Dependent Link Padding Algorithms for Low Latency Anonymity Systems

W. Wang, M. Motani, V. Srinivasan

CCS 2008

Presented by B. Choi
cs6461 at MTU

Motivation

- Low latency anonymity systems are vulnerable to traffic analysis attacks
- One way to thwart such an attack is to use dummy traffic
- Understanding of the cost and effectiveness is low
- Where to start?

Things to think ...

- Scope: entire network, tier-1 AS, tier-2 AS, tier-3 AS, ...
  - Tarzan?
- Effectiveness of dummy traffic
  - Linkability from a suspect input to any suspect output to be:
    - Minimized?
    - Randomized
    - Equalized?
- Cost: genuine traffic vs. dummy traffic

Background

- Independent link padding
  - Scope: one hop
  - Output pattern: pre-determined regardless of input
  - Straightforward output patterns: constant, exponential (Poisson)
- Dependent link padding
  - Scope: one hop
  - Output pattern: determined online depending on input
  - How to produce output with given input?

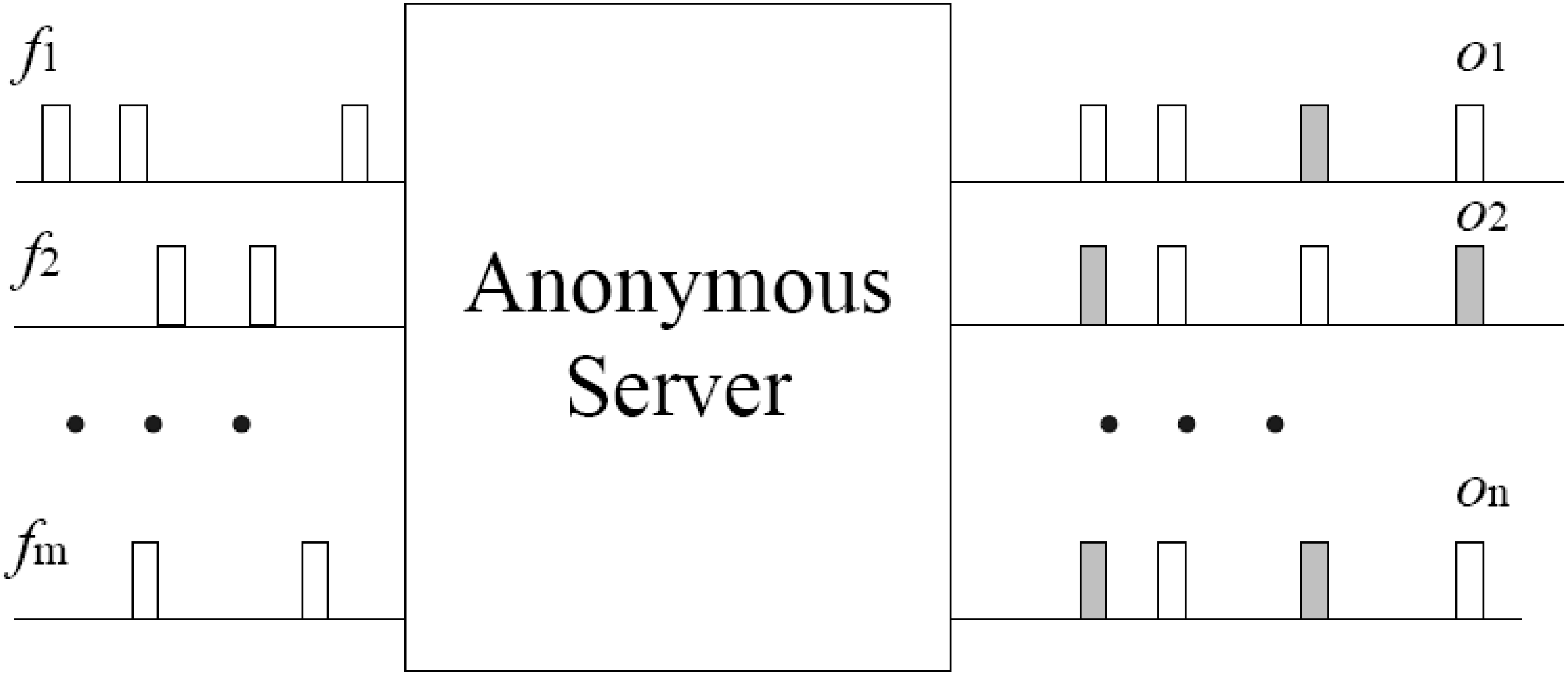
Intuition

- Independent link padding:
  - Very strong resistance against traffic analysis
  - Low bandwidth utilization
- Dependent link padding
  - Maybe strong enough to resist traffic analysis
  - Flexible bandwidth utilization
  - Can there be a good framework on DLP?

