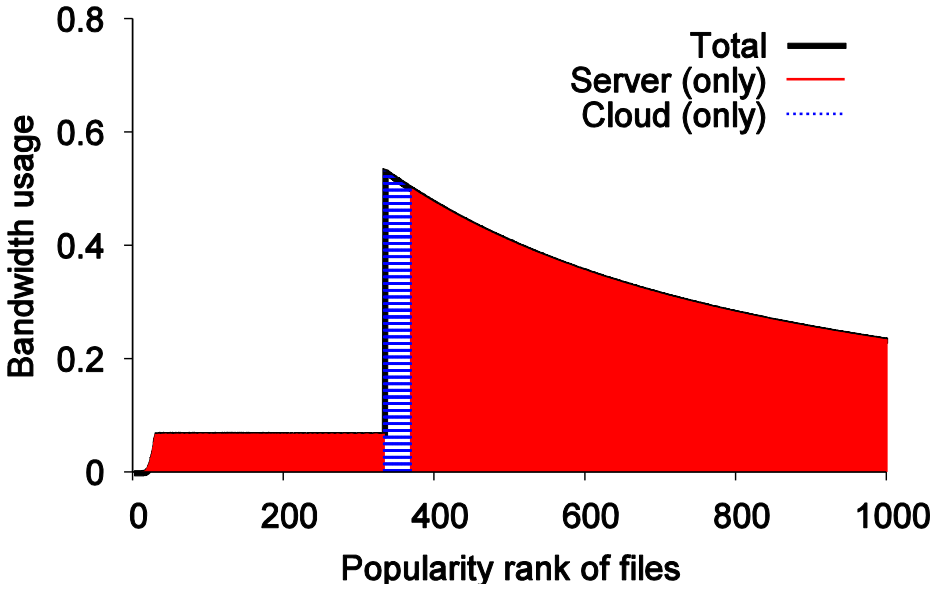


Bundling (only)

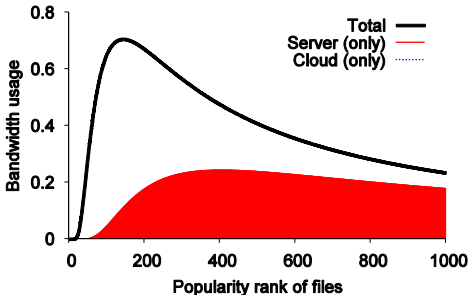


Hybrid

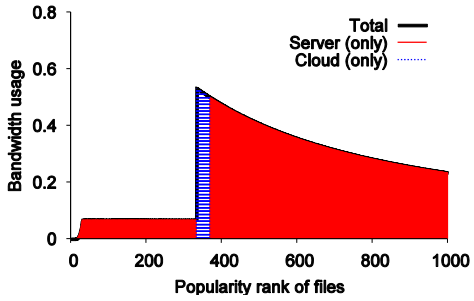
Bundling only (UT > L)



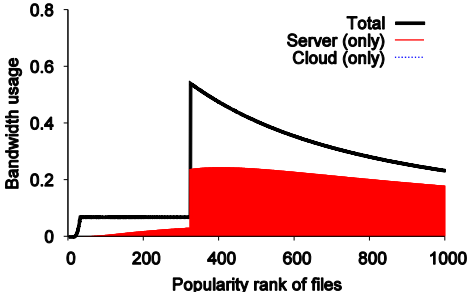
- Inflate most popular
- Intermediate to cloud



Seeding (only)

