Real-World ARQ Performance: TCP

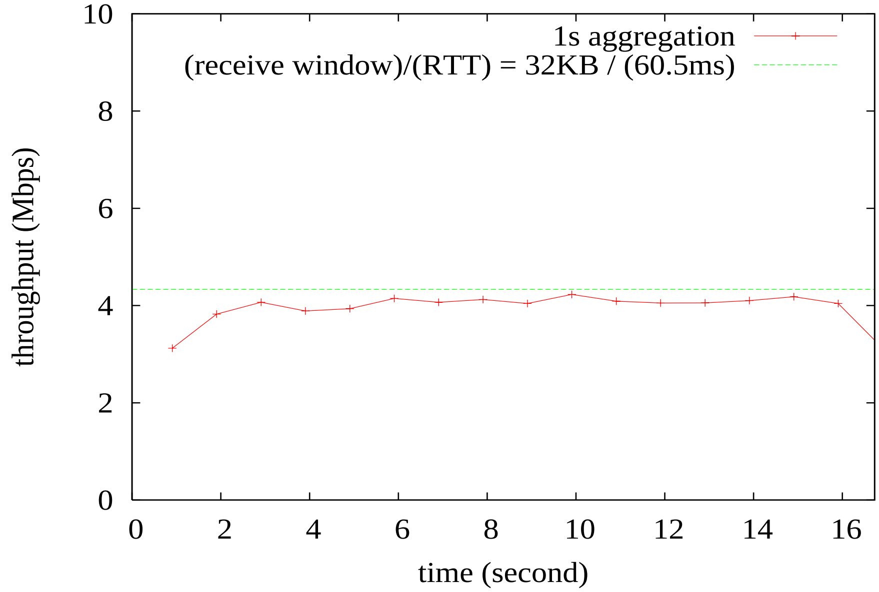
Ex.: Purdue $\rightarrow$ UCSD

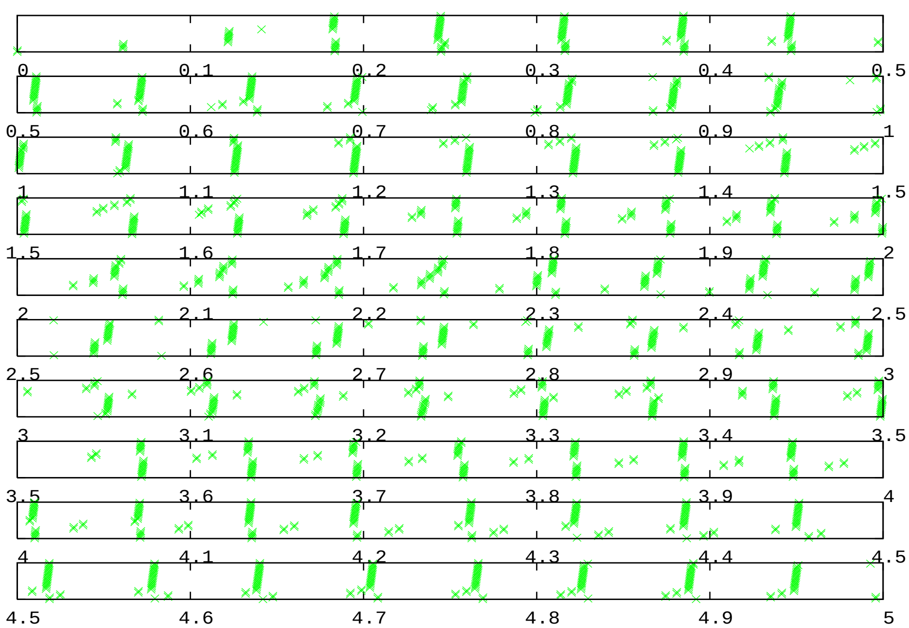
- Purdue (NSL): web server
- UCSD: web client

```
traceroute to planetlab3.ucsd.edu (132.239.17.226), 30 hops max, 40 byte packets
 1  switch-lwsn2133-z1r11 (128.10.27.250)  0.483 ms  0.344 ms  0.362 ms
 2  lwsn-b143-c6506-01-tcom (128.10.127.251)  0.488 ms  0.489 ms  0.489 ms
 3  172.19.57.1 (172.19.57.1)  0.486 ms  0.488 ms  0.489 ms
 4  tel-210-m10i-01-campus.tcom.purdue.edu (192.5.40.54)  0.614 ms  0.617 ms  0.615 ms
 5  gigapop.tcom.purdue.edu (192.5.40.134)  1.743 ms  1.679 ms  1.727 ms
 6  * * *
 7  * * *
 8  * * *
 9  hpr-lax-hpr--nlr-packenet.cenic.net (137.164.26.130)  56.919 ms  56.919 ms  57.658 ms
10  hpr-ucsd-10ge--lax-hpr.cenic.net (137.164.27.165)  60.326 ms  60.198 ms  60.196 ms
11  nodeb-720--ucsd-t320-gw-10ge.ucsd.edu (132.239.255.132)  60.326 ms  60.370 ms  75.130 ms
```