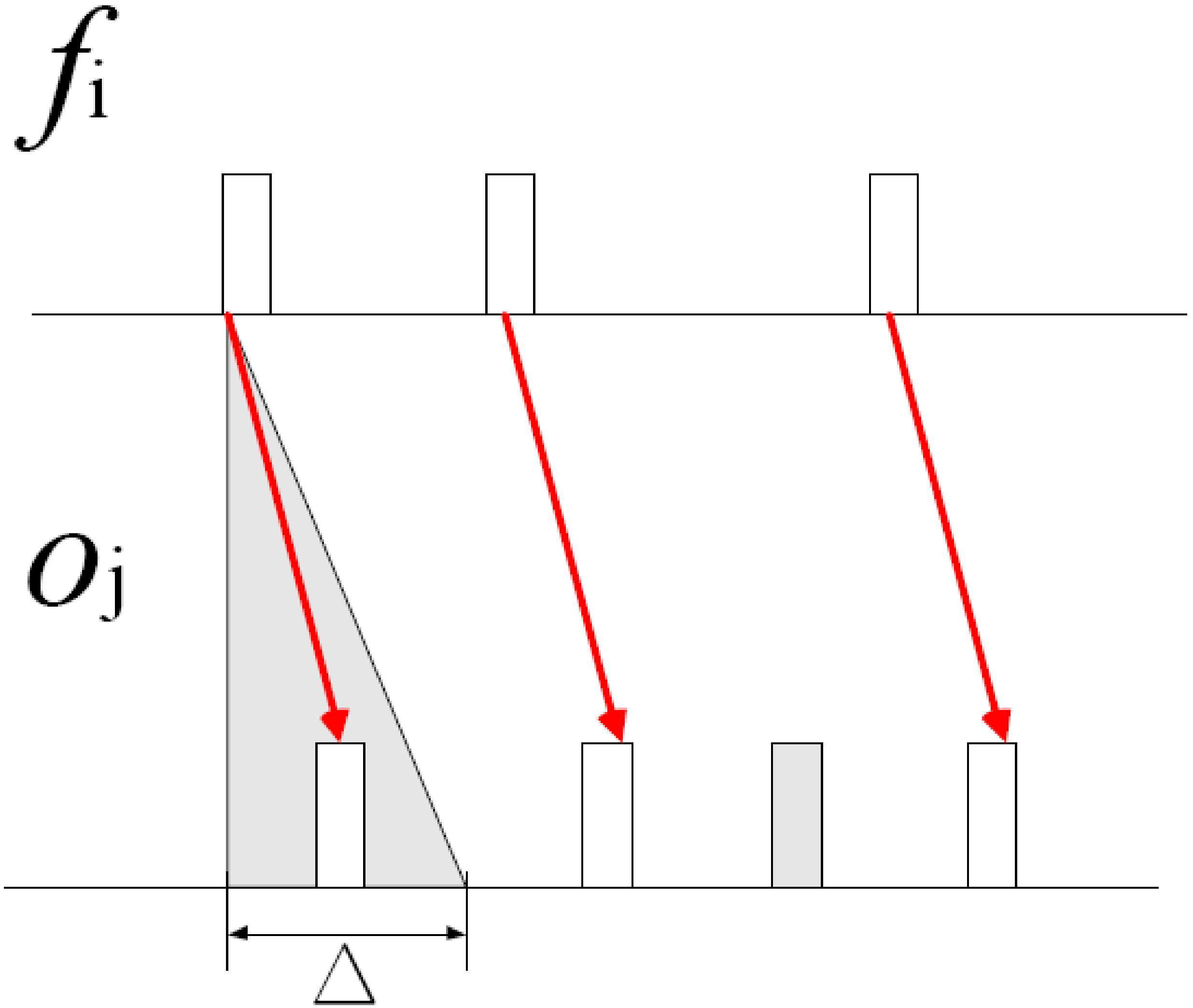
Assumptions

- Input flows are about of the same rate in Poisson
- All packets belong to a flow (link) are sent to the same output flow (link)
- Single anonymity server (mix) with a strict delay bound
- The mix does not drop any packet
- All output links show the same output to maximize the anonymity

