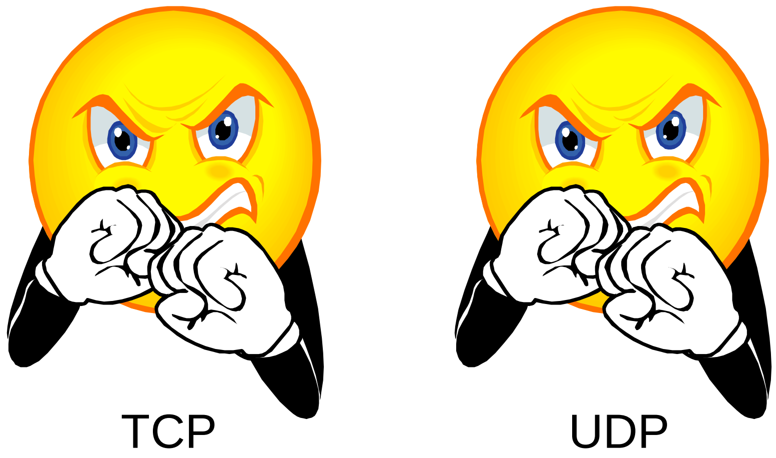
Stochastic TCP Friendliness And TCP-Friendly CBR-Like Rate Control

Lisong Xu, joint work with Jie Feng
Computer Science and Engineering, University of Nebraska-Lincoln

What is TCP friendliness?



- History: Proposed in the late 1990's [Floyd 99] as a design guideline for the traffic control of UDP.
- Reason: UDP without any traffic control could lead to TCP starvation and congestion collapse in the Internet.

Deterministic TCP friendliness

- A commonly used TCP friendliness (referred to as Deterministic TCP Friendliness, or DTF) requires that in a short time interval (such as a few RTTs)

  Avg Rate of a UDP flow = Avg Rate of a TCP flow.

- For example, TFRC (TCP Friendly Rate Control), TEAR (TCP Emulation At Receiver), Binomial, RAP (Rate Adaptation Protocol), SIMD (Square Increase Multiplicative Decrease)…

Problems of DTF

- DTF has a stringent requirement for UDP traffic.
- For example, the following protocols are not DTF by definition:
  - Any congestion control protocol that maintains a long term smooth rate (e.g. in minutes),
  - Any admission control protocol.
- DTF considerably limits the design space of TCP-friendly traffic control protocols.

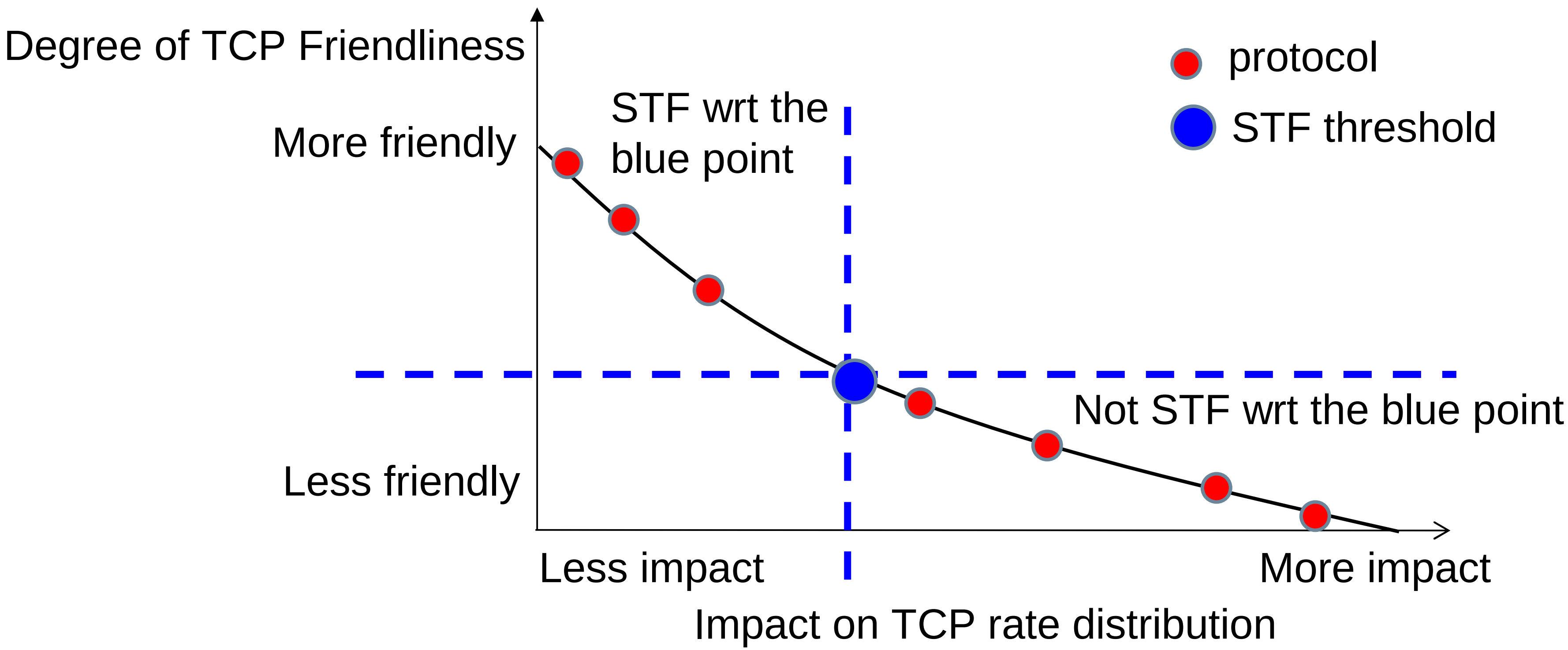
Our contributions

1. Propose Stochastic TCP Friendliness (STF) that
   - Greatly expands the design space of TCP-friendly traffic control protocols
   - Still effectively prevents the Internet from TCP starvation and congestion collapse.

2. Demonstrate that based on STF, more efficient protocols can be developed for some applications.

Stochastic TCP friendliness

- STF measures the degree of TCP friendliness of a traffic control protocol by its impact on the rate distribution of all TCP flows.



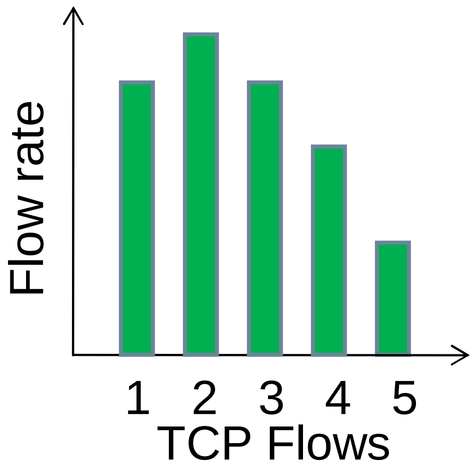
DTF vs. STF

- Intuitively, DTF considers the specific impact of UDP traffic on the rate of each TCP flow, whereas STF considers the statistic impact of UDP traffic on the rate distribution of all TCP flows.

- Illustrating Example: A single bottleneck shared by 1 UDP flow and 5 TCP flows arriving at different times and sending different amount of data.

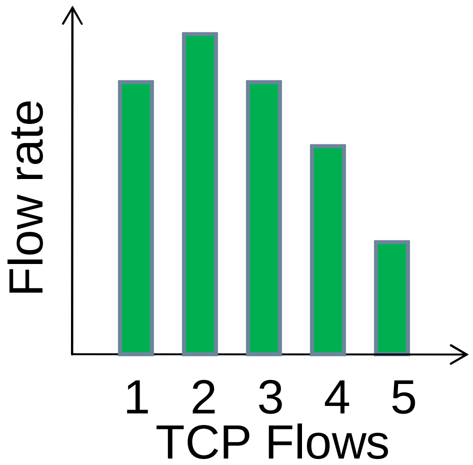
DTF vs. STF (cont.)

