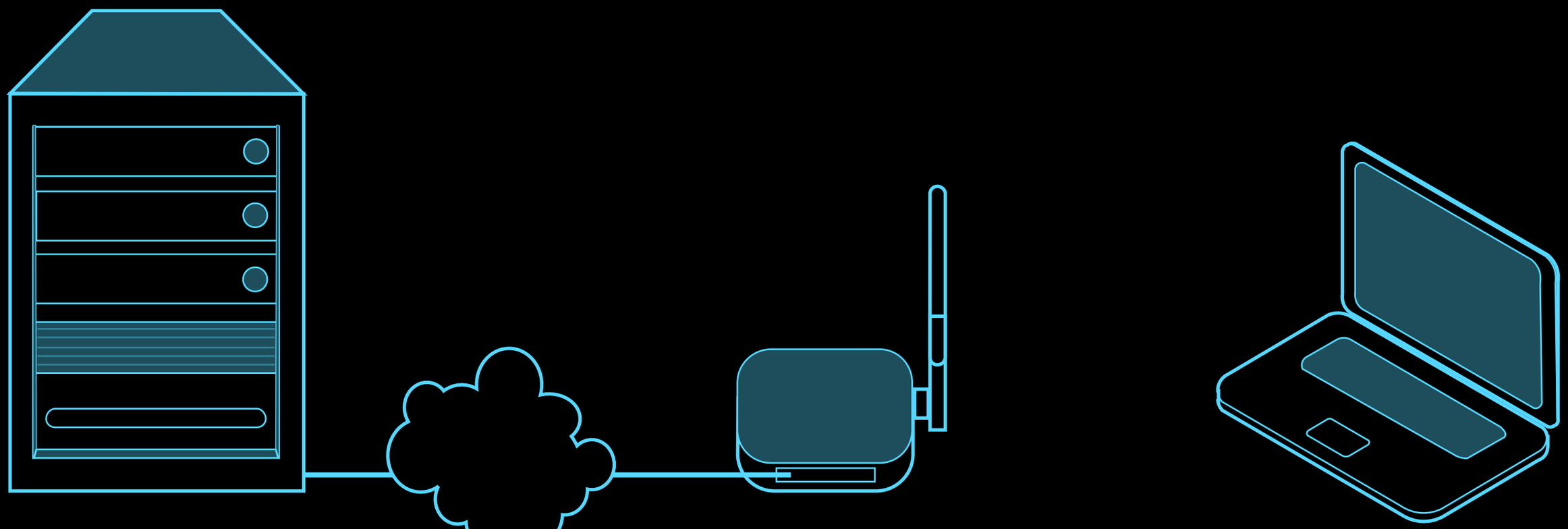
UCL

HACK: HIERARCHICAL ACKS FOR EFFICIENT WIRELESS MEDIUM UTILIZATION

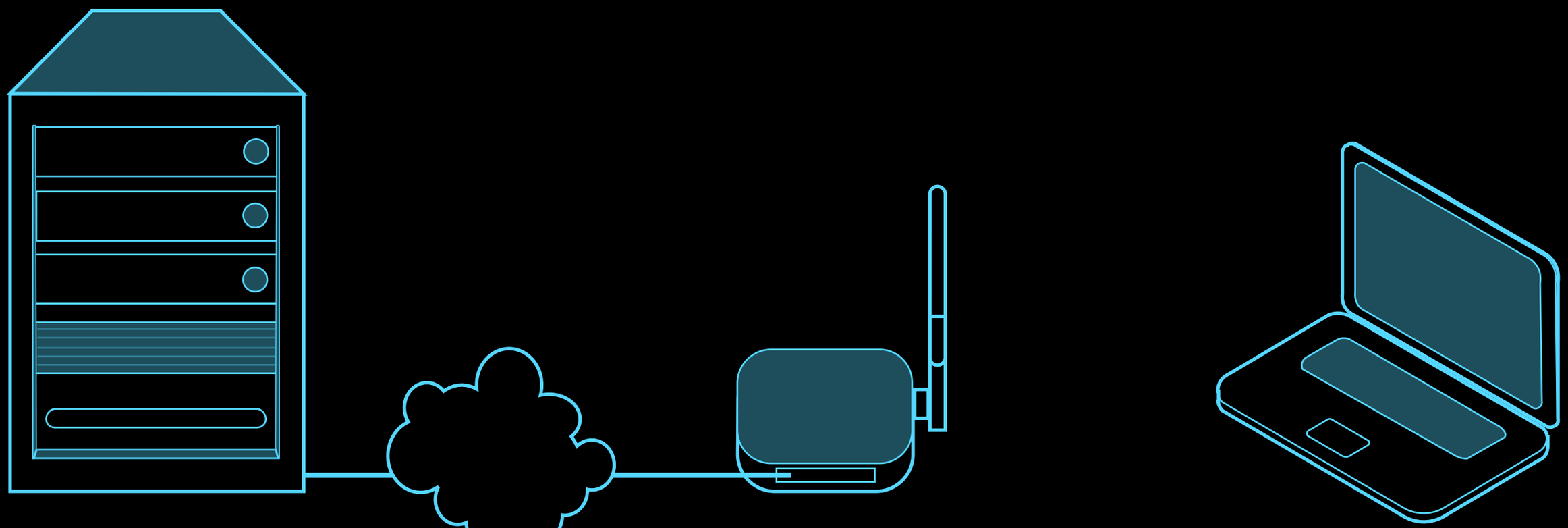
LYNNE SALAMEH, ASTRIT ZHUSHI, MARK HANDLEY,
KYLE JAMIESON, BRAD KARP.

UNIVERSITY COLLEGE LONDON

WIFI MOSTLY USED FOR TCP DOWNLOADS



WIFI MOSTLY USED FOR TCP DOWNLOADS



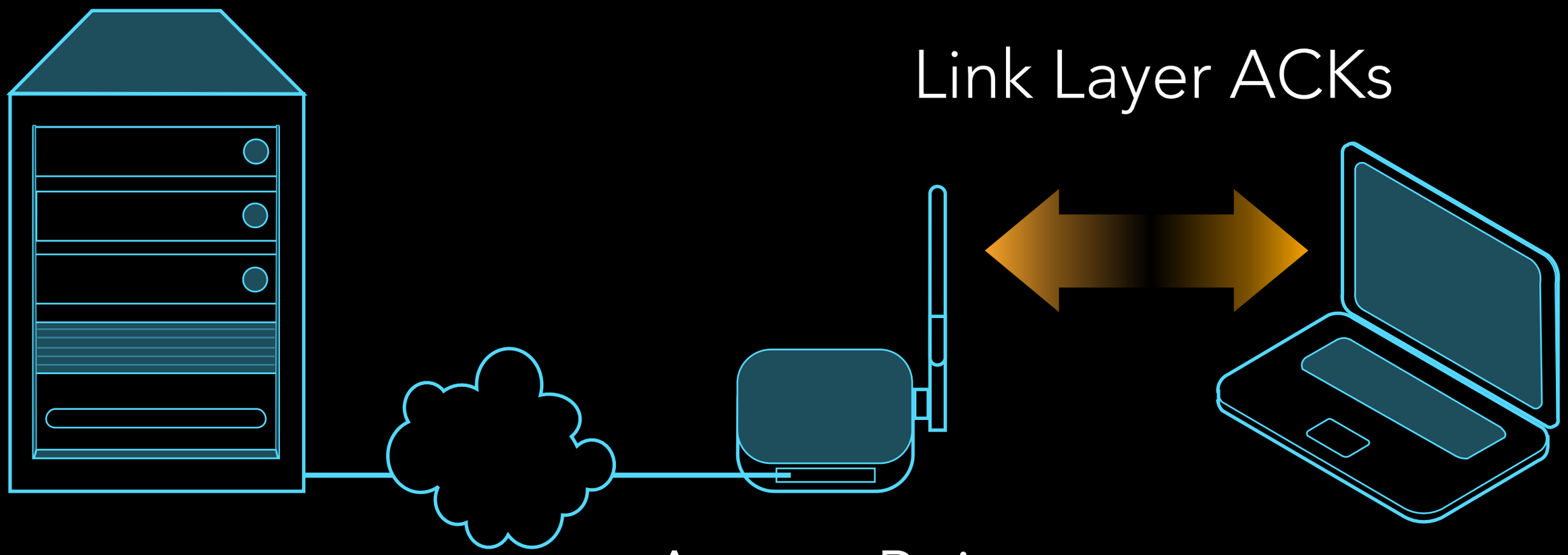
Server

Access Point (AP)

