

# Quantifying Disincentives in P2P Networks



Michal Feldman

Kevin Lai

John Chuang

Ion Stoica

SIMS and Computer Science  
UC Berkeley



# Outline

- Introduction
- P2P Performance Model
- Potential vs. Actual Disincentives
- Eliminating Disincentives via Prioritization
- Conclusions



# Introduction

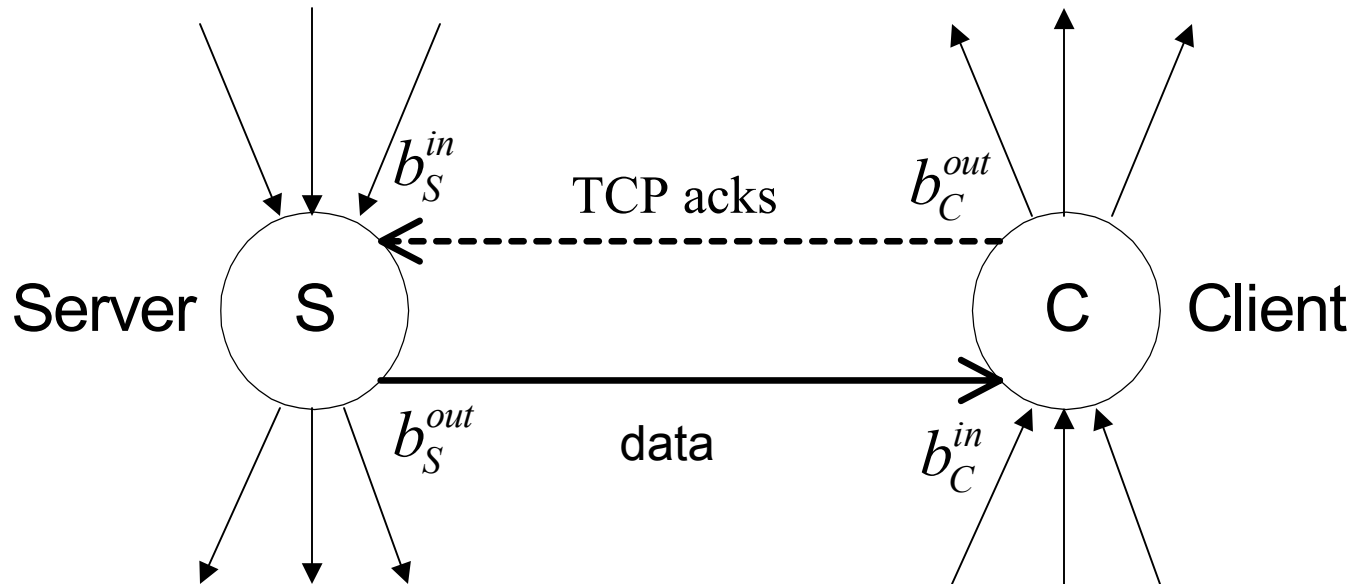
- Cooperation (sharing) improves the performance of peer-to-peer file-sharing systems
- Studies show low level of sharing
- Need better understanding of performance-related disincentives
- How performance varies as a function of:
  - The sharing level
  - Whether a user shares or not



# Outline

- Introduction
- **P2P Performance Model**
- Potential vs. Actual Disincentive
- Eliminating Disincentives via Prioritization
- Conclusions

# Performance Model



Performance metric: average download latency



# Model Assumptions

- Download time is dominated by transfer time
- Bottleneck is always at the edge of the n/w
- Traffic follows TCP protocol
- Searches experience no delay; require negligible BW
- Files have the same size, popularity and spatial distribution
- Generated load is evenly distributed
- Number of uploads per node is proportional to its outgoing bandwidth



# Outline

- Introduction
- P2P Performance Model
- **Potential vs. Actual Disincentive**
- Eliminating Disincentives via  
Prioritization
- Conclusions



# Disincentive to Share

- Useful to distinguish between:
  - **Potential** disincentive
  - **Actual** disincentive
- Potential disincentive:
  - client's uploads may delay his downloads
- Actual disincentive:
  - Existence depends on the location of the transfer bottleneck





# Potential Disincentive to Share

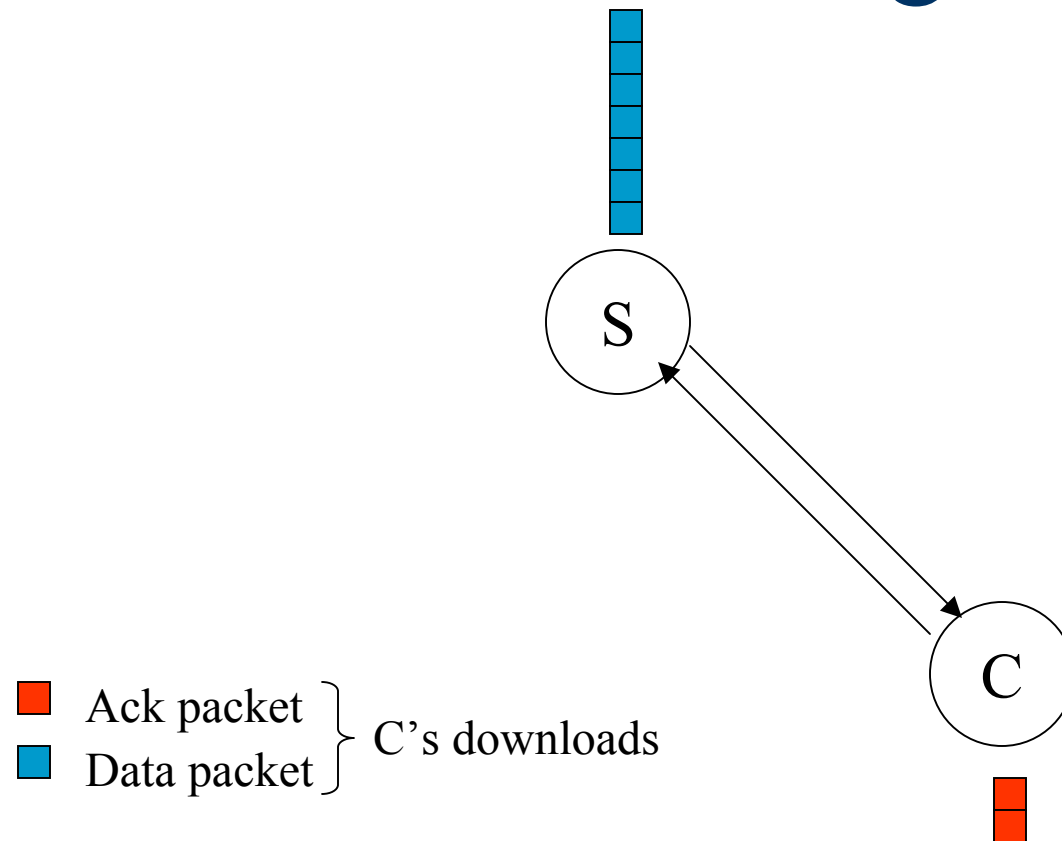
- Incoming link utilization ( $\rho$ ) drops if user uploads:

	in bw	out bw	link utilization
ADSL	1.5Mb/s	128Kb/s	0.2
Ethernet	10Mb/s	10Mb/s	0.8

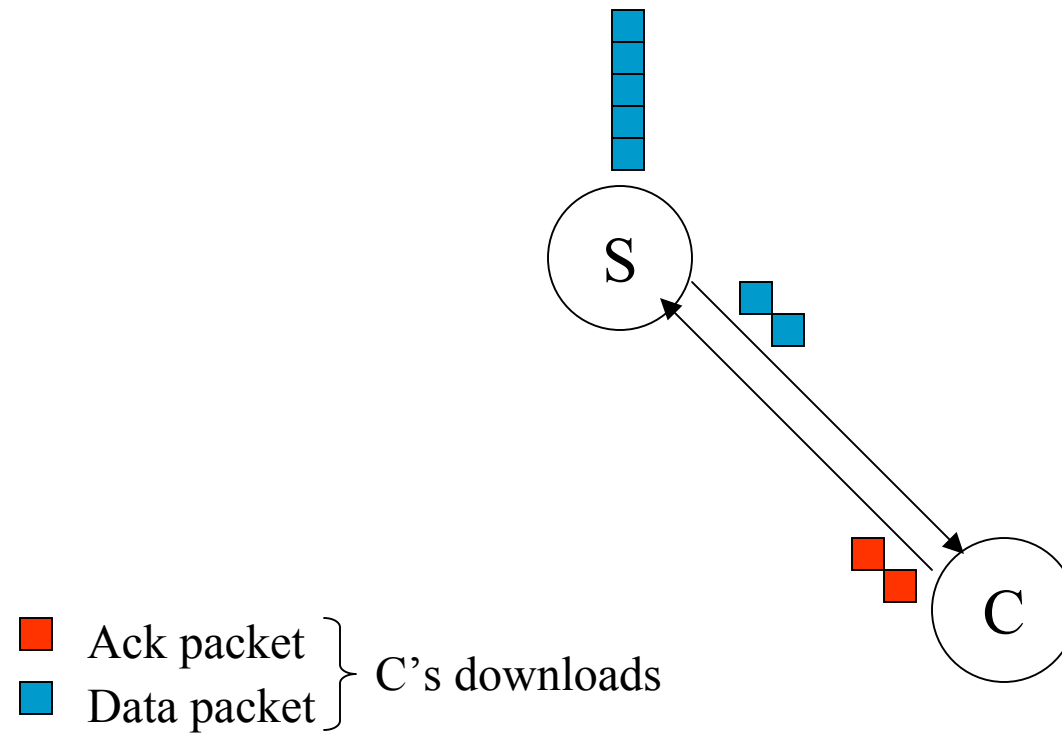
- Due to contention between upload data and TCP acks
- Magnitude of effect depends on node characteristics