1. When the UDP flow is controlled by TCP congestion control algorithm



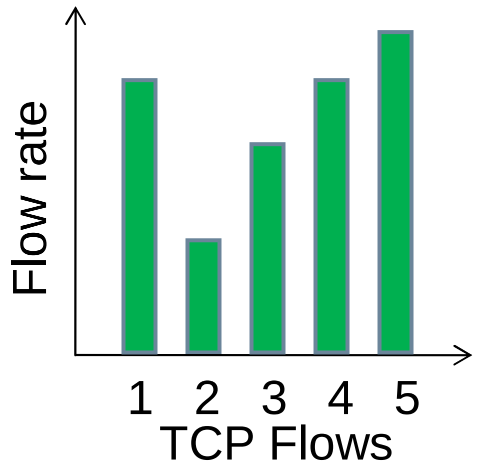
Reference case

2. When the UDP flow is controlled by deterministic TCP friendliness



Little impact on the rate of each flow

3. When the UDP flow is controlled by stochastic TCP friendliness



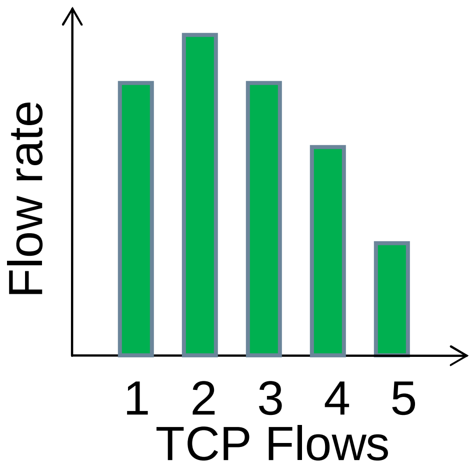
Little impact on the distribution of all flows

Measure the impact on distribution

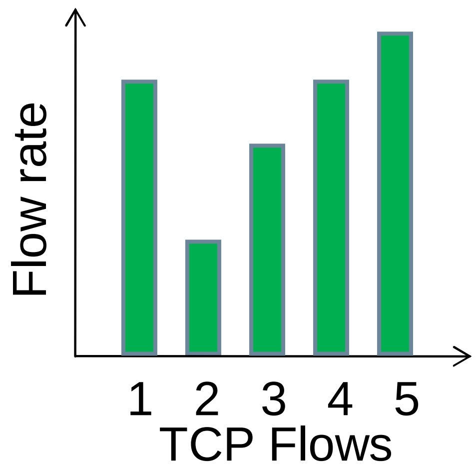
- We measure the impact of protocol π by comparing X^{TCP} and X^{π} in stochastic order.
- Define F = set of all TCP utility functions, such that

$$E[\ f(X^{TCP})\ ] \ \leq\ E[\ f(X^{\pi})\ ],\ \forall\ f \in F.$$

- The larger the size of F, the less the impact of π.



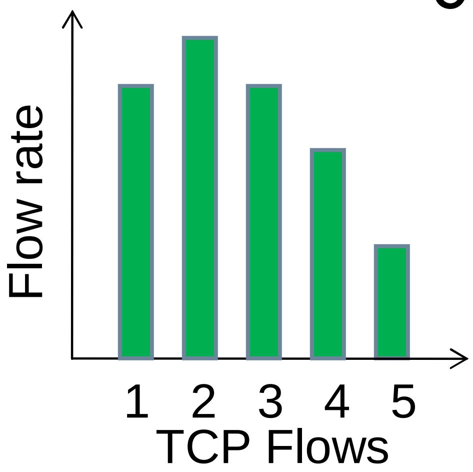
Random variable X^{TCP} denotes the average rate of a TCP flow when UDP traffic is controlled by <u>TCP algorithm.</u>



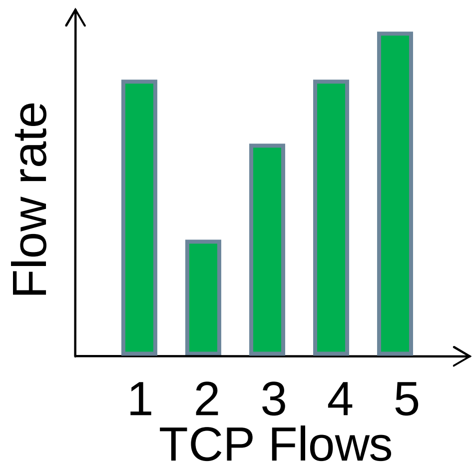
Random variable X^{π} denotes the average rate of a TCP flow when UDP traffic is controlled by <u>protocol π</u>

Formal definition of STF

- Definition: Protocol π is STF with respect to TCP utility function set F, if $E[\, f(X^{TCP})\,] \leq E[\, f(X^{\pi})\,]\ \forall\ f \in F$.
- Protocol π improves the overall satisfaction of TCP users measured with any utility function in F.
- The larger the size of F, the less the impact of π, and then the higher the degree of its TCP friendliness.



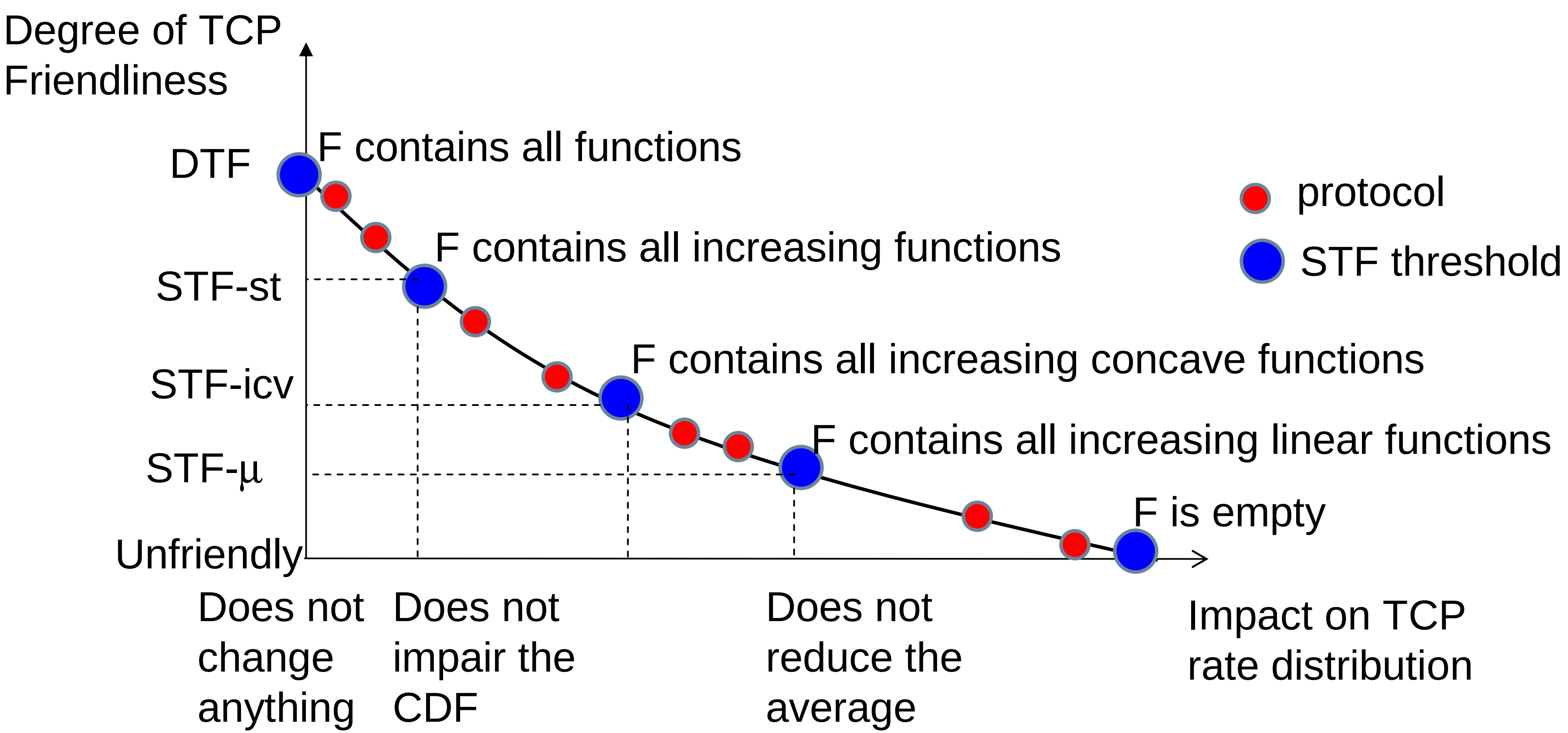
Random variable X^{TCP} denotes the average rate of a TCP flow when UDP traffic is controlled by TCP algorithm.