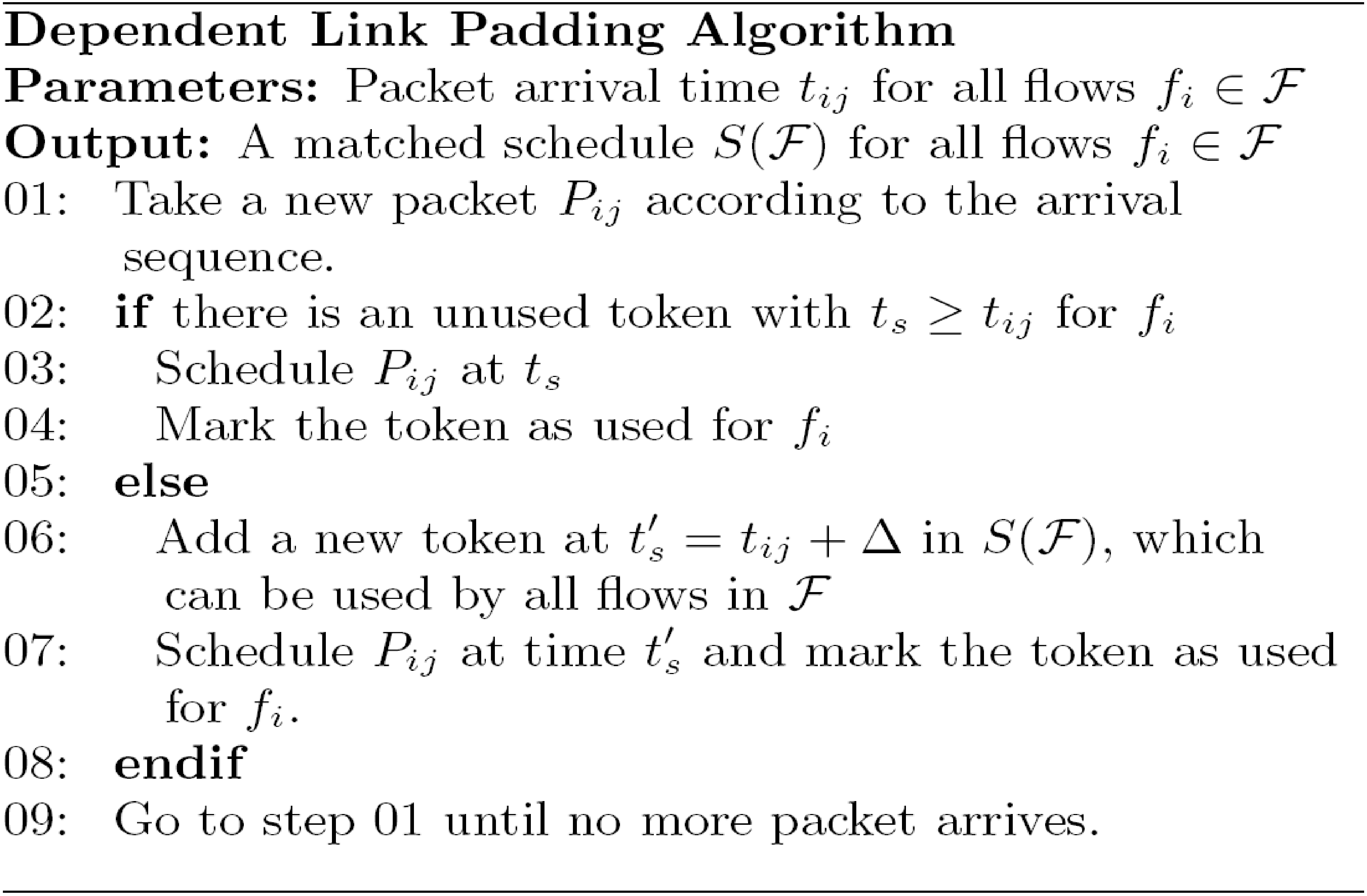
Mix



Matching packets



Proposed DLP algorithm



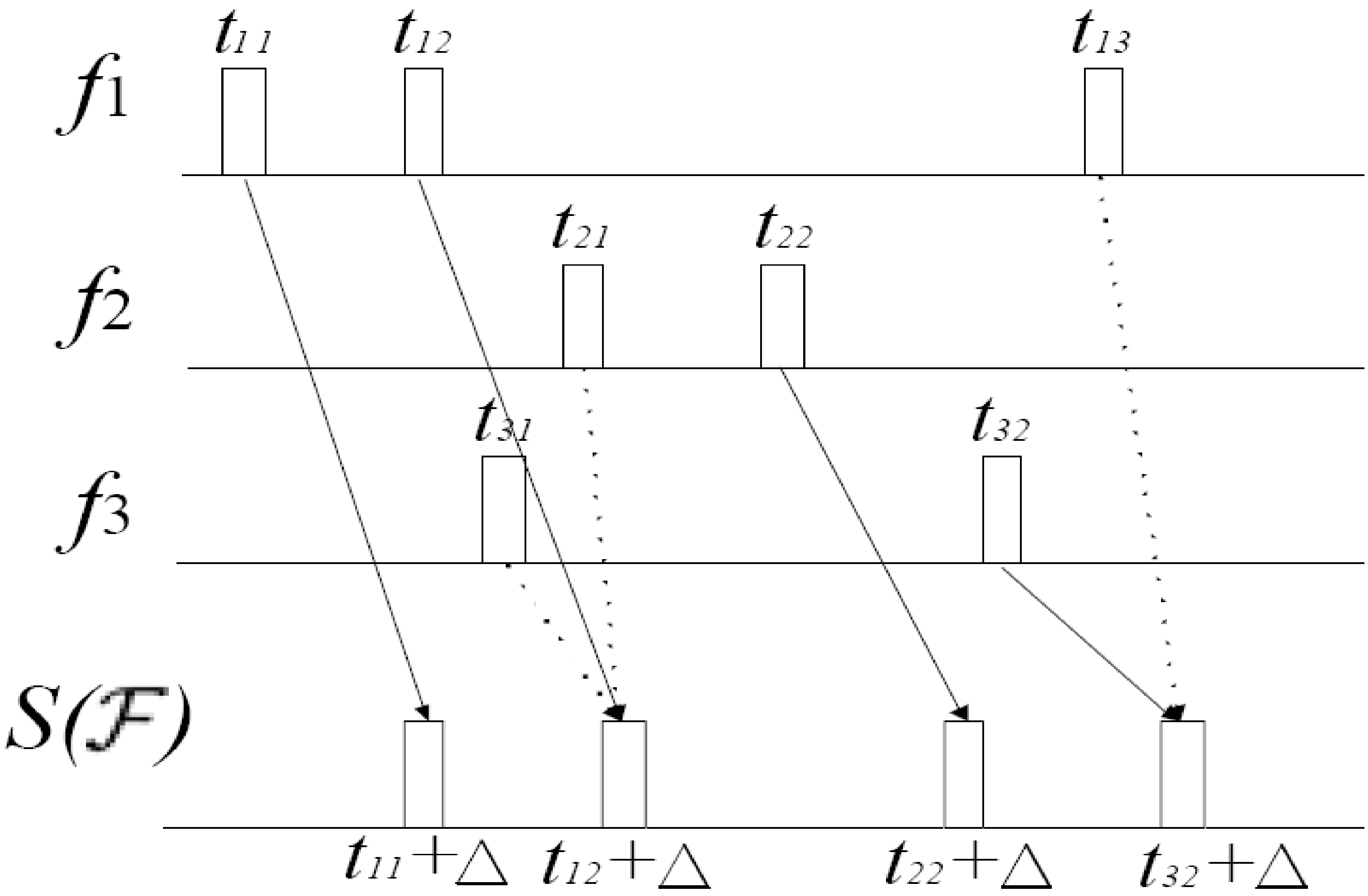
Dependent Link Padding Algorithm

Parameters: Packet arrival time t_{ij} for all flows $f_i \in \mathcal{F}$

Output: A matched schedule $S(\mathcal{F})$ for all flows $f_i \in \mathcal{F}$

```
01:  Take a new packet $P_{ij}$ according to the arrival
      sequence.
02:  if there is an unused token with $t_s \geq t_{ij}$ for $f_i$
03:    Schedule $P_{ij}$ at $t_s$
04:    Mark the token as used for $f_i$
05:  else
06:    Add a new token at $t'_s = t_{ij} + \Delta$ in $S(\mathcal{F})$, which
        can be used by all flows in $\mathcal{F}$
07:    Schedule $P_{ij}$ at time $t'_s$ and mark the token as used
        for $f_i$.
08:  endif
09:  Go to step 01 until no more packet arrives.
```