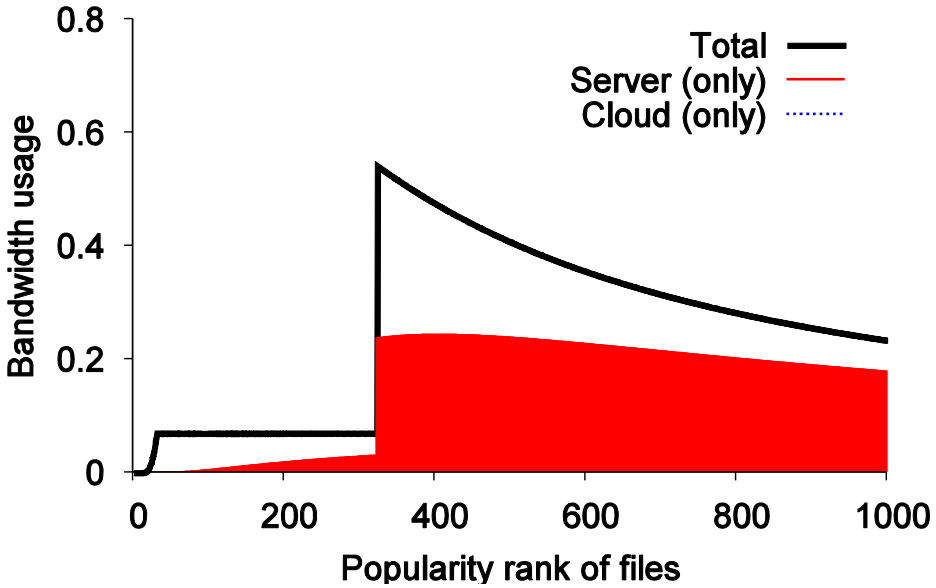


Bundling (only)

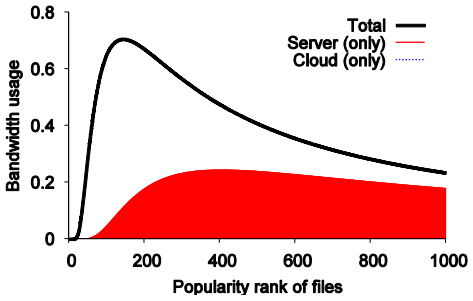


Hybrid

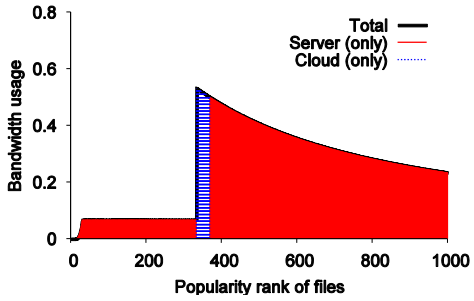
Hybrid (UT > L)



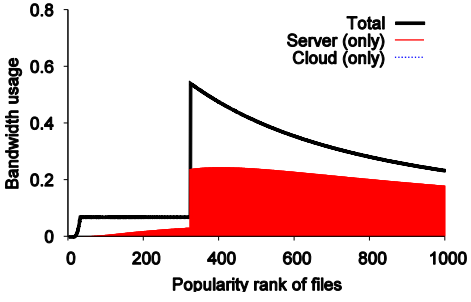
- Inflate most popular
- Depends less on cloud



Seeding (only)



Bundling (only)

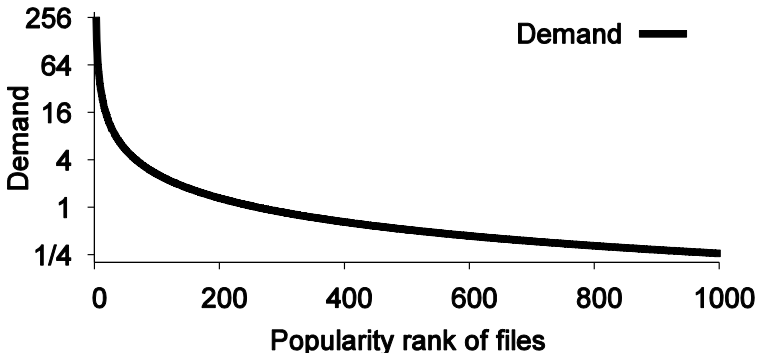


Hybrid

Policy comparison

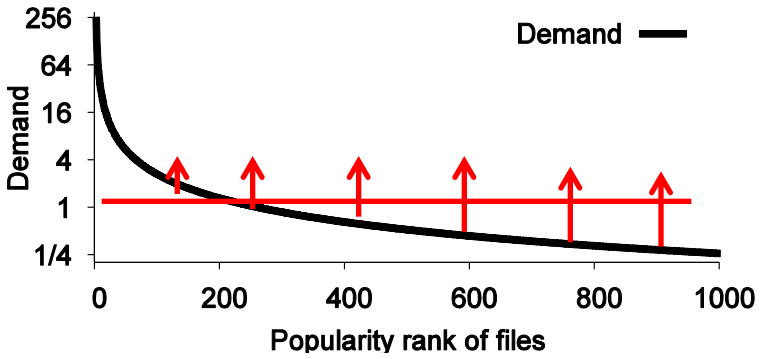
- How did we decide which files to “inflate”?
- Baseline inflation policies with different complexity
  - Proportional: $\phi_i \propto \lambda_i$ (based on random peer interest, to help friends, for example)
  - Random: $\phi_i \propto 1$ (same for all)
  - Basic: yes/no decision using base allocation; same to all
  - Fine: Greedy search (with “basic” as starting point)
  - Other baseline inflation policies [IFIP Networking ‘10]