Random variable X^{π} denotes the average rate of a TCP flow when UDP traffic is controlled by protocol π

Class of STF definitions

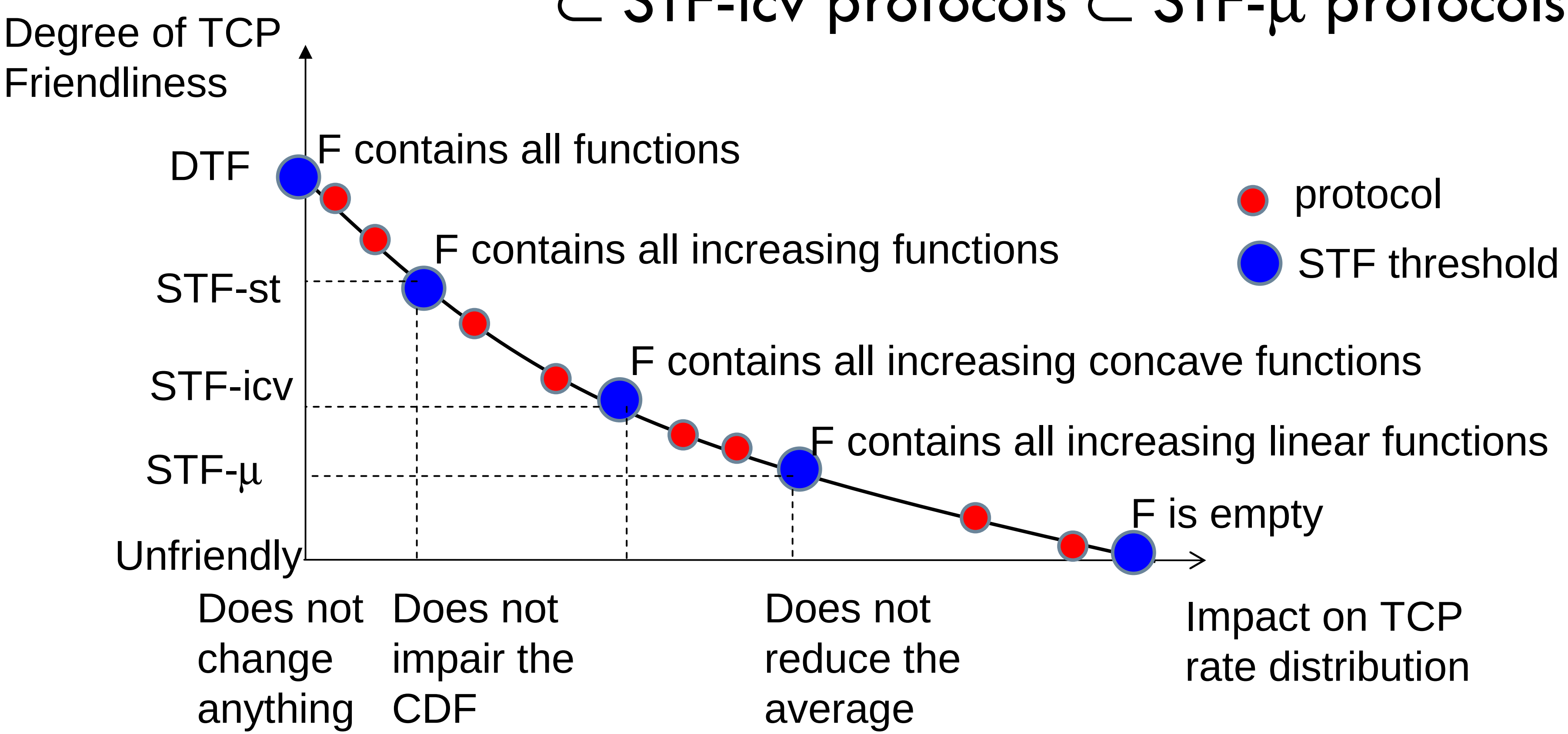
- We define five degrees of TCP friendliness.



Class of STF definitions (cont.)

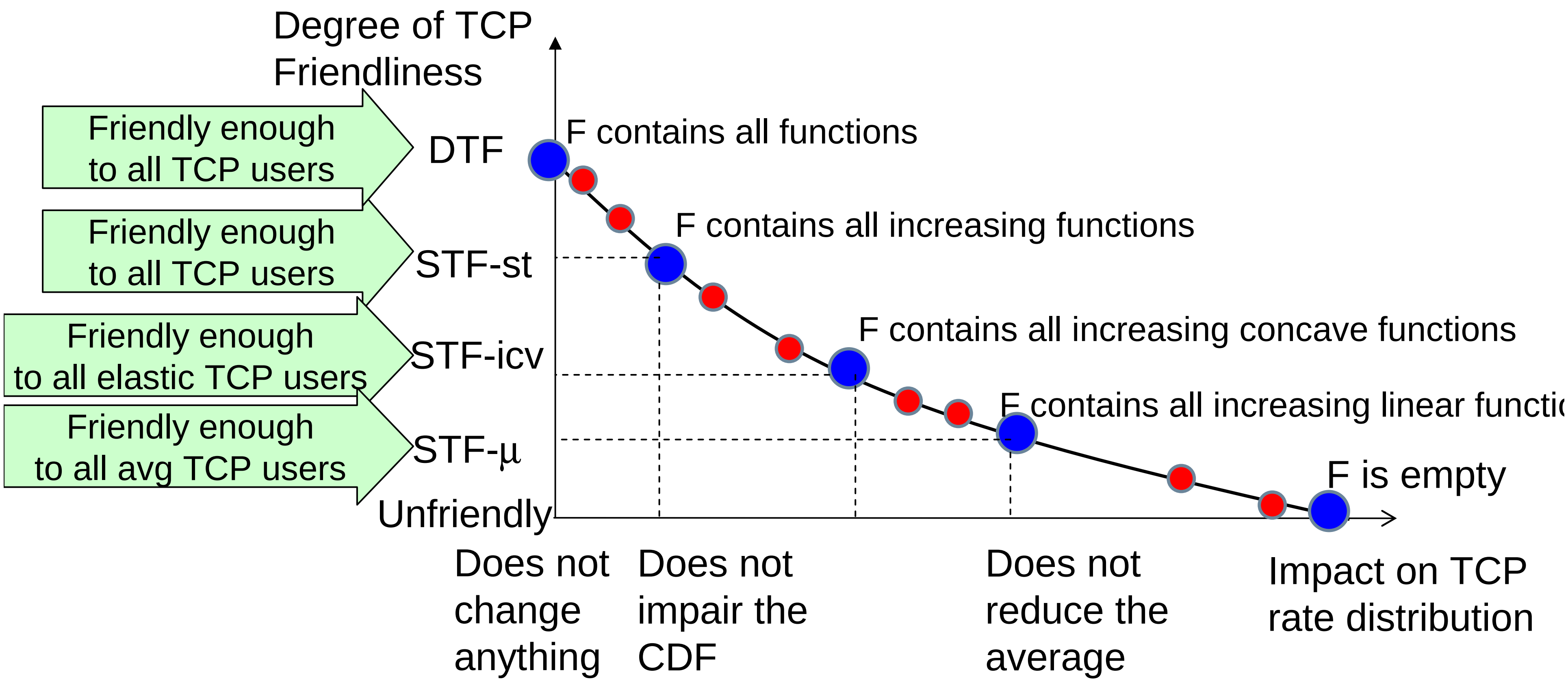
- DTF protocols ⊂ STF-st protocols

  ⊂ STF-icv protocols ⊂ STF-μ protocols



Which STF is friendly enough?