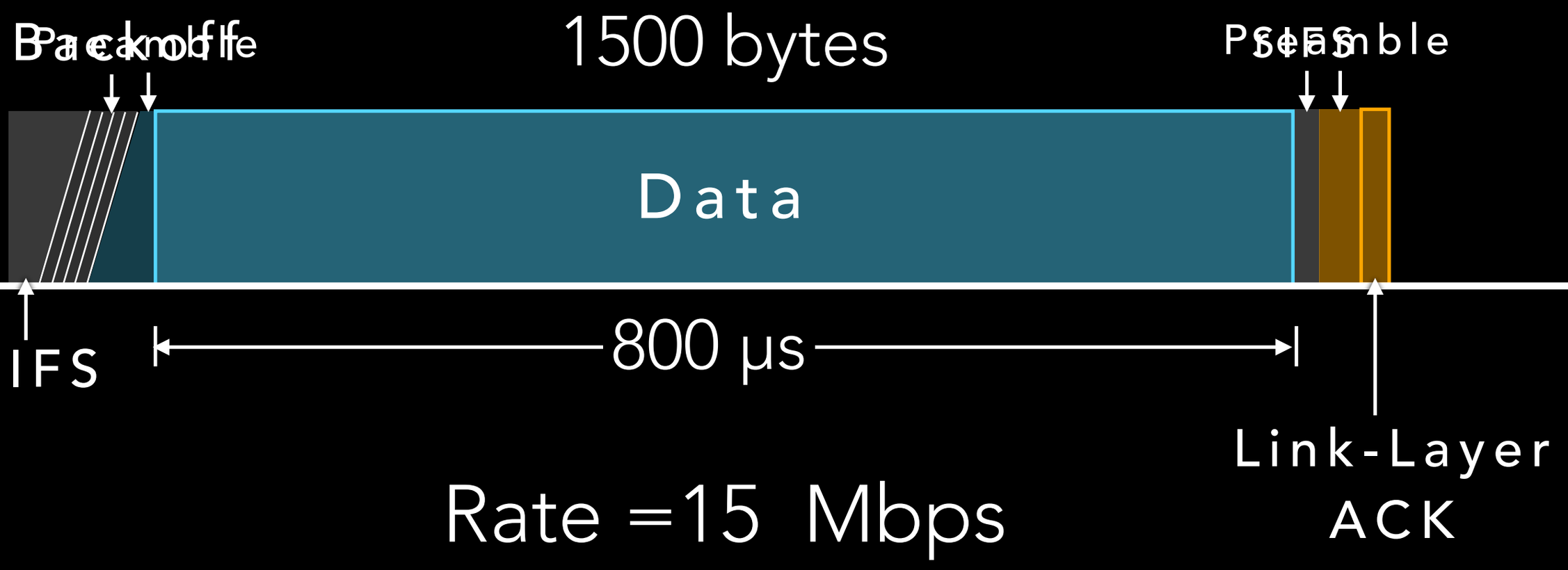
Client

TCP ACKs

WIFI MOSTLY USED FOR TCP DOWNLOADS



WIFI MEDIUM ACQUISITION INCURS OVERHEAD



* Diagram to Scale

WIFI MEDIUM ACQUISITION INCURS OVERHEAD

Rate =15 Mbps

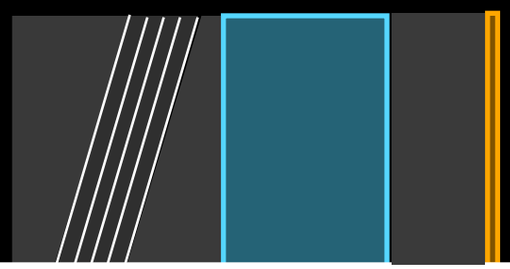
~80% Utilization

WIFI MEDIUM ACQUISITION INCURS HIGH OVERHEAD

Rate = 60 Mbps

~51% Utilization

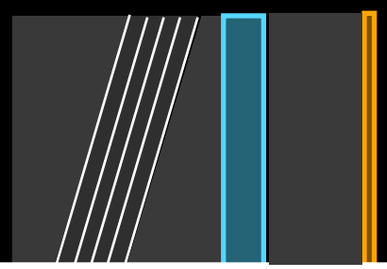
WIFI MEDIUM ACQUISITION INCURS HIGH OVERHEAD