Policy comparison



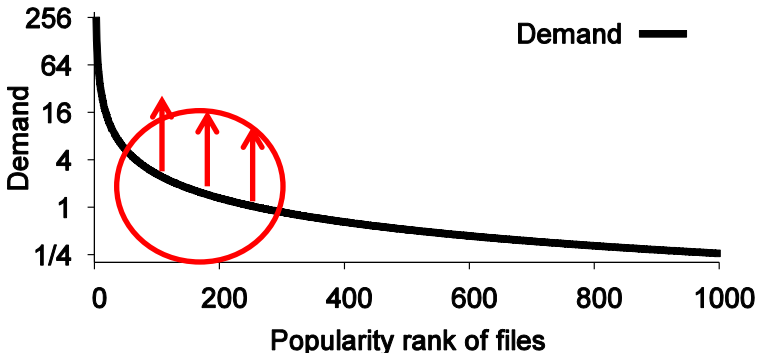
- How did we decide which files to “inflate”?
- Baseline inflation policies with different complexity
  - Proportional: $\phi_i \propto \lambda_i$ (based on random peer interest, to help friends, for example)
  - Random: $\phi_i \propto 1$ (same for all)
  - Basic: yes/no decision using base allocation; same to all
  - Fine: Greedy search (with “basic” as starting point)
  - Other baseline inflation policies [IFIP Networking ‘10]

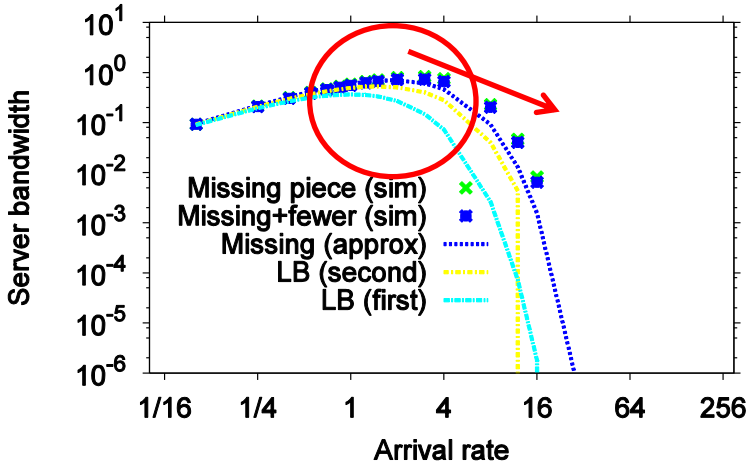
Policy comparison



- How did we decide which files to “inflate”?
- Baseline inflation policies with different complexity
  - Proportional: $\phi_i \propto \lambda_i$ (based on random peer interest, to help friends, for example)
  - Random: $\phi_i \propto 1$ (same for all)
  - Basic: yes/no decision using base allocation; same to all
  - Fine: Greedy search (with “basic” as starting point)
  - Other baseline inflation policies [IFIP Networking ‘10]