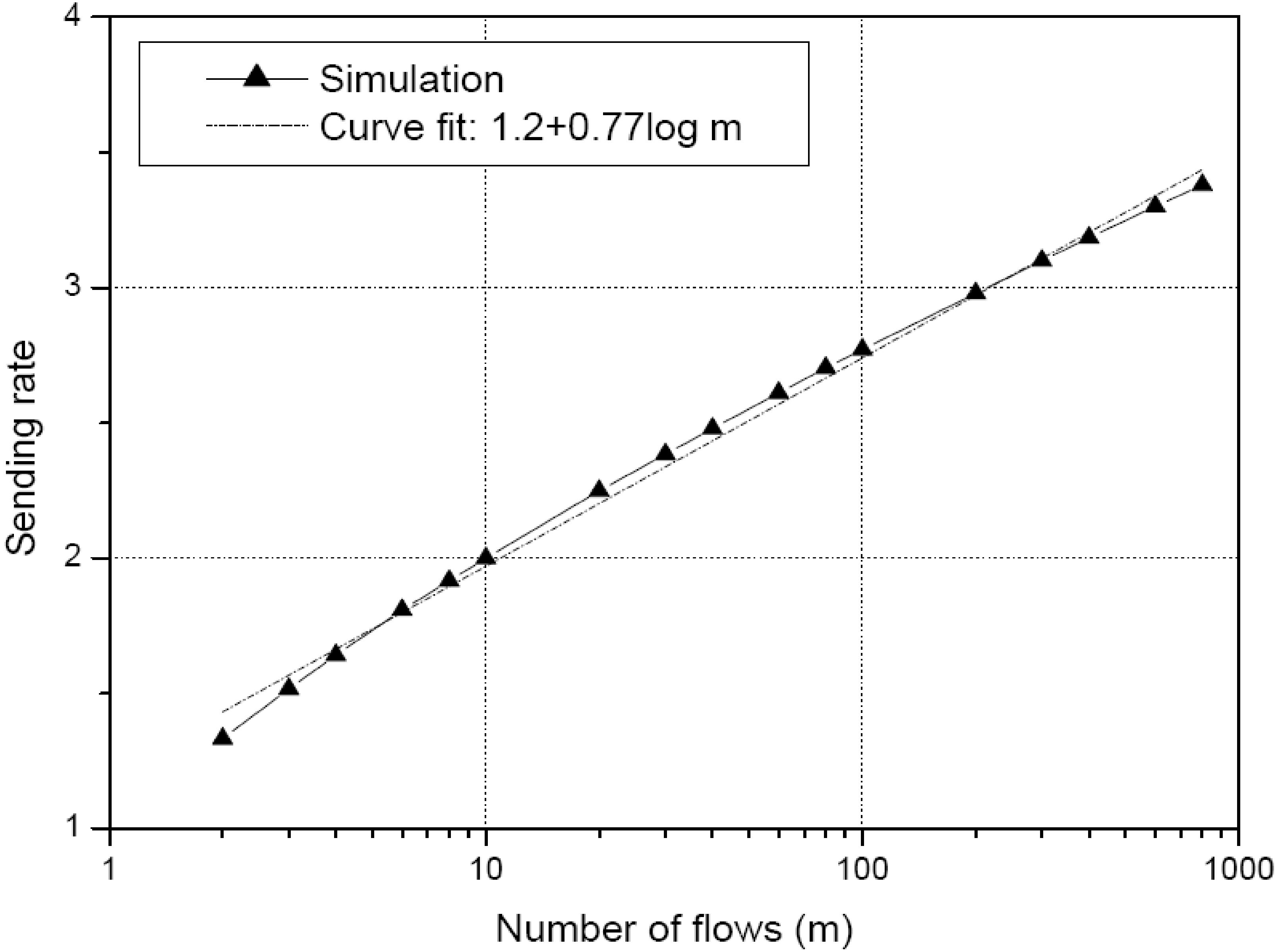
Example of output



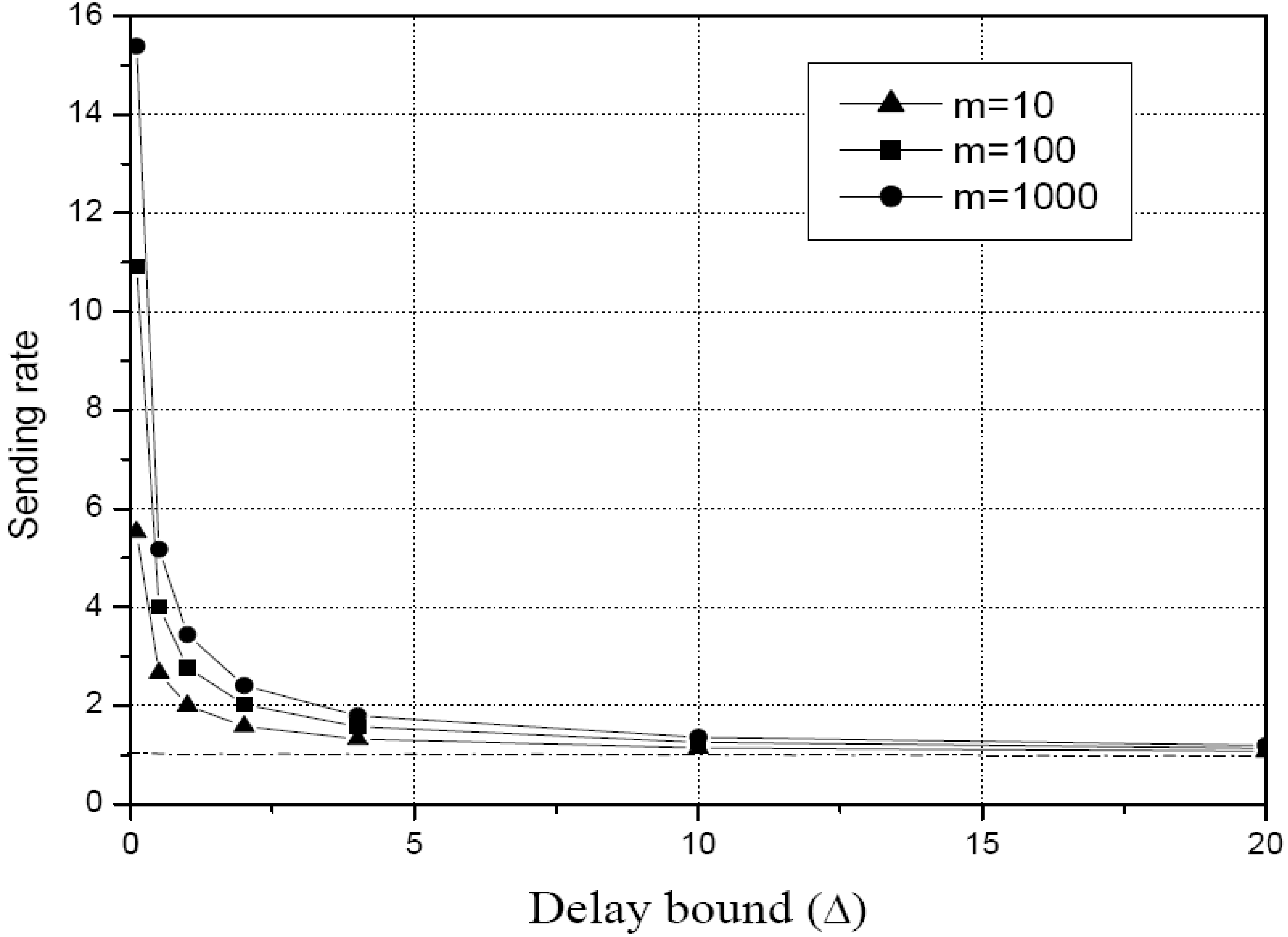
Claims

- The dummy traffic is minimized (max efficiency)
- Sending rate proportional to log(m)
  - M: the number of input flows
- Multi-hop: upper-bounded delay x hops
-

