THE UNIVERSITY OF TOKYO

A Solution Approach for AS Relationships-aware Overlay Routing

<draft-asai-cross-domain-overlay-00>

Hirochika Asai <panda@hongo.wide.ad.jp>, U. Tokyo

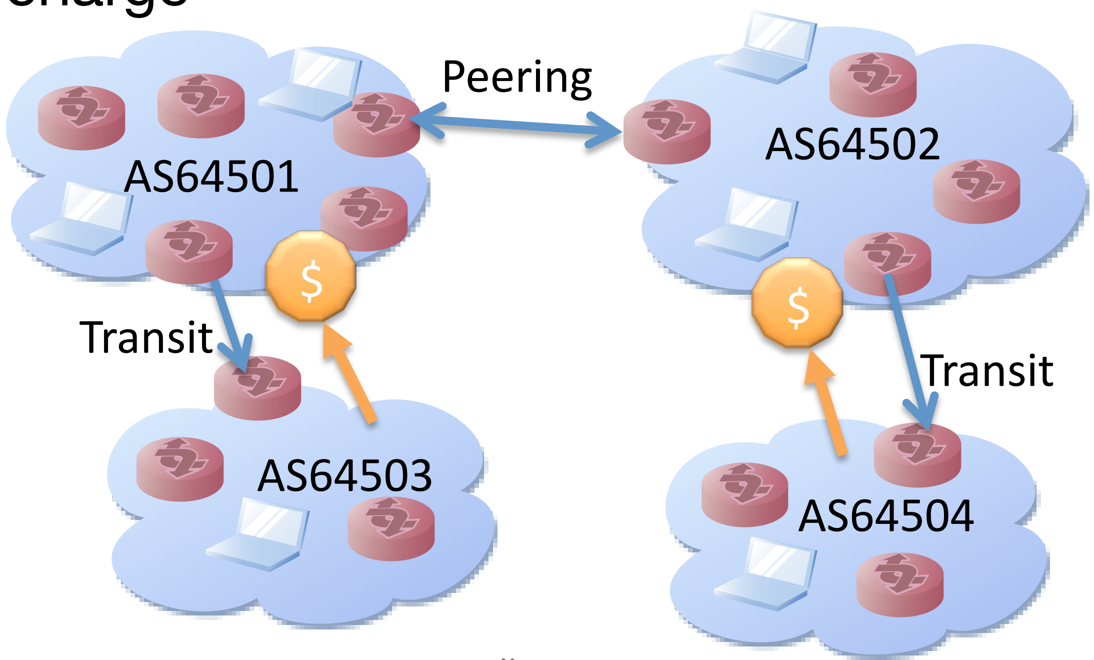
Hiroshi Esaki <hiroshi@wide.ad.jp>, U. Tokyo

Tsuyoshi Momose <tmomose@cisco.com>, Cisco Systems

July 28th, 2010, P2P RG, IETF 78

The Internet

- Autonomous systems (ASes)
  - e.g., ISPs, companies, and universities
- Inter-AS economics
  - transit charge



Motivation

- Reduction in transit traffic/charge
  - Why?
    - Most of ISPs providing their network to P2P nodes are residential ones.
      - i.e., not tier-1, but customers
    - Transit traffic costs more compared to peering or intra-domain traffic.

Approach

- Be aware of commercial relationships between ASes
    - in overlay networks
- Similar to ALTO
    - but
        - focusing mainly on cross-domain traffic
        - with hiding ISP's confidential information as much as possible (i.e., minimum ISP cooperation)

Detail with valley-free path model

Note; edge ASes accommodate CDN peers.

higher preference

	Source Edge: p2c	Source Edge: p2p	Source Edge: c2p
Destination Edge: p2c	+,+	+,0	+,-
Destination Edge: p2p	0,+	0,0	0,-
Destination Edge: c2p	-,+	-,0	-,-

the worst path: both ASes pay transit charge.