- "Which STF is friendly enough?" depends on "What are the utility functions of TCP users?"



Our contributions

1. Propose Stochastic TCP Friendliness (STF) that
   - Considerably expands the design space of TCP-friendly traffic control protocols
   - Still effectively prevents the Internet from TCP starvation and congestion collapse.

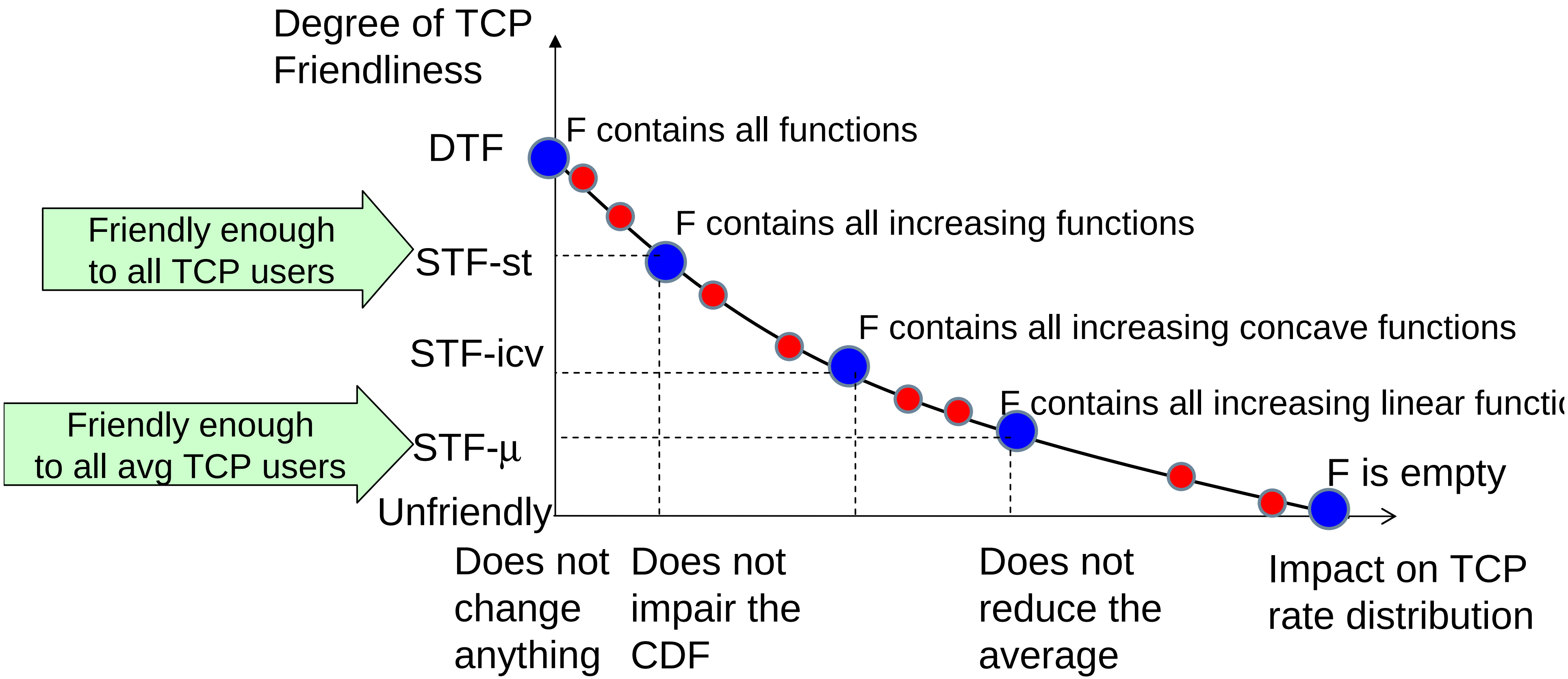
2. Demonstrate that based on STF, more efficient protocols can be developed for some applications.

TCP-friendly CBR-like rate control

- Motivation:
  - Media streaming applications usually prefer a smooth sending rate for better user-perceived media quality.
  - Example: A measurement study [sigcomm06] shows that Skype users are sensitive to the rate variation in an interval of 30 seconds.
  - By definition, all DTF protocols (e.g. TFRC, TEAR, etc) can only maintain a smooth sending rate in a short interval (e.g. a few RTTs).
- Design TFCBR to have a smooth sending rate in a longer time interval.

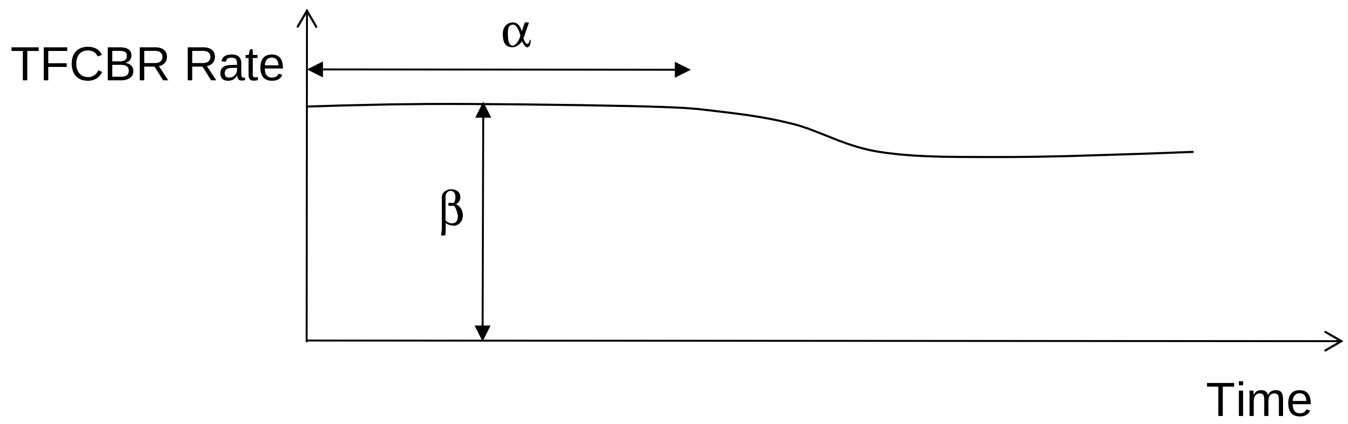
TFCBR Design Goals

- 1: A smooth rate in an interval of 1-minute, then not DTF.
- 2: STF-st in most cases, STF-μ in all cases.