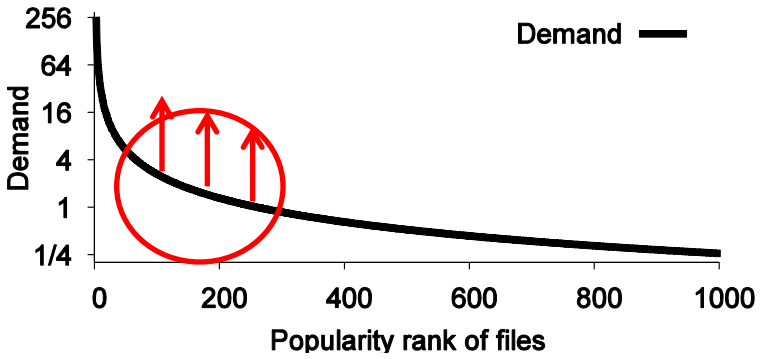
Policy comparison

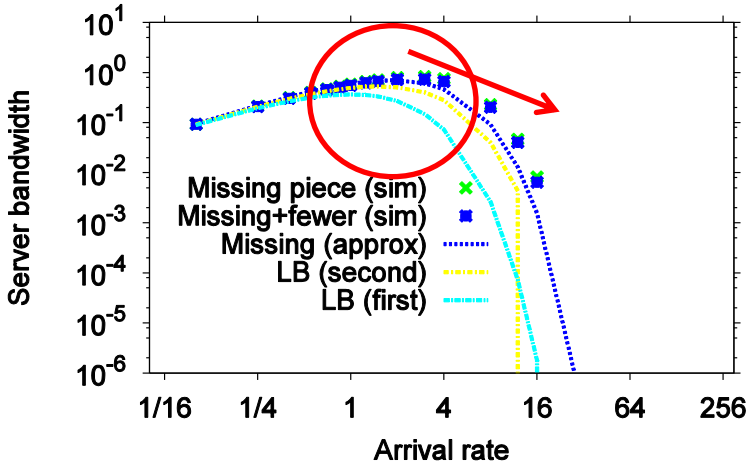




- How did we decide which files to "inflate"?
- Baseline inflation policies with different complexity
  - Proportional: $\phi_i \propto \lambda_i$ (based on random peer interest, to help friends, for example)
  - Random: $\phi_i \propto 1$ (same for all)
  - Basic: yes/no decision using base allocation; same to all
  - Fine: Greedy search (with "basic" as starting point)
  - Other baseline inflation policies [IFIP Networking '10]

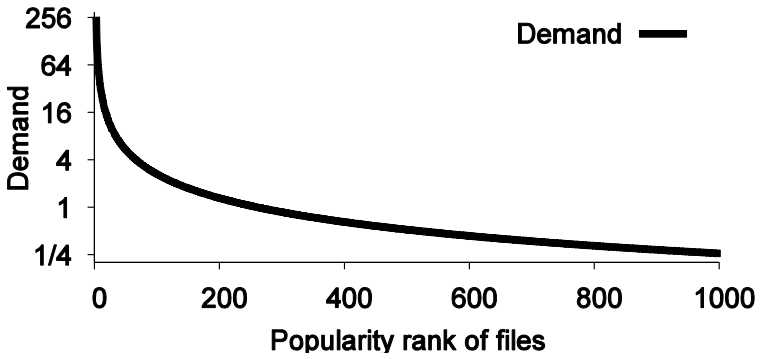
Policy comparison

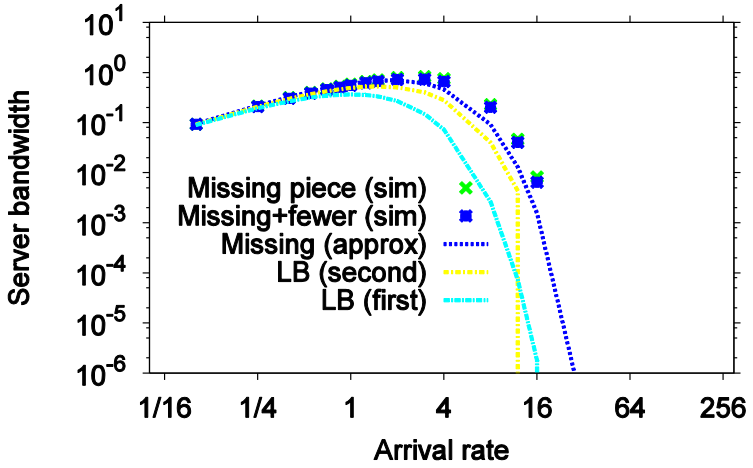




- How did we decide which files to "inflate"?
- Baseline inflation policies with different complexity
  - Proportional: $\phi_i \propto \lambda_i$ (based on random peer interest, to help friends, for example)
  - Random: $\phi_i \propto 1$ (same for all)
  - Basic: yes/no decision using base allocation; same to all
  - Fine: Greedy search (with "basic" as starting point)
  - Other baseline inflation policies [IFIP Networking '10]

