

Simulación de control de congestión de TCP

Enlace compartido — Actividad opcional

2025-09-24 11:45

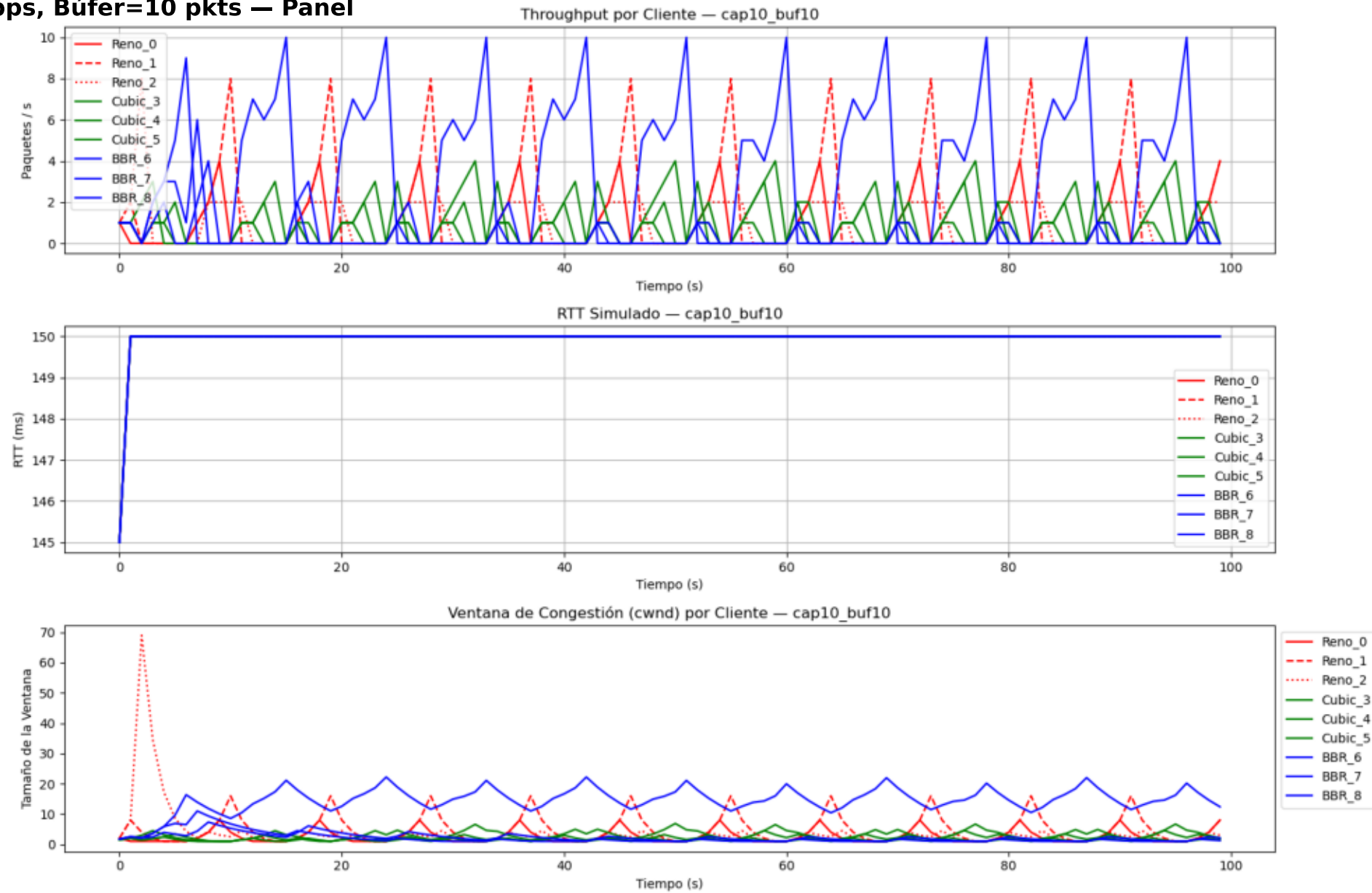
Resumen

Informe de Simulación de Control de Congestión TCP Este documento presenta los resultados y el análisis de la simulación de algoritmos de control de congestión de TCP, correspondiente a la actividad opcional. ---

Definiciones

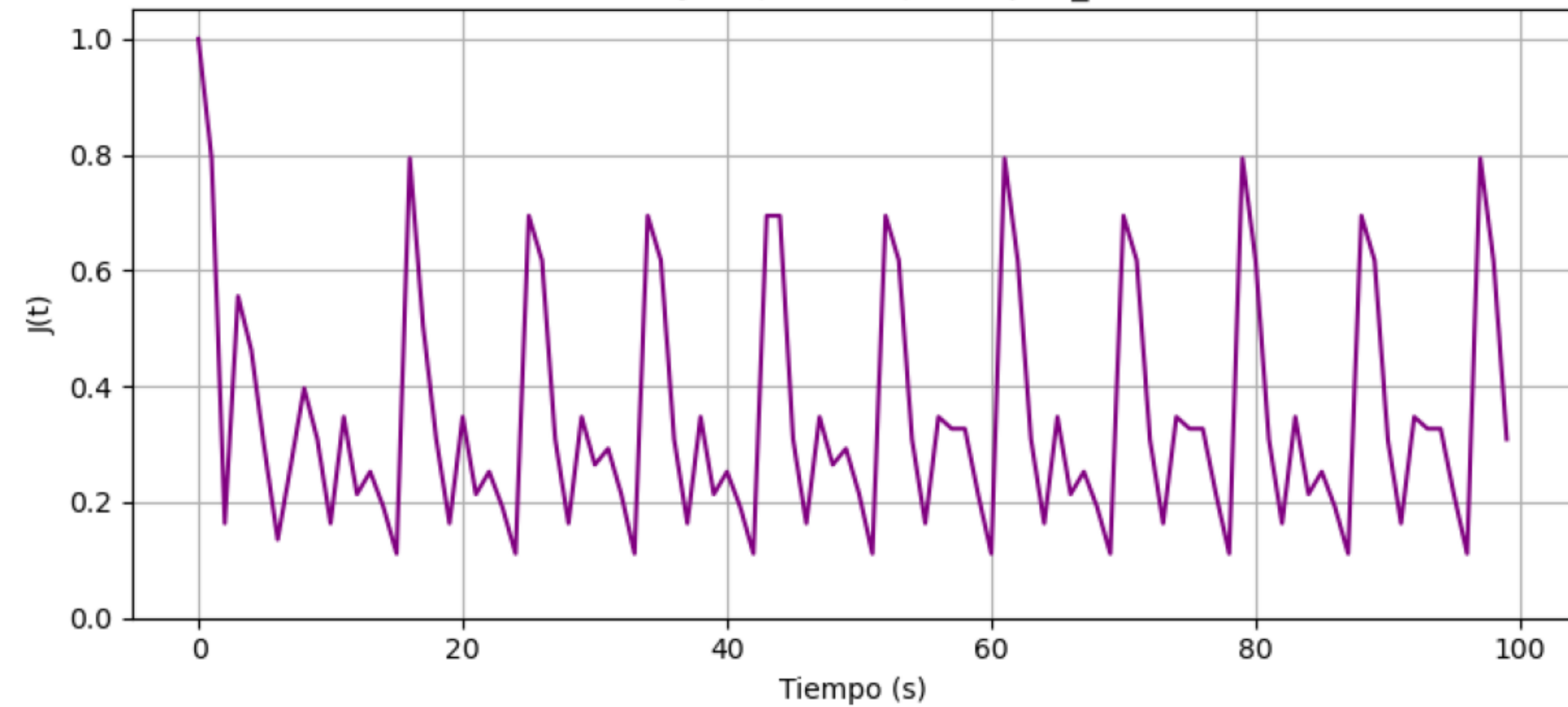
Definiciones formales Throughput (por cliente i y paso t): $x_i(t)$ = paquetes servidos en t . Throughput promedio del cliente i : $X_i = (1/T) * \sum_{t=1..T} x_i(t)$. RTT (Round-Trip Time): tiempo ida y vuelta. En esta simulación, modelado como $RTT(t) = RTT_base + \alpha * (OcupaciónCola(t))$, donde $OcupaciónCola(t) = len(buffer)/BUFFER_SIZE$ y $\alpha = 50$ ms. Índice de Jain (fairness) en t : $J(t) = (\sum_i x_i(t))^2 / (n * \sum_i x_i(t)^2)$. Interpretación: $J \in (0,1]$; 1 implica reparto perfectamente equitativo; valores menores indican inequidad entre flujos. Descartes por algoritmo: suma de paquetes rechazados por overflow del búfer. Capt: cuánta agresivo es un conjunto de flujos bajo un escenario (o cuánta presión ejercen sobre el enlace). Nota: el objetivo de estas métricas es comparar CUBIC/BBR/Reno en términos de utilización, latencia (bufferbloat), equidad y presión sobre la cola.