Conclusion: **high potential disincentive** to share

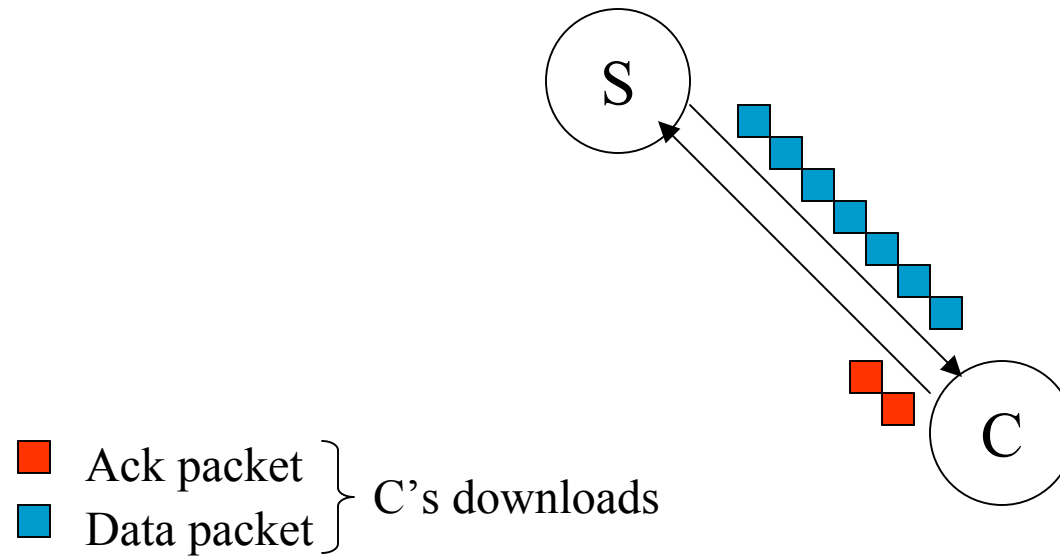
# Case 1: Non-Sharing Client



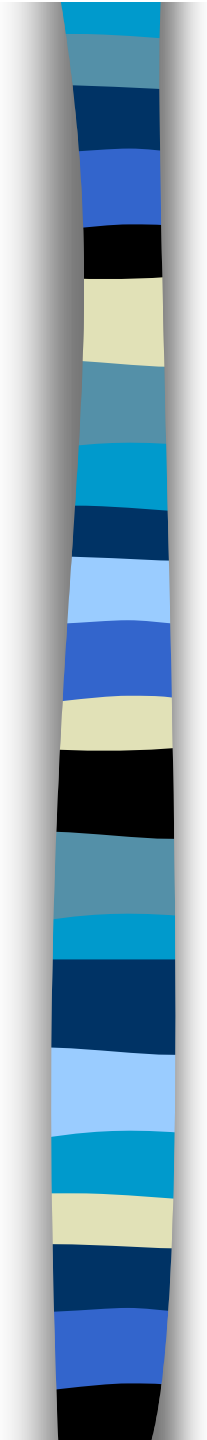
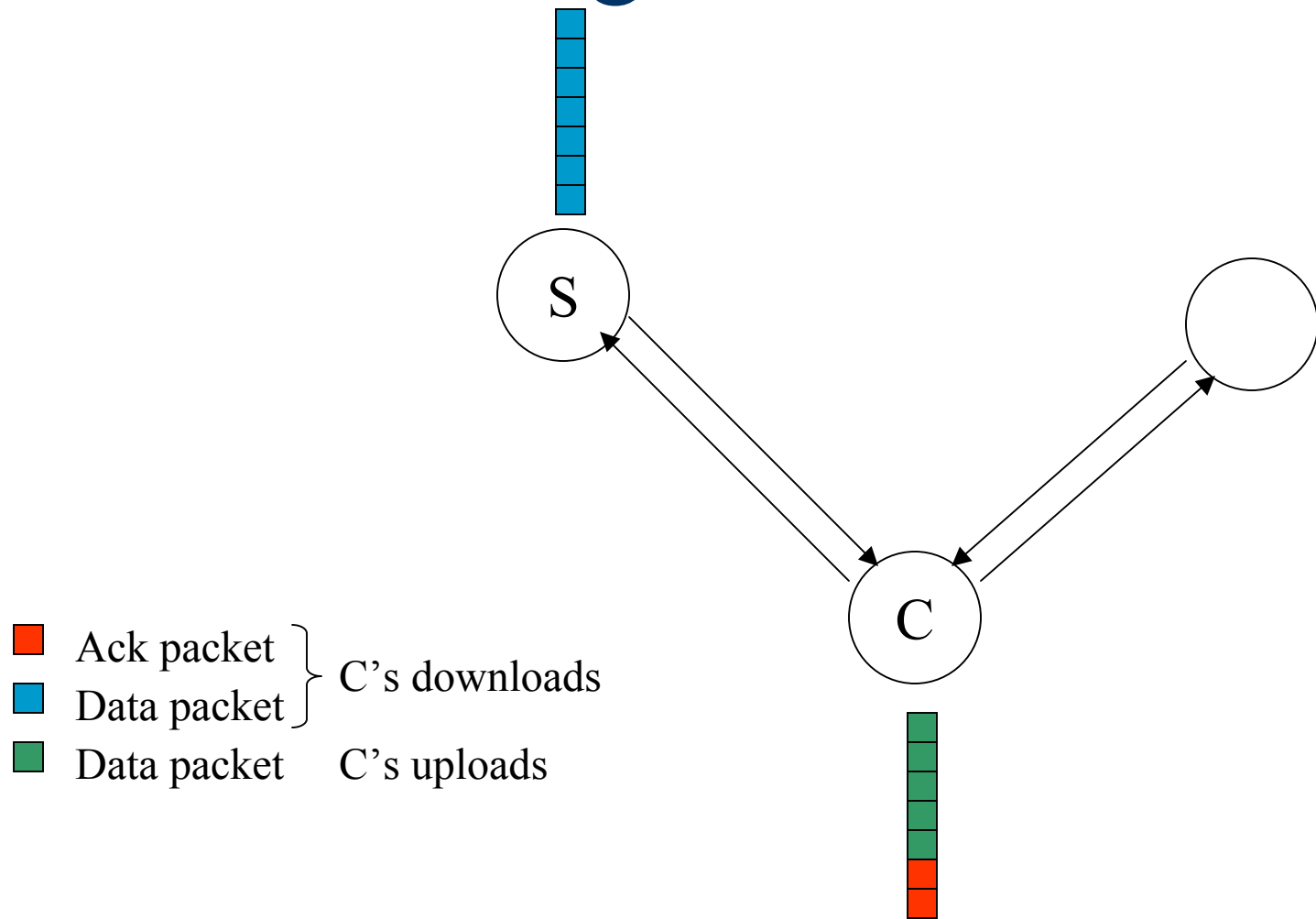
# Case 1: Non-Sharing Client