Rate = 150 Mbps

~29% Utilization

WIFI MEDIUM ACQUISITION INCURS HIGH OVERHEAD

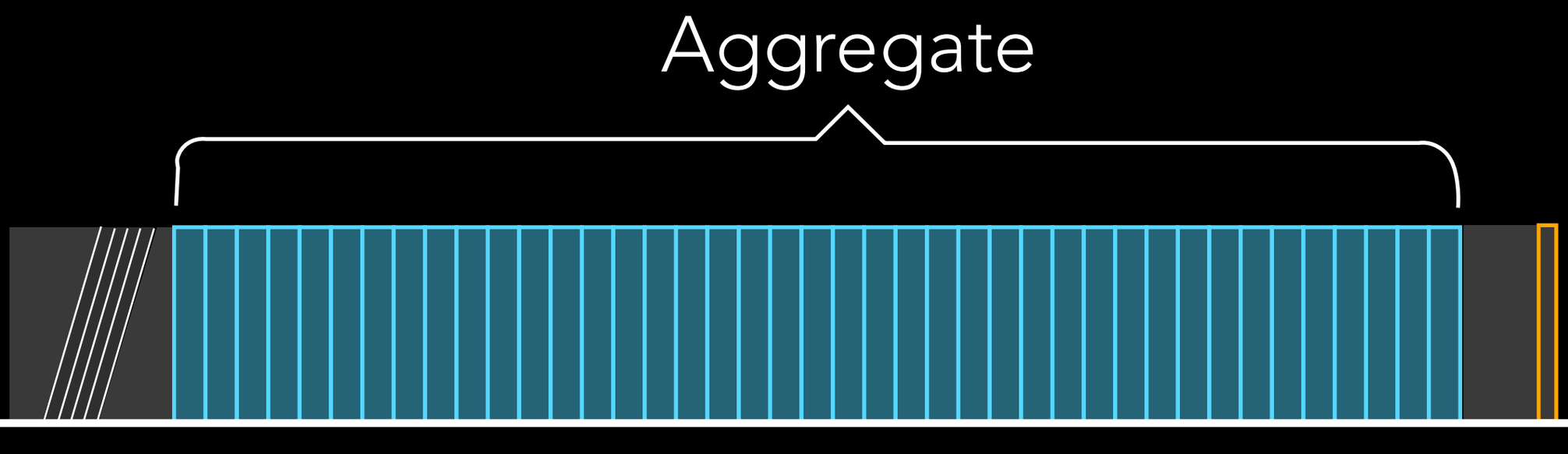


ACK

Rate = 600 Mbps

~9% Utilization

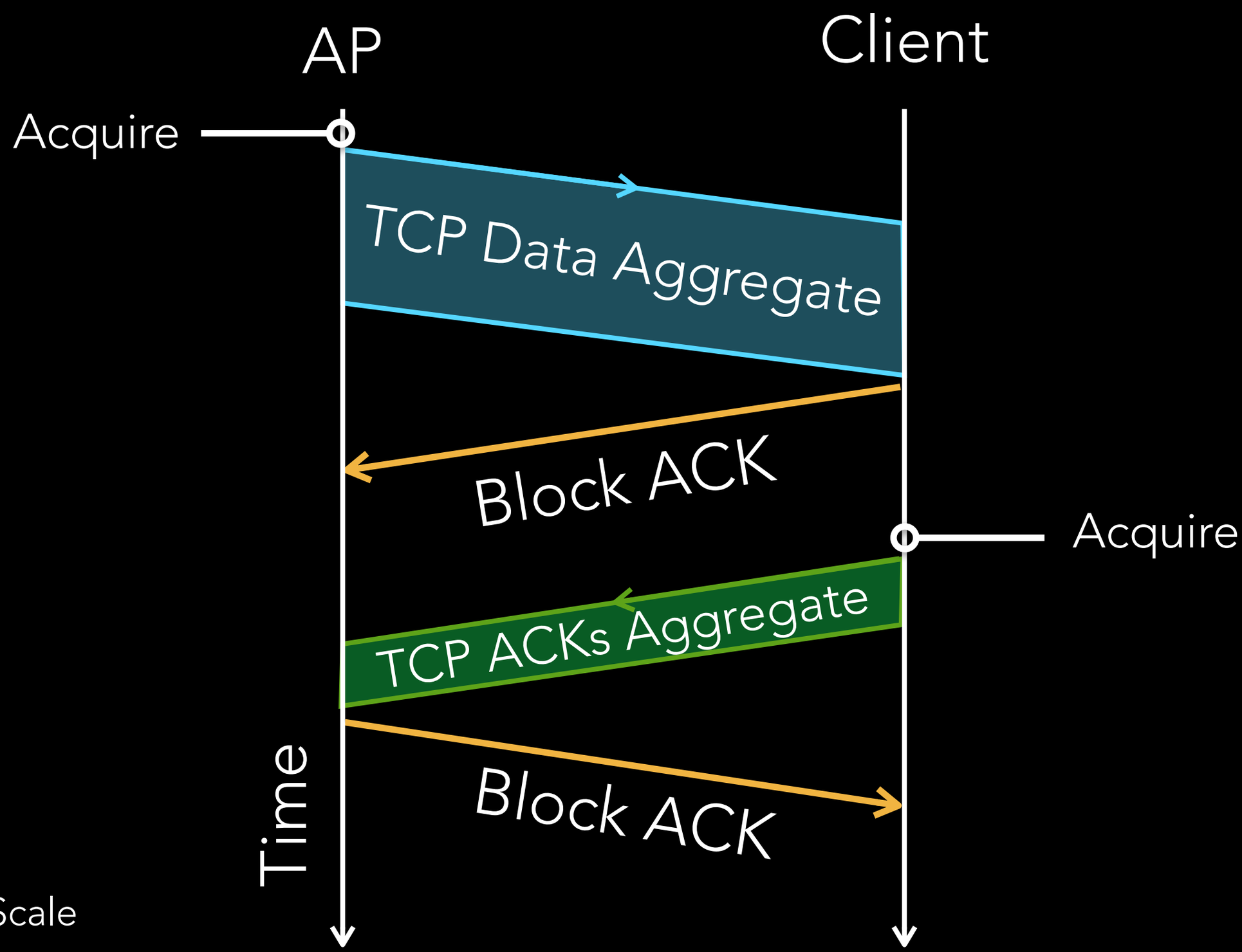
AGGREGATION INCREASES UTILIZATION AT HIGH RATES



Rate = 600 Mbps

~80% Utilization

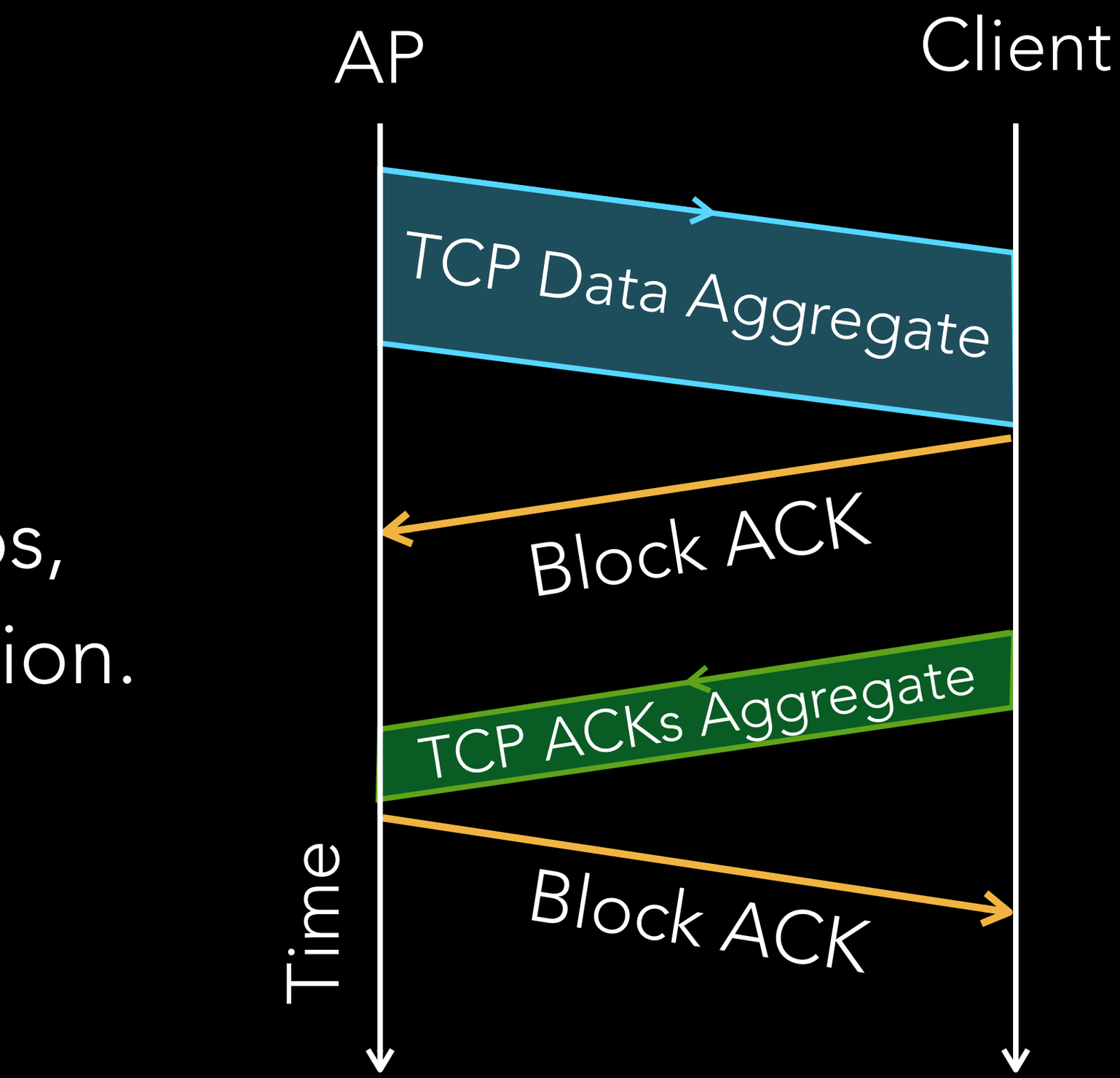
TCP DOUBLES MEDIUM ACQUISITIONS



* Not to Scale

TCP DOUBLES MEDIUM ACQUISITIONS

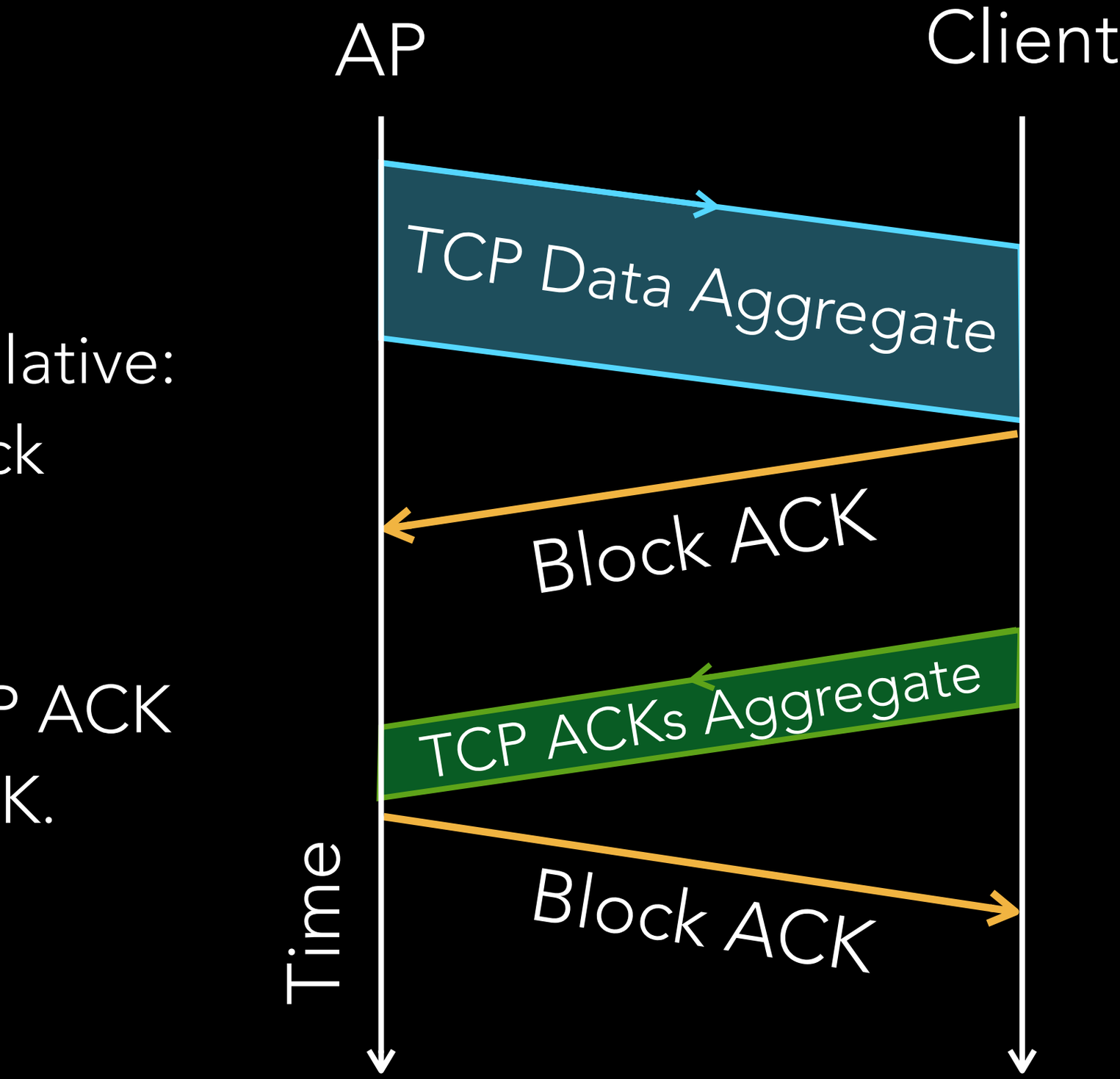
At 600 Mbps,
~61% utilization.