ops, Búfer=10 pkts — Panel



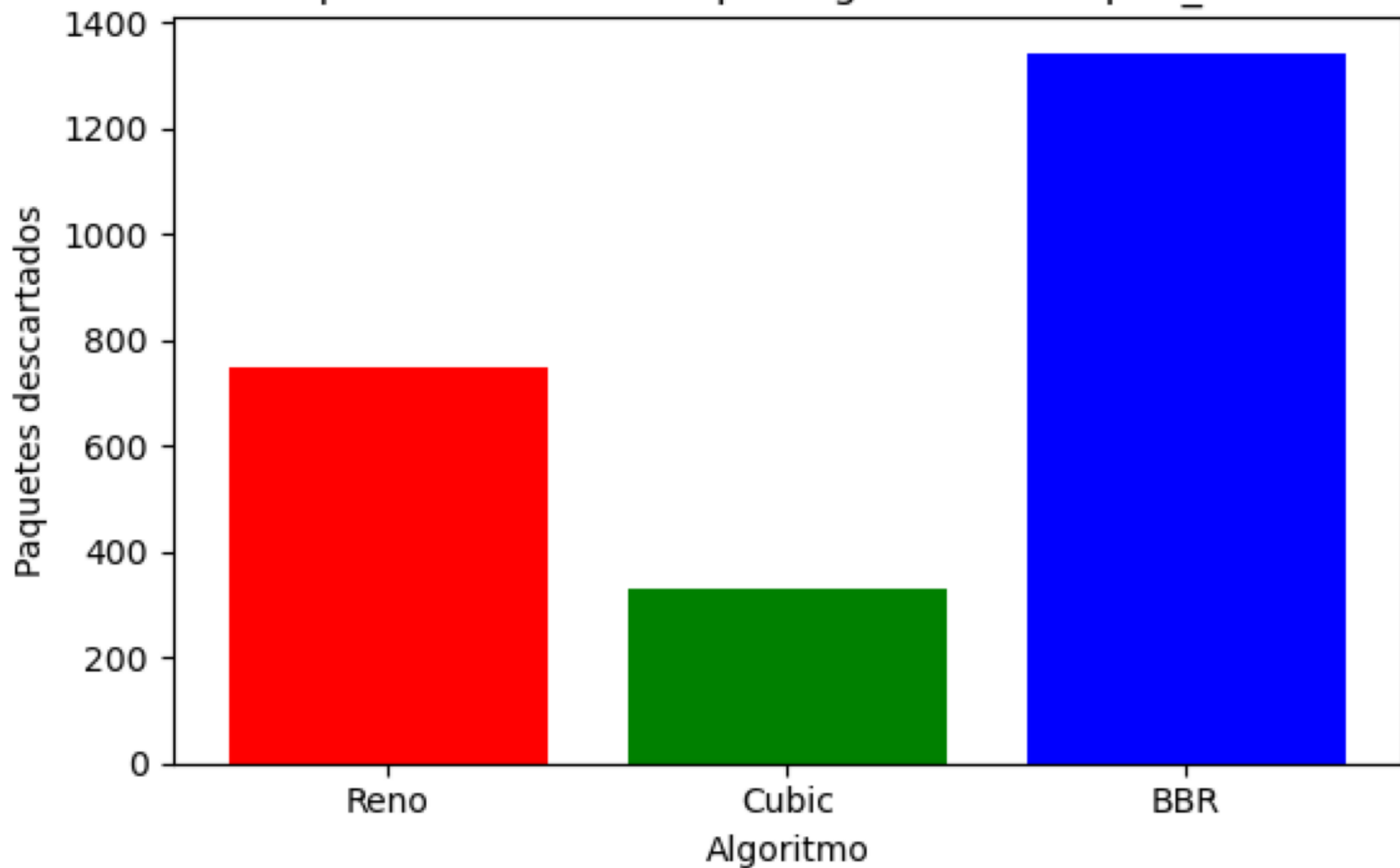
Búfer=10 pkts — Fairness (Jain)