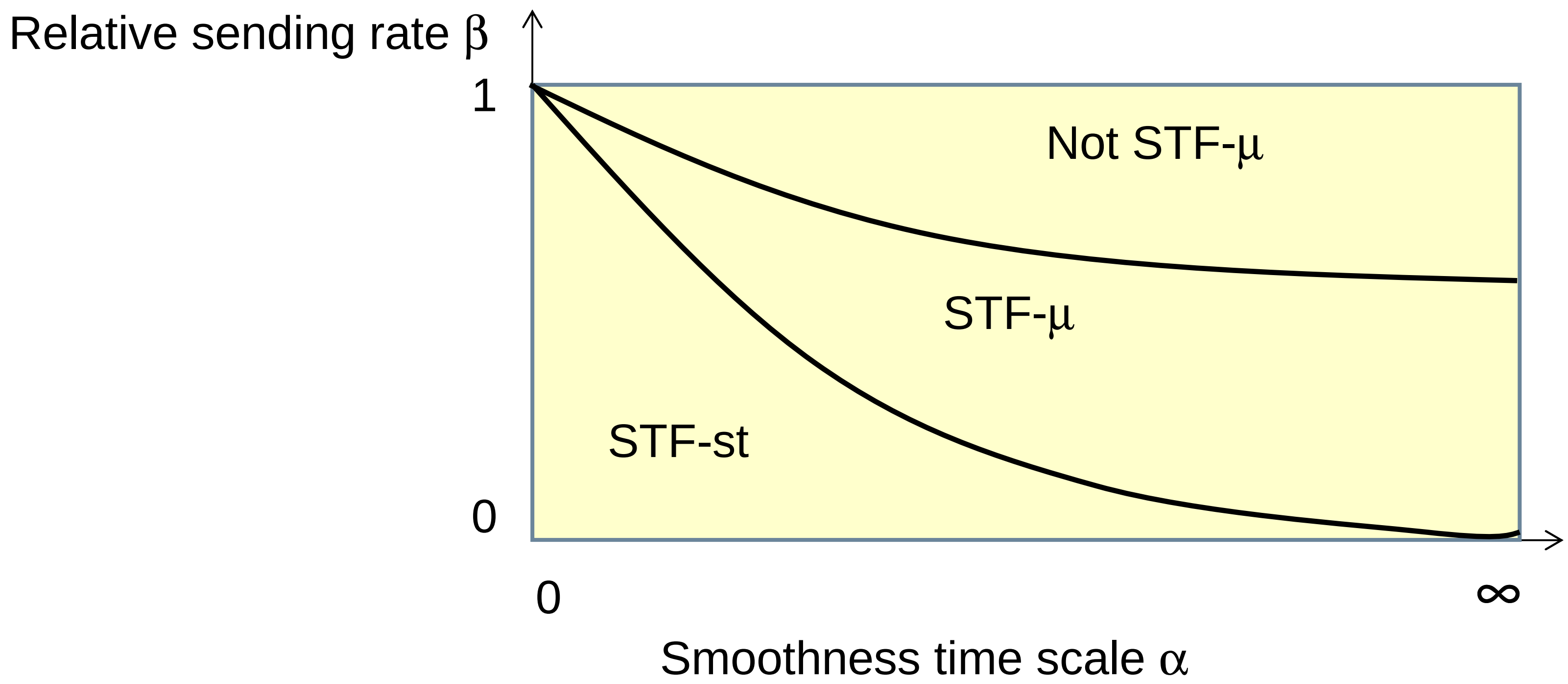
TFCBR Parameters

- TFCBR controls its sending rate with two parameters:
  - Smoothness time scale α determines on what time scale TFCBR maintains a smooth sending rate. Unit: seconds
  - Relative sending rate β = (the average TFCBR rate)/(the average TCP rate in the same network environment) on a time scale of α seconds. $\beta \in (0,1)$



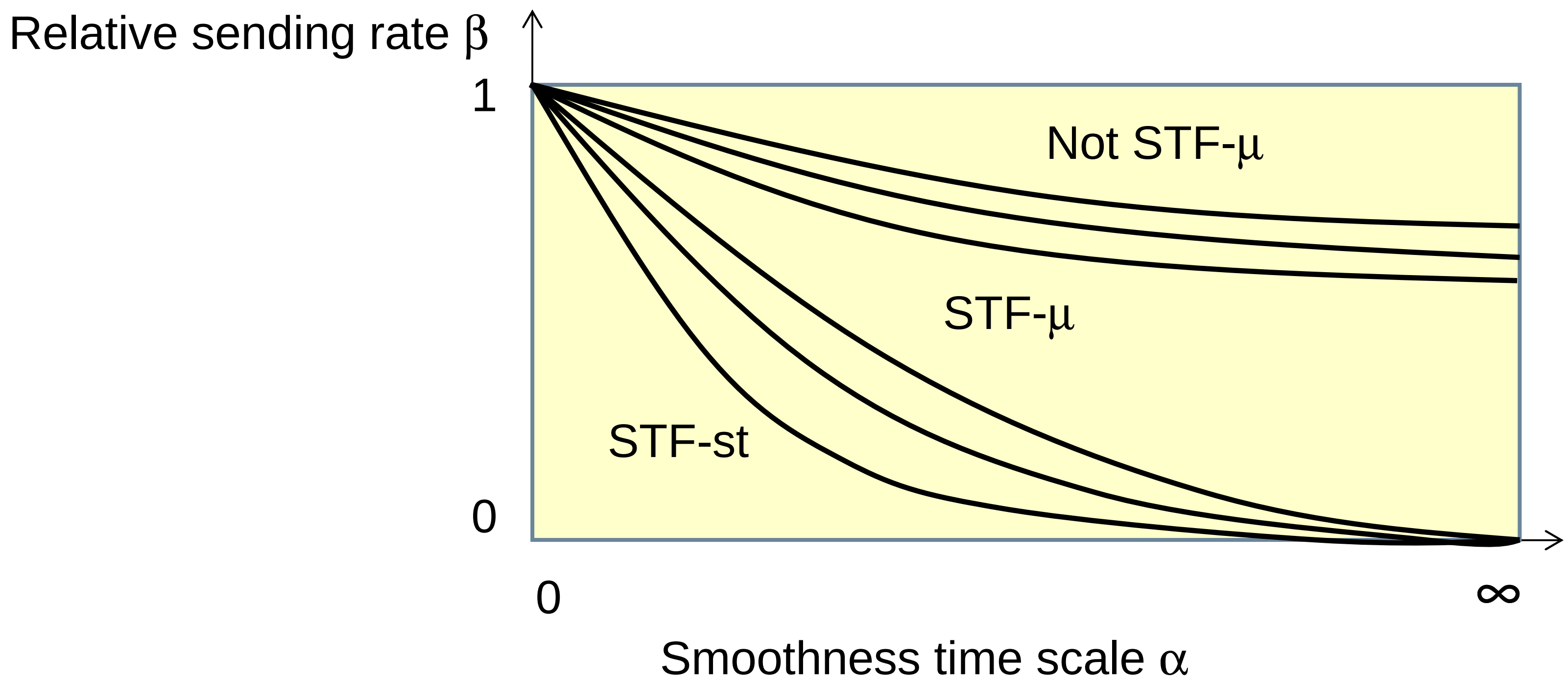
TFCBR Parameter Space

- To choose the values of α and β, we study the TFCBR parameter space.