DO WE NEED TO ACQUIRE THE MEDIUM FOR FEEDBACK?

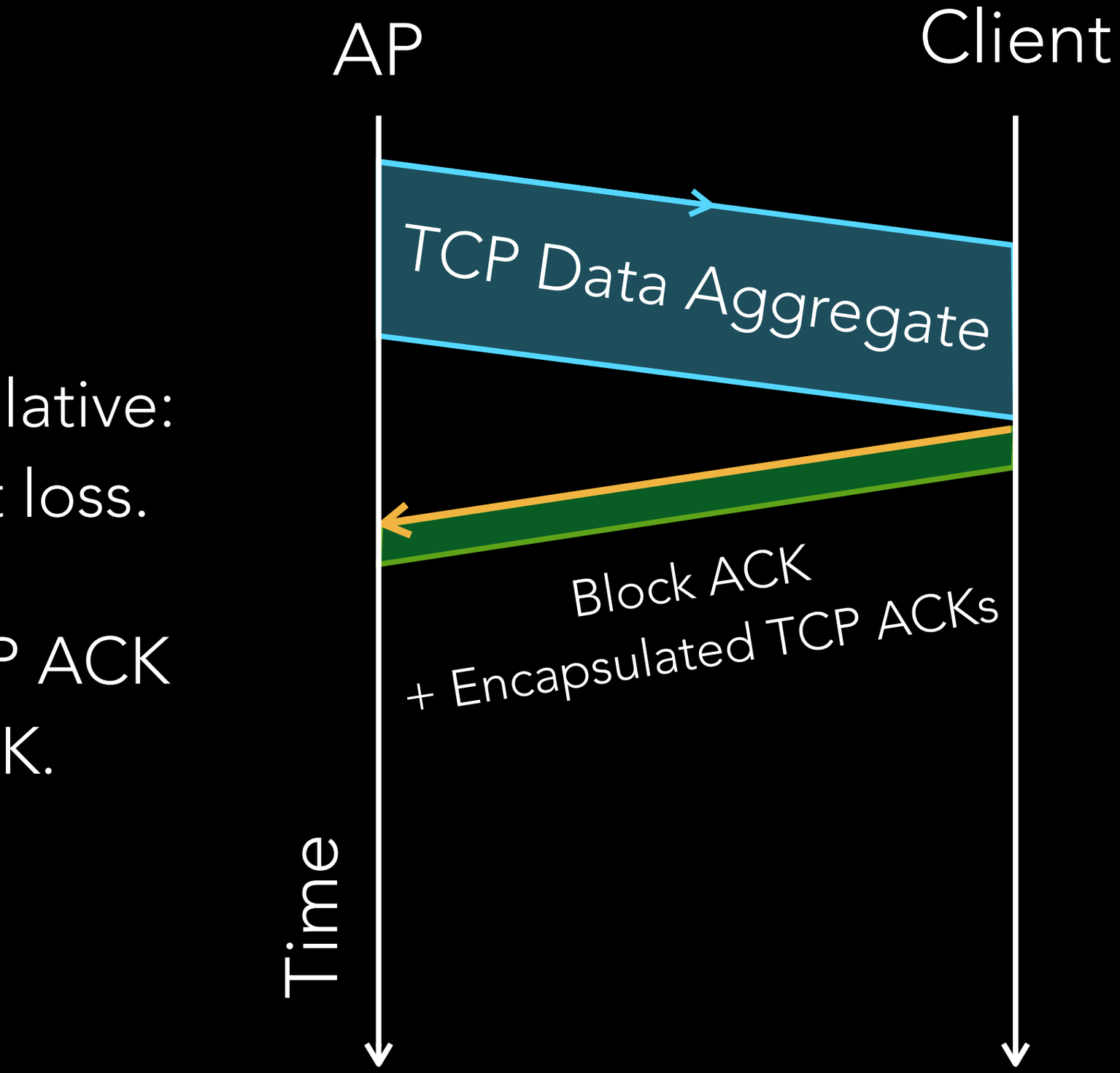
- TCP ACKs cumulative: no need for Block ACK.

- Encapsulate TCP ACK in Link-Layer ACK.



DO WE NEED TO ACQUIRE THE MEDIUM FOR FEEDBACK?

- TCP ACKs cumulative: robust to packet loss.
- Encapsulate TCP ACK in Link-Layer ACK.



HACK: HIERARCHICAL

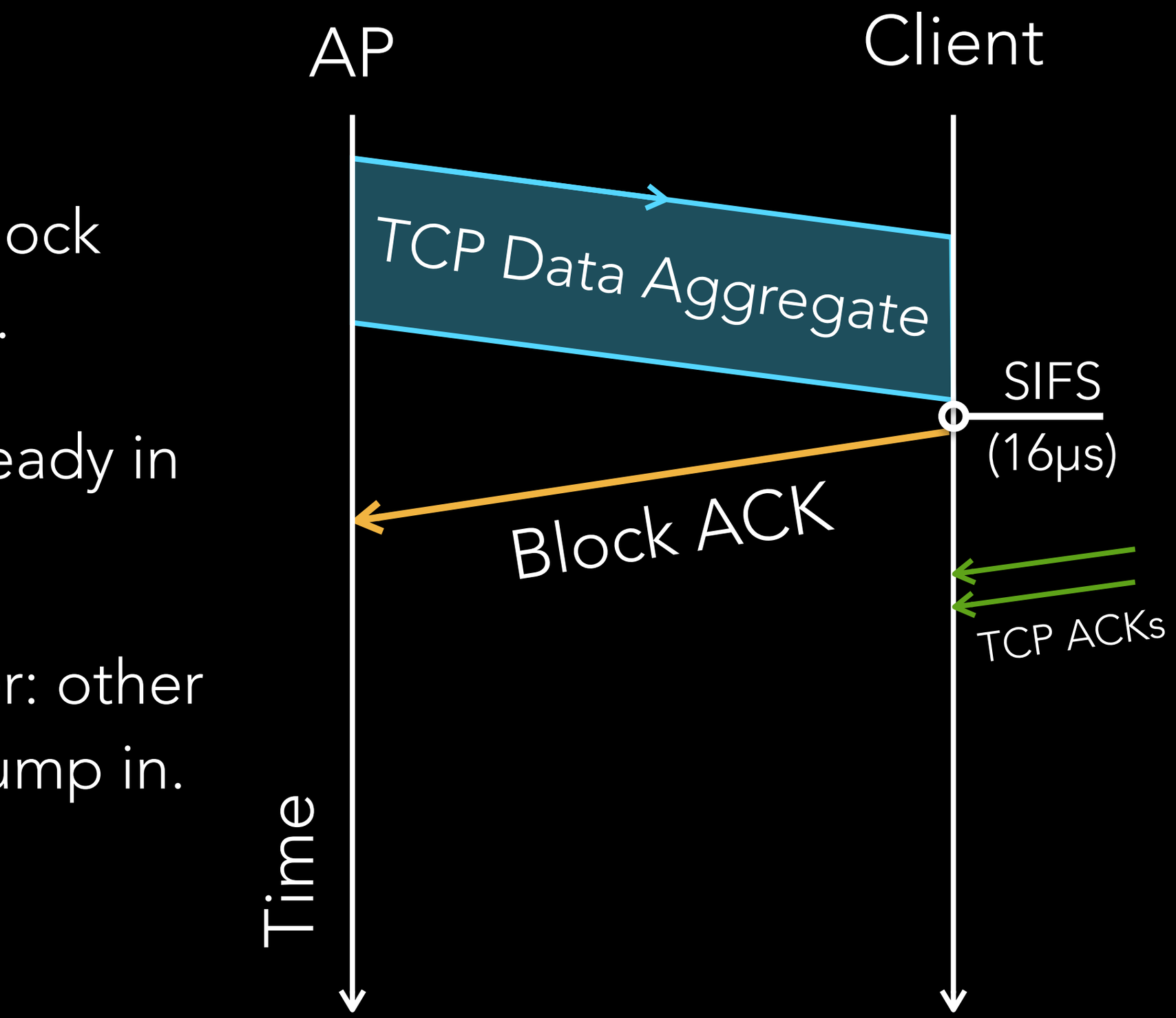
ACKNOWLEDGEMENTS

GOALS

- Coexistence with stock 802.11.
- Simple NIC changes.
- No TCP changes.
- Robustness to loss.