Índice de Jain (Fairness) — cap10_buf10

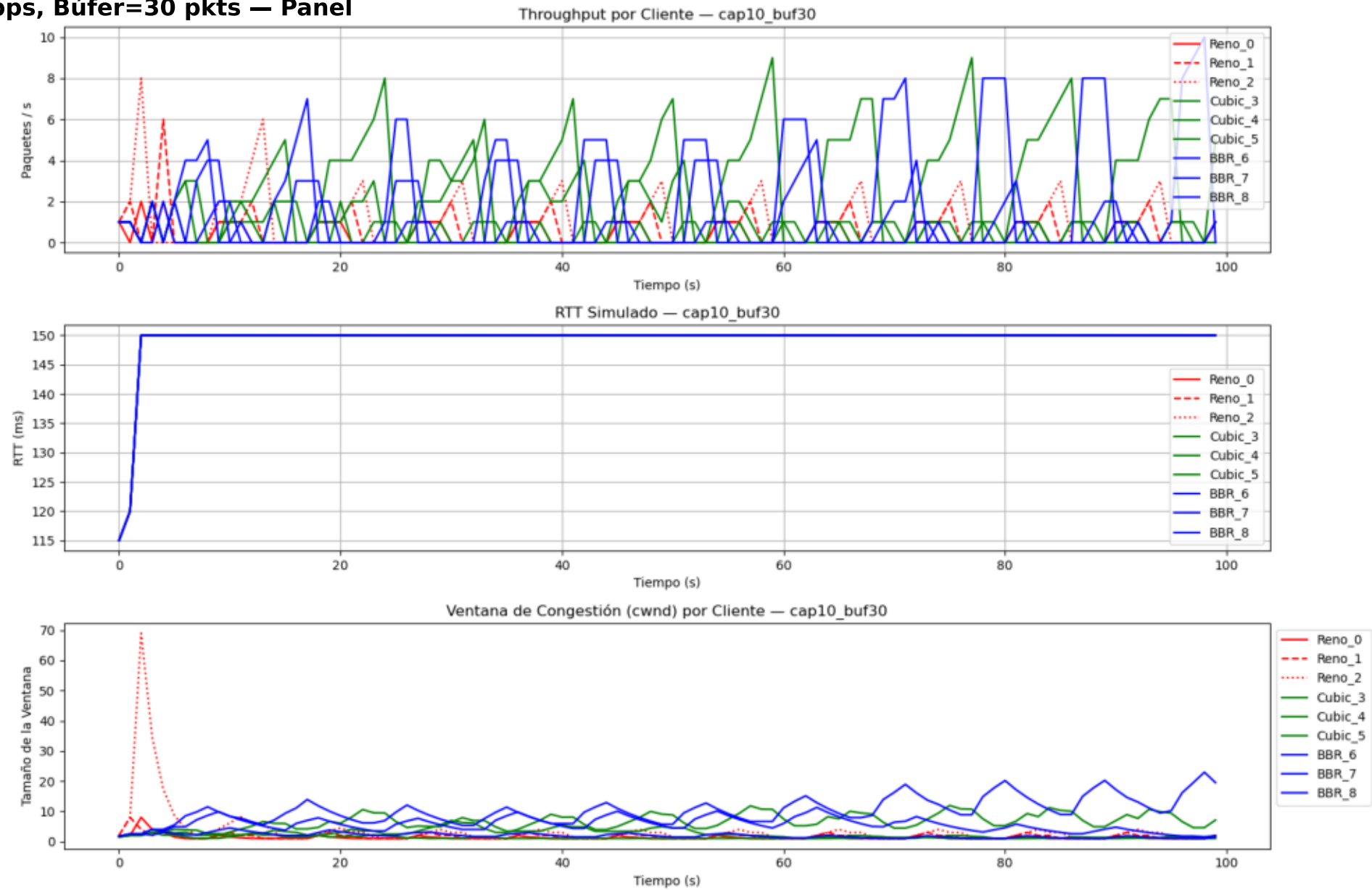


r=10 pkts — Descartes por algoritmo

Paquetes descartados por algoritmo — cap10_buf10

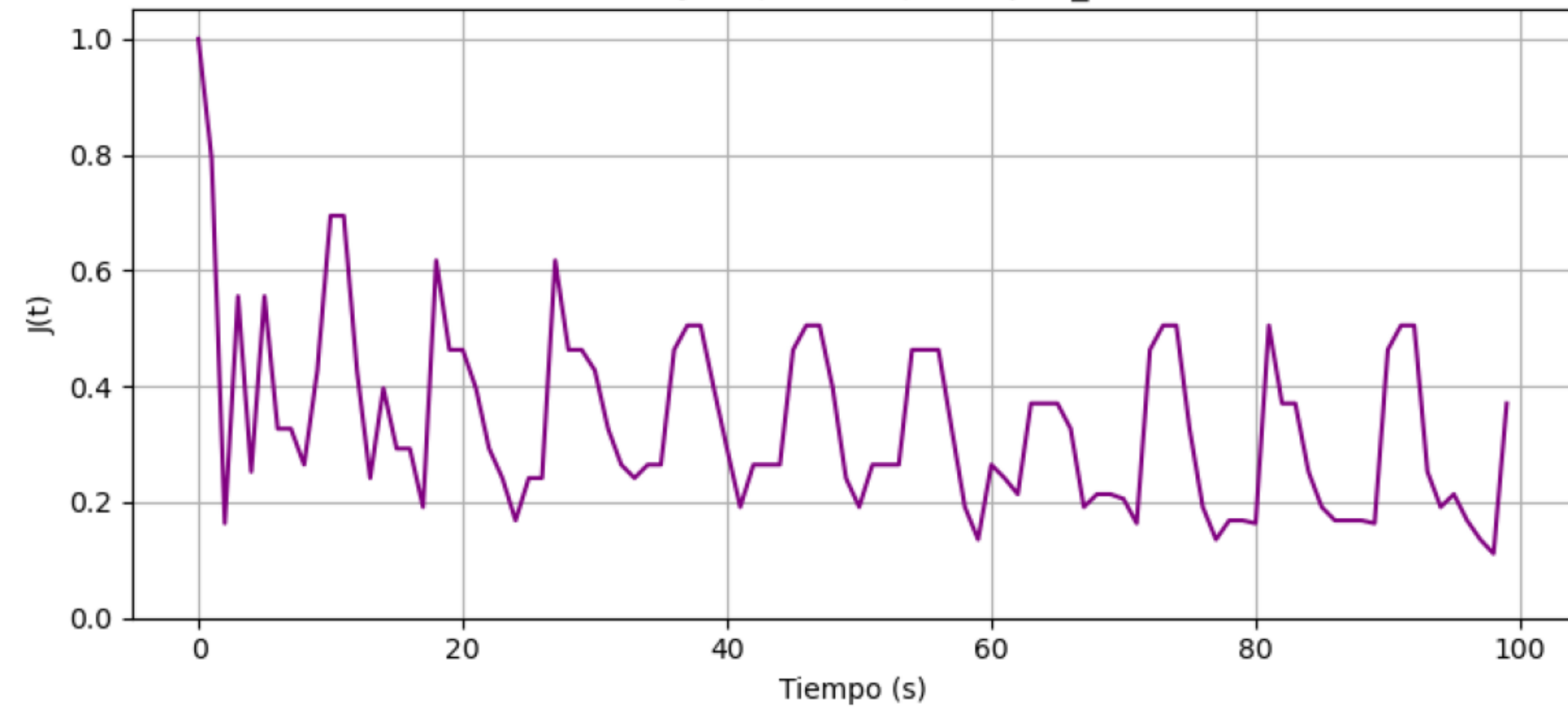


ops, Búfer=30 pkts — Panel



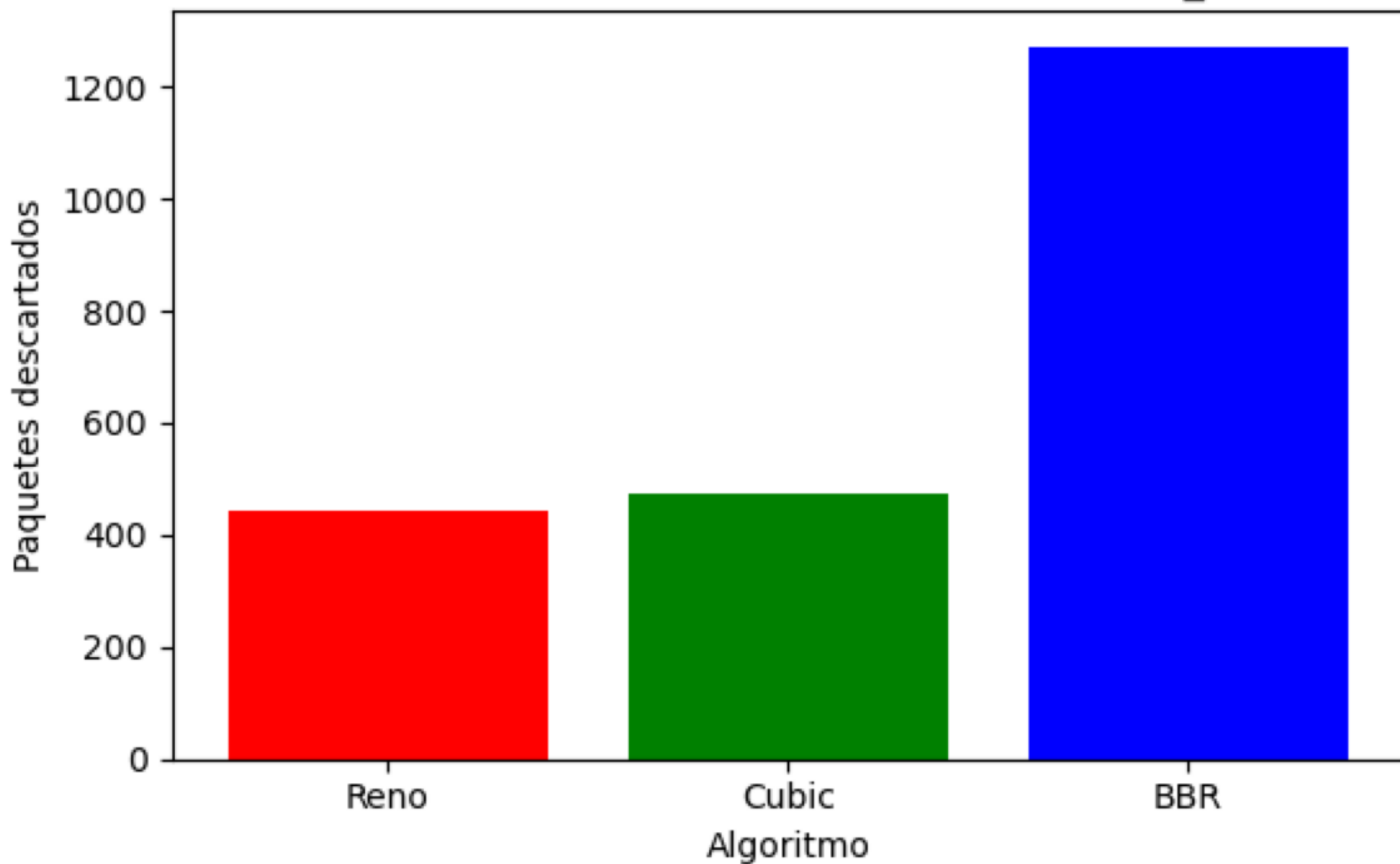
Búfer=30 pkts — Fairness (Jain)