$\longrightarrow$ RTT $\approx$ 60.5 msec

$\longrightarrow$ receiver window size: 32 KB


1s aggregation
(receive window)/(RTT) = 32KB / (60.5ms)
throughput (Mbps)
time (second)
0
2
4
6
8
10
12
14
16



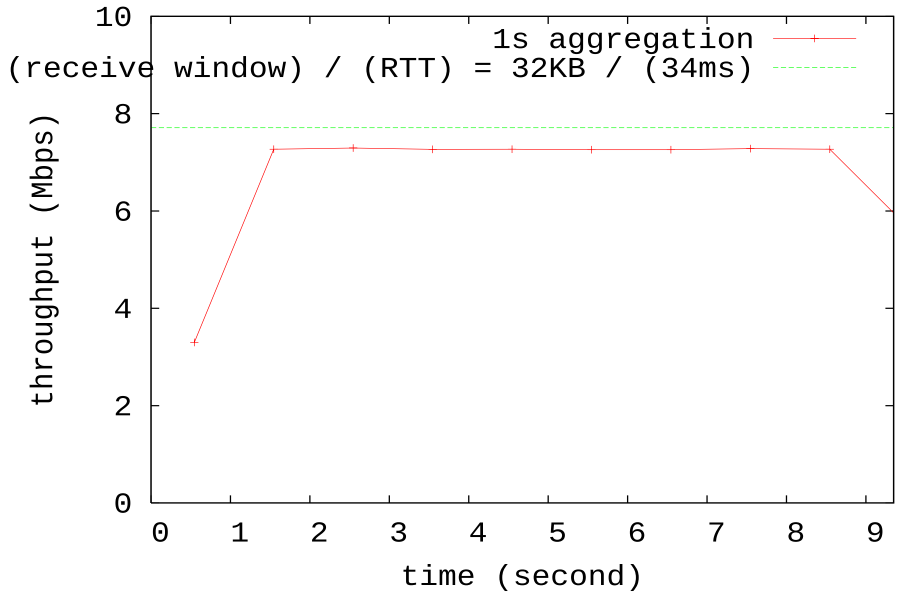
Ex.: Purdue $\rightarrow$ Rutgers

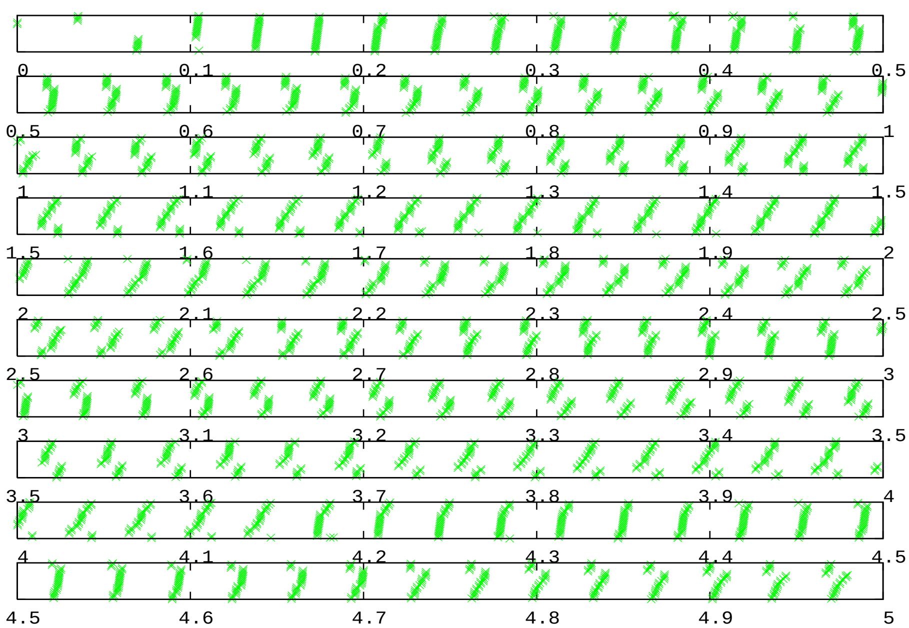
- Purdue: web server
- Rutgers: web client

```
traceroute to planetlab1.rutgers.edu (165.230.49.114), 30 hops max, 40 byte packets
 1  switch-lwsn2133-z1r11 (128.10.27.250)  12.336 ms  0.339 ms  0.362 ms
 2  lwsn-b143-c6506-01-tcom (128.10.127.251)  0.489 ms  0.491 ms  0.488 ms
 3  172.19.57.1 (172.19.57.1)  0.490 ms  0.488 ms  0.489 ms
 4  tel-210-m10i-01-campus.tcom.purdue.edu (192.5.40.54)  0.614 ms  0.615 ms  0.614 ms
 5  switch-data.tcom.purdue.edu (192.5.40.166)  2.864 ms  2.865 ms  2.864 ms
 6  abilene-ul.indiana.gigapop.net (192.12.206.249)  2.988 ms  13.608 ms  3.113 ms
 7  chinng-iplsng.abilene.ucaid.edu (198.32.8.76)  6.740 ms  6.875 ms  6.859 ms