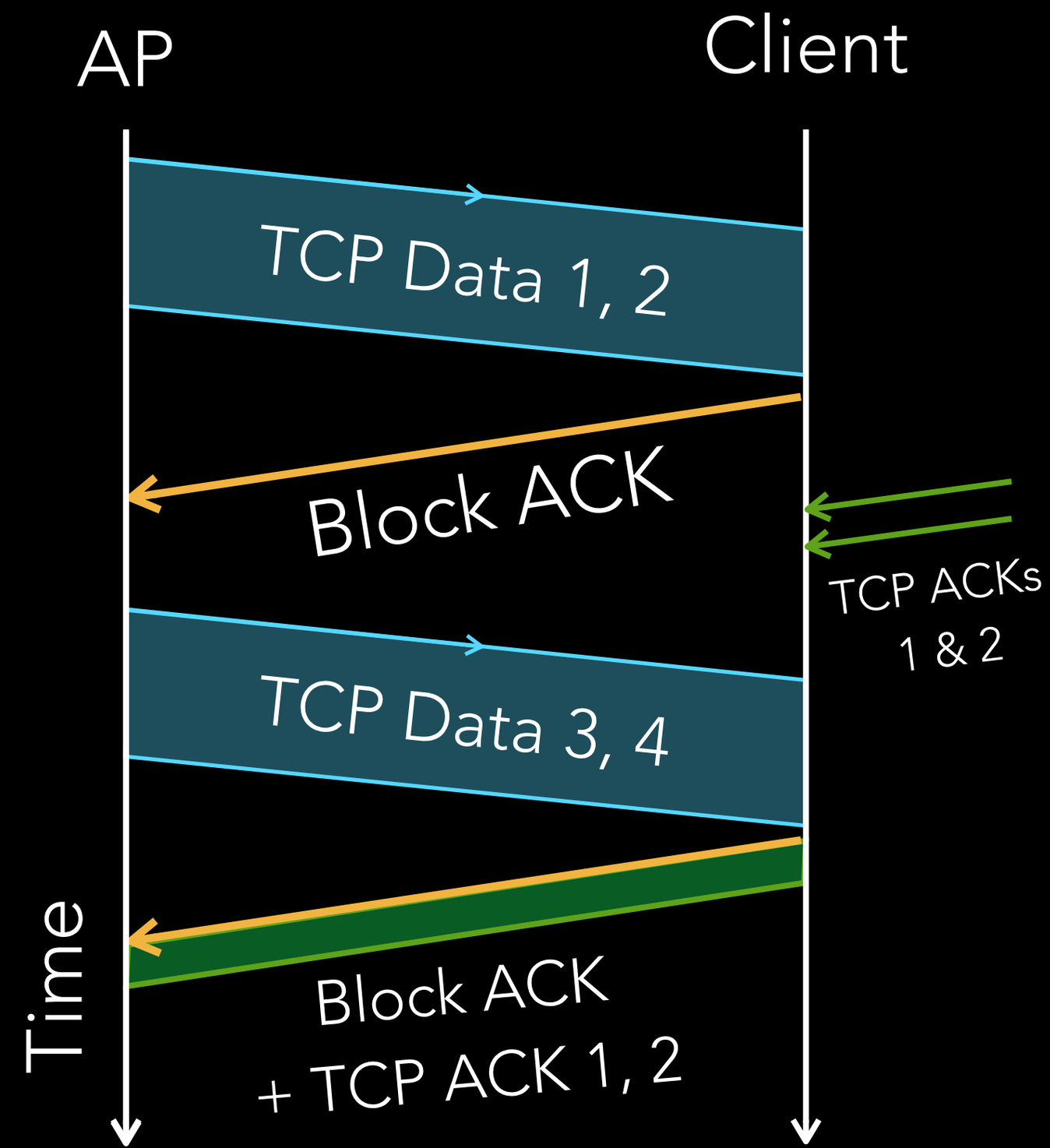
BLOCK ACKS MUST MEET HARD DEADLINE

- Need to send Block ACK within SIFS.
- TCP ACKs not ready in time.
- Can’t wait longer: other senders might jump in.



TCP ACKS DON'T HAVE A HARD DEADLINE

- TCP ACKs can afford to wait.
- Append them to the next Link-Layer ACK.



CONSECUTIVE TCP ACKS ARE REDUNDANT

TCP ACK Header
(20 Bytes)

CONSECUTIVE TCP ACKS ARE REDUNDANT

TCP ACK 1

CONSECUTIVE TCP ACKS ARE REDUNDANT

TCP ACK 1

TCP ACK 2

CONSECUTIVE TCP ACKS ARE REDUNDANT

TCP ACK 1

TCP ACK 2

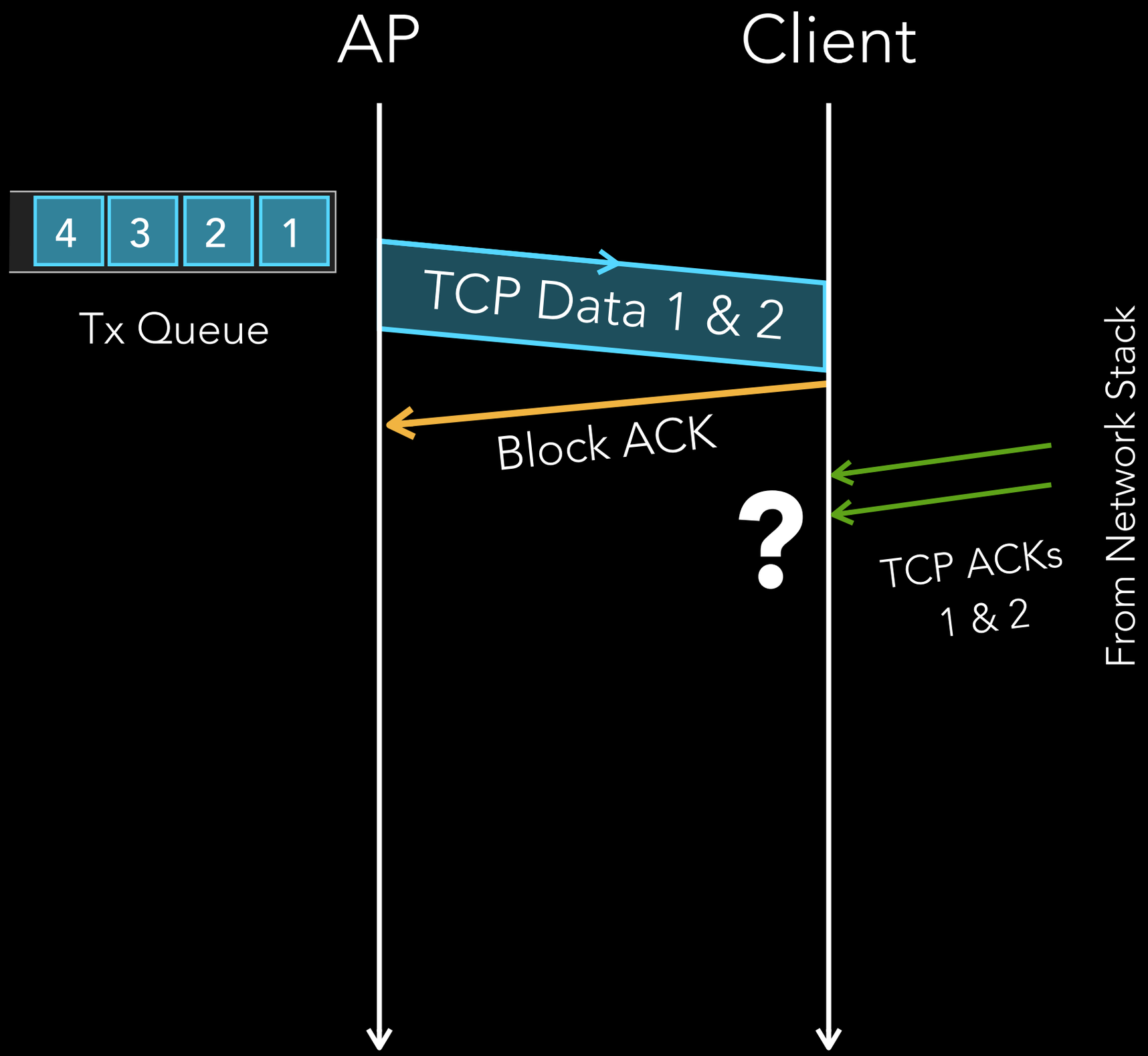
HACK USES ROHC FOR EFFICIENT COMPRESSION