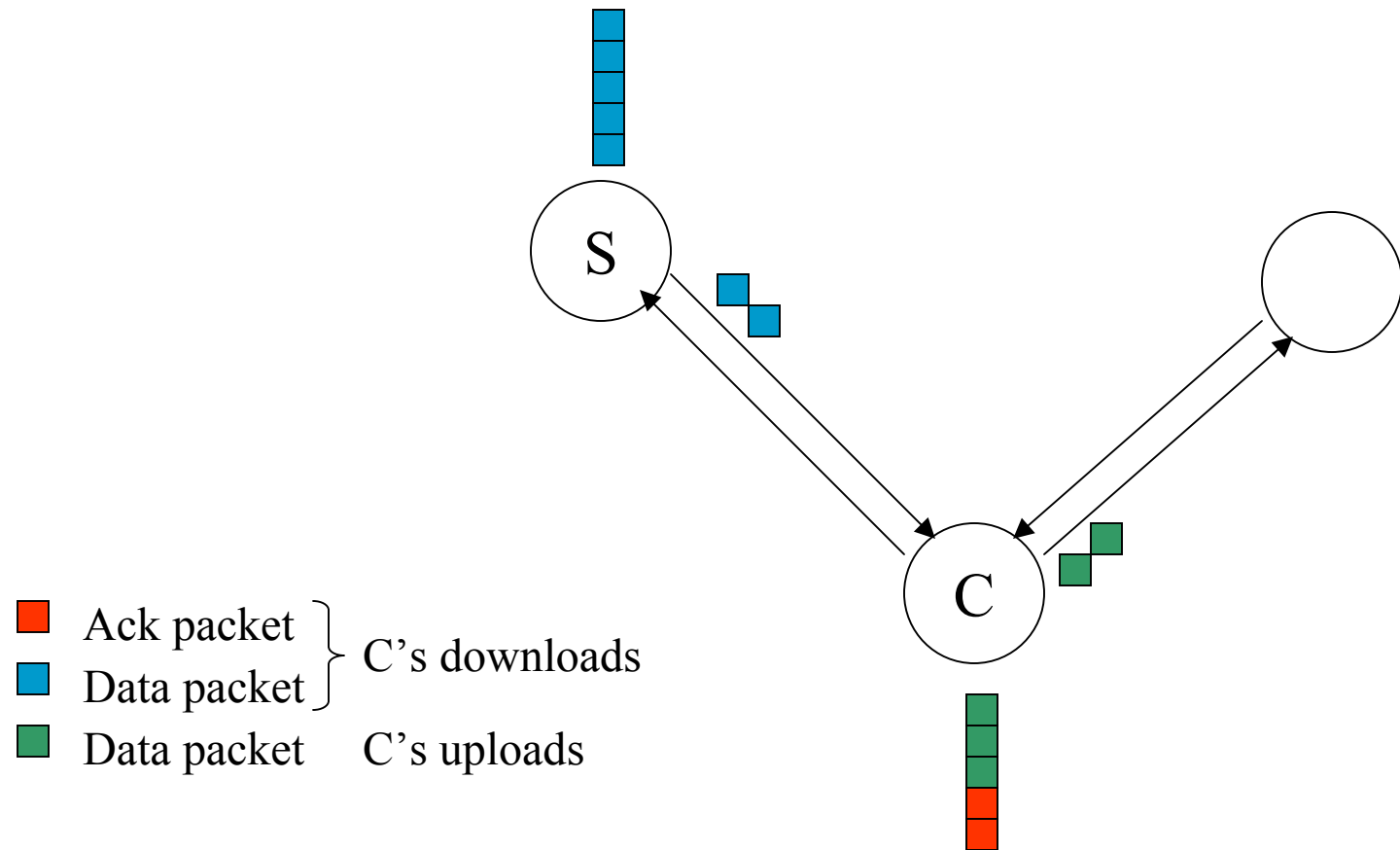
# Case 1: Non-Sharing Client



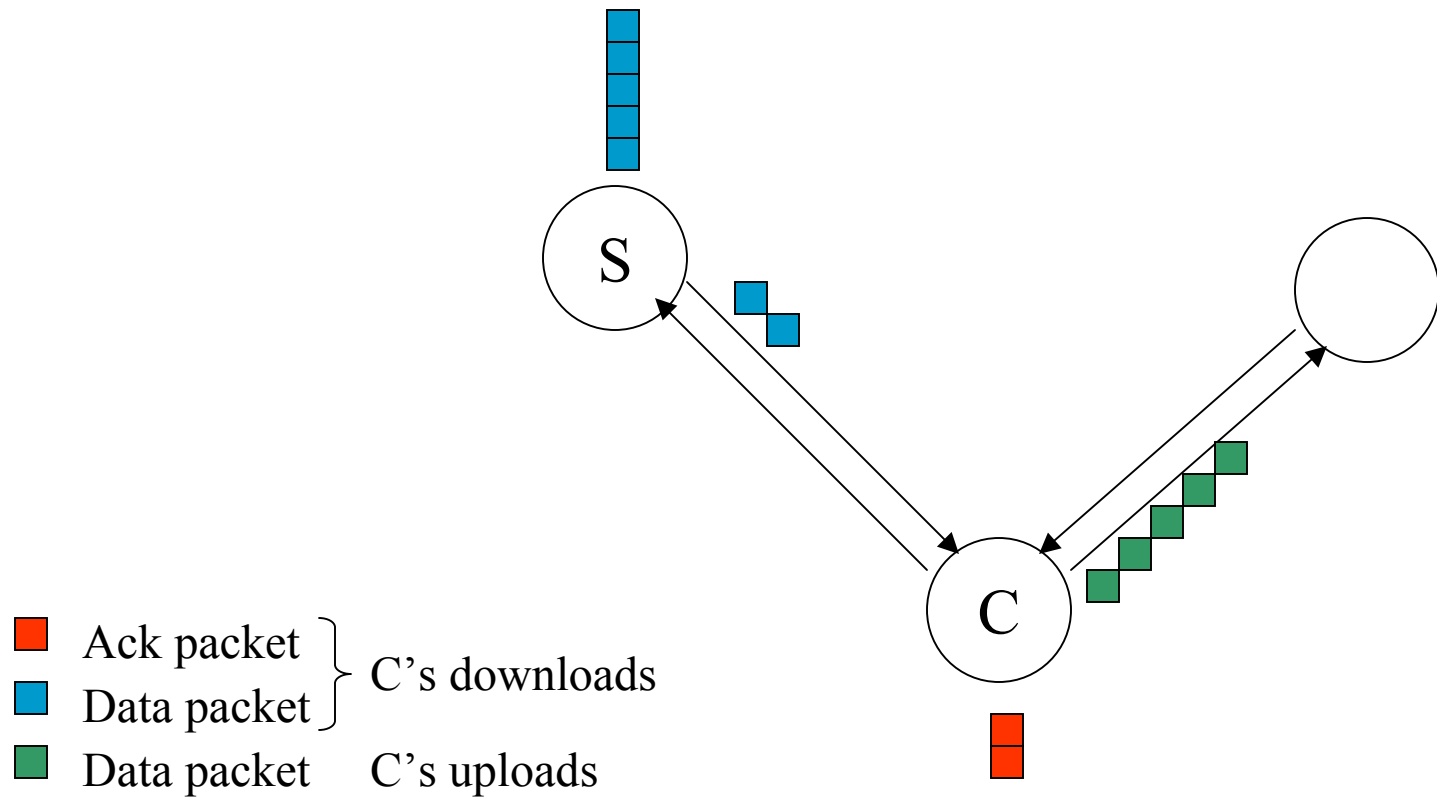
# Case 2: Sharing Client



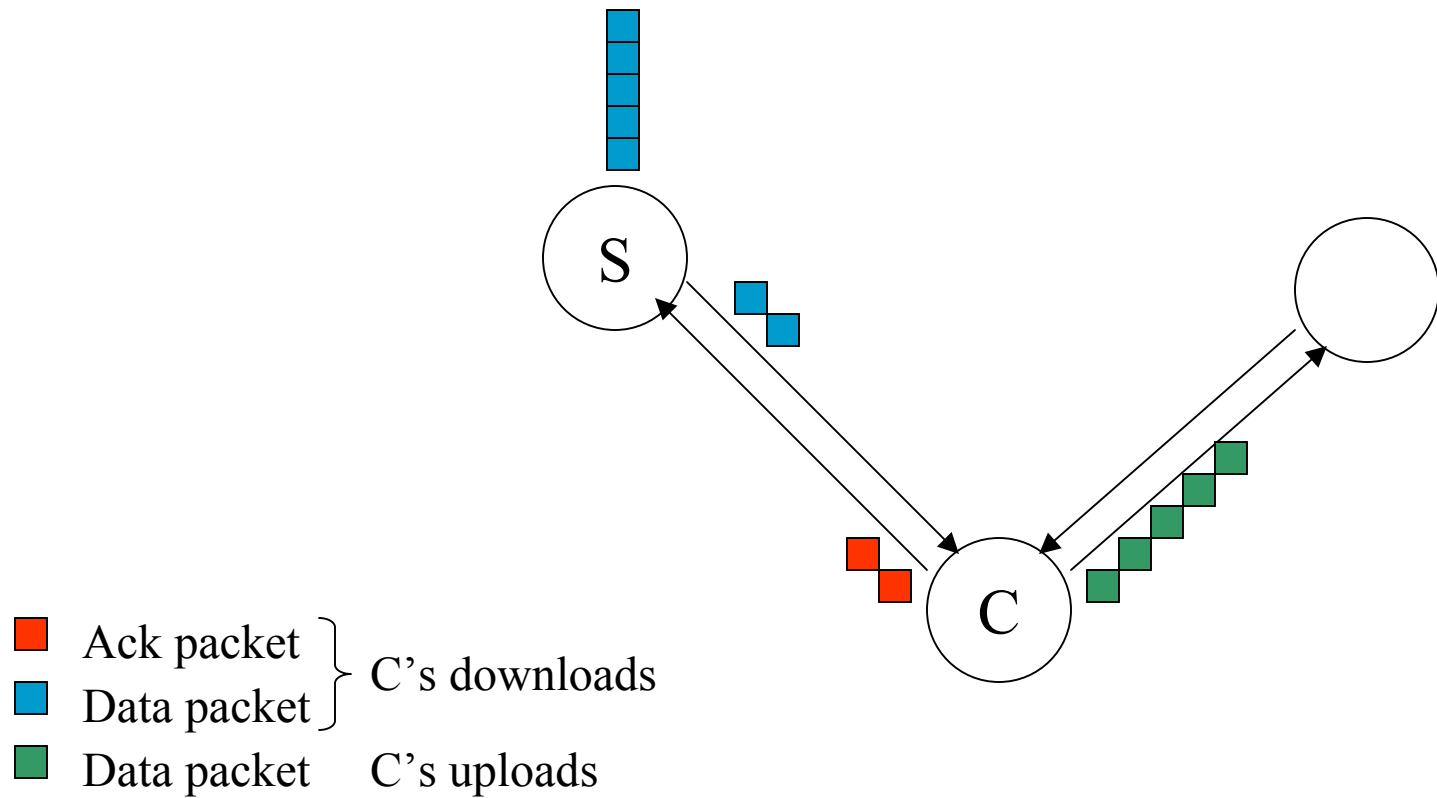
# Case 2: Sharing Client



# Case 2: Sharing Client

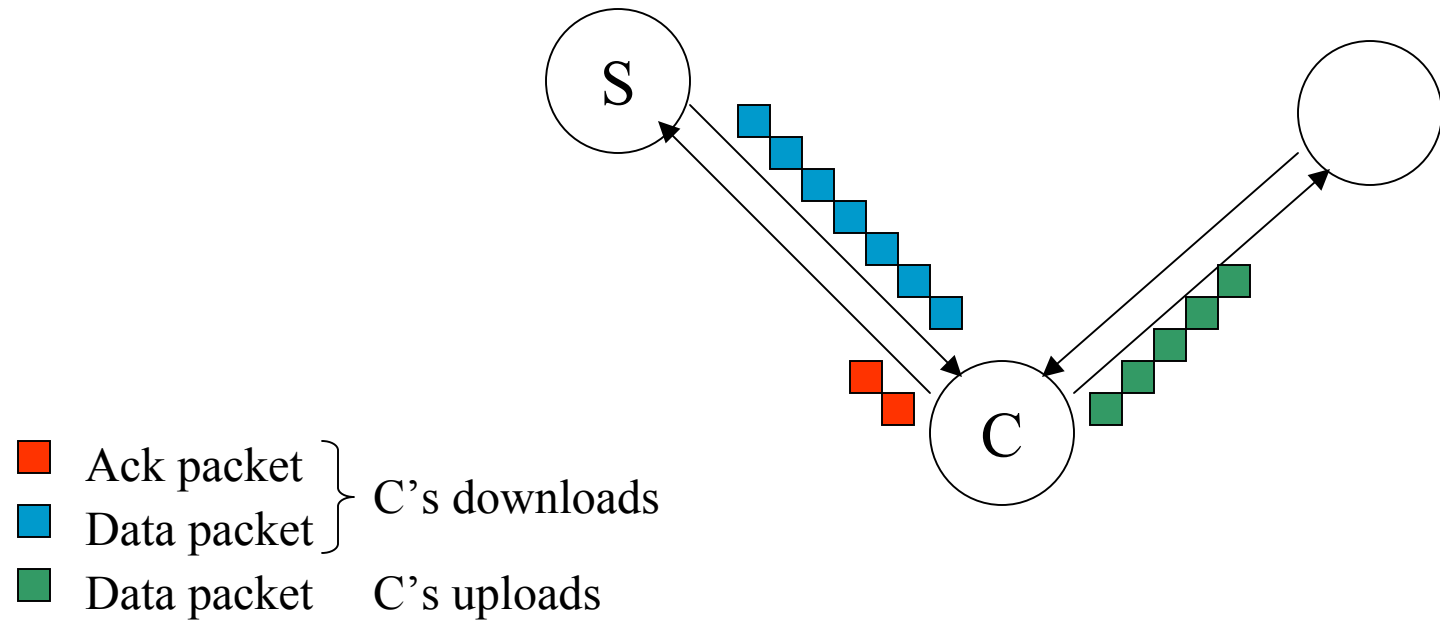


# Case 2: Sharing Client

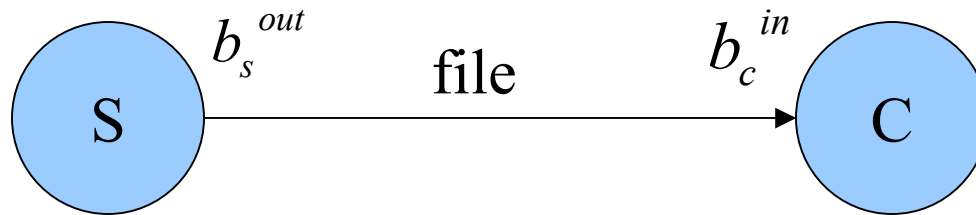




# Case 2: Sharing Client



# Actual Disincentive to Share



$\rho$  : utilization factor

$P_{sharing}$  : sharing level

$L$  : average load

$EB$ : effective bandwidth

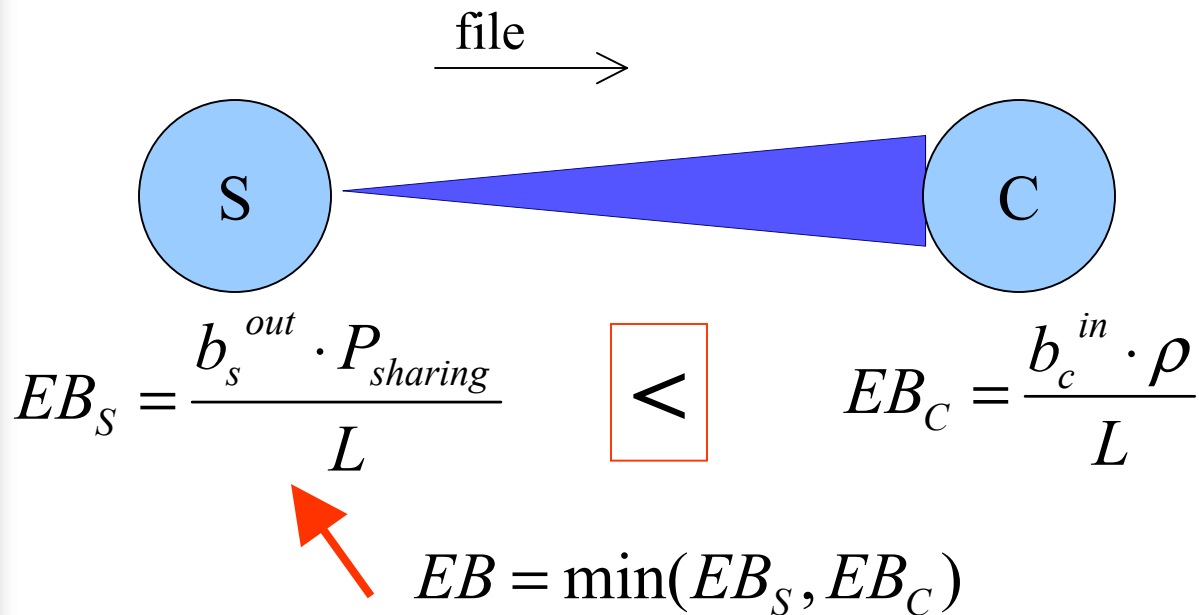