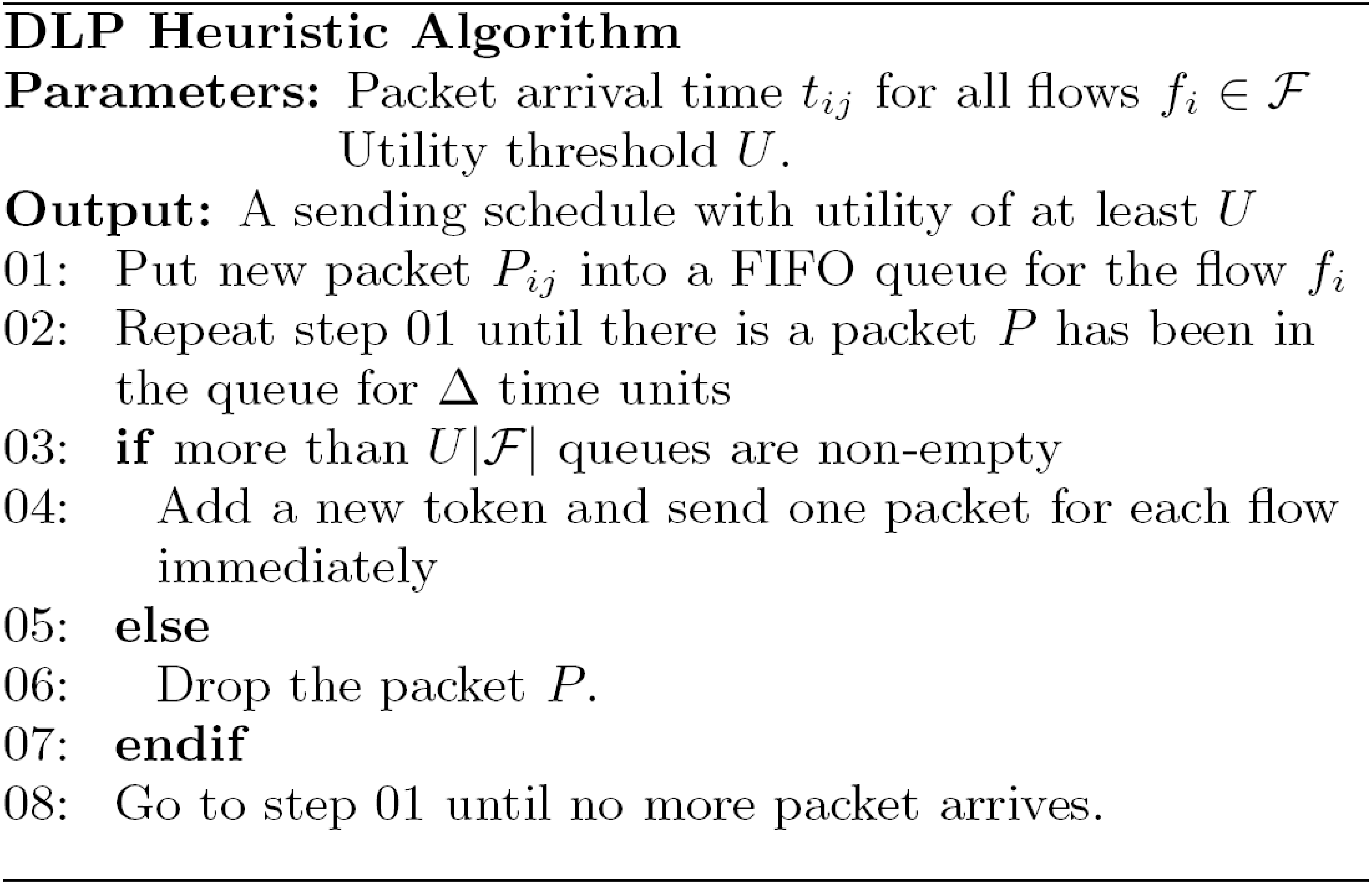
Experiment on the sending rate



Experiment on delay bound



Comparison with ILPs



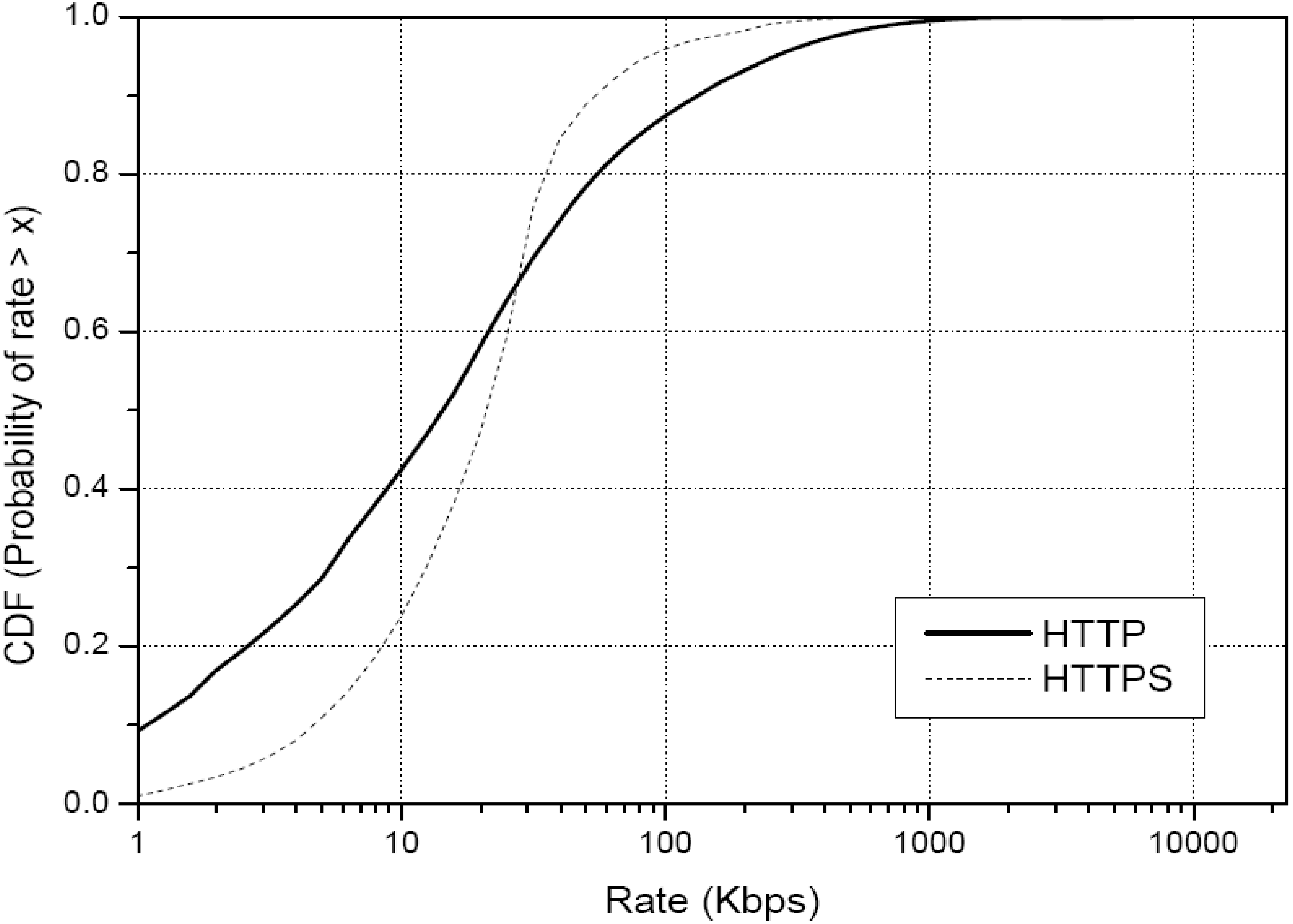
DLP Heuristic Algorithm

Parameters: Packet arrival time t_{ij} for all flows $f_i \in \mathcal{F}$
Utility threshold U.