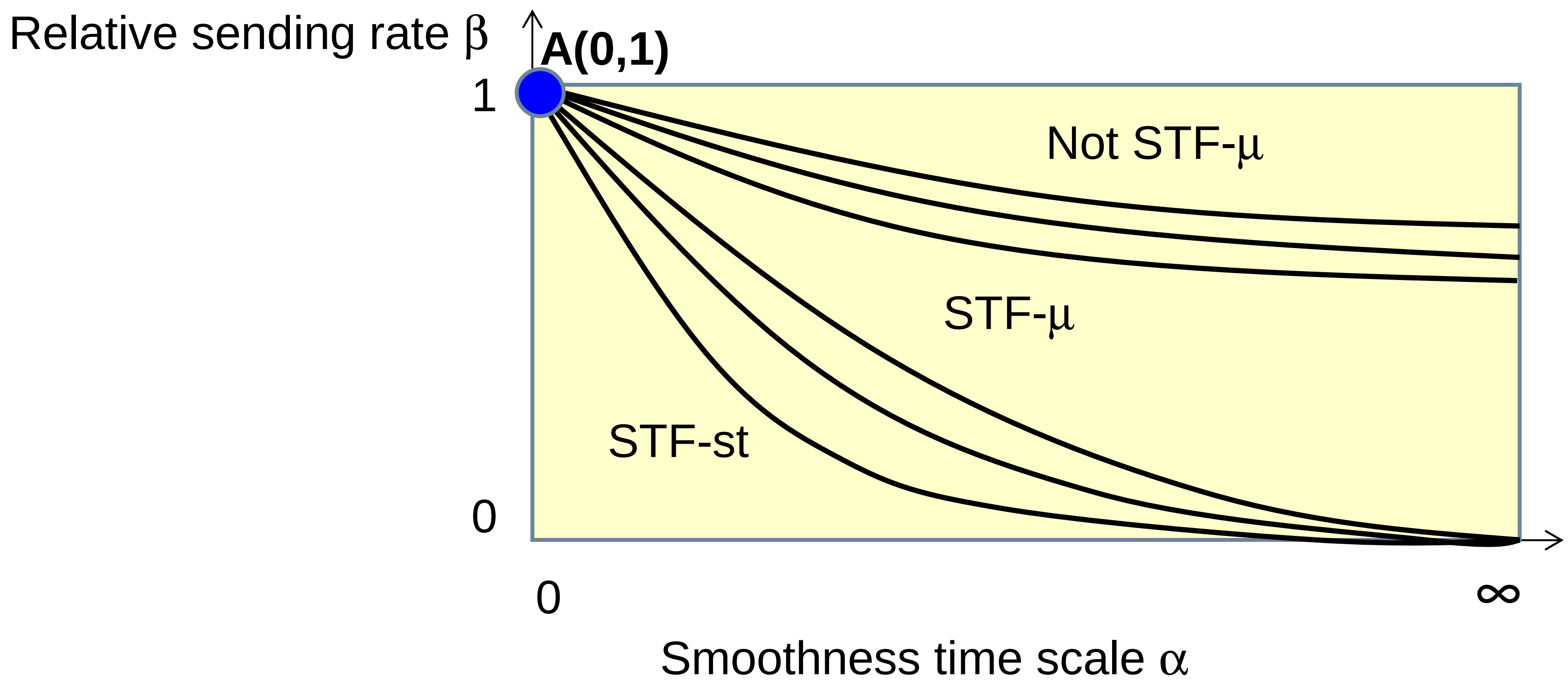
Determining the three areas

- The exact STF-st and STF-μ areas depend on many network parameters, such as number of TCP flows, number of UDP flows, link capacity, link delay,…



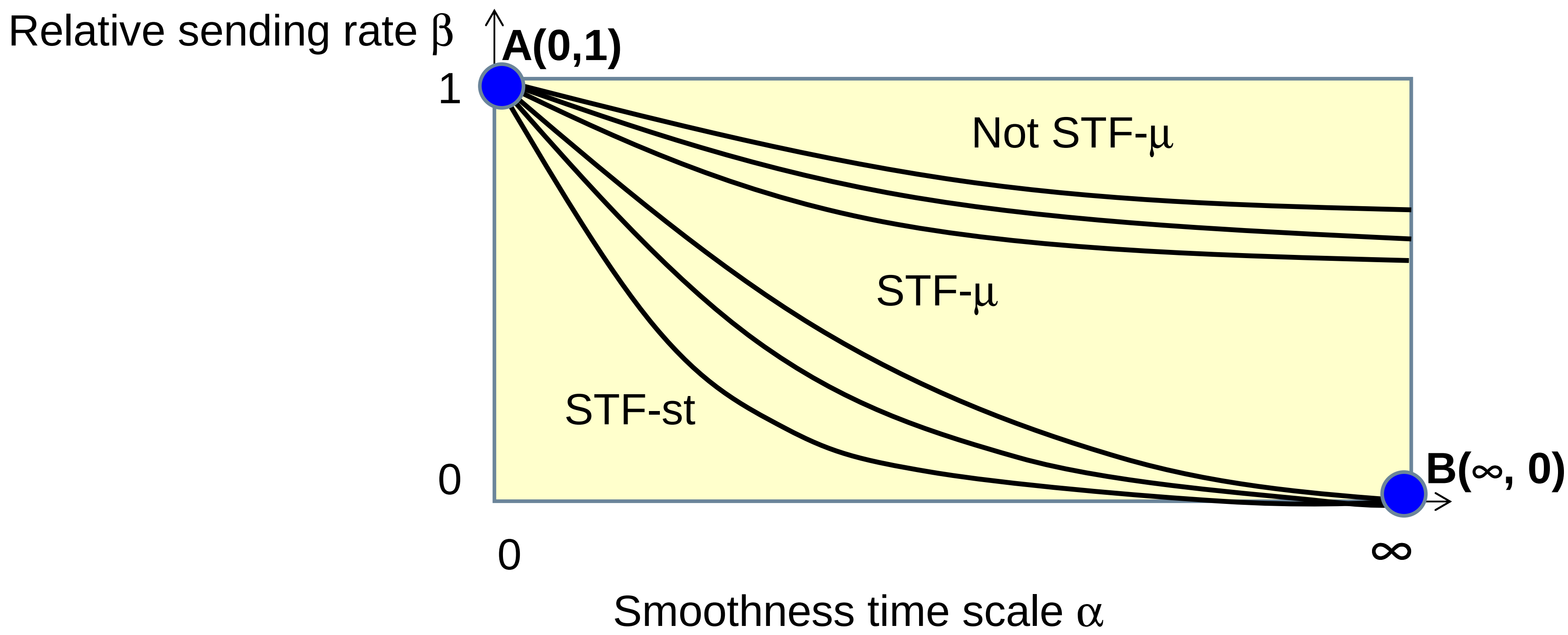
STF-st condition

- When α=0 and β=1, TFCBR behaves exactly the same as TCP, and then it is both STF-st and STF-μ.



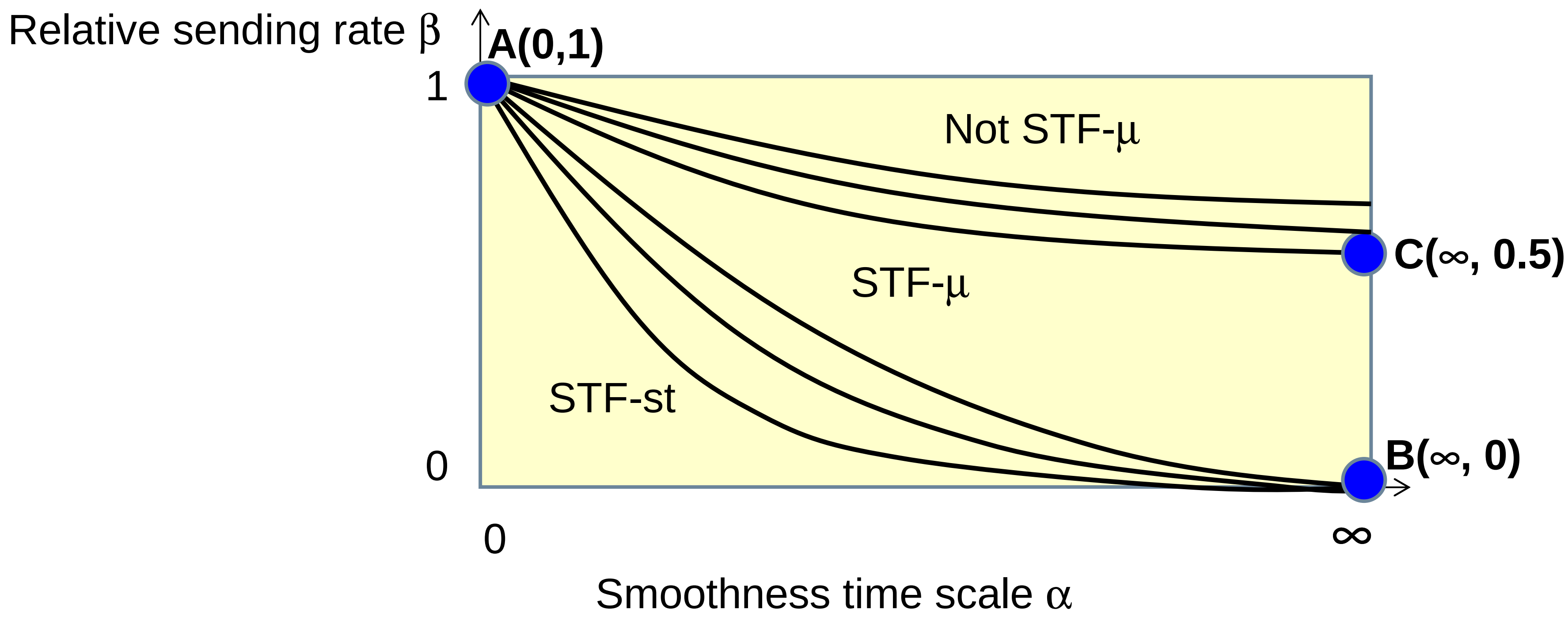
STF-st condition (cont.)