p2c: provider to customer
c2p: customer to provider
p2p: peer to peer

Requirements (1/2): AS relationships

- Information on AS relationships
  - Inference methods (N.B., assuming ISPs do not want to reveal AS relationships)
    - Path analysis [Gao2001, etc]
    - Adjacency analysis [Asai2010]
      - A method for provisioning this information to peers
        - » Inferred then provisioned from server(s)
          - ***written in the draft***

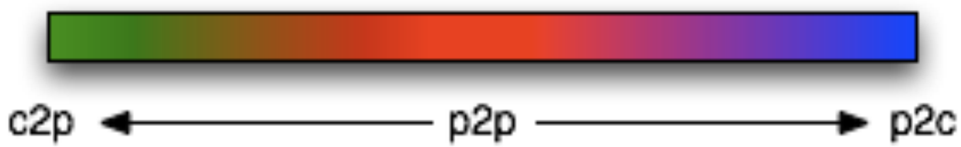
Requirements (2/2): Delivery cost for a certain path

- Delivery cost computation
  - End-to-end path (AS path)
    - A method for provisioning this information to peers
      - traceroute by peers
      - Provisioned by servers in ASes
        - ***written in the draft***
  - Function
    - i.e., *f(P)* = (transit charge on residential ASes)
      - *f*: function, *P*: AS path

AS relationships inference

- Heuristics: Common approach
  - **Degree** (i.e., # of neighbors)
    - High degree ⇔ large
      - tends to be provider
    - Low degree ⇔ small
      - tends to be customer