 8  ge-0-0-0.10.rtr.chic.net.internet2.edu (64.57.28.1)  7.113 ms  6.975 ms  6.986 ms
 9  so-3-0-0.0.rtr.wash.net.internet2.edu (64.57.28.13)  29.349 ms  24.086 ms  23.626 ms
10  ge-1-0-0.418.rtr.chic.net.internet2.edu (64.57.28.10)  44.786 ms  28.822 ms  28.839 ms
11  local.internet2.magpi.net (216.27.100.53)  30.723 ms  30.818 ms  30.744 ms
12  phl-02-09.backbone.magpi.net (216.27.100.229)  31.045 ms  36.644 ms  30.839 ms
13  remote.njedge.magpi.net (216.27.98.42)  33.221 ms  33.021 ms  33.087 ms
14  er01-hill-ext.runet.rutgers.net (198.151.130.233)  33.229 ms  33.207 ms  33.217 ms
```

$\longrightarrow$ RTT $\approx$ 34 msec

$\longrightarrow$ receiver window size: 32 KB


1s aggregation
(receive window) / (RTT) = 32KB / (34ms)
throughput (Mbps)
time (second)
0
2
4
6
8
10
0 1 2 3 4 5 6 7 8 9


0 0.1 0.2 0.3 0.4 0.5
0.5 0.6 0.7 0.8 0.9 1
1 1.1 1.2 1.3 1.4 1.5
1.5 1.6 1.7 1.8 1.9 2
2 2.1 2.2 2.3 2.4 2.5
2.5 2.6 2.7 2.8 2.9 3
3 3.1 3.2 3.3 3.4 3.5
3.5 3.6 3.7 3.8 3.9 4
4 4.1 4.2 4.3 4.4 4.5
4.5 4.6 4.7 4.8 4.9 5

Ex.: Purdue → Korea University (Seoul)