$$EB_S = \frac{b_s^{out} \cdot P_{sharing}}{L}$$

$$EB_C = \frac{b_c^{in} \cdot \rho}{L}$$

$$EB = \min(EB_S, EB_C)$$

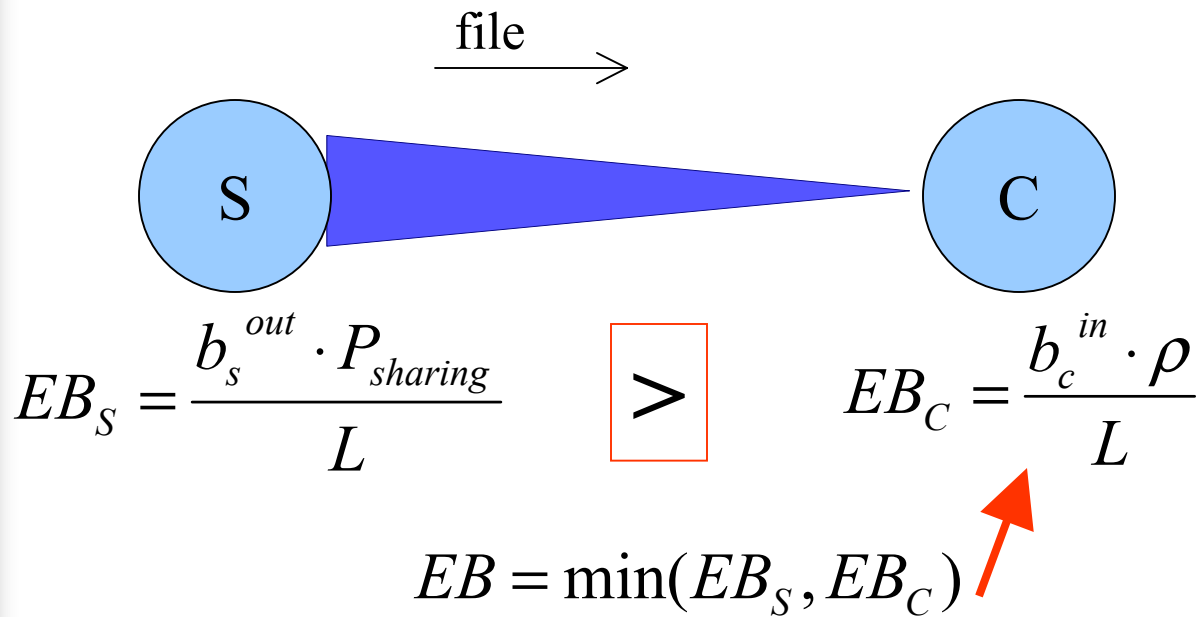
Where is the bottleneck?

# Case 1: Server is the Bottleneck



- Low level of sharing
- Latency independent of utilization factor,  $\rho$
- No actual disincentive for client to share !

## Case 2: Client is the Bottleneck



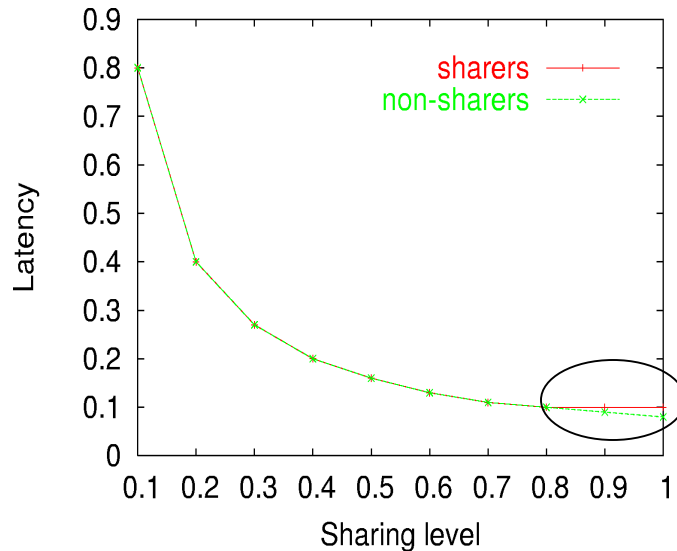
- High level of sharing
- Latency depends on utilization factor,  $\rho$
- Actual disincentive for client to share !