Diff. in degree

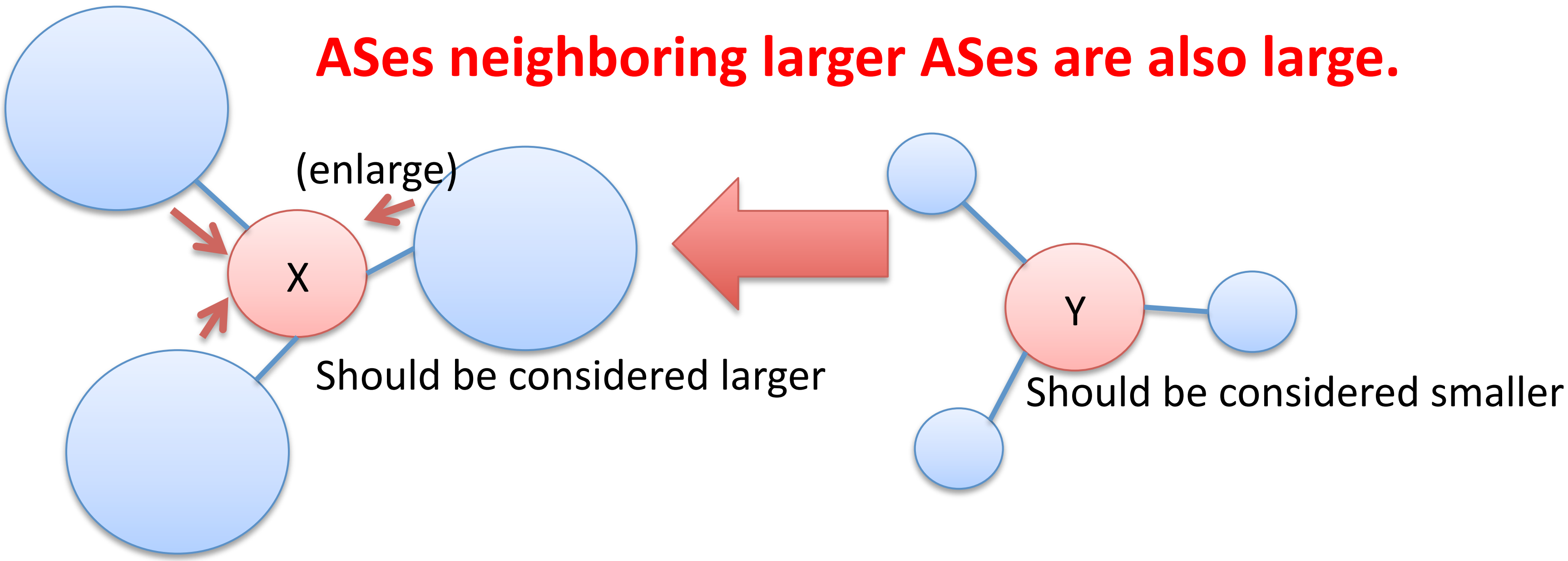


p2c: provider to customer
c2p: customer to provider
p2p: peer to peer

AS relationships inference (cont'd): Improving degree-based approach

Take into account the size of *n*-hop neighbor ASes

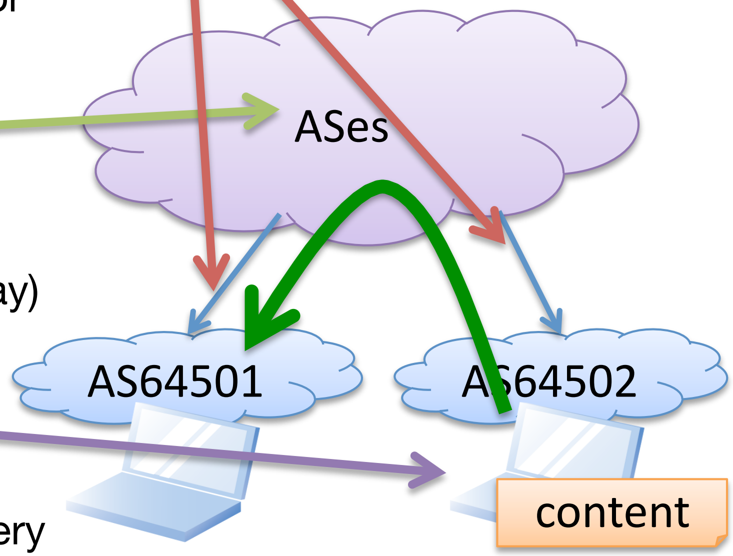
→ **"Magnitude"**



c.f. H. Asai et al., "Estimating AS Relationships for Application-Layer Traffic Optimization," 2010

Example of cost function (e.g., in CDN)

- Metrics for peer selection
  - (1) Inter-AS transit cost
    - Estimated AS relationships
      - for reducing transit traffic and charge for ASes accommodating peers
  - (2) Network cost
    - AS hop count (or distance)
      - for AS-level localization (well-known way)
  - (3) Quality
    - #simultaneous uploads
      - for avoiding deteriorating content delivery throughput (very naïve parameter…)