- Based on a flow-level queueing model and validated by NS-2 simulation, we show that TFCBR with $\alpha=\infty$ is STF-st for any network conditions if $\beta=0$.



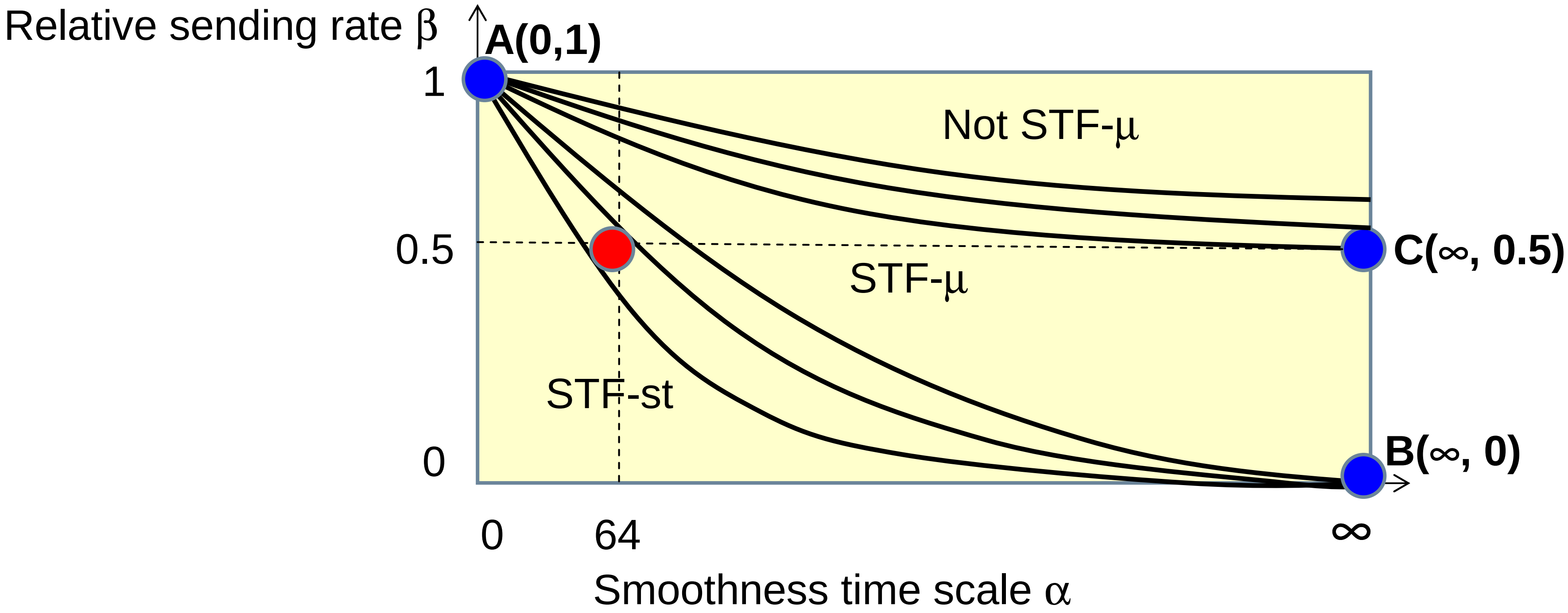
STF-μ condition

- Based on a flow-level queueing model and validated by NS-2 simulation, we show that TFCBR with $\alpha=\infty$ is STF-μ for any network conditions if $\beta \leq 0.5$



Parameter Setting

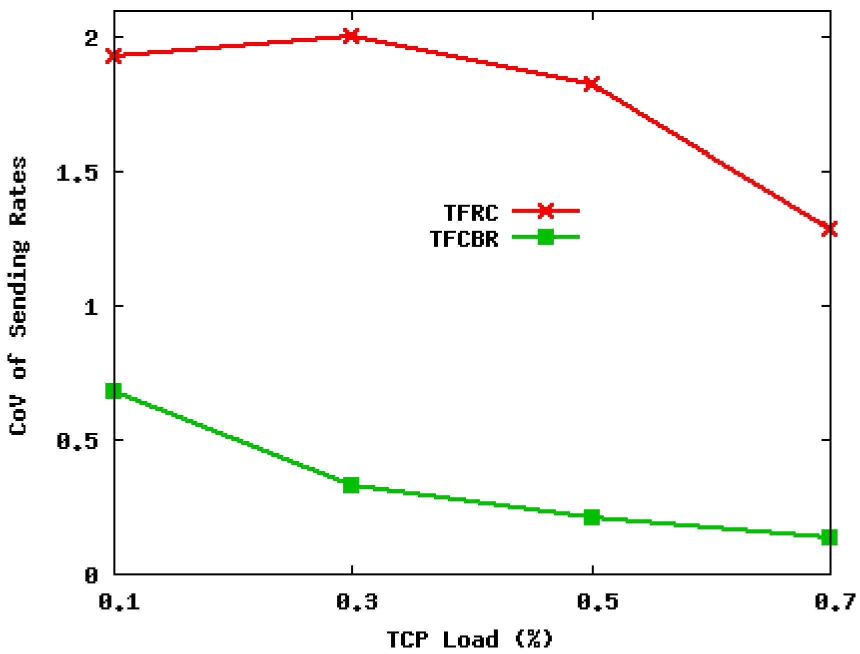
- TFCBR chooses ●(64, 0.5). i.e. α=64 seconds, β=0.5
  - Smooth rate in an interval of 64 seconds.
  - STF-st in most cases, and STF-μ in all cases.



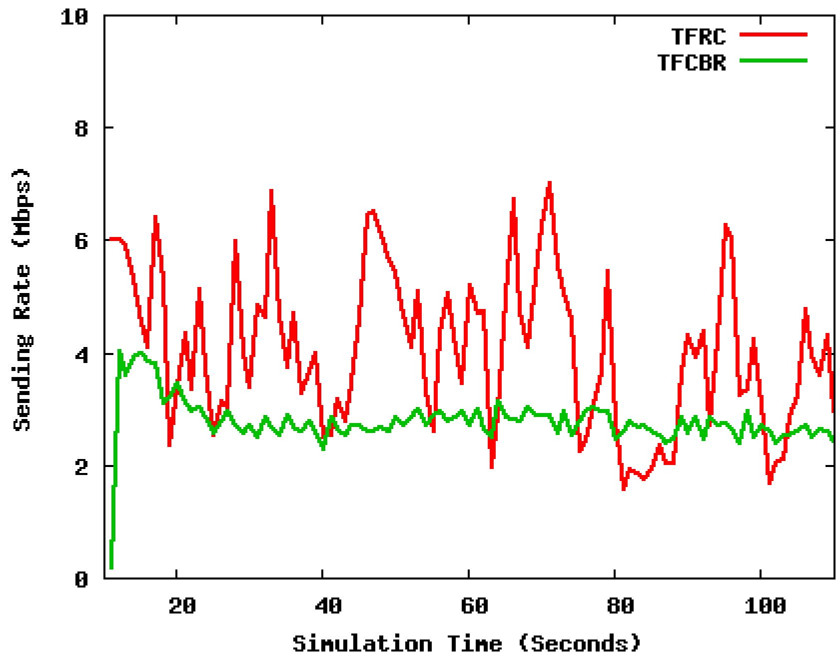
Simulation

- We use NS-2 to simulate
  - Three types of TCP flows
    - Short-lived TCP flows arriving as a Poisson process
    - Short-lived TCP flows generated by using PackMime
    - Long-lived TCP flows
  - Three types of traffic control protocols for UDP flows
    - DTF protocols: TFRC (TCP-Friendly Rate Control)
    - STF protocols: TFCBR
    - Non-TCP-Friendly protocols that maintain the same rate as a TCP flow only in a long time interval.