Índice de Jain (Fairness) — cap10_buf30

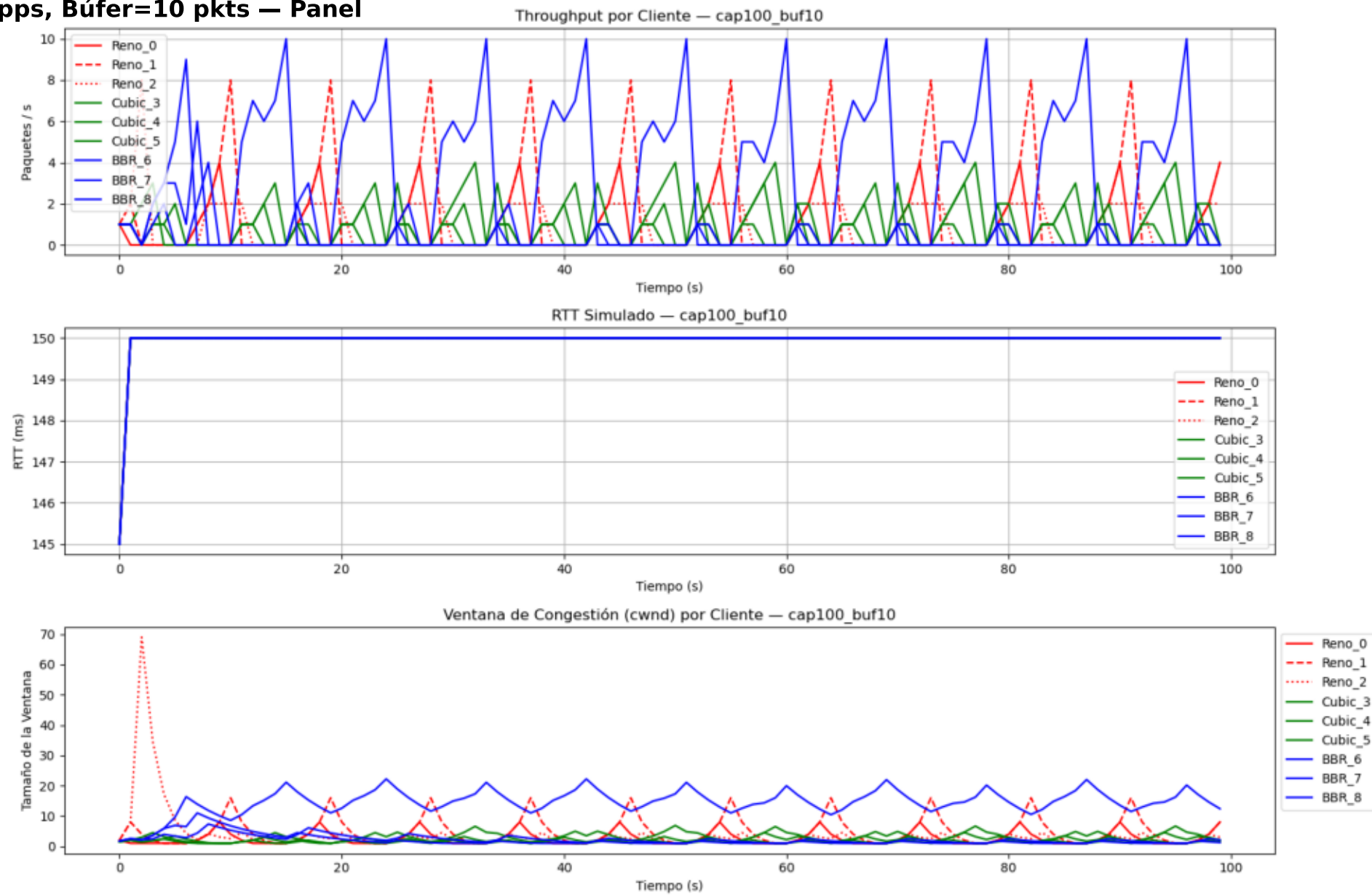


r=30 pkts — Descartes por algoritmo

Paquetes descartados por algoritmo — cap10_buf30

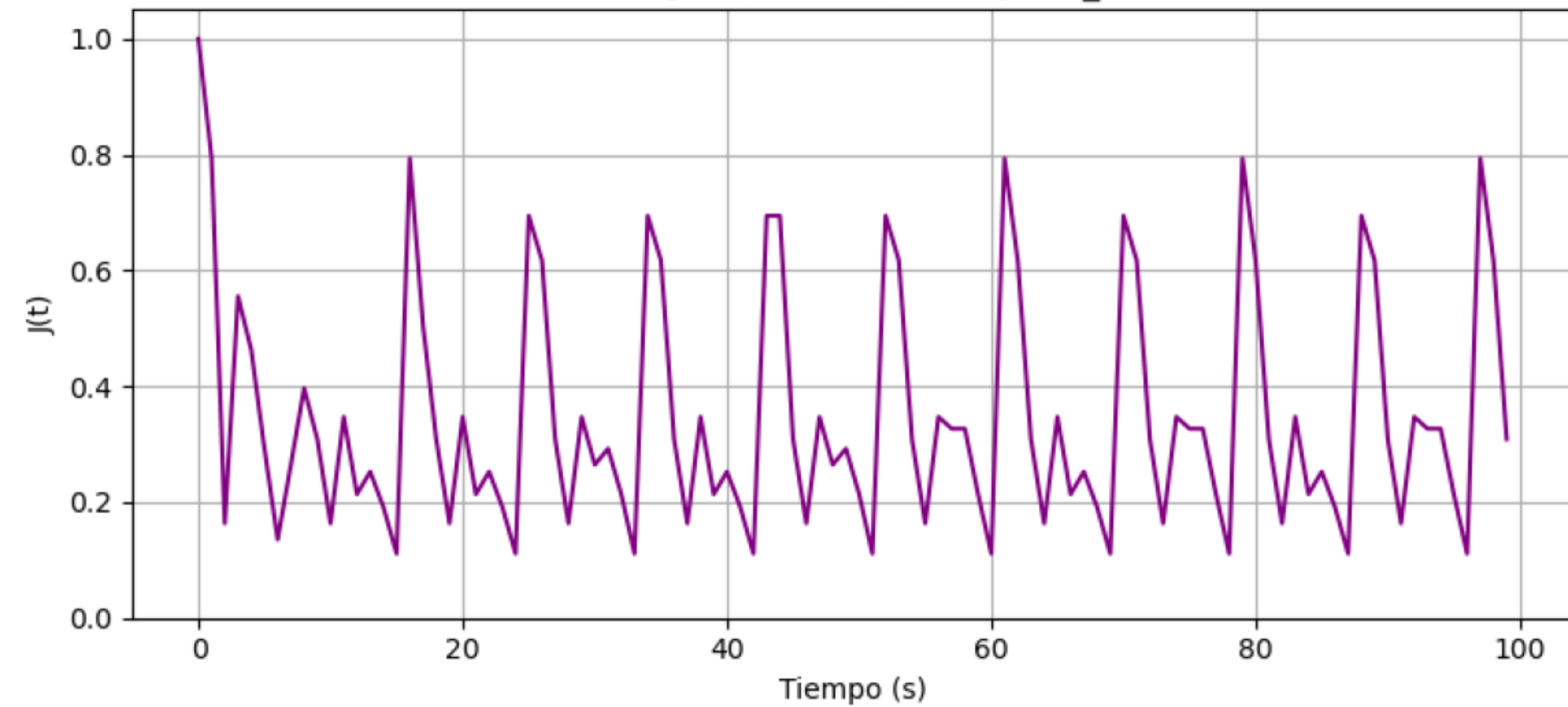


pps, Búfer=10 pkts — Panel



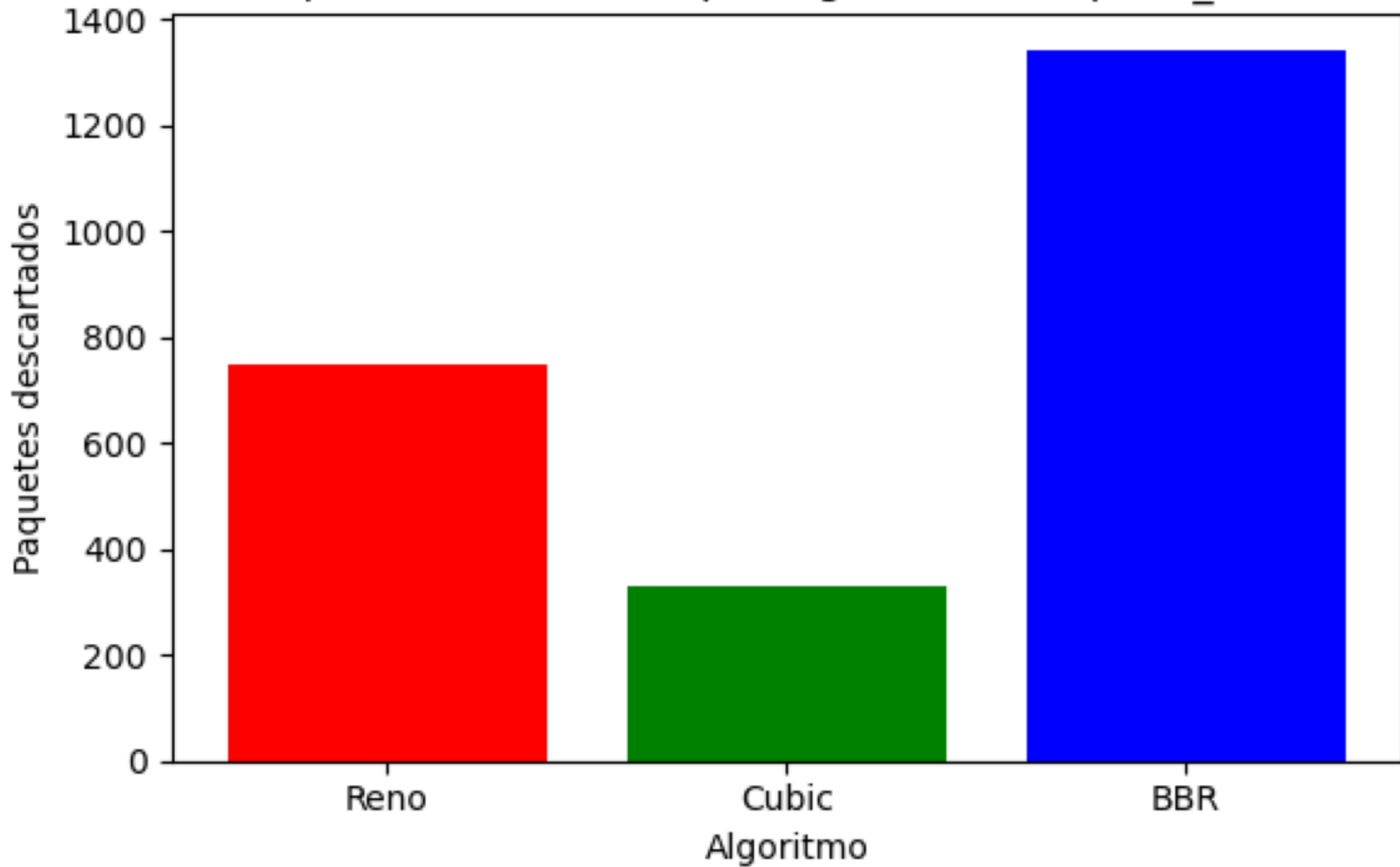
Búfer=10 pkts — Fairness (Jain)