Simulation result: High-cost transit traffic reduction

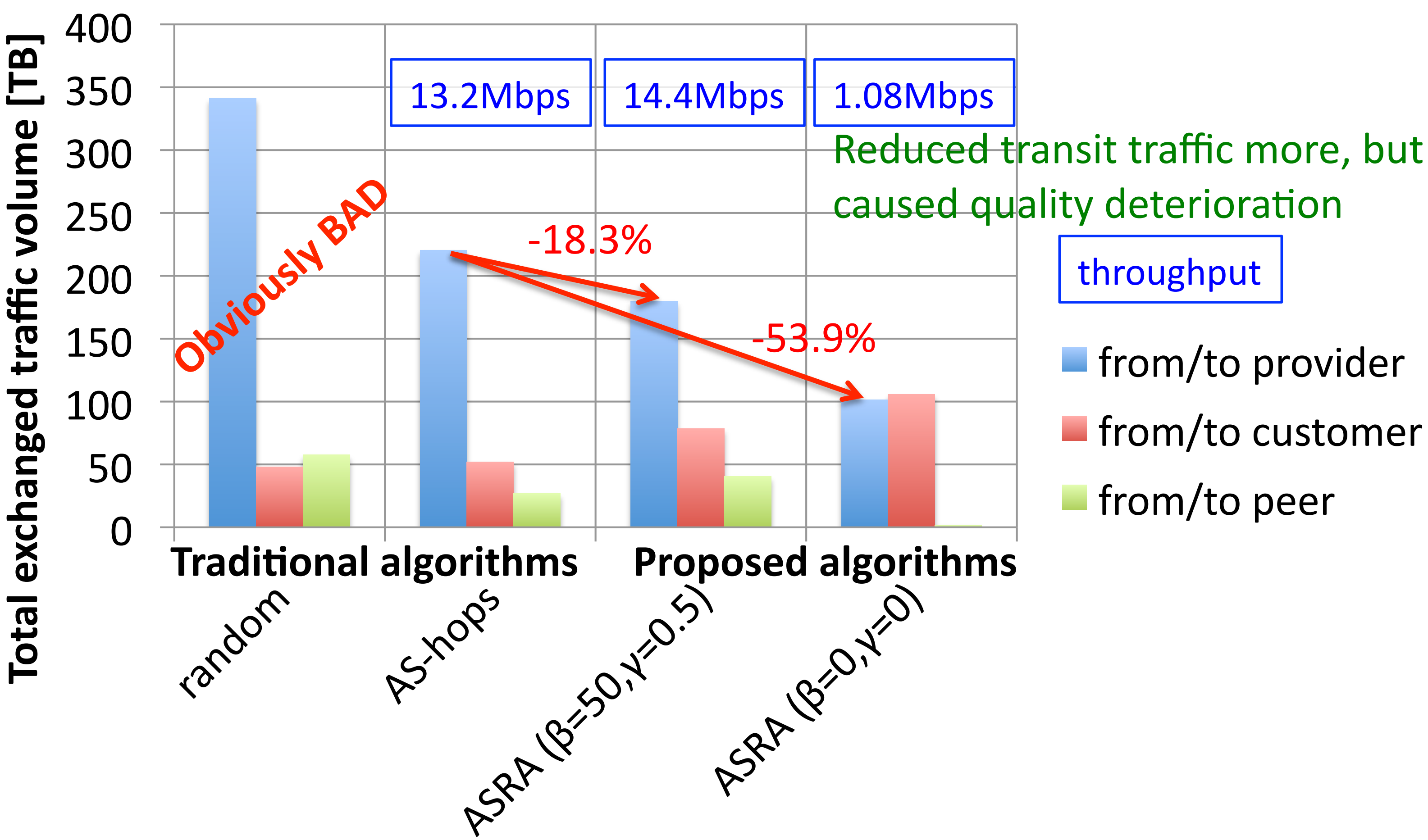


Fig. Breakdown of inter-domain traffic on ASes accommodating CDN peers

Simulation result: Transit charge reduction

N.B., provider ASes charges customer ASes for transit traffic based on the exchanged traffic volume.