TFCBR is smooth in an interval of 1-minute?

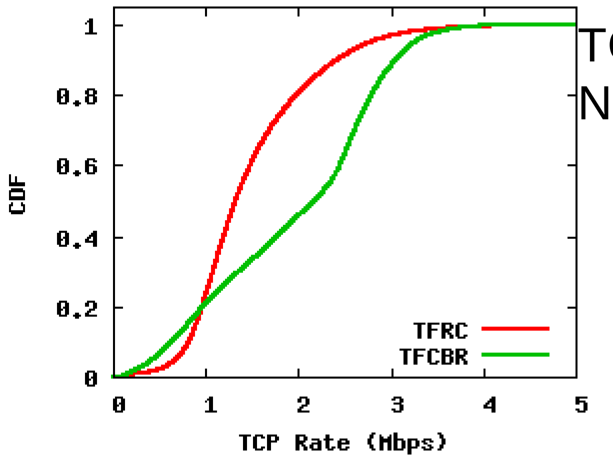


Coefficient of Variation (CoV) of TFRC and TFCBR in an interval of 1 minute when TCP load is 10% to 70% of the bottleneck link capacity.

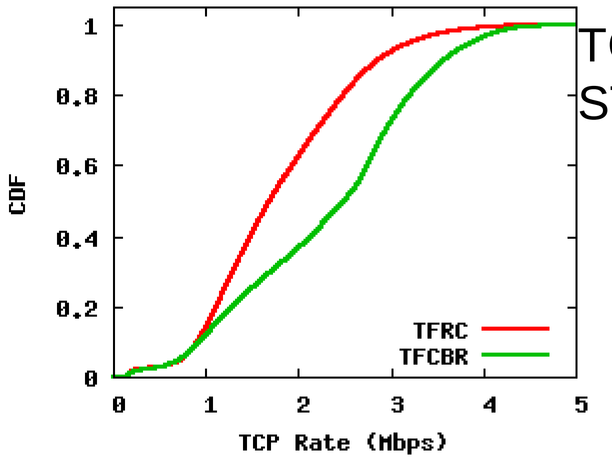


One-second average rate of TFRC and TFCBR when TCP load is 30% of the bottleneck link capacity.

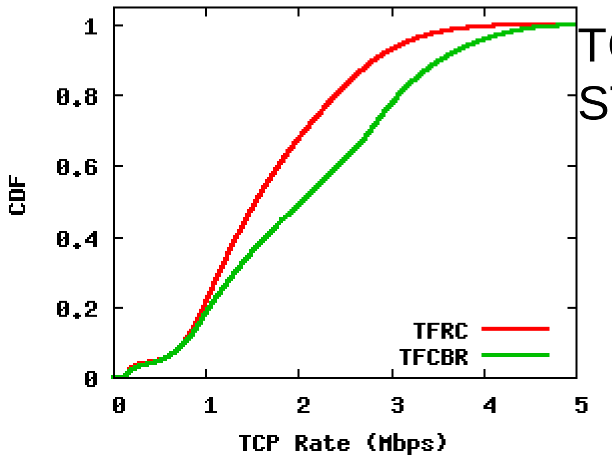
TFCBR is STF-st in most cases?



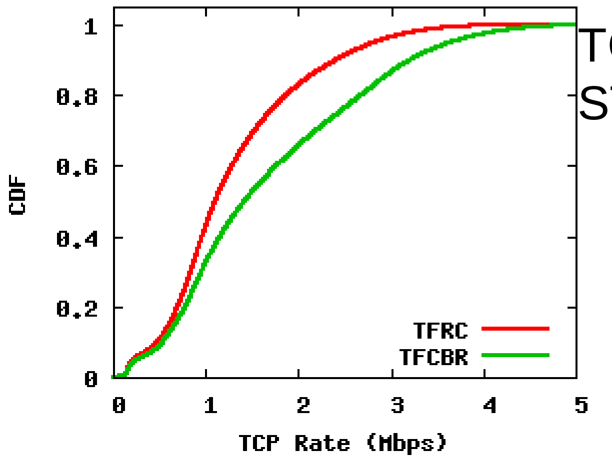
TCP load=10%
NOT STF-st



TCP load=30%
STF-st

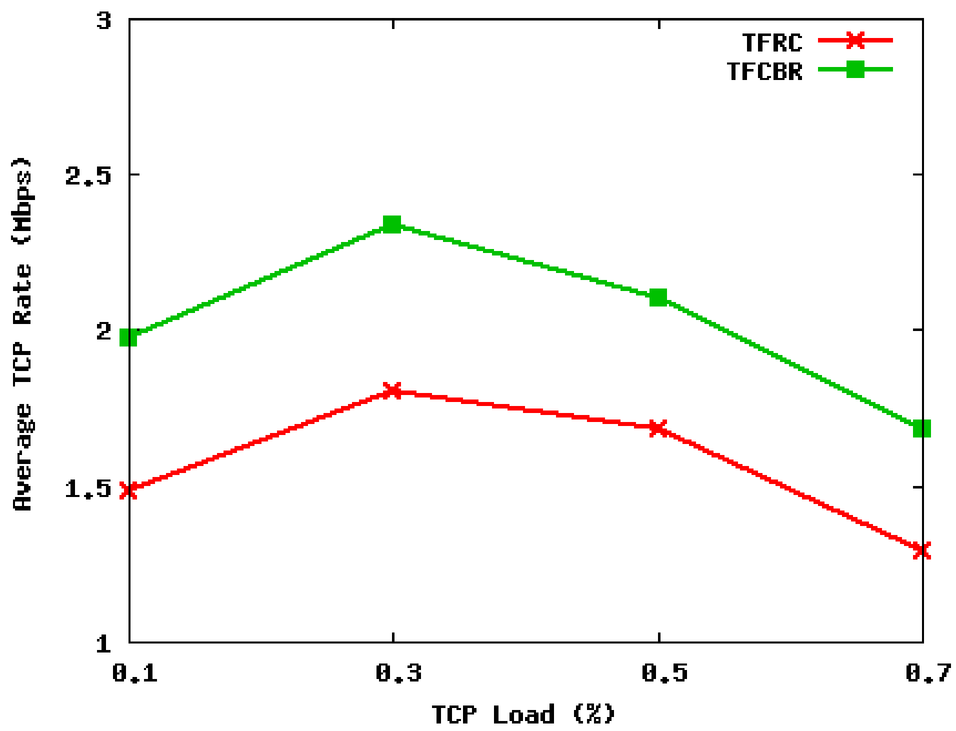


TCP load=50%
STF-st



TCP load=70%
STF-st

TFCBR is STF-μ in all cases?



Average rate of all TCP flows when competing with TFRC and with TFCBR

Conclusions

- Propose STF to considerably expand the design space of TCP-friendly traffic control protocols.
- Propose TFCBR to demonstrate that more efficient protocols based on STF can be developed for some applications.
- This paper considers only the friendliness problem (i.e. the impact of UDP on TCP), future work should also consider the fairness problem (i.e. impact of them on each other).

Acknowledgement

- Our work is inspired by the TCP and UDP fairness work by Prof. Dahming Chiu and Prof. Adrian Tam at The Chinese University of Hong Kong.
- Our work is supported by NSF CAREER Award 0644080.