

IO Core Refresh

Net module

ATS Summit Fall 2021

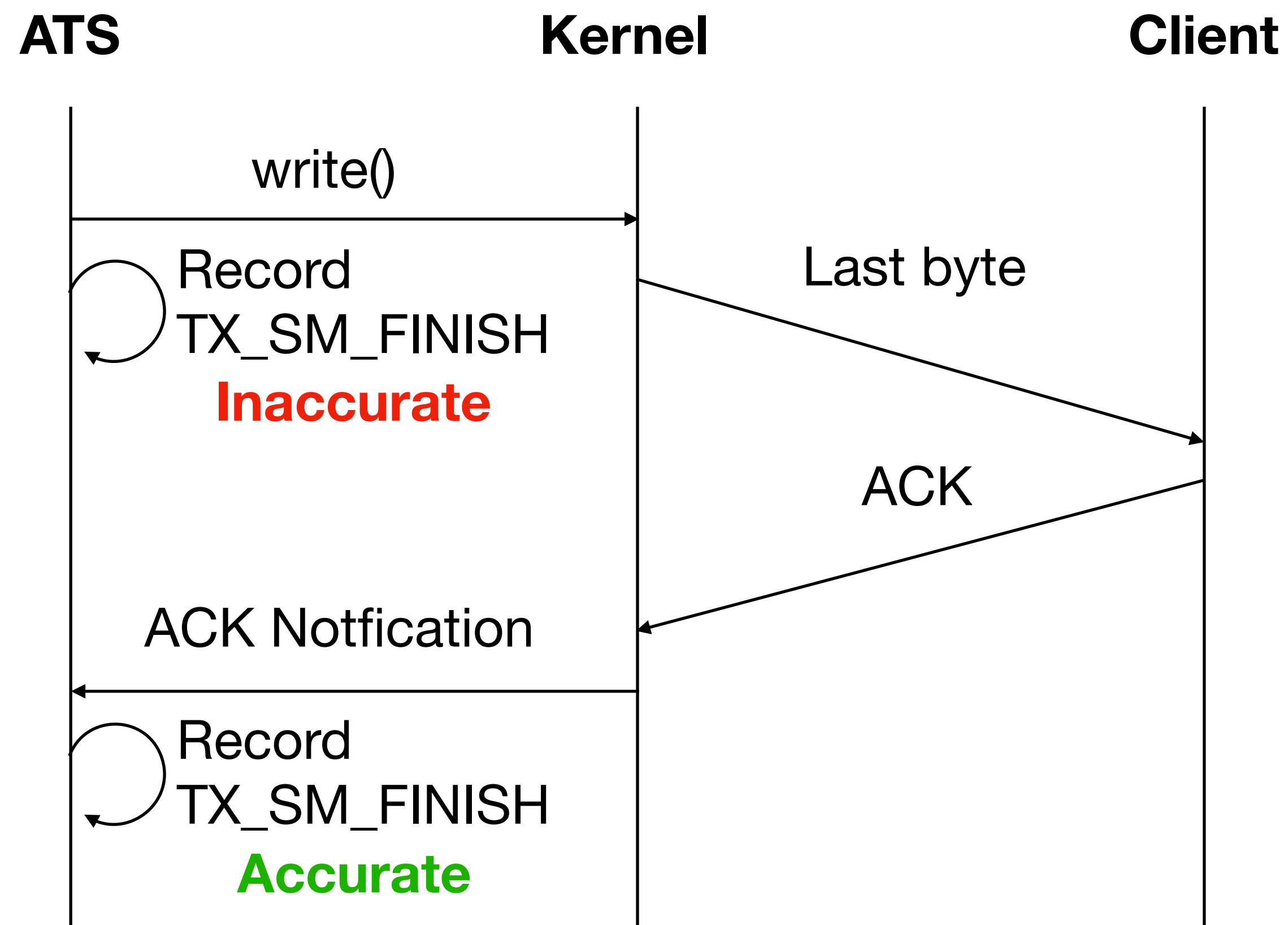
Masakazu Kitajo <maskit@apache.org>

Agenda

- Features & Issues
 - Timestamping
 - MSG_ZEROCOPY
 - Kernel TLS
 - io_uring
- Discussion (if we have time)