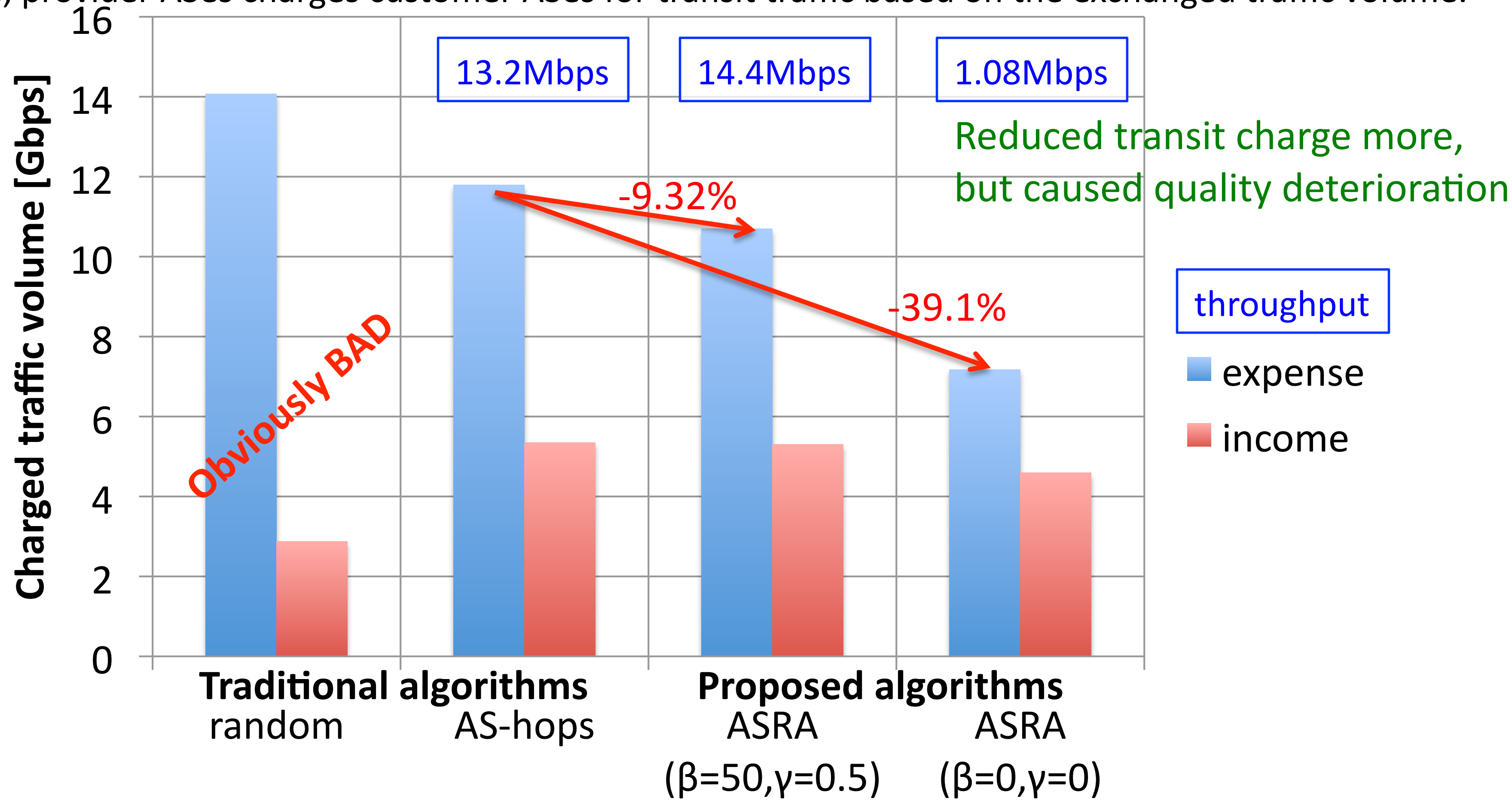


Fig. Charged transit traffic volume with the 95th percentile charging policy

Conclusion

- Approach for reducing transit traffic
  - Pros.
    - Not require ISPs' information (available by end-to-end)
  - Cons.
    - Efficiency depends on inference methods.
- TODO
  - Field experiment to evaluate the system
  - Discussion on deployment possibility and next step
    - Can ISPs provide AS paths?
      - Yes → How to deploy
      - No → traceroute-based approach or other alternatives?

Thank you for your attention
Questions or Comments?

BACKUP SLIDES: SIMULATION SETUP

Improved AS relationships inference

(1) Define a weighted AS adjacency matrix

$$^{n}A := \left({}^{n}a_{v_i v_j}\right)$$

(i) $n = 0$

$$^{n}a_{v_i v_j} = \begin{cases} 1 & : \text{if AS } v_i \text{ and AS } v_j \text{ are adjacent} \\ 0 & : \text{otherwise} \end{cases}$$

(ii) $n \geq 1, n \in \mathbb{Z}$

$$^{n}a_{v_i v_j} = \begin{cases} {}^{(n-1)}\rho_{v_j} & : \text{if AS } v_i \text{ and AS } v_j \text{ are adjacent} \\ 0 & : \text{otherwise} \end{cases}$$