TCP ACK 1

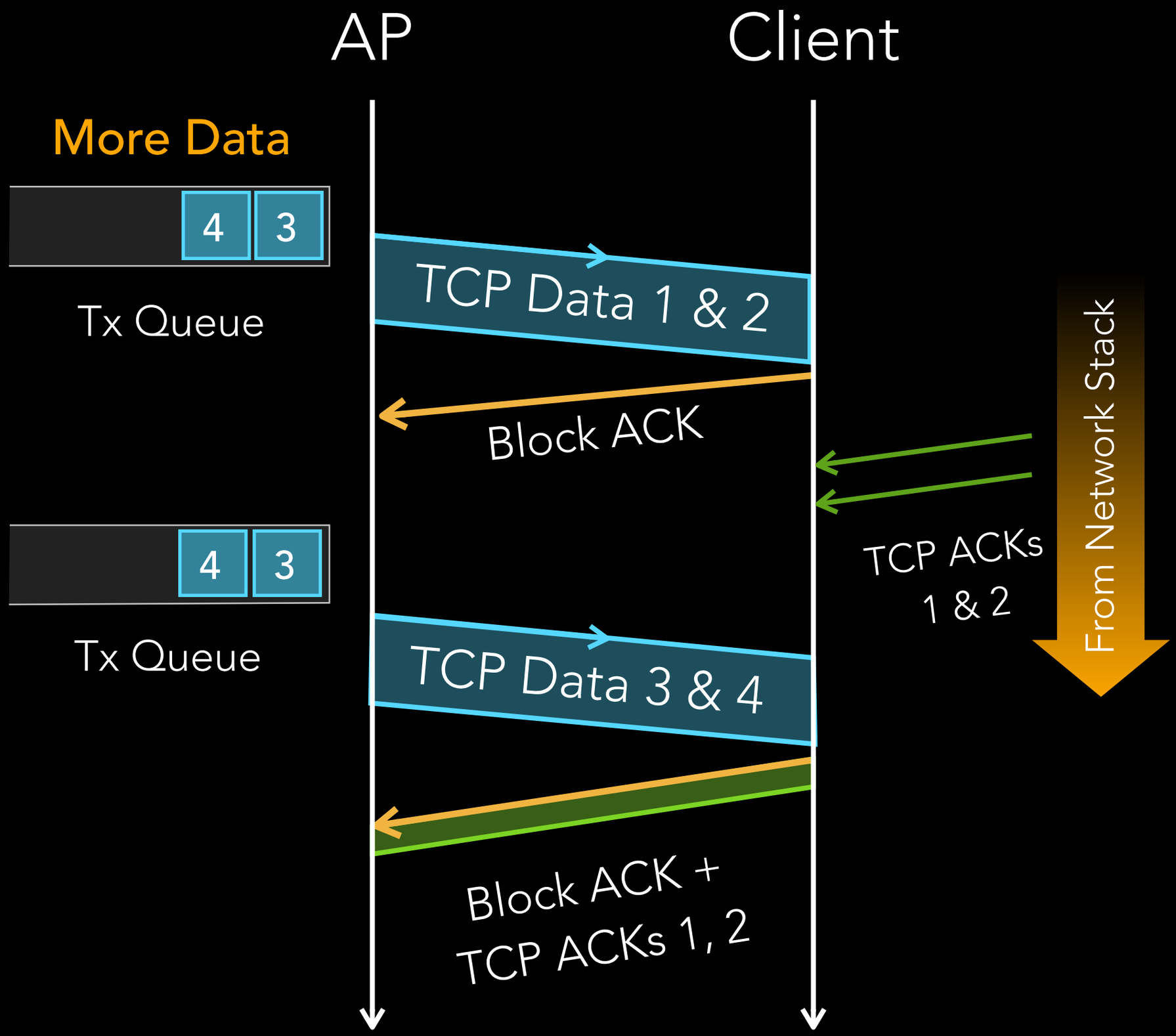
TCP ACK 2

- HACK uses Robust Header Compression (ROHC, RFC 6846)
- ROHC shrinks a TCP ACK to about 5 bytes

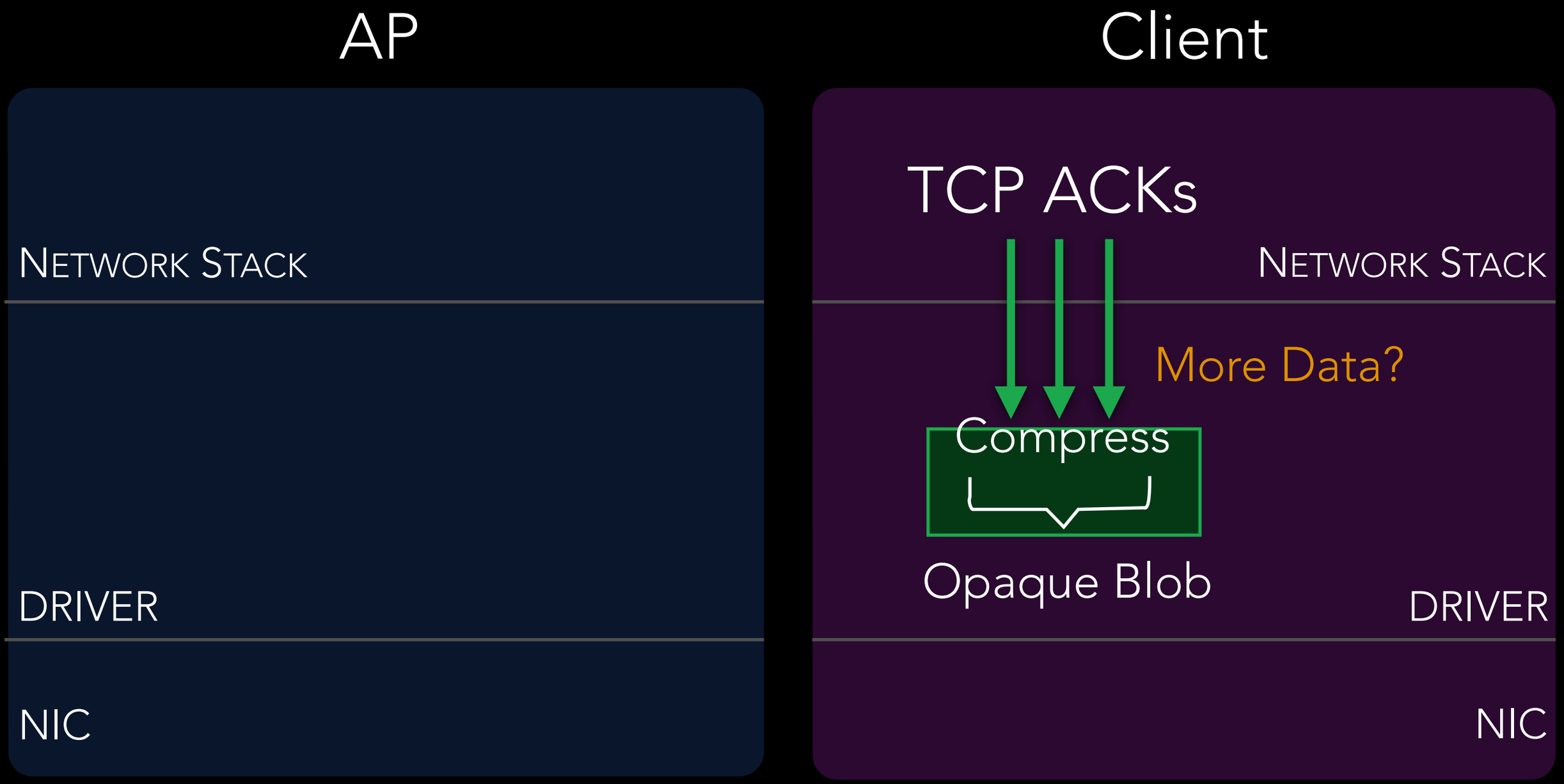
TO HACK OR NOT TO HACK?