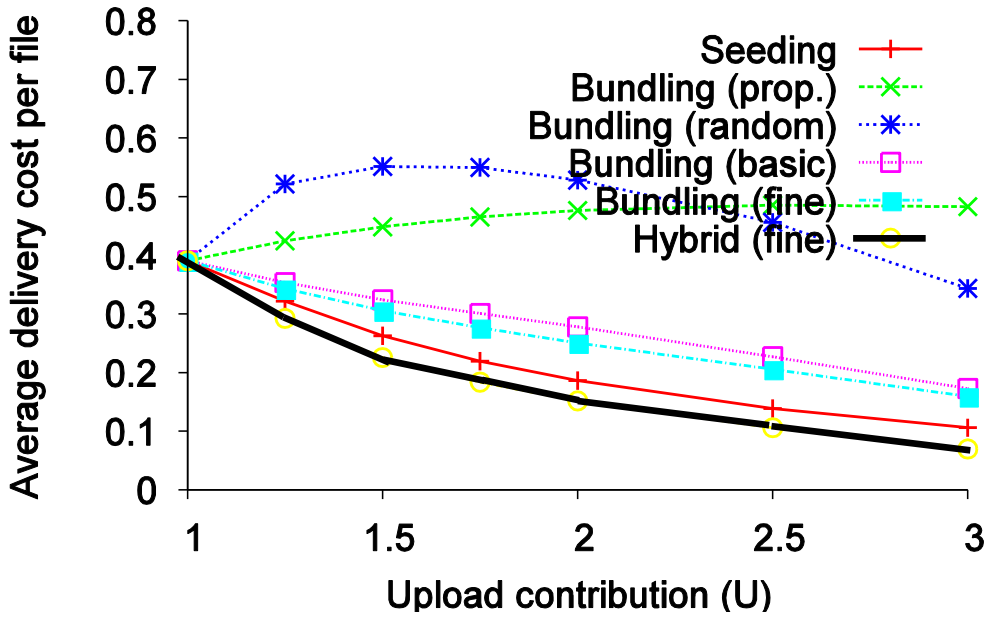
Policy comparison





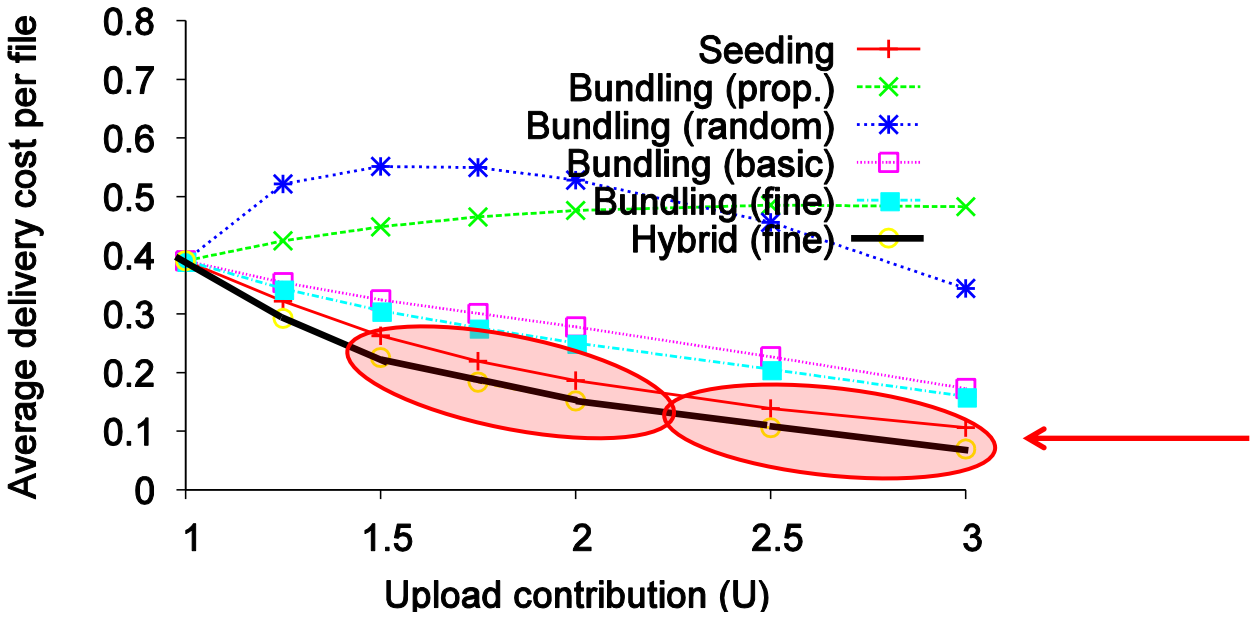
- How did we decide which files to “inflate”?
- Baseline inflation policies with different complexity
  - Proportional: $\phi_i \propto \lambda_i$ (based on random peer interest, to help friends, for example)
  - Random: $\phi_i \propto 1$ (same for all)
  - Basic: yes/no decision using base allocation; same to all
  - Fine: Greedy search (with “basic” as starting point)
  - Other baseline inflation policies [IFIP Networking ‘10]