(2) Convert the weighted AS adjacency matrix to a traffic transition matrix

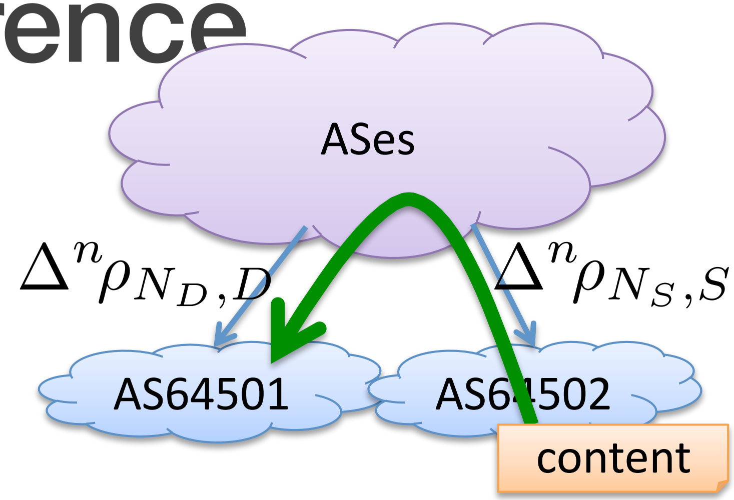
$$^{n}T := \left(\frac{{}^{n}a_{v_i v_j}}{\sum_{v_k} {}^{n}a_{v_i v_k}} \right)$$

(3) Calculate the left eigenvector of the traffic transition matrix corresponding to the maximum eigenvalue

$$^{n}\boldsymbol{\rho} = \left[{}^{n}\rho_{v_1}, \cdots, {}^{n}\rho_{v_m}\right]^t$$: the left eigenvector

$$(\|{}^{n}\boldsymbol{\rho}\| = 1)$$

Peer selection preference



Peer Selection Preference

$$p(P) := \alpha c_i(P) + \beta c_t(P) + \gamma q(P)$$

α, β, γ: parameters

(1) Internal CDN cost; i.e., AS relationships

$$c_i(P) := \begin{cases} \epsilon \Delta^n \rho_{N_D,D} + (1-\epsilon)\Delta^n \rho_{N_S,S} + 2|\max{}^n \boldsymbol{\rho}| & (S \neq D) \\ 0 & (S = D) \end{cases}$$

ε: weighting factor, ε = 0.5

given (by traceroute etc.)