# Actual Disincentive to Share

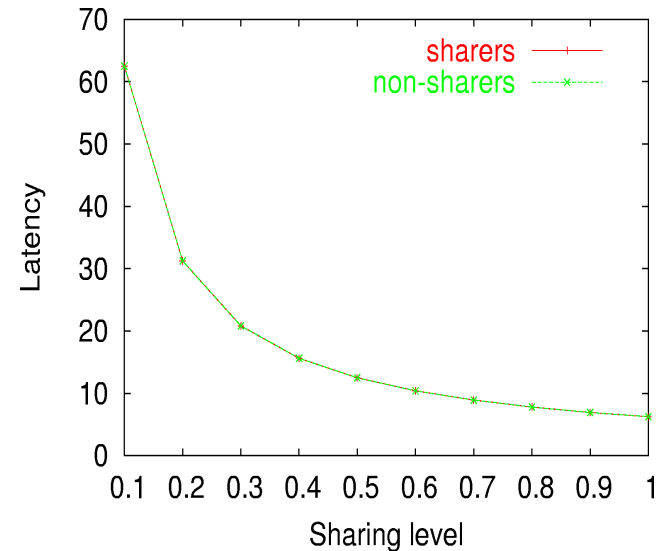
- Existence of **actual** disincentive for client depends on the location of the transfer **bottleneck**:
  - Server side: no actual disincentive
  - Client side: actual disincentive
- Bandwidth bottleneck switches between the server side and the client side depending on the sharing level

# Actual Disincentives in Homogeneous systems



## Ethernet (symmetric)

- Low sharing level: NO disincentive
- High sharing level: insignificant disincentive

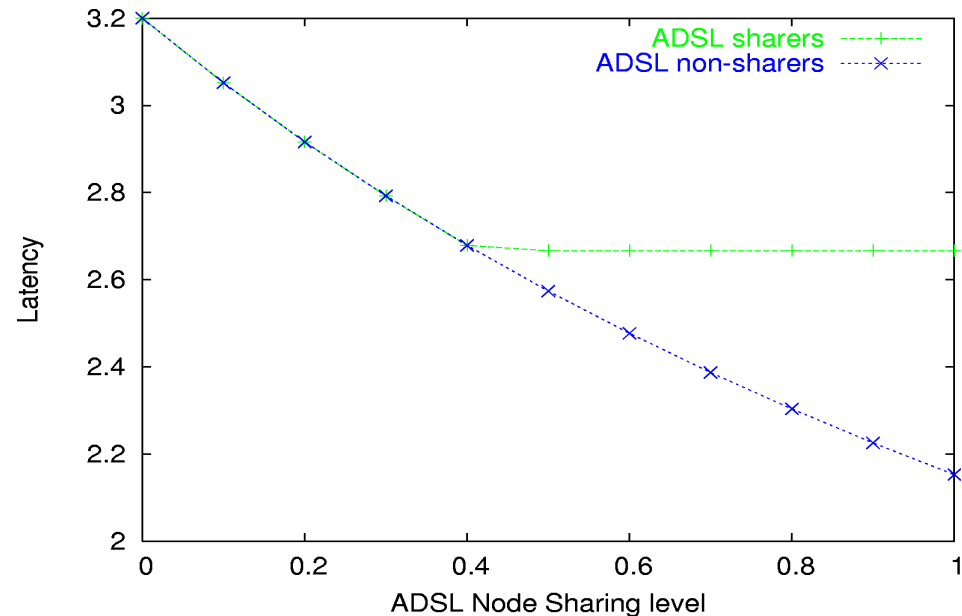


## ADSL (asymmetric)

NO disincentive

# Actual Disincentives in Heterogeneous System

- System composed of 95% ADSL, 5% Ethernet nodes
- Ethernet nodes: no disincentive
- ADSL nodes: disincentives exist only for high level of sharing



Conclusion: **no significant actual disincentive** in homogeneous system or heterogeneous system with low sharing level



# Performance-Related Disincentives

- High **potential** disincentive
- Negligible **actual** disincentive
- Yet, users do not share !
- **Perceived** disincentive may be shaped by **potential** disincentive more than by **actual** disincentive
- **Perceived** disincentive determines whether a user shares or not

How can we change the **potential** disincentive?





# Outline

- Introduction
- P2P Performance Model
- Potential vs. Actual Disincentive
- **Eliminating Disincentives via Prioritization**
- Conclusions



# Prioritizing TCP Acknowledgements

- Source of disincentive: contention between acks and data
- Proposed solution: prioritize acks over data on the outgoing link
- Effect:

- Receiver's incoming:

$$bw_c^{in} \cdot \rho \rightarrow bw_c^{in} \quad (\rho = 0.2 / 0.8) \quad \uparrow$$

- Sender's outgoing:

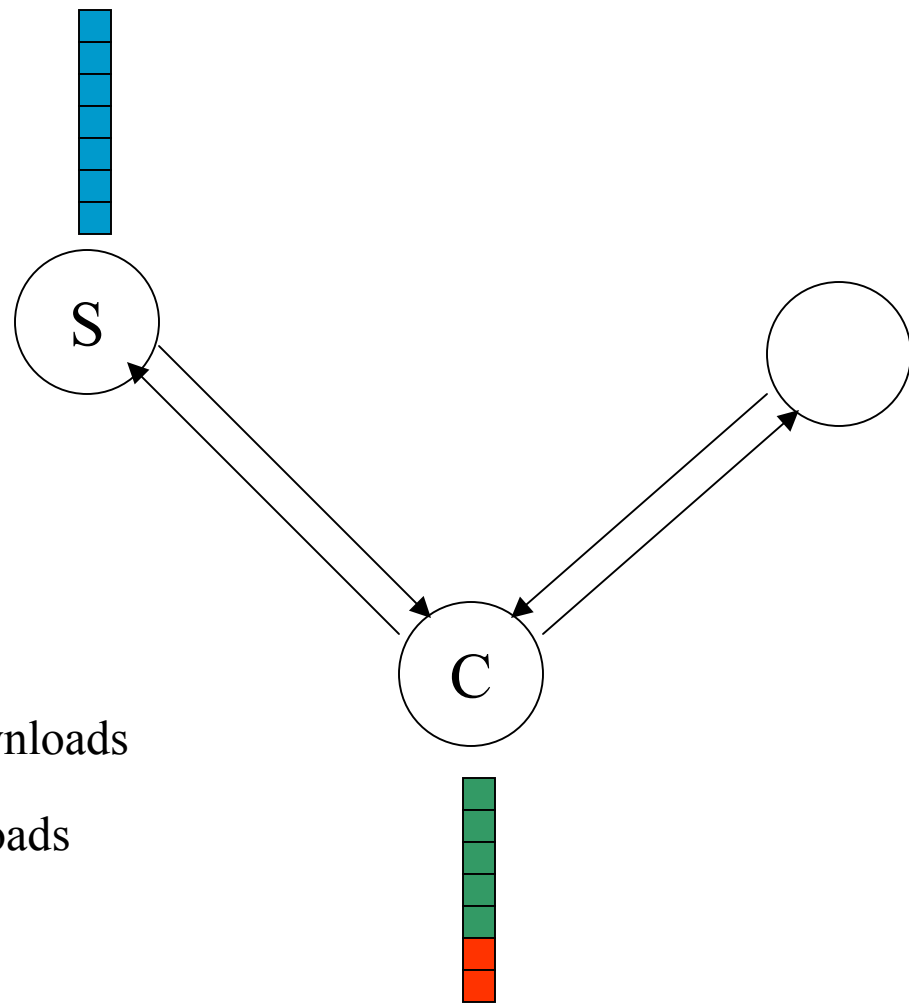
$$bw_s^{out} \rightarrow bw_s^{out} \cdot \beta \quad (\beta = 0.9) \quad \downarrow$$




- Net effect: ?