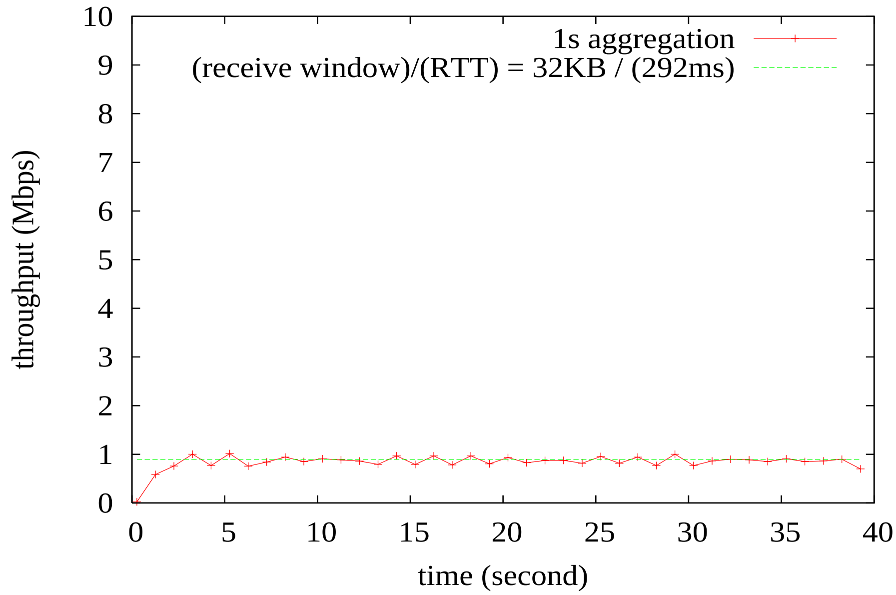
- Purdue: web server
- KU: web client

```
 1  switch-lwsn2133-z1r11 (128.10.27.250)  0.513 ms  10.061 ms  0.358 ms
 2  lwsn-b143-c6506-01-tcom (128.10.127.251)  0.487 ms  0.476 ms  0.364 ms
 3  172.19.57.1 (172.19.57.1)  0.489 ms  0.475 ms  0.490 ms
 4  tel-210-m10i-01-campus.tcom.purdue.edu (192.5.40.54)  0.613 ms  0.600 ms  0.614 ms
 5  switch-data.tcom.purdue.edu (192.5.40.166)  7.982 ms  7.969 ms  14.596 ms
 6  abilene-ul.indiana.gigapop.net (192.12.206.249)  8.977 ms  7.721 ms  6.857 ms
 7  kscyng-iplsng.abilene.ucaid.edu (198.32.8.81)  36.860 ms  25.873 ms  29.075 ms
 8  dnvrng-kscyng.abilene.ucaid.edu (198.32.8.13)  24.218 ms  23.125 ms  36.317 ms
 9  snvang-dnvrng.abilene.ucaid.edu (198.32.8.1)  47.815 ms  78.440 ms  54.048 ms
10  losang-snvang.abilene.ucaid.edu (198.32.8.94)  55.080 ms  55.131 ms  60.674 ms
11  transpac-1-lo-jmb-702.lsanca.pacificwave.net (207.231.240.136)  55.165 ms  55.212 ms  59.1
12  tokyo-losa-tp2.transpac2.net (192.203.116.146)  175.068 ms  170.832 ms  170.444 ms
13  tyo-gate1.jp.apan.net (203.181.248.249)  170.488 ms  170.893 ms  171.818 ms
14  sg-so-02-622m.bb-v4.noc.tein2.net (202.179.249.5)  277.150 ms  275.966 ms  276.136 ms
15  kr.pr-v4.noc.tein2.net (202.179.249.18)  278.422 ms  276.486 ms  280.132 ms
16  61.252.48.182 (61.252.48.182)  276.170 ms  279.606 ms  279.421 ms
17  202.30.43.45 (202.30.43.45)  271.663 ms  269.492 ms  268.761 ms
18  honeungl3-seoul.kreonet.net (134.75.120.2)  269.781 ms  269.913 ms  273.516 ms
19  203.241.173.74 (203.241.173.74)  272.663 ms  278.774 ms  270.902 ms
```