Timestamping

Notification of transfer completion



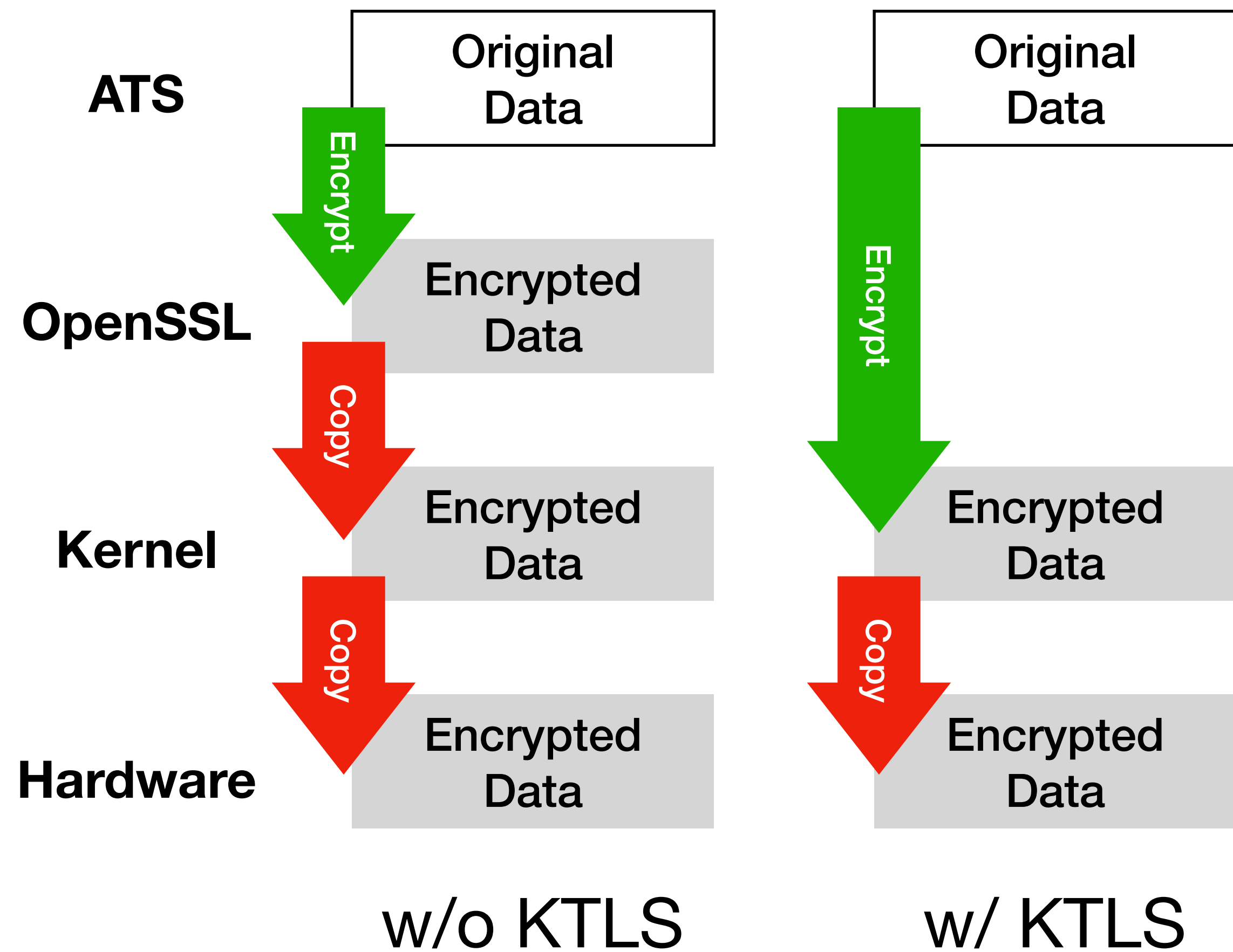
Transaction time (sec.)

	w/o TS	w/ TS	Actual
128KB	0.009	0.365	2.2
256KB	0.008	1.8	2.6
512KB	0.010	6.6	7.5
1024KB	6.3	11.0	11.8

PING: 320ms

Kernel TLS

Handling TLS encryption & decryption on Kernel



Performance

	w/o KTLS	w/ KTLS
Time	47.10 s	47.14 s
Req/sec	2123.20 req/s	2121.47 req/s
Throughput	265.98MB/s	265.77MB/s

h2load -n 100000 -c 10 -t5 --h1 https://test-server/128kb

Linux: 5.14 (Fedora 34)
ATS: master + tiny changes

Kernel TLS

Issues

SSL_write

Default BIO

sock_write:

```
# ifndef OPENSSSL_NO_KTLS
if (BIO_should_ktls_ctrl_msg_flag())
    ktls_send_ctrl_message()
# endif

write()
```

Private functions!