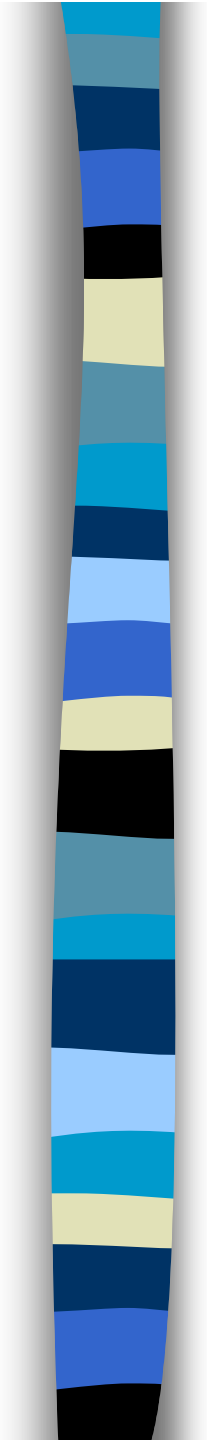
$\rho$  : incoming  
link utilization

$\beta$  : outgoing  
link utilization

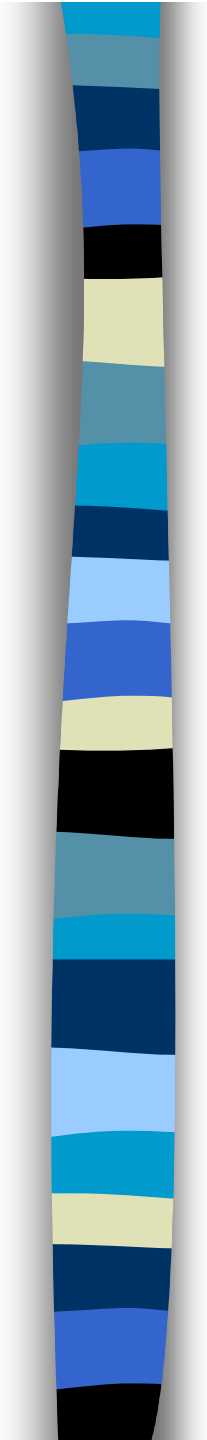
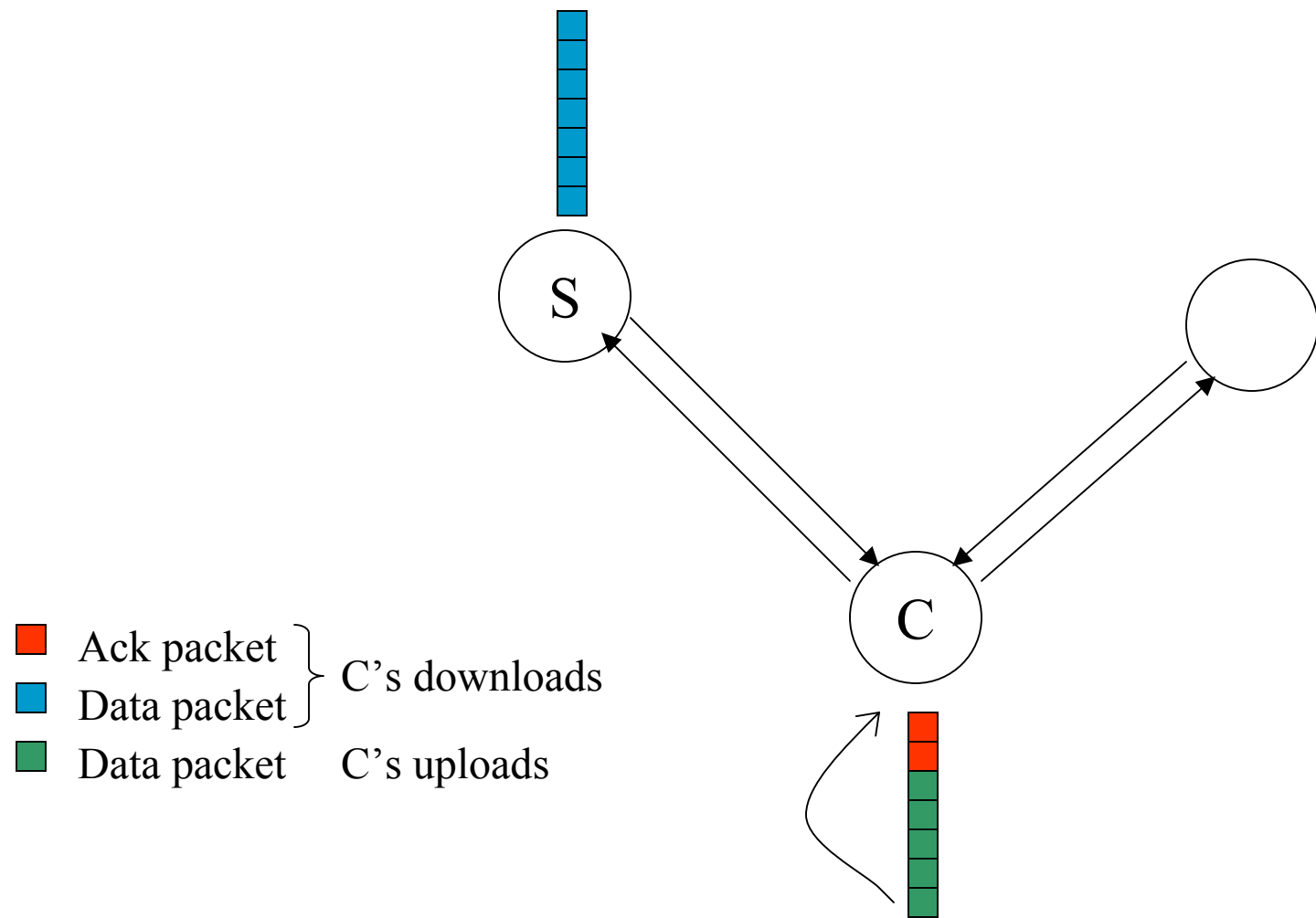
# Prioritization Effect on Client



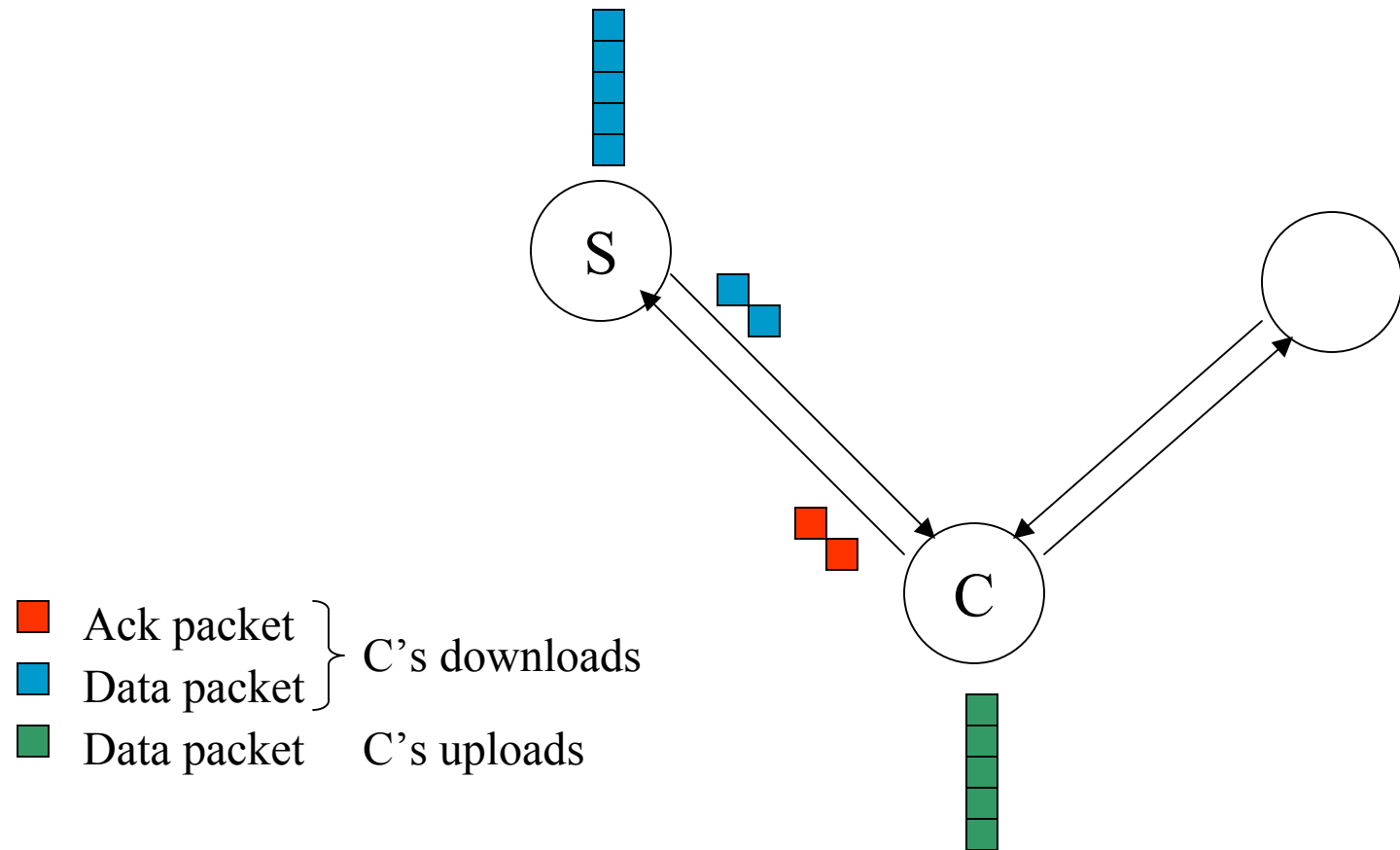
-  Ack packet } C's downloads
-  Data packet } C's downloads
-  Data packet } C's uploads