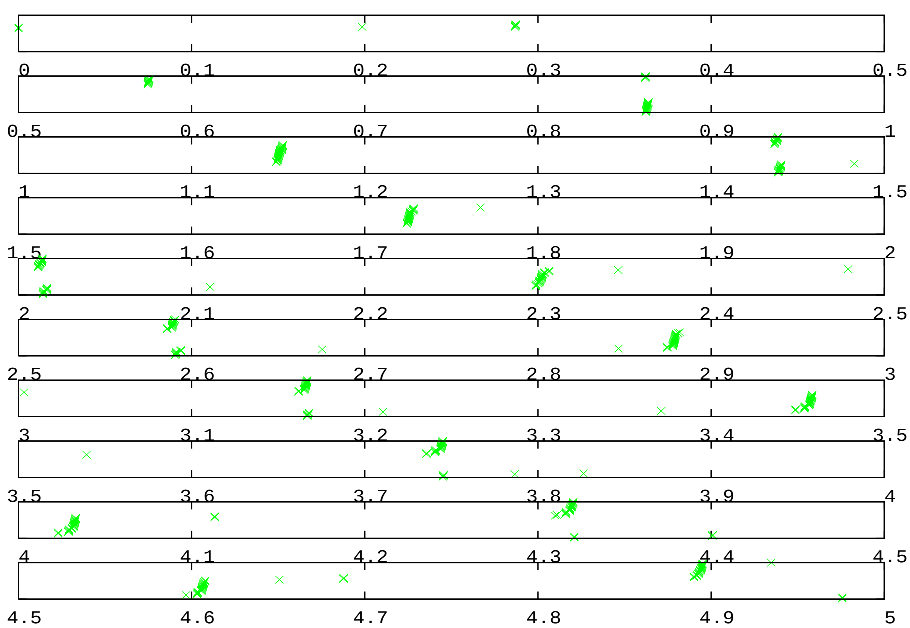
$\longrightarrow$ RTT $\approx$ 292 msec

$\longrightarrow$ long route to Korea (via Singapore)

$\longrightarrow$ receiver window size: 32 KB



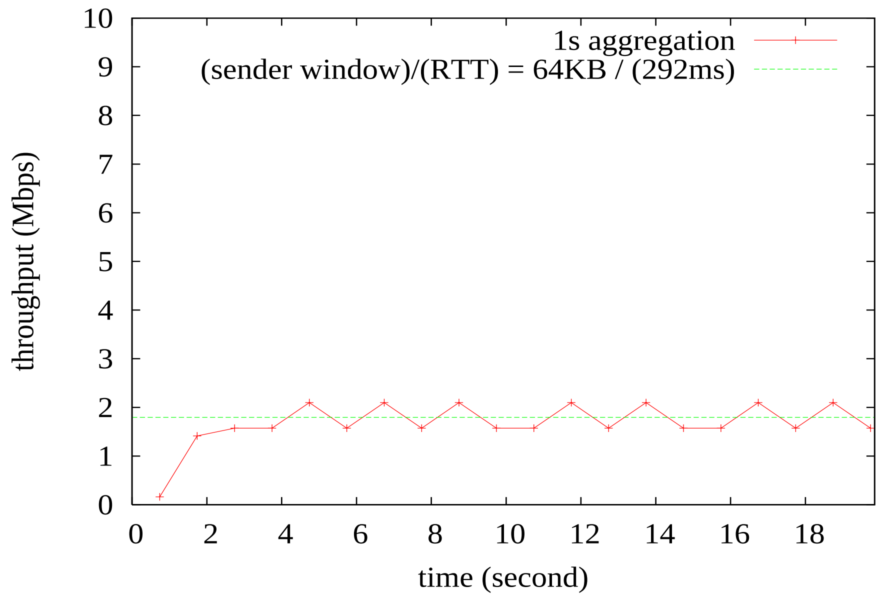
throughput (Mbps)
time (second)
1s aggregation
(receive window)/(RTT) = 32KB / (292ms)
0
5
10
15
20
25
30
35
40
1
2
3
4
5
6
7
8
9
10



0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8
2.9
3
3.1
3.2
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
4.1
4.2
4.3
4.4
4.5
4.6
4.7
4.8
4.9
5