Índice de Jain (Fairness) — cap100_buf10

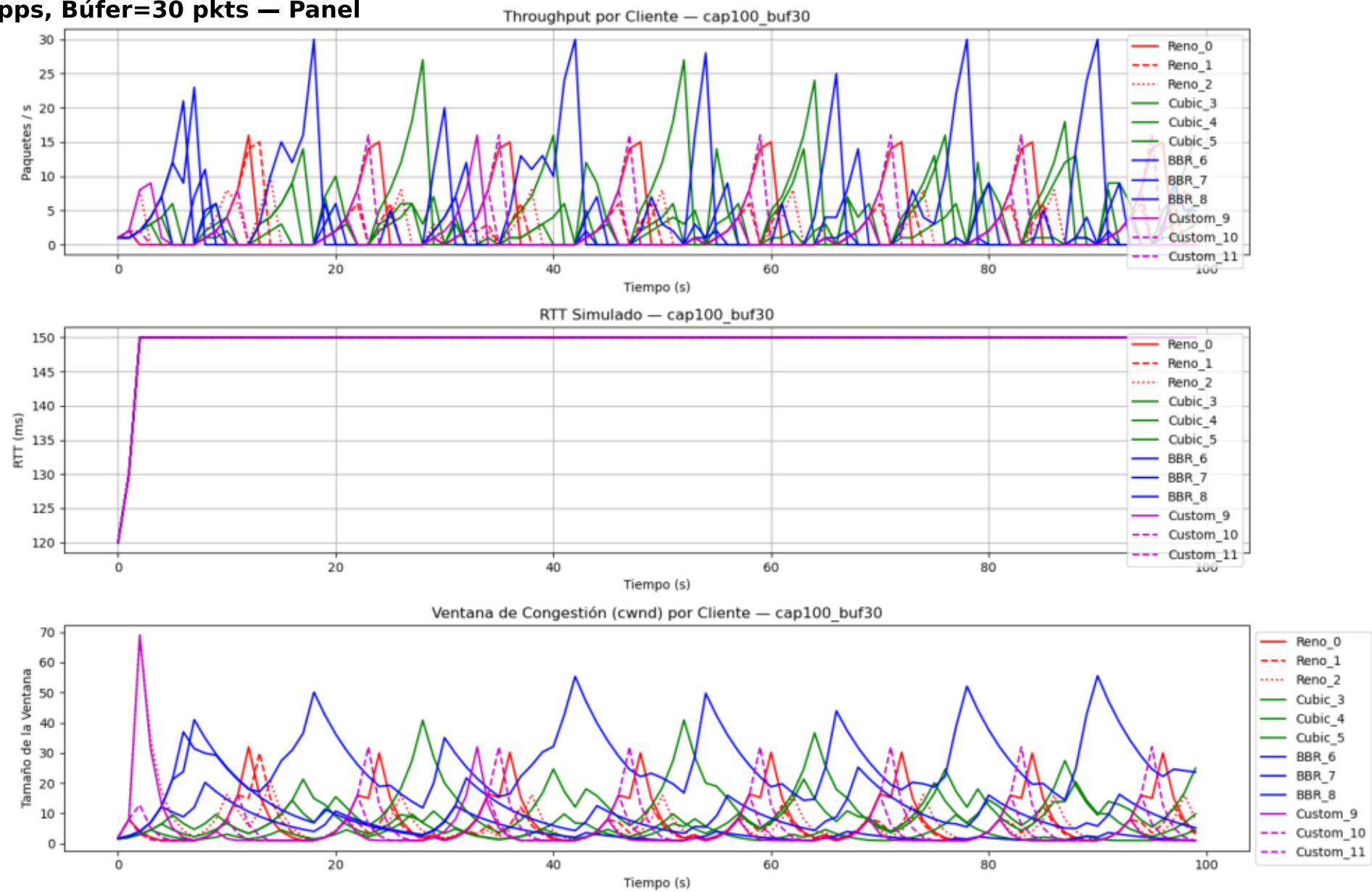


r=10 pkts — Descartes por algoritmo

Paquetes descartados por algoritmo — cap100_buf10