Policy comparison



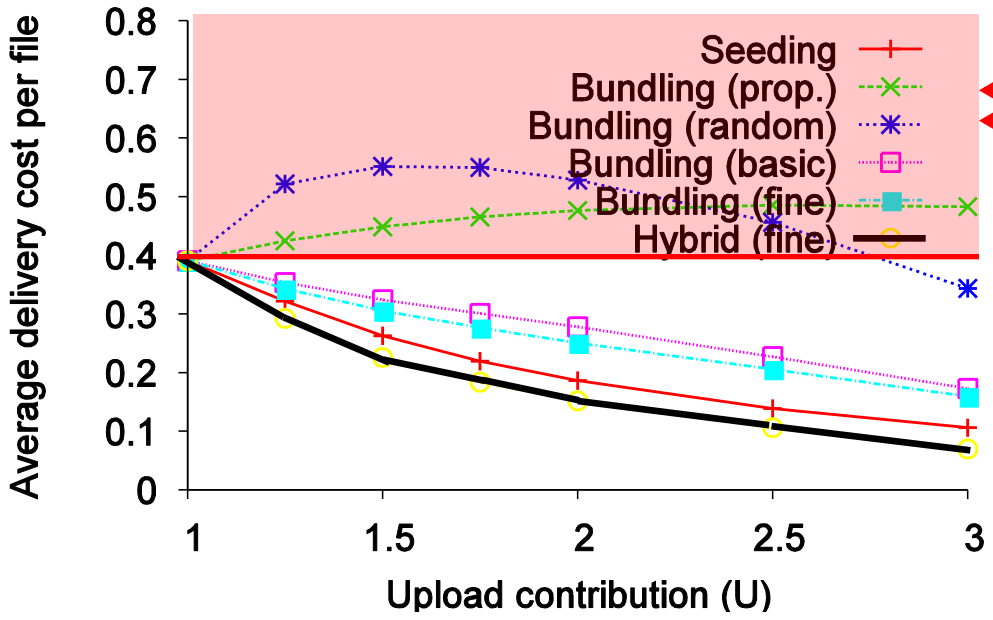
- Baseline inflation policies with different complexity
  - Up to 20% benefit using hybrid approach
  - Reckless use of bundling can be costly
  - Simpler (basic) policies achieves most of the benefits

Policy comparison



- Baseline inflation policies with different complexity
  - Up to 20% benefit using hybrid approach
  - Reckless use of bundling can be costly
  - Simpler (basic) policies achieves most of the benefits