Increase receiver window size: 128 KB

$\longrightarrow$ 4-fold increase

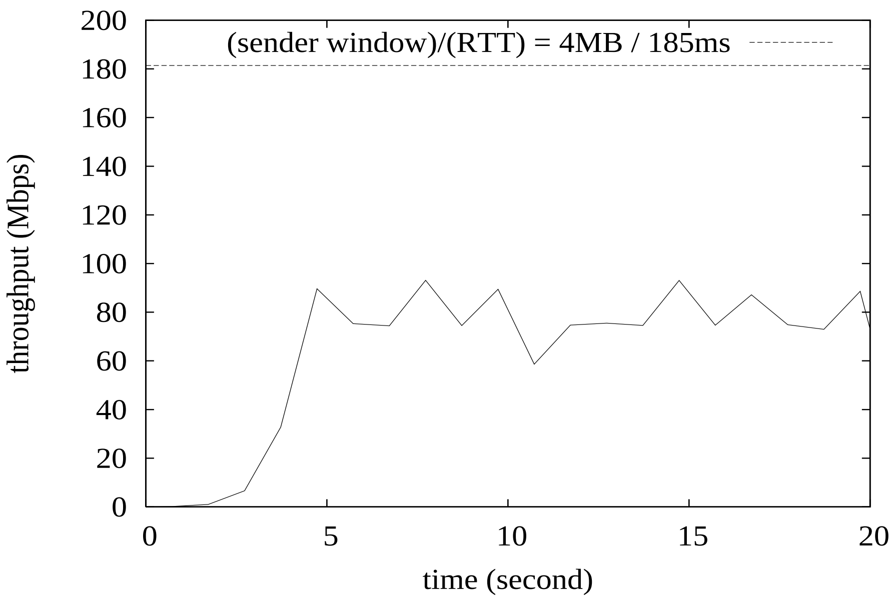


$\longrightarrow$ why only 2-fold throughput increase?

Increase receiver window size: 8 MB

$\longrightarrow$ also increase sender buffer size to 4 MB

$\longrightarrow$ RTT $\approx$ 185 msec (short route to Korea)



$\longrightarrow$ around 90 Mbps

$\longrightarrow$ download time for 10 MB file?

$\longrightarrow$ can be confused with DoS (denial-of-service) attack

$\longrightarrow$ why less than 180 Mbps?

Link Layer: Wired Media

Multi-Access Communication

Bandwidth sharing: two approaches

- contention-free

  → e.g., TDMA, FDMA, mixture of FDMA + TDMA, CDMA, SDMA, token ring

  → used in telephony and broadband data networks

- contention-based

  → e.g., CSMA/CD, CSMA/CA

  → used in Ethernet, WLAN

Also called MAC (medium access control)

→ 48-bit NIC address: MAC address

- broadband: FDMA, FDMA + TDMA, CDMA, SDMA
- baseband: TDMA, contention-based multiple access, token ring

Contention-free MAC:

→ how to keep discussion orderly?

→ moderator

→ i.e., prior allocation of bandwidth

→ centralized

Contention-based MAC:

→ single carrier frequency shared by many

→ no moderator

→ much less structured

→ decentralized

→ interference between conversations

→ i.e., collision

Basic features:

- To send data, just send: multiple access (MA)

  → if link seems idle, just send: carrier sense (CS)

- If two or more users send at the same time, data can become junk (i.e., bit flips)

  → collision

  → collision need not always cause bit flips

  → sufficiently stronger signal packet wins: capture effect

Can get pretty "chaotic"

→ used in real systems?

→ pure MA (no CS): ALOHA (1970s)

→ packet network connecting Univ. of Hawaii campuses

→ ∼30 years before boom of wireless data networks

→ today's technology: hasn't changed much

Additional features (optional):

- NIC can detect if someone else is using the channel

  → called carrier sense (CS)

  → rule: if someone else talks, don't talk

  → impose gentlemanly (i.e., cooperative) behavior

  → not always practically feasible

- NIC can detect if collision has occurred

  → called collision detection (CD)

  → not always technically feasible

In real systems:

⟶ Ethernet: MA, CS + CD

⟶ WLAN: MA, CS + CA

Why not just use TDMA, FDMA, or CDMA?

Benefits of contention-based MAC:

- when not too many users, faster response time

  → minimal coordination overhead

  → e.g.: in TDMA need to request and reserve slots

- decentralized

  → minimal configuration overhead

  → to join: just send

  → except for security concerns (e.g., Purdue's PAL)

  → a good idea?

Drawbacks of contention-based MAC:

- when many users, degraded throughput

  → collision can waste slot bandwidth

- lack of QoS (quality of service) assurances

  → "you get what you get"; called best effort

  → problematic for real-time traffic, e.g., telephony

  → IEEE 802.11 WLAN has provisions to suppport telephony: not used in practice

When to use what?

⟶ contention-free vs. contention-based