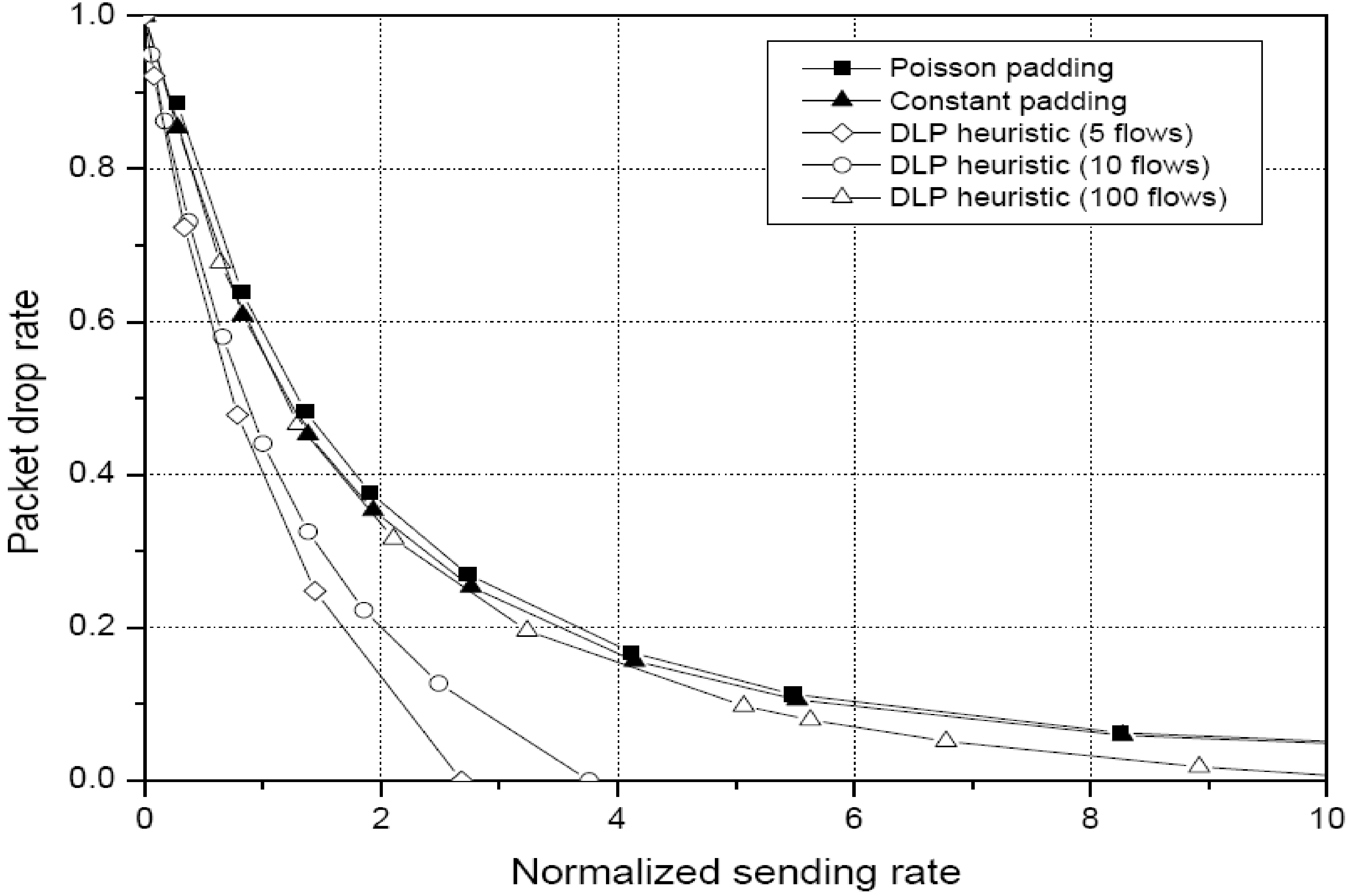
Output: A sending schedule with utility of at least U

```
01: Put new packet $P_{ij}$ into a FIFO queue for the flow $f_i$
02: Repeat step 01 until there is a packet $P$ has been in
    the queue for $\Delta$ time units
03: if more than $U|\mathcal{F}|$ queues are non-empty
04:     Add a new token and send one packet for each flow
        immediately
05: else
06:     Drop the packet $P$.
07: endif
08: Go to step 01 until no more packet arrives.
```

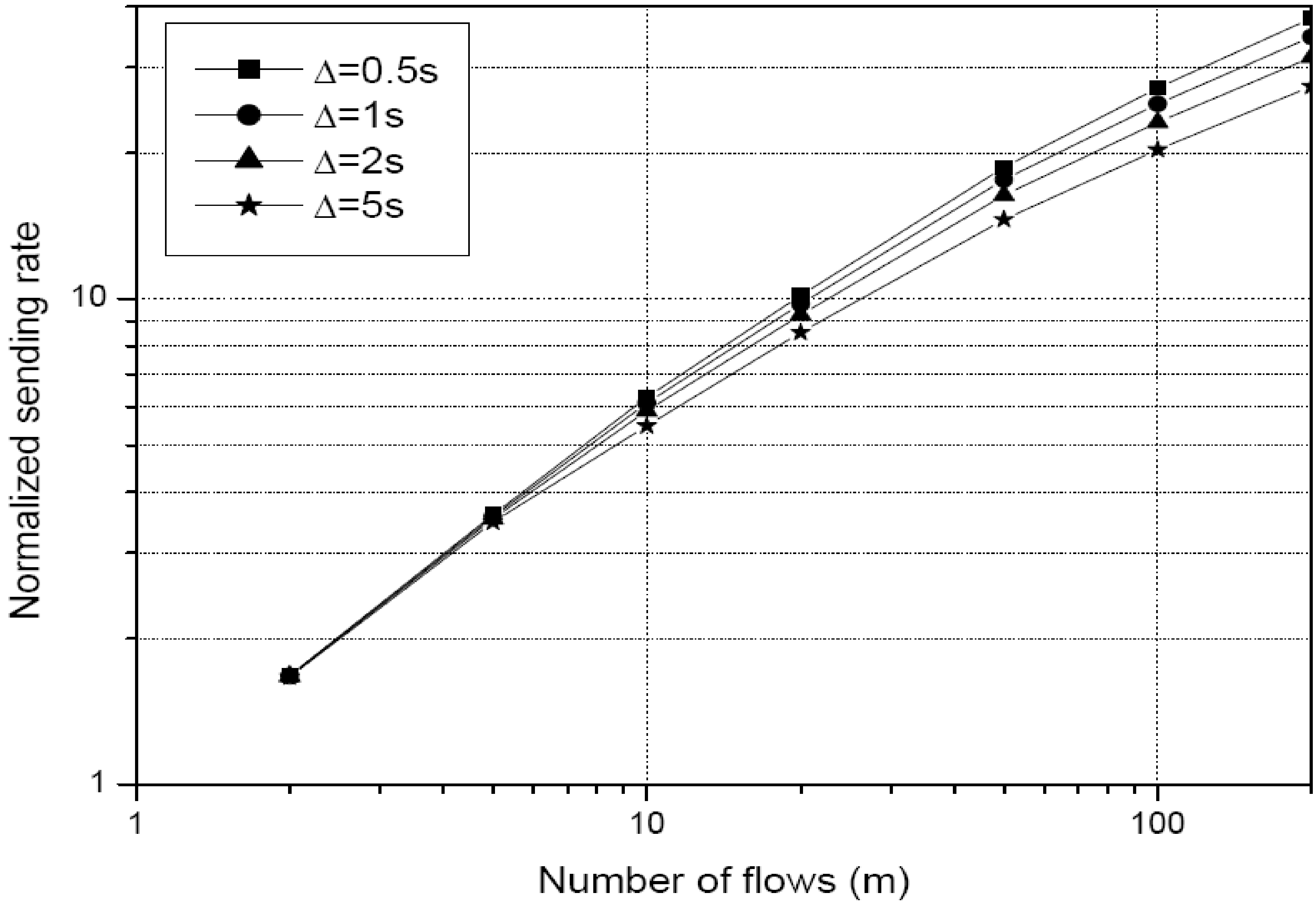
Real Traffic (2003)