Custom BIO

sock_write:

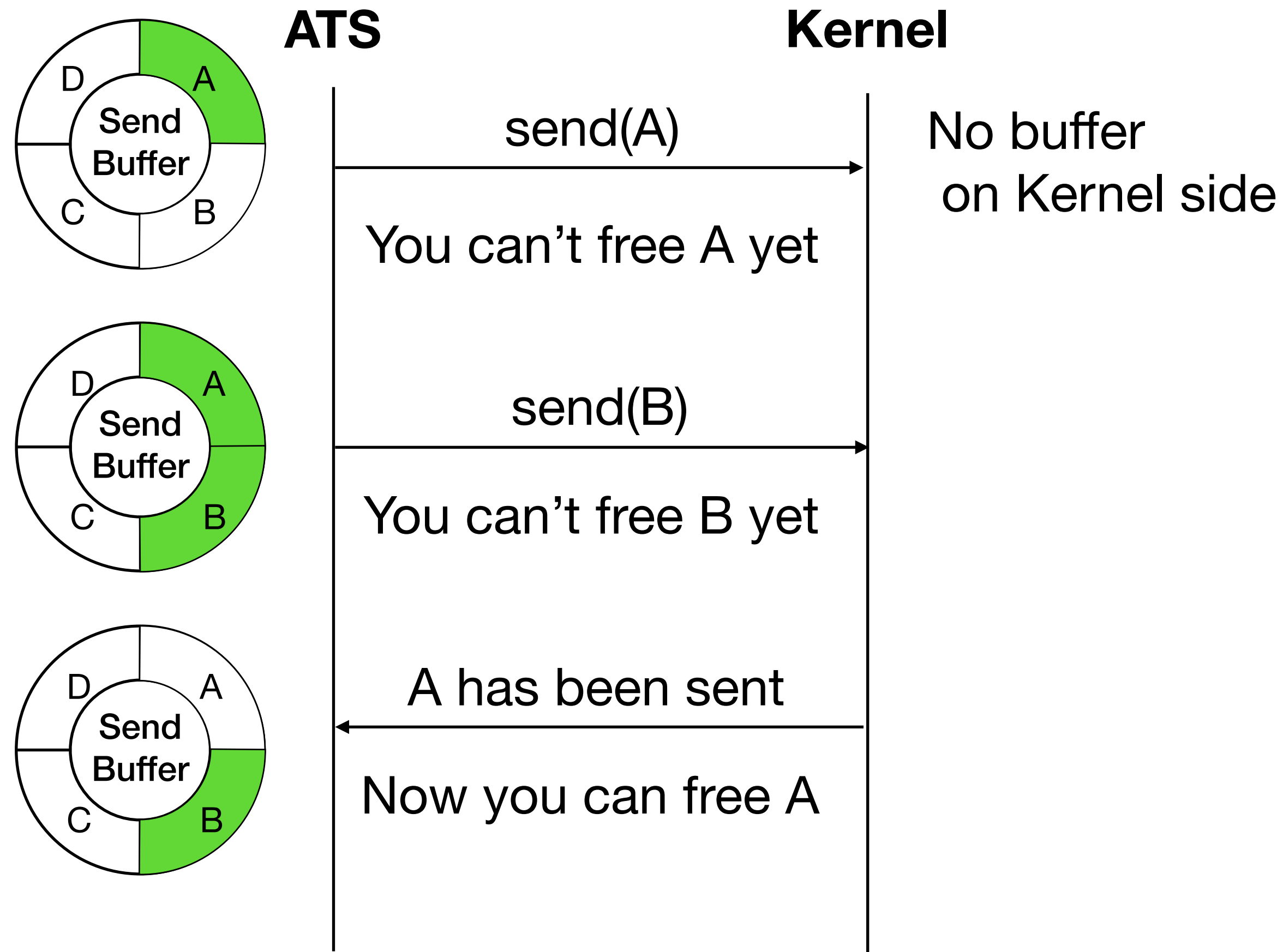
```
# ifndef OPENSSSL_NO_KTLS
if (BIO_should_ktls_ctrl_msg_flag())
    ktls_send_ctrl_message()
# endif

sendmsg()
```