(2) Total network cost

$$c_t(P) := H_{S,D}$$

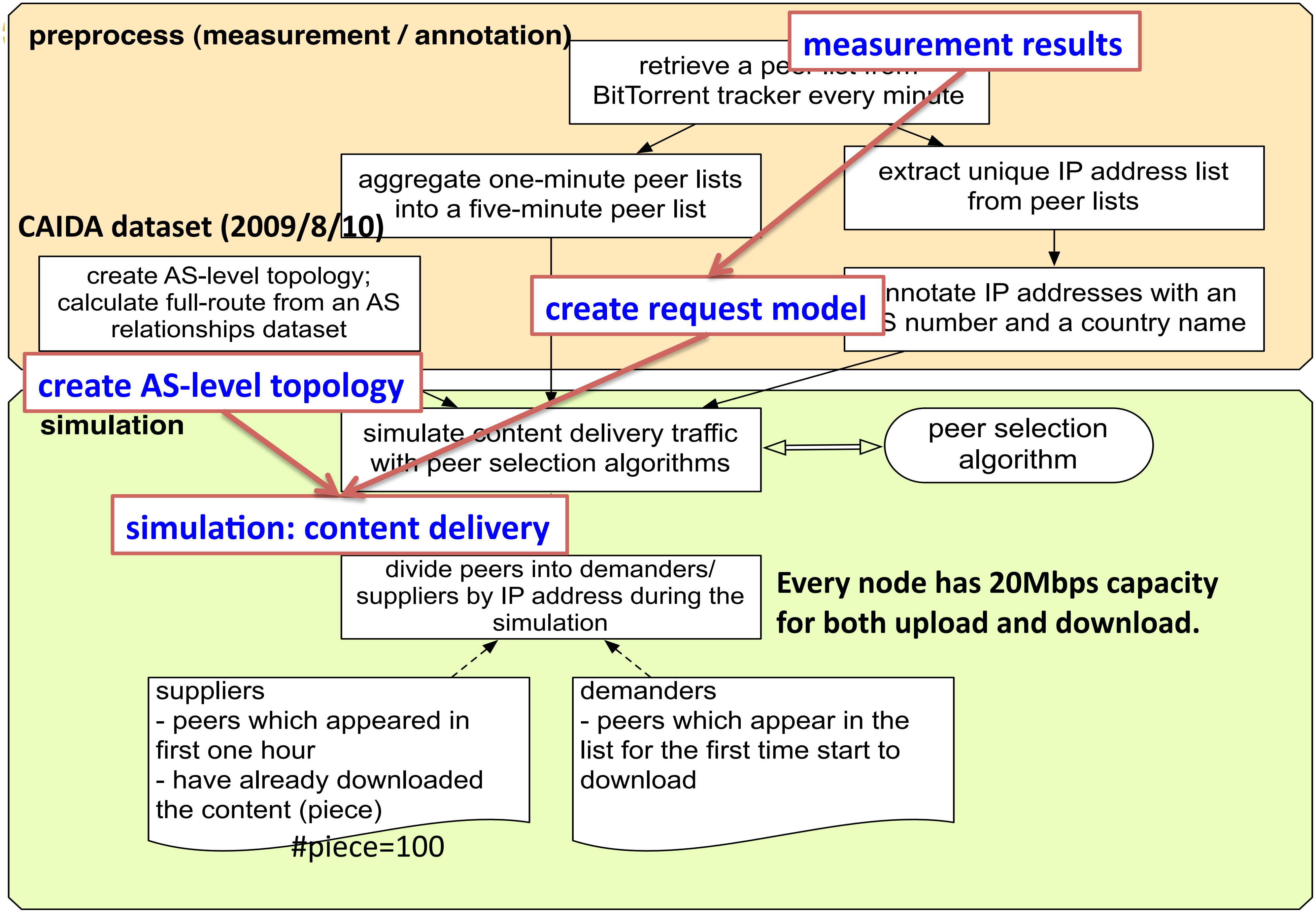
(3) Quality

$$q(P) := b_s^2$$

P: path from peer *s* in AS *S* to peer *d* in AS *D*
N_S: neighbor (next hop) AS of AS *S*
N_D: neighbor (previous hop) AS of AS *D*
$H_{S,D}$: AS hop count from AS *S* to AS *D*
b_s: the number of simultaneous uploads on peer *s*

Simulation setup

- Evaluation model
  - Request pattern
    - based on measured peer distribution in *BitTorrent*
  - Internet topology
    - CAIDA AS Relationships Dataset (10/08/2009)
- Algorithms
  - random: select one uniformly at random
  - AS-hops: minimize AS hop count
  - proposed algorithm ($\alpha = 1$, $\varepsilon = 0.5$): minimize preference
    - (β, γ) = {(0, 0), (0, 0,5), (50, 0), (50, 0.5)}


preprocess (measurement / annotation)
measurement results
retrieve a peer list from BitTorrent tracker every minute
aggregate one-minute peer lists into a five-minute peer list
extract unique IP address list from peer lists
CAIDA dataset (2009/8/10)
create AS-level topology; calculate full-route from an AS relationships dataset
create request model
annotate IP addresses with an AS number and a country name
create AS-level topology
simulation
simulate content delivery traffic with peer selection algorithms
peer selection algorithm
simulation: content delivery
divide peers into demanders/ suppliers by IP address during the simulation
Every node has 20Mbps capacity for both upload and download.
suppliers
- peers which appeared in first one hour
- have already downloaded the content (piece)
#piece=100
demanders
- peers which appear in the list for the first time start to download