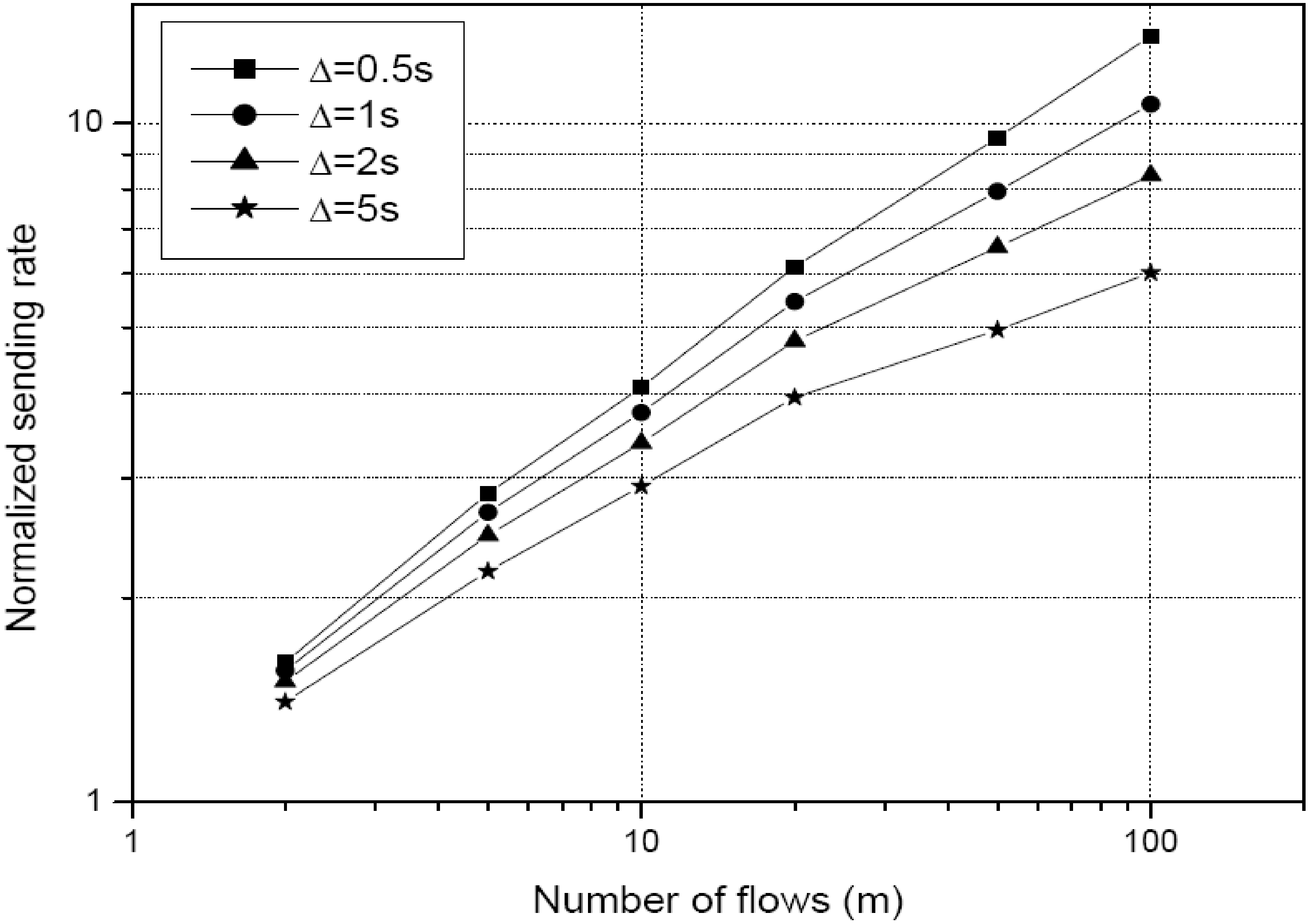
Packet drop rates



Drawback of DLP



Drawback of DLP


■ Δ=0.5s
● Δ=1s
▲ Δ=2s
★ Δ=5s
Normalized sending rate
Number of flows (m)