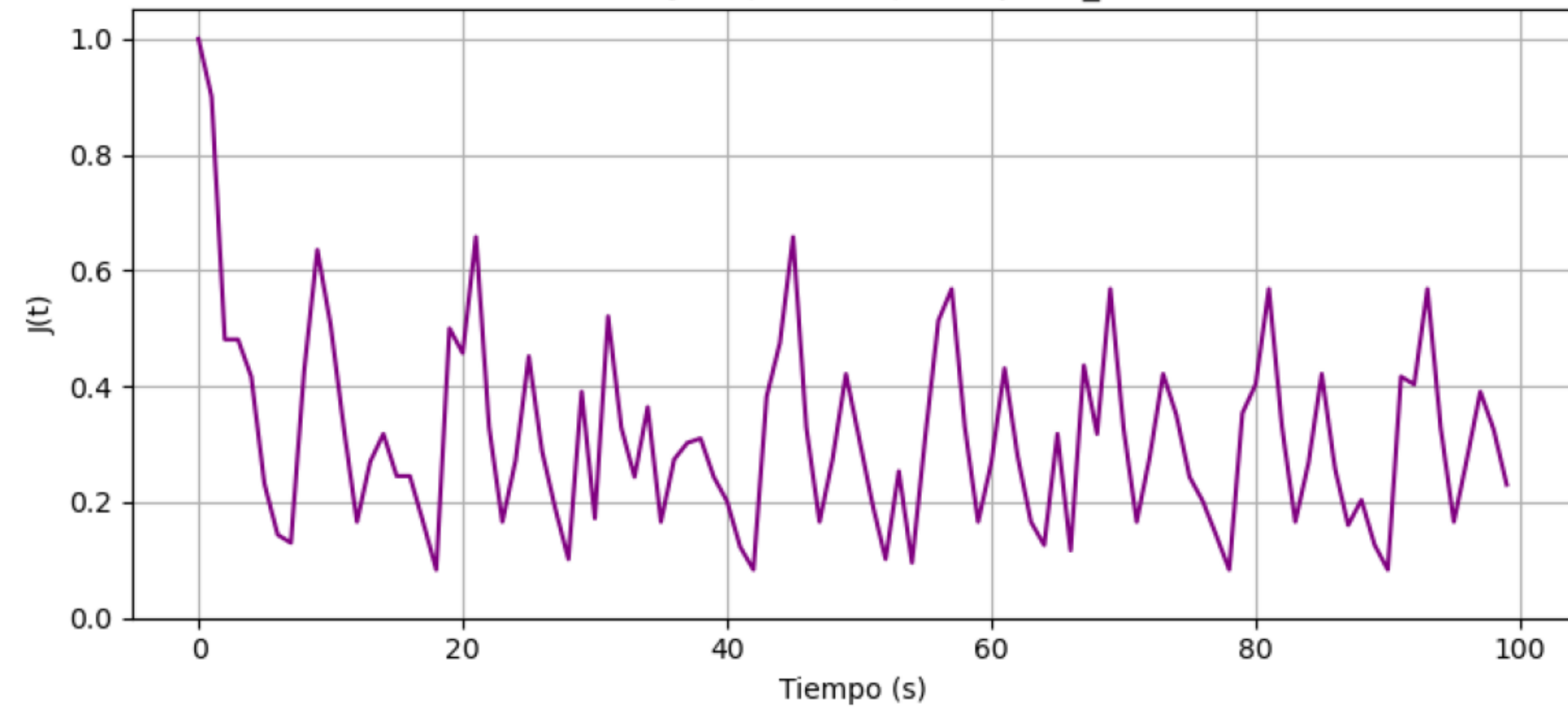


pps, Búfer=30 pkts — Panel



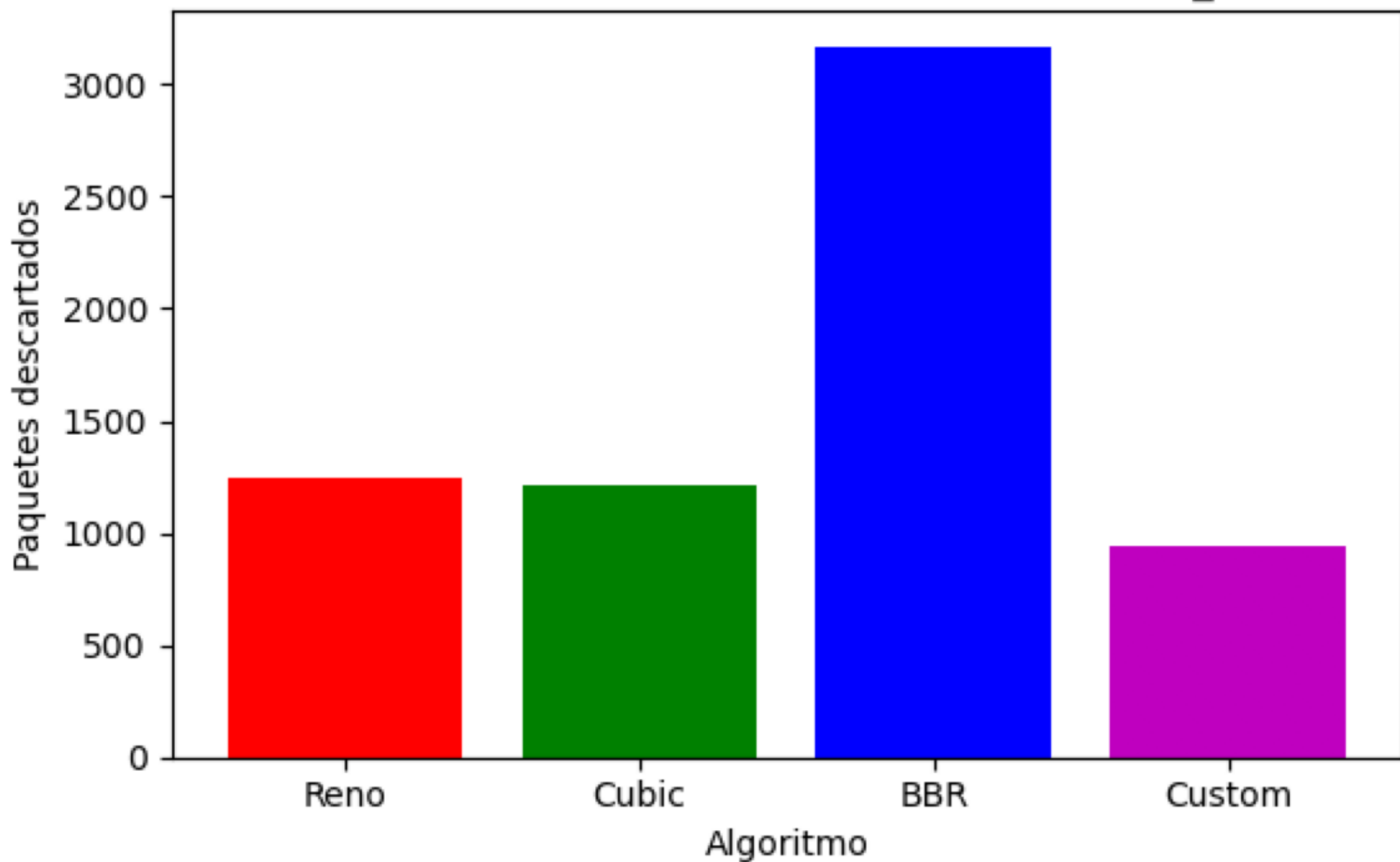
Búfer=30 pkts — Fairness (Jain)