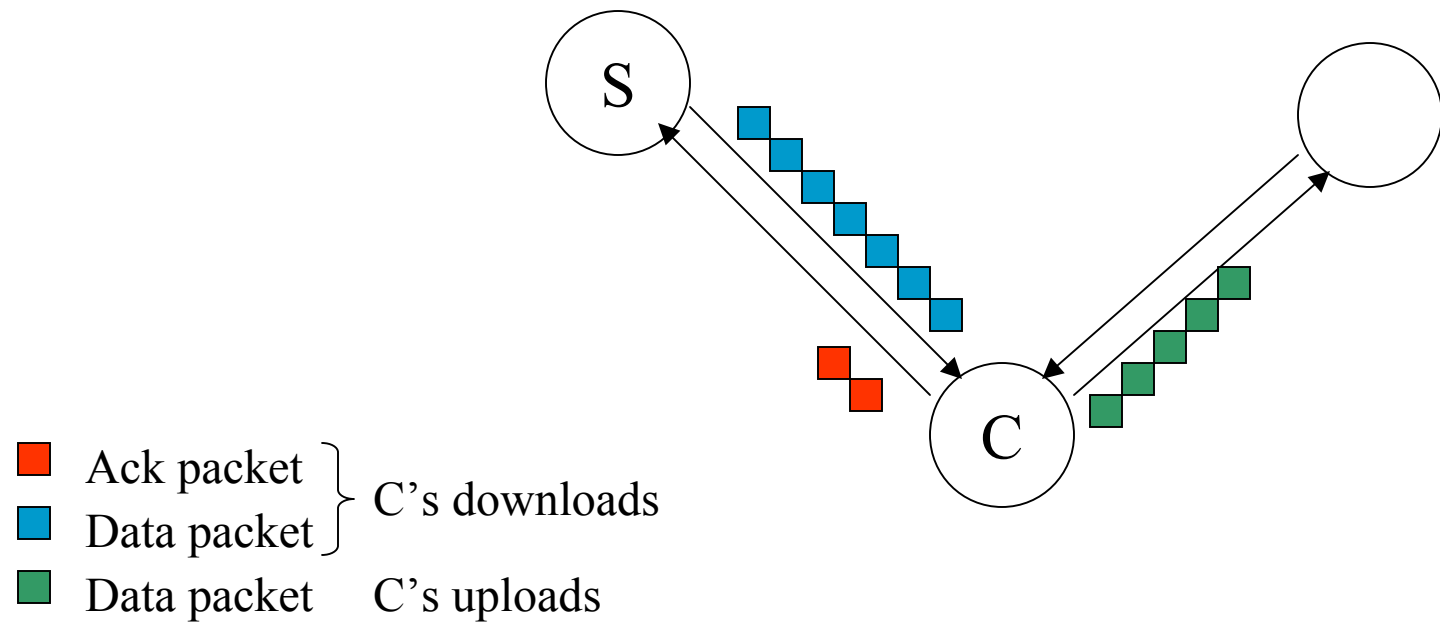
# Prioritization Effect on Client



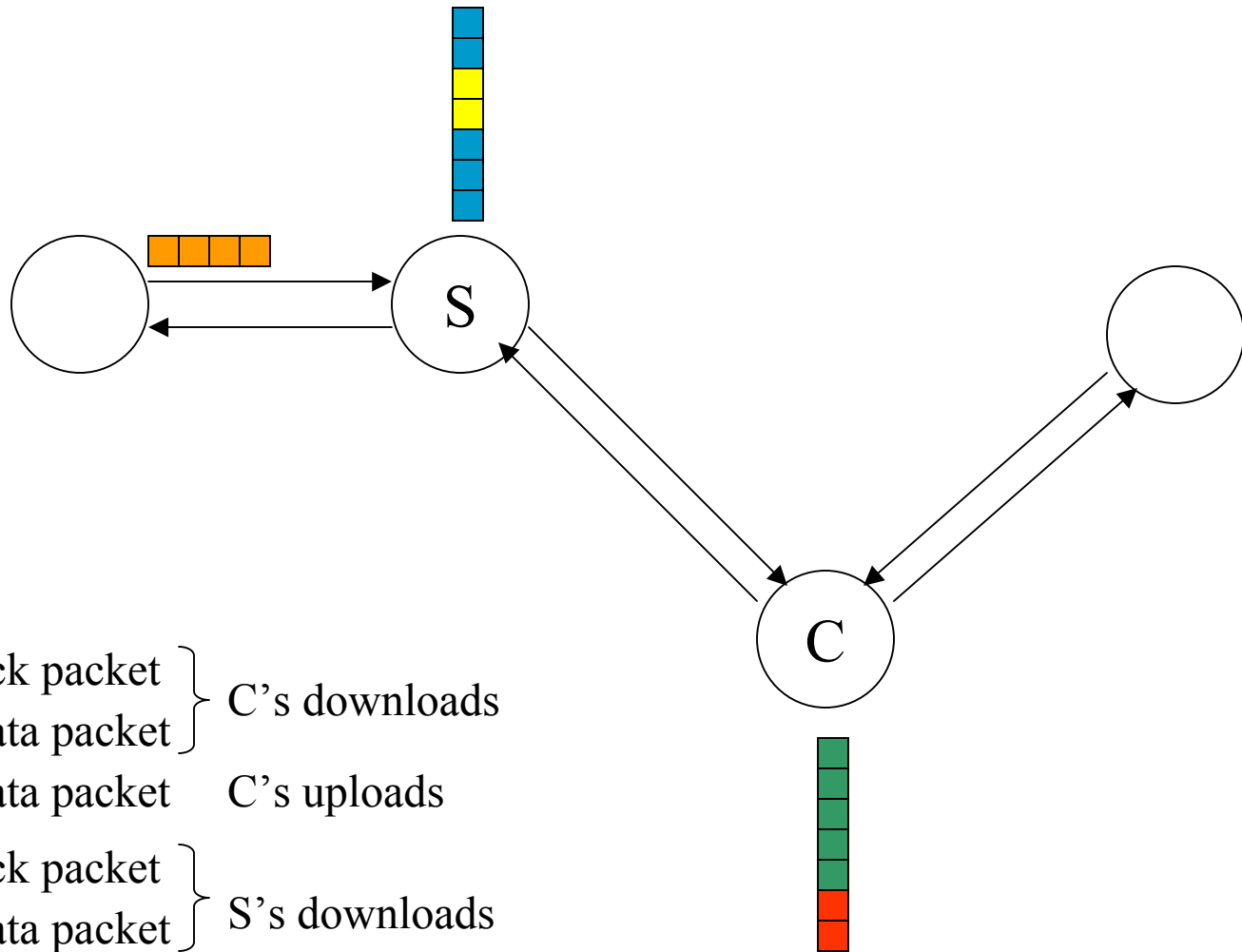
# Prioritization Effect on Client



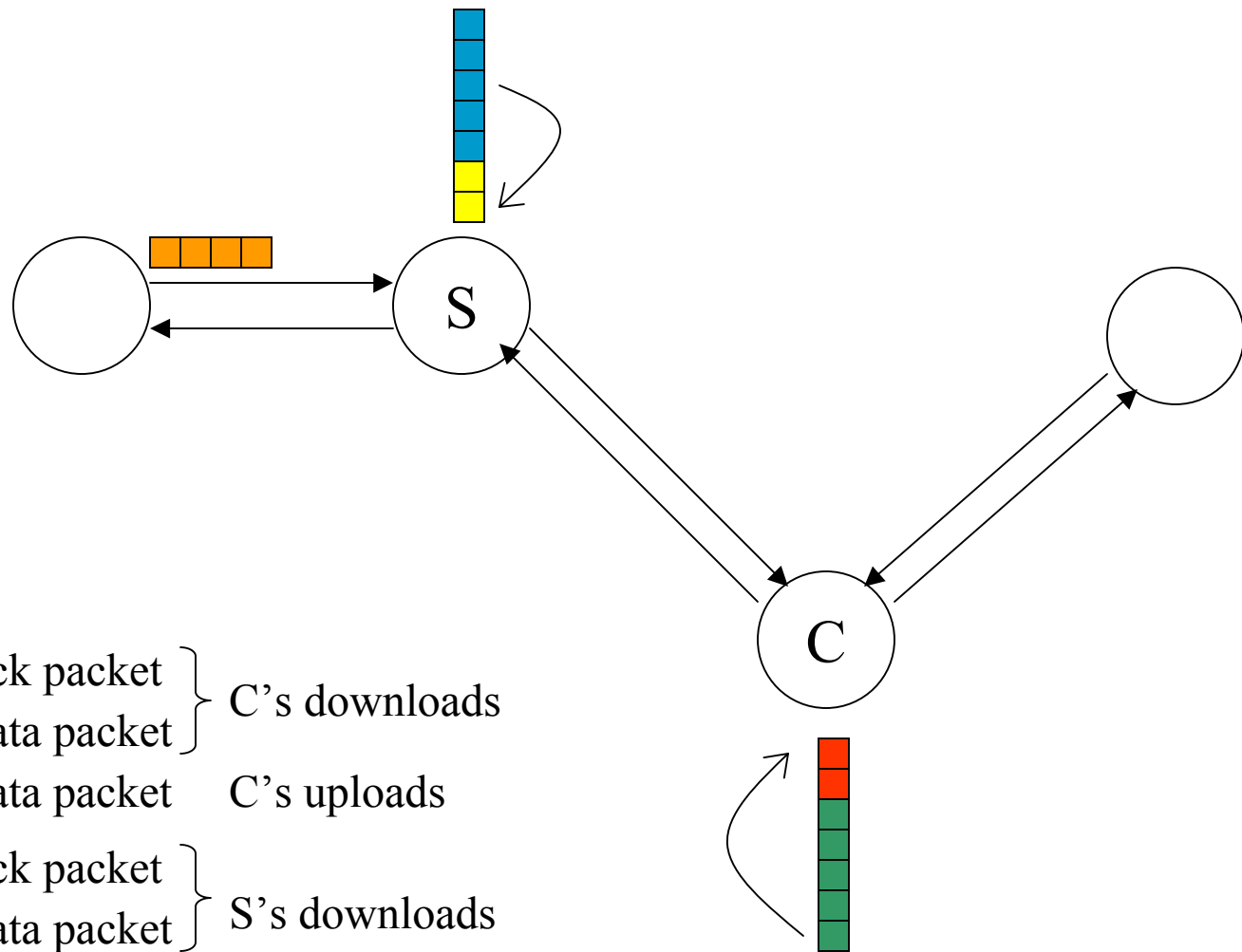
# Prioritization Effect on Client



# Prioritization: No Free Lunch

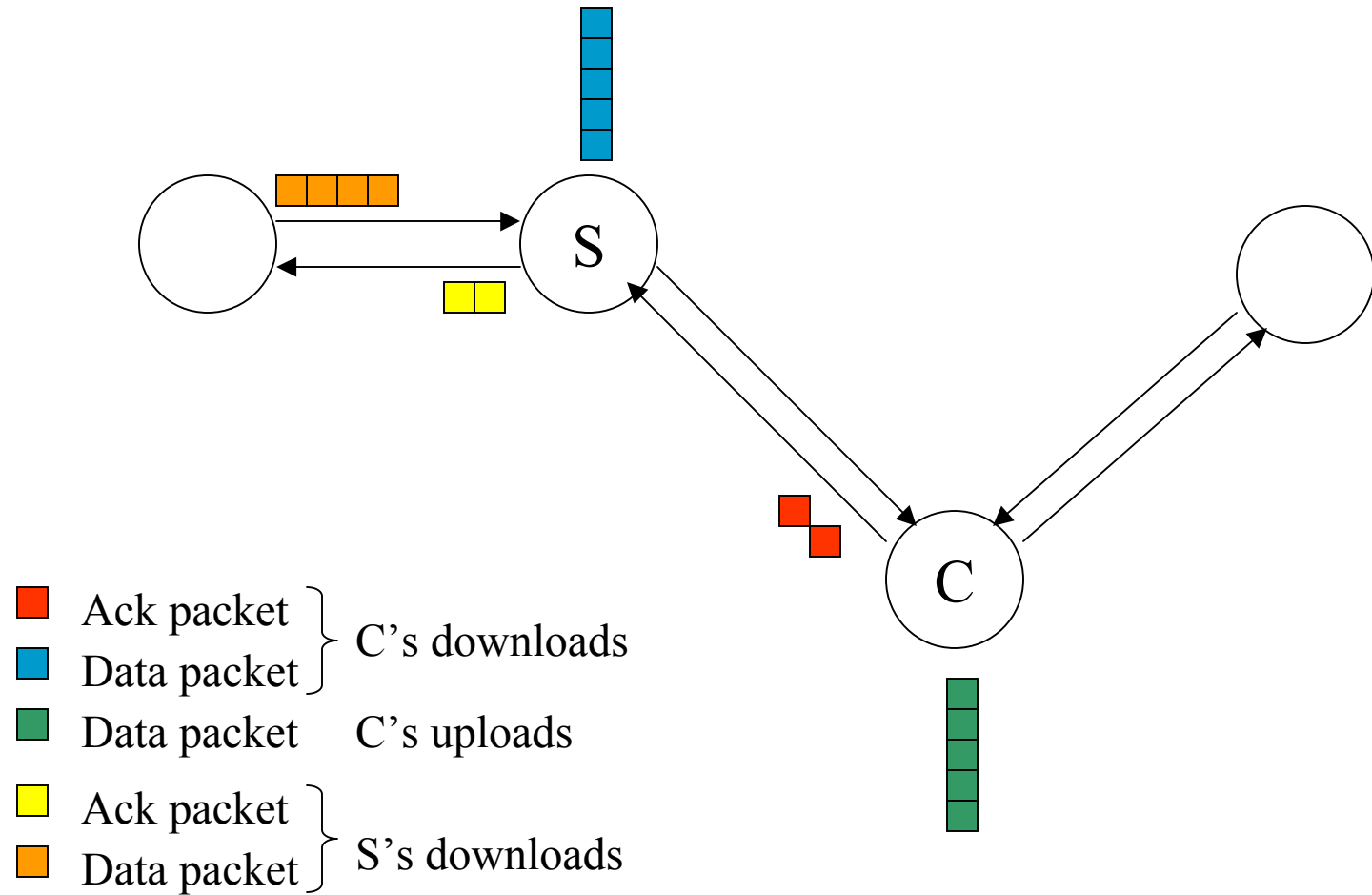


# Prioritization: No Free Lunch

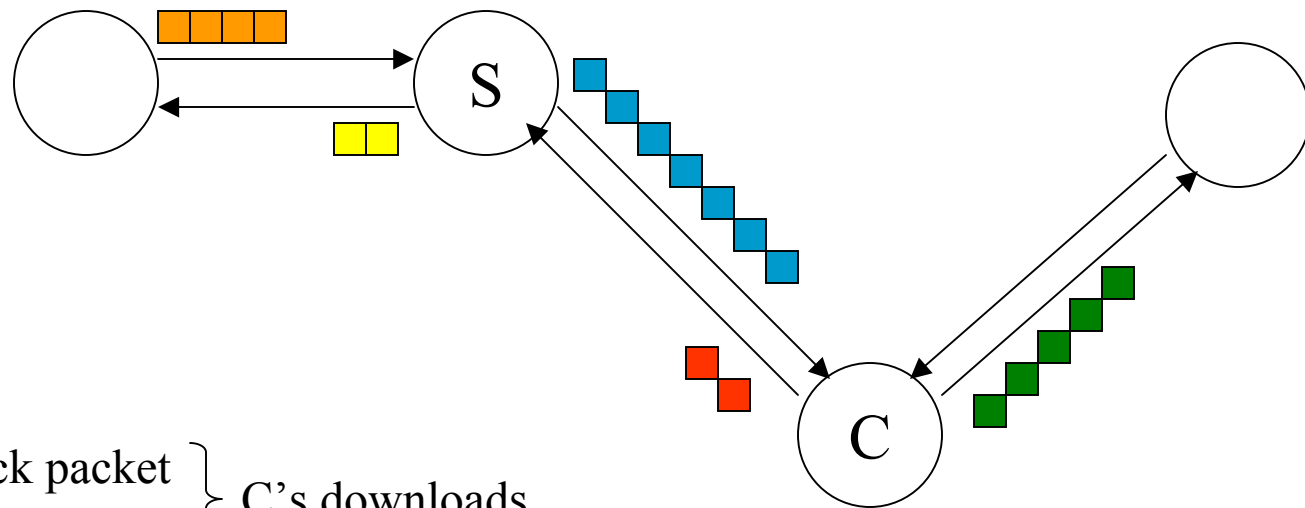




# Prioritization: No Free Lunch

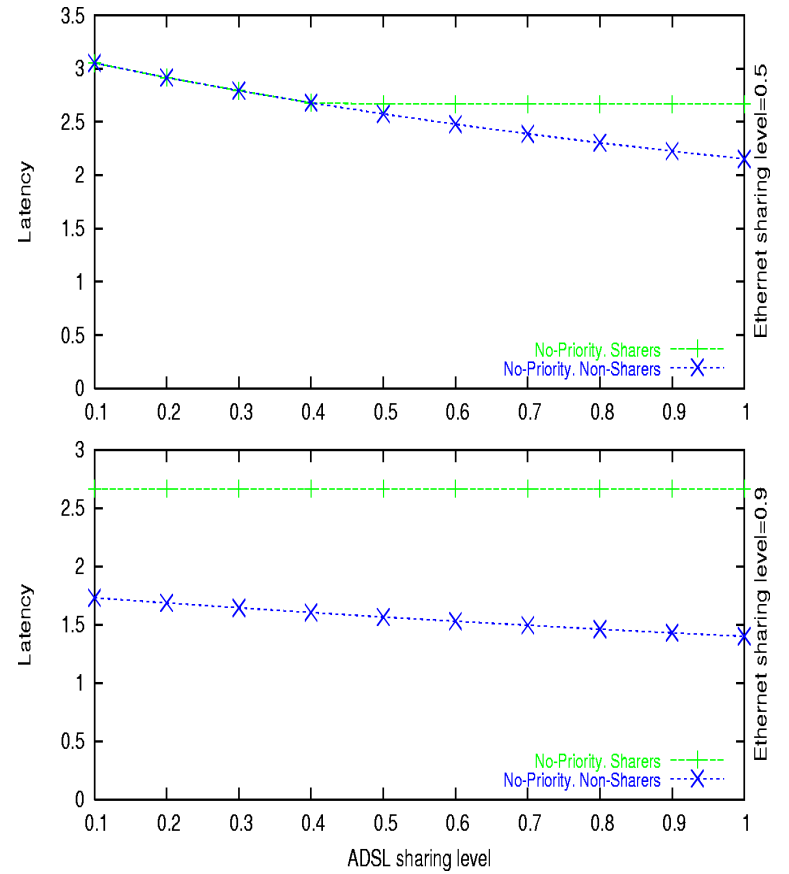


# Prioritization: No Free Lunch



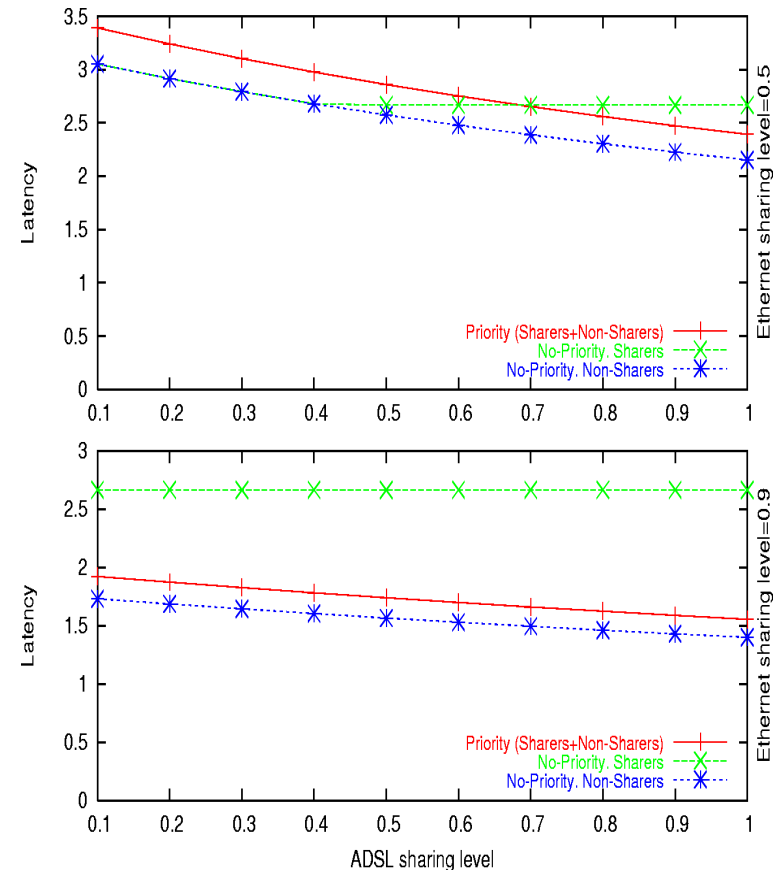
- Ack packet } C's downloads
- Data packet } C's downloads
- Data packet } C's uploads
- Ack packet } S's downloads
- Data packet } S's downloads

# Prioritization Effect



# Prioritization Effect

- Prioritization eliminates performance gap between sharers and non-sharers
- However, it may have a cost
- Prioritization may result in better or worse performance
  - Non-sharers: always worse
  - Sharers: better or worse





## Prioritization Effect

- Prioritization effect is unclear
- However, results are for a fixed sharing level
- When perceived disincentive is eliminated, we expect sharing level to increase
- If sharing level increases by  $\frac{1}{\beta}$ , overall performance increases
- For a large  $\beta$  (e.g.  $\beta = 0.9$ ), a relatively small increase in sharing level (1.11) increases system performance



# Outline

- Introduction
- P2P Performance Model
- Potential vs. Actual Disincentive
- Eliminating Disincentives via Prioritization
- **Conclusions**



# Conclusions

- There is high potential disincentive to share
- There is low actual disincentive to share in a homogeneous system or a heterogeneous system with low sharing level
- Prioritization
  - eliminates potential disincentive. i.e. sharers and non-sharers get same download latency
  - may result in better or worse performance (for a fixed sharing level)
  - When perceived disincentive is eliminated, we expect sharing level to increase
  - With a relatively small increase in sharing level, prioritization always increases system performance