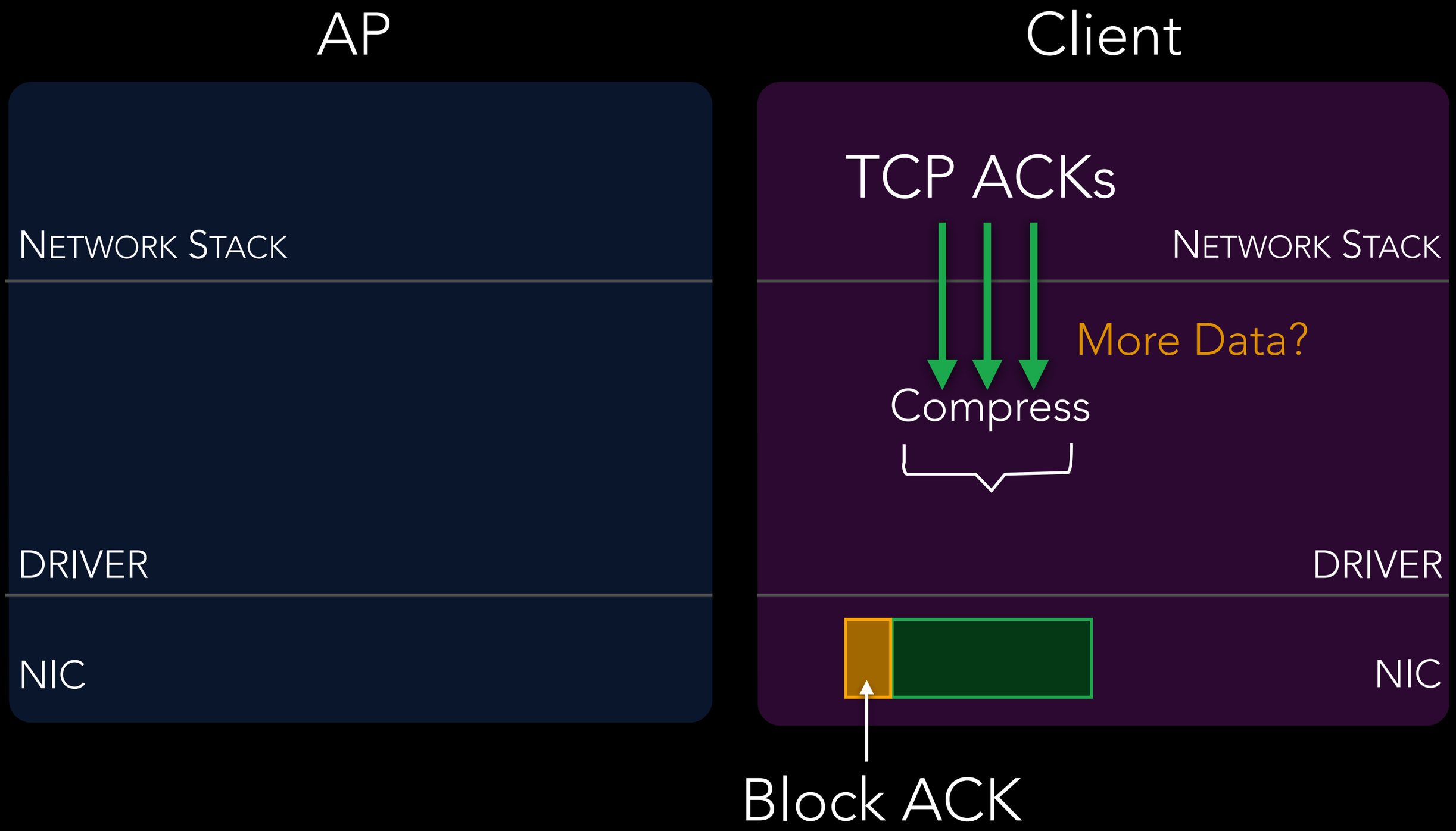
TO HACK OR NOT TO HACK?