Índice de Jain (Fairness) — cap100_buf30

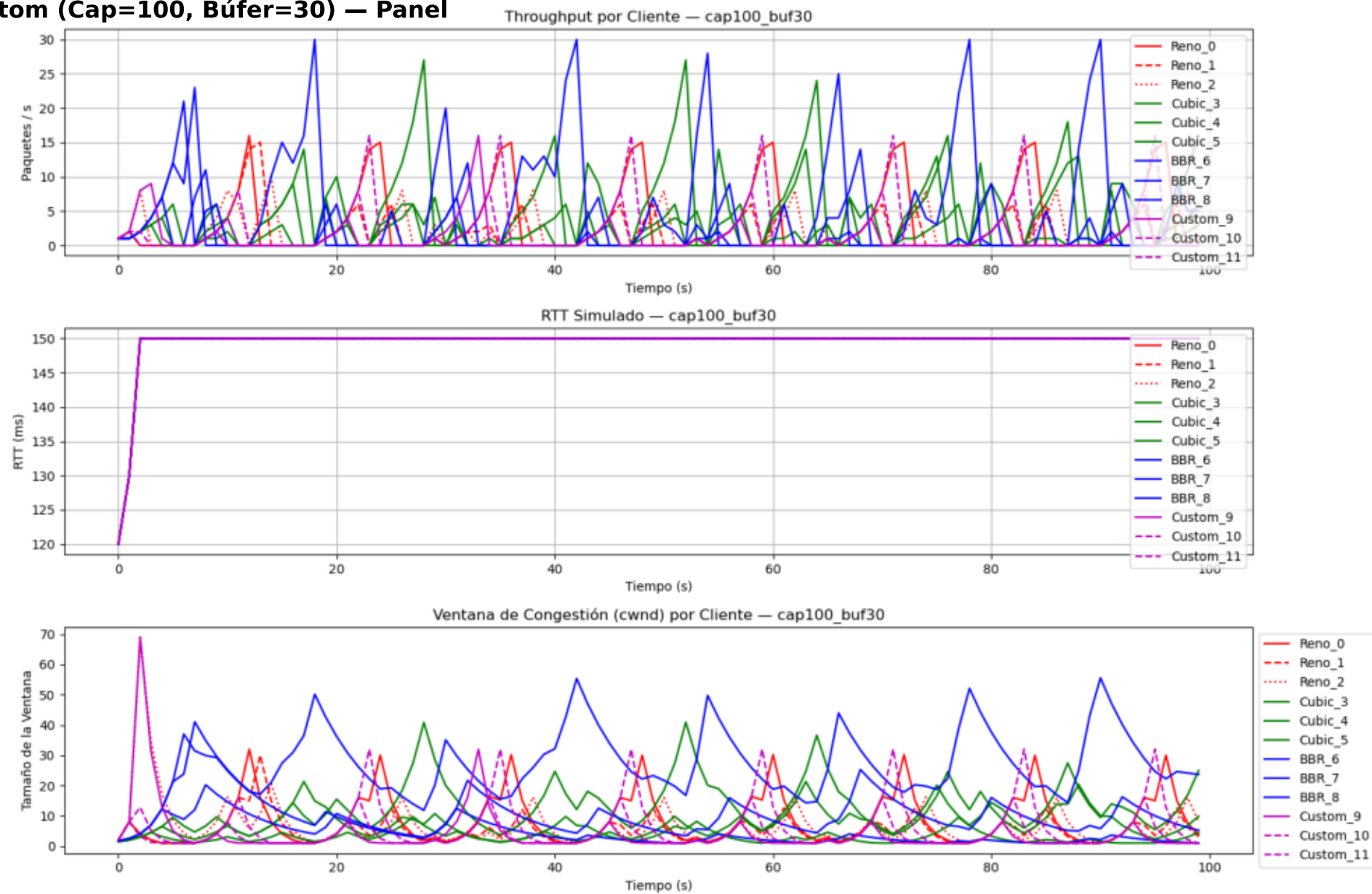


r=30 pkts — Descartes por algoritmo

Paquetes descartados por algoritmo — cap100_buf30

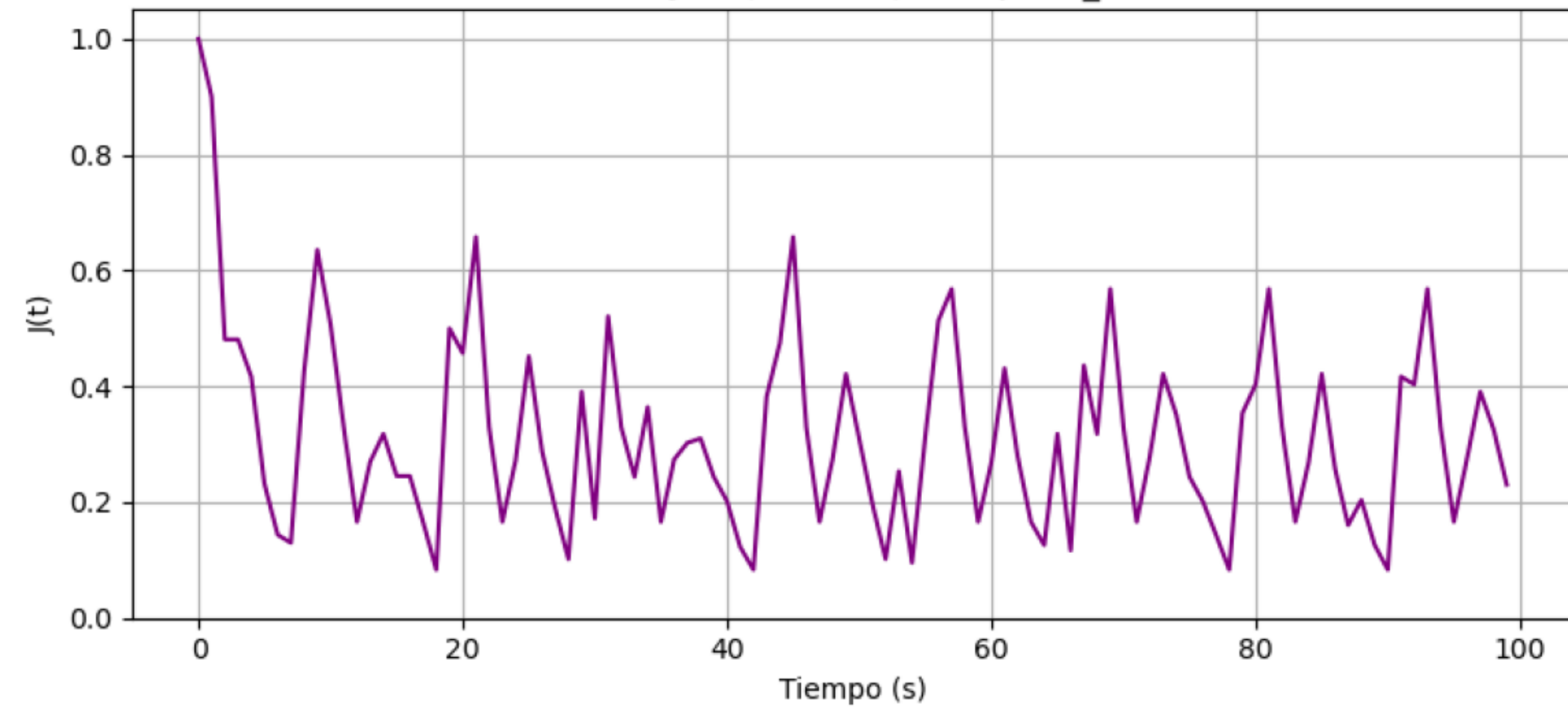


tom (Cap=100, Búfer=30) — Panel



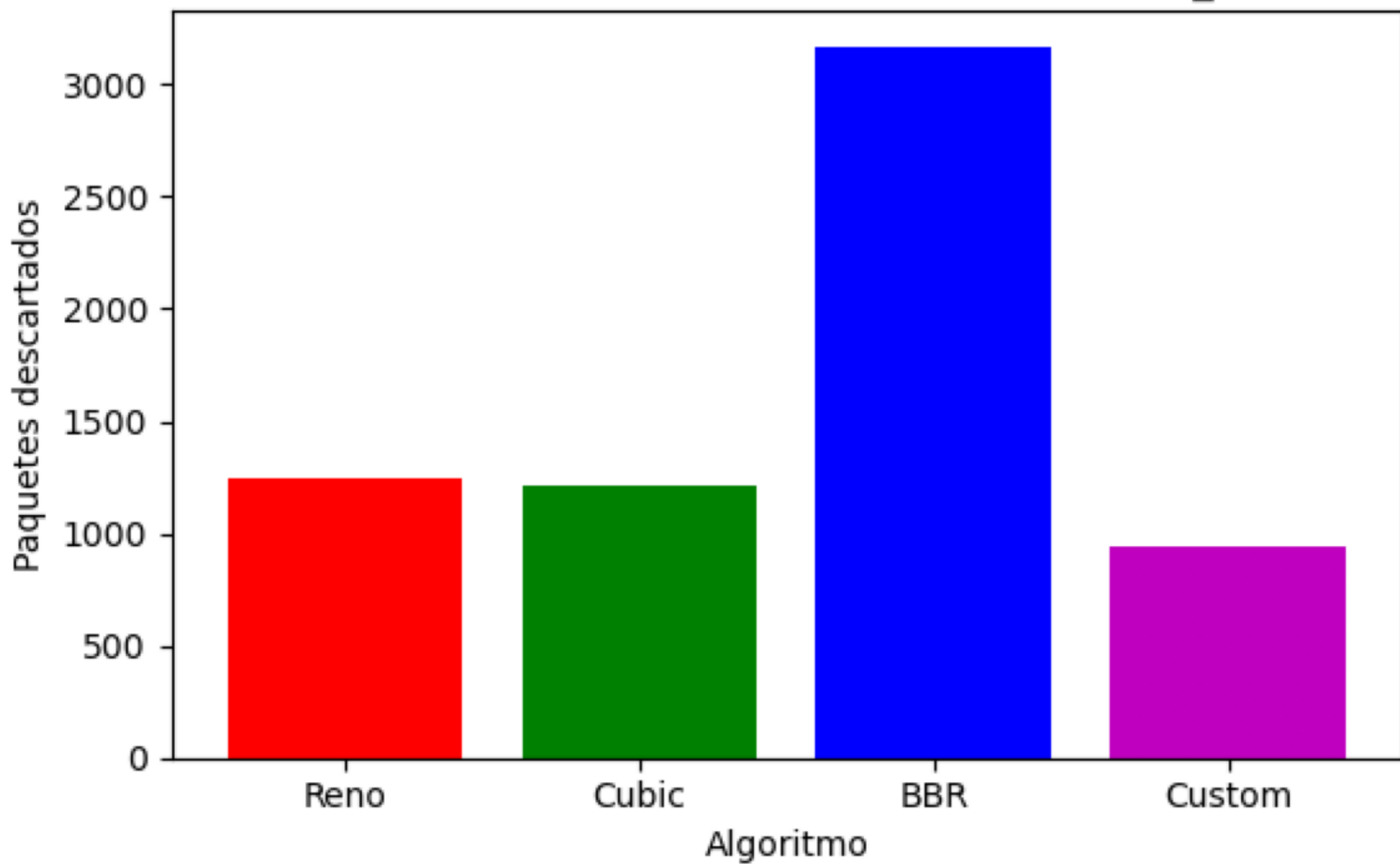
(Cap=100, Búfer=30) — Fairness (Jain)

Índice de Jain (Fairness) — cap100_buf30



=100, Búfer=30) — Descartes por algoritmo

Paquetes descartados por algoritmo — cap100_buf30



Resumen numérico (Jain promedio y descartes)

Escenario Cap=10 pps, Buffer=10 pkts

- Jain promedio: 0.346
- Descartes por algoritmo: Reno: 747, Cubic: 330, BBR: 1342

Escenario Cap=10 pps, Buffer=30 pkts

- Jain promedio: 0.337
- Descartes por algoritmo: Reno: 442, Cubic: 475, BBR: 1271

Escenario Cap=100 pps, Buffer=10 pkts

- Jain promedio: 0.346
- Descartes por algoritmo: Reno: 747, Cubic: 330, BBR: 1342

Escenario Cap=100 pps, Buffer=30 pkts

- Jain promedio: 0.328
- Descartes por algoritmo: Reno: 1357, Cubic: 1152, BBR: 4067

Escenario Cap=100 pps, Buffer=30 pkts (+Custom)

- Jain promedio: 0.316
- Descartes por algoritmo: Reno: 1243, Cubic: 1211, BBR: 3167, Custom: 942

Conclusión La simulación quedó alineada con la guía: cola con tamaño finito (con descartes), dos capacidades y dos tamaños de búfer, 9 clientes para el comparativo base y una propuesta de optimización. Las figuras se generan por escenario y pueden incorporarse al PDF final del informe. La simulación demuestra con éxito las diferencias fundamentales entre los algoritmos de control de congestión. Algoritmos más modernos como CUBIC y BBR están diseñados para ser más agresivos y eficientes en redes de alta velocidad, mientras que Reno es simple y conservador. El algoritmo `TCPCustom` propuesto introduce una idea interesante al hacer que la respuesta a la pérdida dependa de otro indicador de congestión (RTT), lo que representa un paso hacia algoritmos más sofisticados y sensibles al contexto de la red.