MSG_ZEROCOPY

Copy avoidance between user space and Kernel space



Performance

	w/o Zero Copy	w/ Zero Copy
Time	6.15 s	4.72 s
Req/sec	1626.62 req/s	2119.57 req/s
Throughput	203.77 MB/s	265.53 MB/s

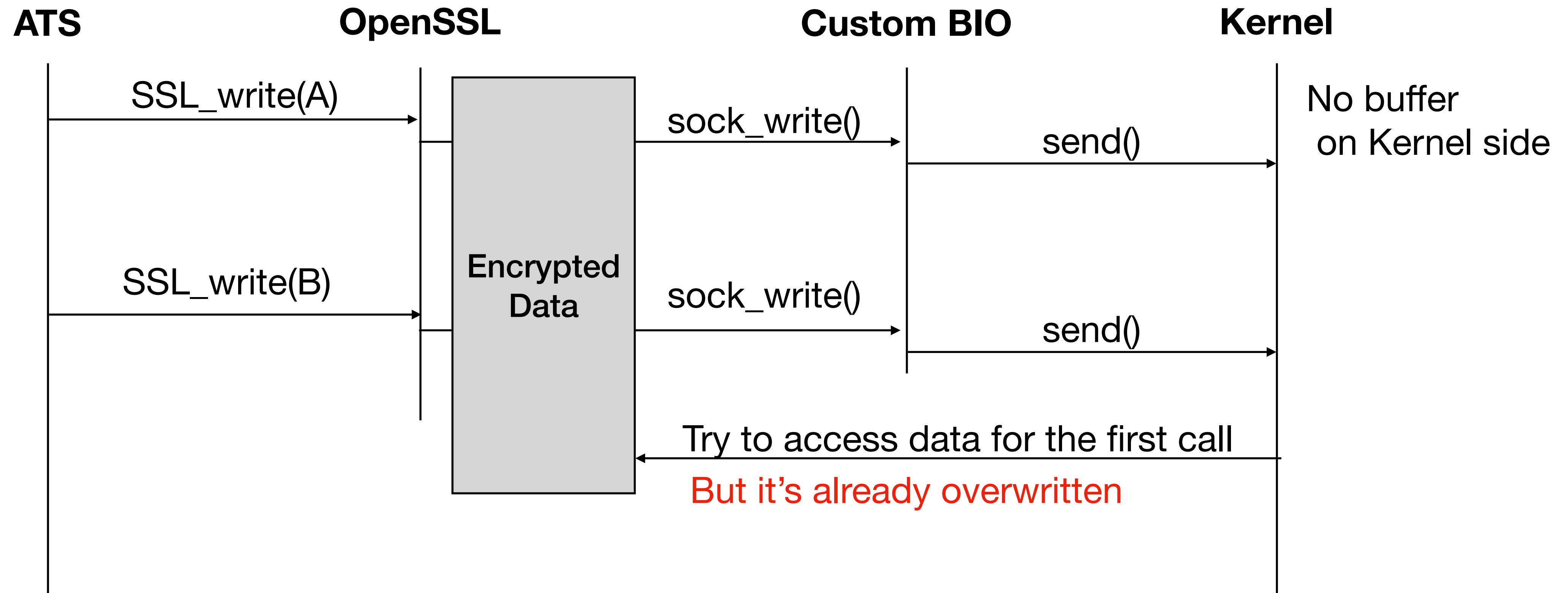
h2load -n 10000 --h1 http://test-server/128kb

Linux: 5.14 (Fedora 34)

ATS: master + hack

MSG_ZEROCOPY

Issues



io_uring

New asynchronous I/O interface

- Available since Kernel 5.1
- Completion model
- It may be slow
 - <https://github.com/axboe/liburing/issues/189>

Issues & Challenges

- Issues
 - Have no event for transfer completion
 - Have no control for sockets and send buffers on TLS connections
 - Have no way to use Kernel TLS and other features at the same
 - Have different concept of I/O
- Challenges
 - Keeping vender neutrality