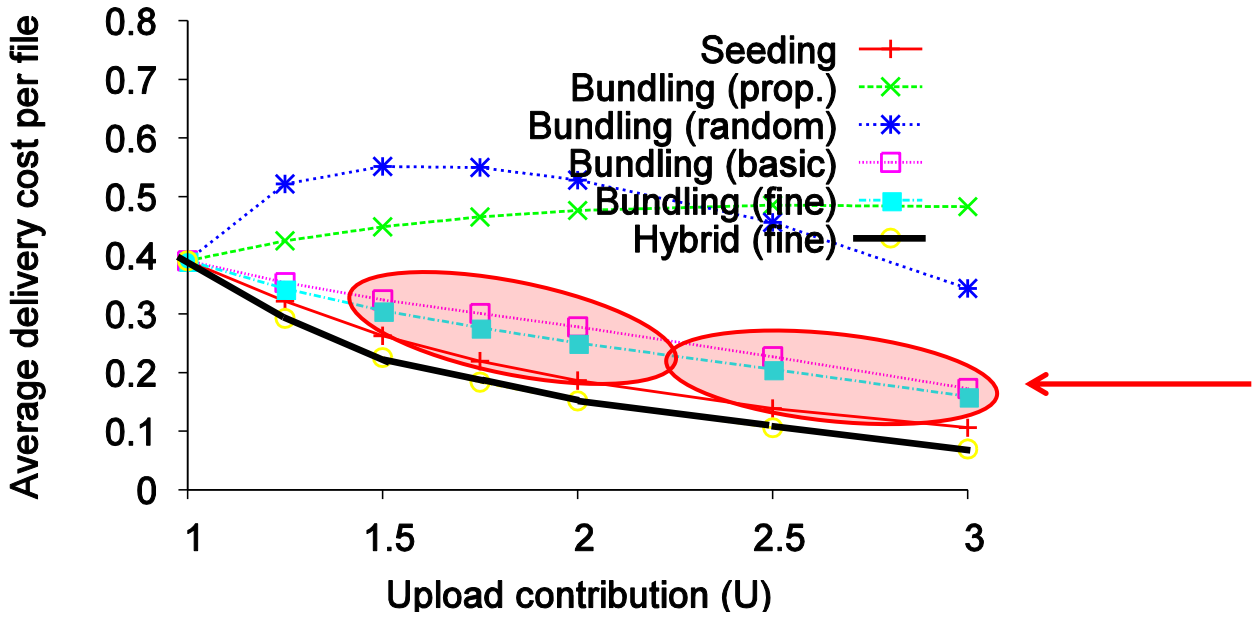
Policy comparison



Some cases worse than no bundling

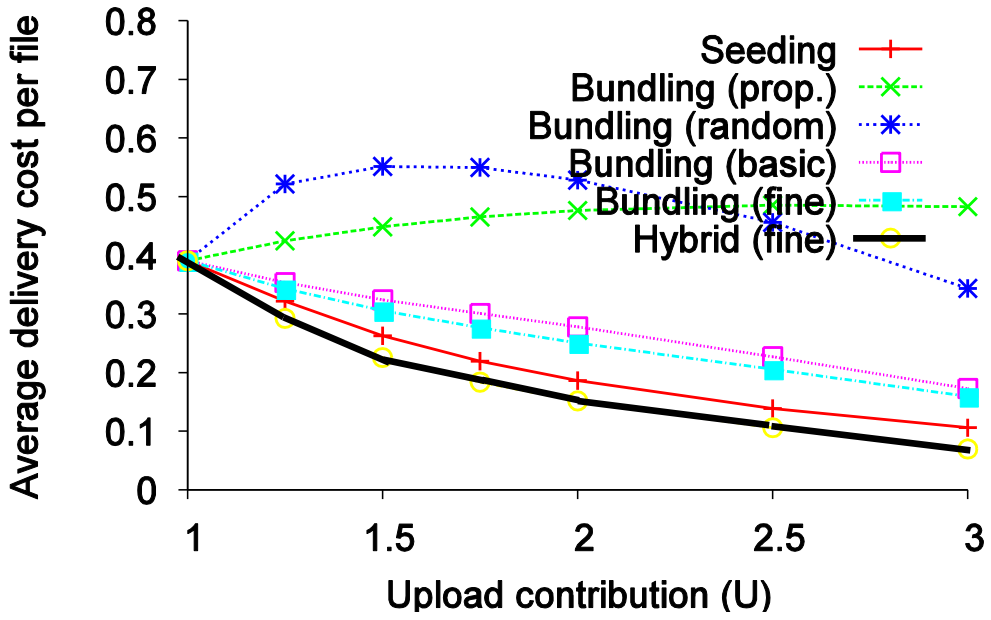
- Baseline inflation policies with different complexity
  - Up to 20% benefit using hybrid approach
  - Reckless use of bundling can be costly
  - Simpler (basic) policies achieves most of the benefits

Policy comparison



- Baseline inflation policies with different complexity
  - Up to 20% benefit using hybrid approach
  - Reckless use of bundling can be costly
  - Simpler (basic) policies achieves most of the benefits

Policy comparison



- Baseline inflation policies with different complexity
  - Up to 20% benefit using hybrid approach
  - Reckless use of bundling can be costly
  - Simpler (basic) policies achieves most of the benefits

Contributions

- Derive and evaluate bounds and approximations of the minimum server bandwidth required to ensure target average download rate
- Compare simple policy classes for which content to push to the cloud and provide insights regarding the importance of careful content selection
- Compare the best usage of the peer upload bandwidth, including policies determining how seeding and torrent inflation should be best utilized
- Also (in paper):
  - Where to direct clients in systems where the cloud provider has a differentiated cost model and charges based on the locality of the clients that are served

Thank you!

- Niklas Carlsson — Linköping University
- György Dan — KTH Royal Institute of Technology
- Derek Eager — University of Saskatchewan
- Anirban Mahanti — NICTA