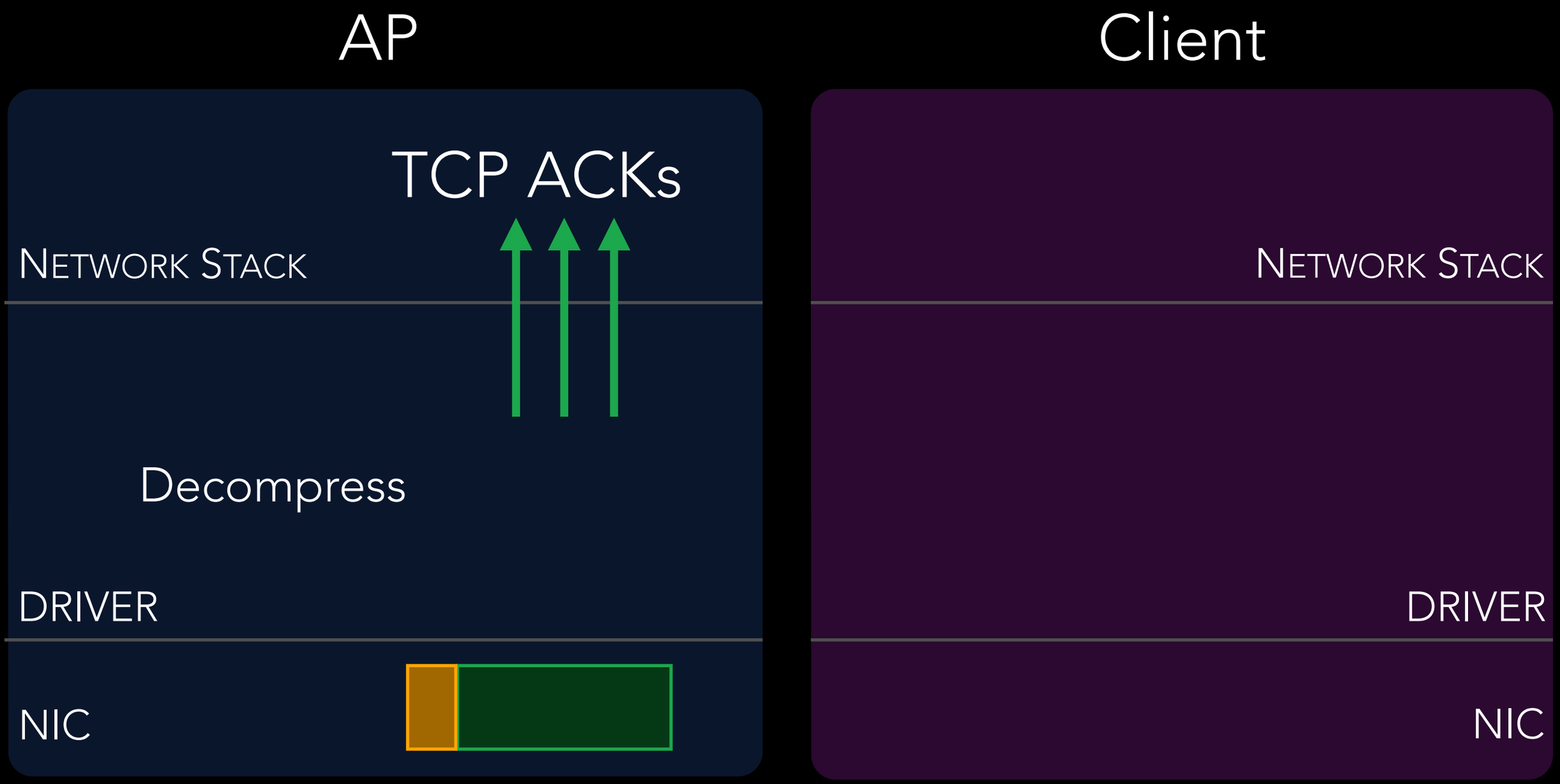
NIC NEEDS NO TCP KNOWLEDGE



NIC NEEDS NO TCP KNOWLEDGE

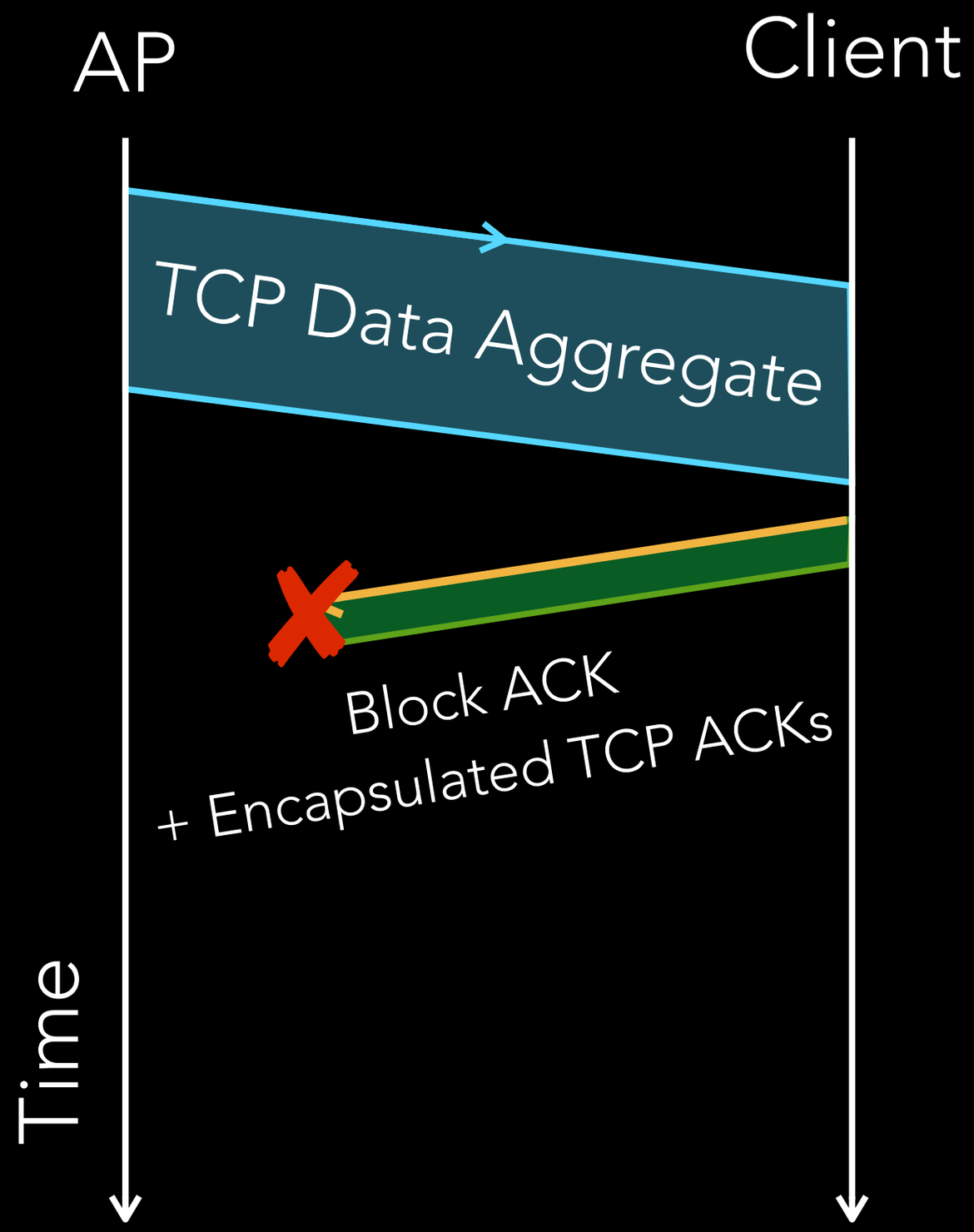


NIC NEEDS NO TCP KNOWLEDGE



WHAT ABOUT LOSS OF BLOCK ACKS?

- TCP can tolerate loss of ACKs.
- ROHC cannot: endpoints should remain in sync.
- HACK should be robust against loss.
- Details in paper.



EVALUATION

Microsoft's Software Radio (SoRa):

- 802.11a (no aggregation).

ns-3 simulator

- 802.11n (aggregation).

TCP GOODPUT EXPERIMENT

SoRa	NS-3
802.11a	802.11n
54 Mbps	150 Mbps

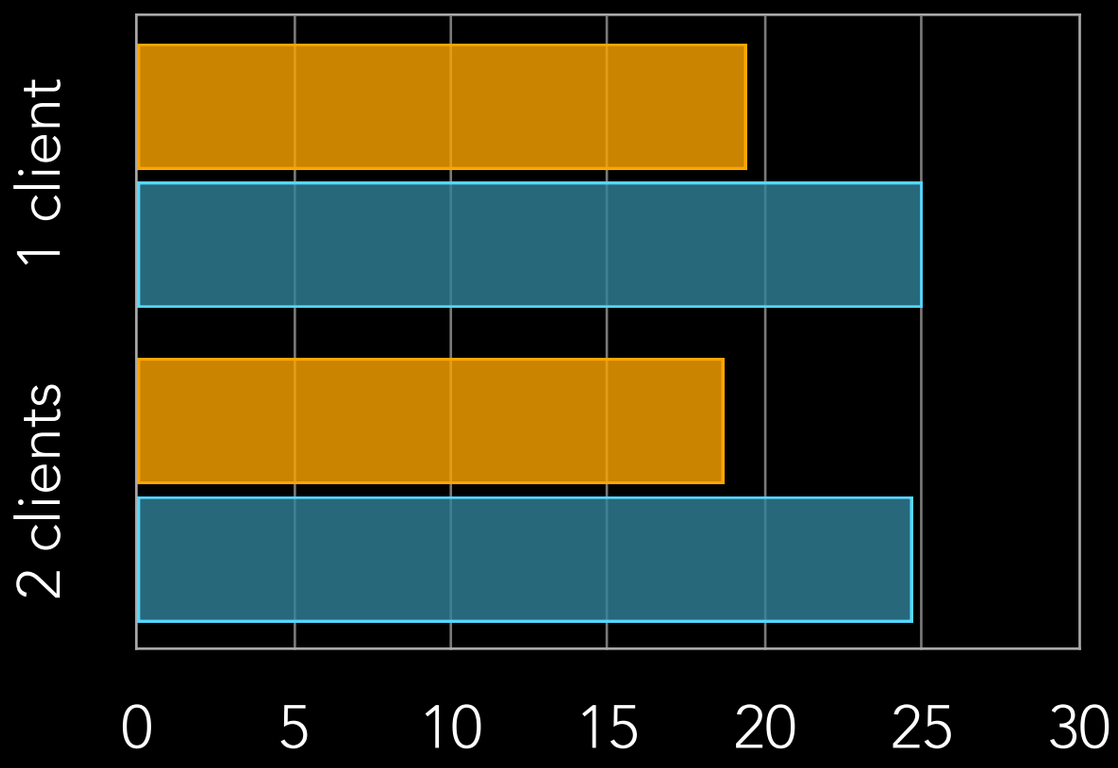
High Quality Link

1, 2 Clients	Up to 10 Clients

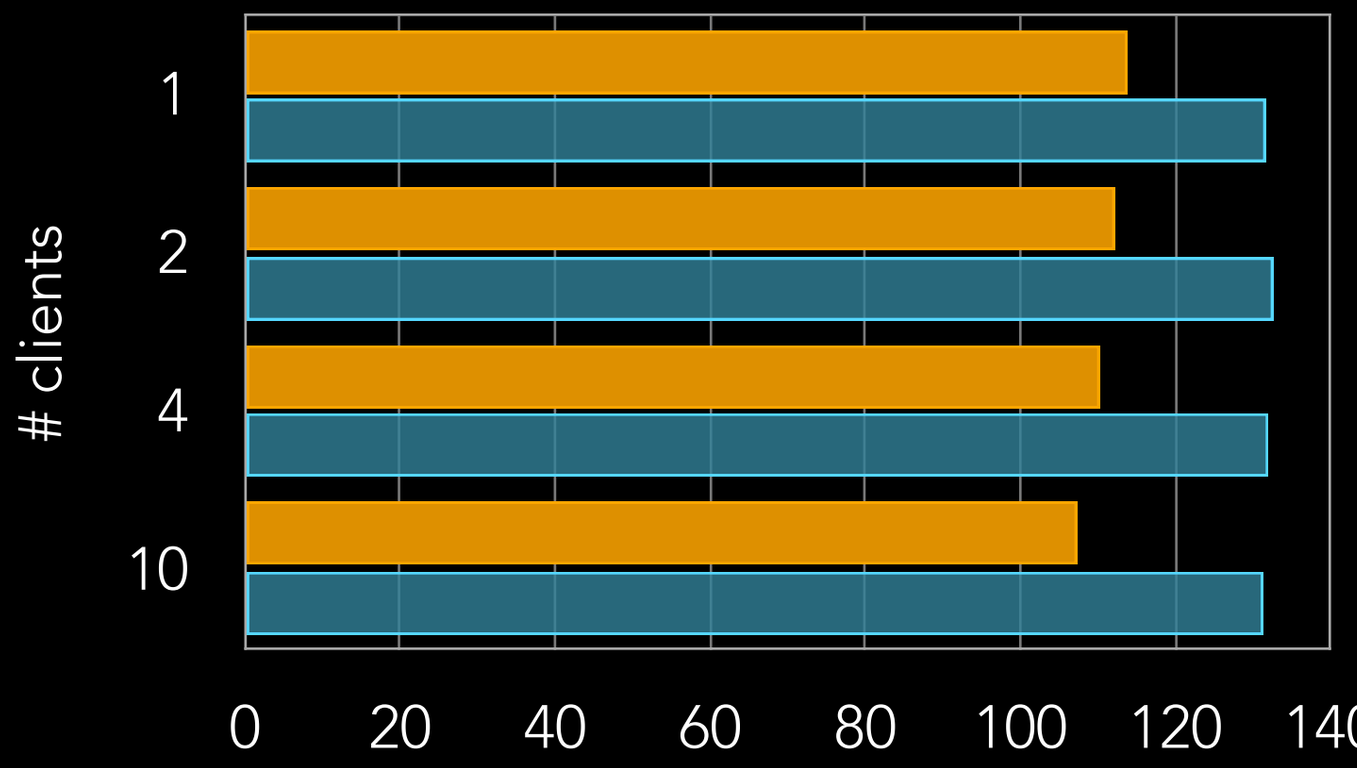
Measure steady state aggregate goodput

HACK IMPROVES TCP GOODPUT

SoRa (802.11a)



ns-3 (802.11n)

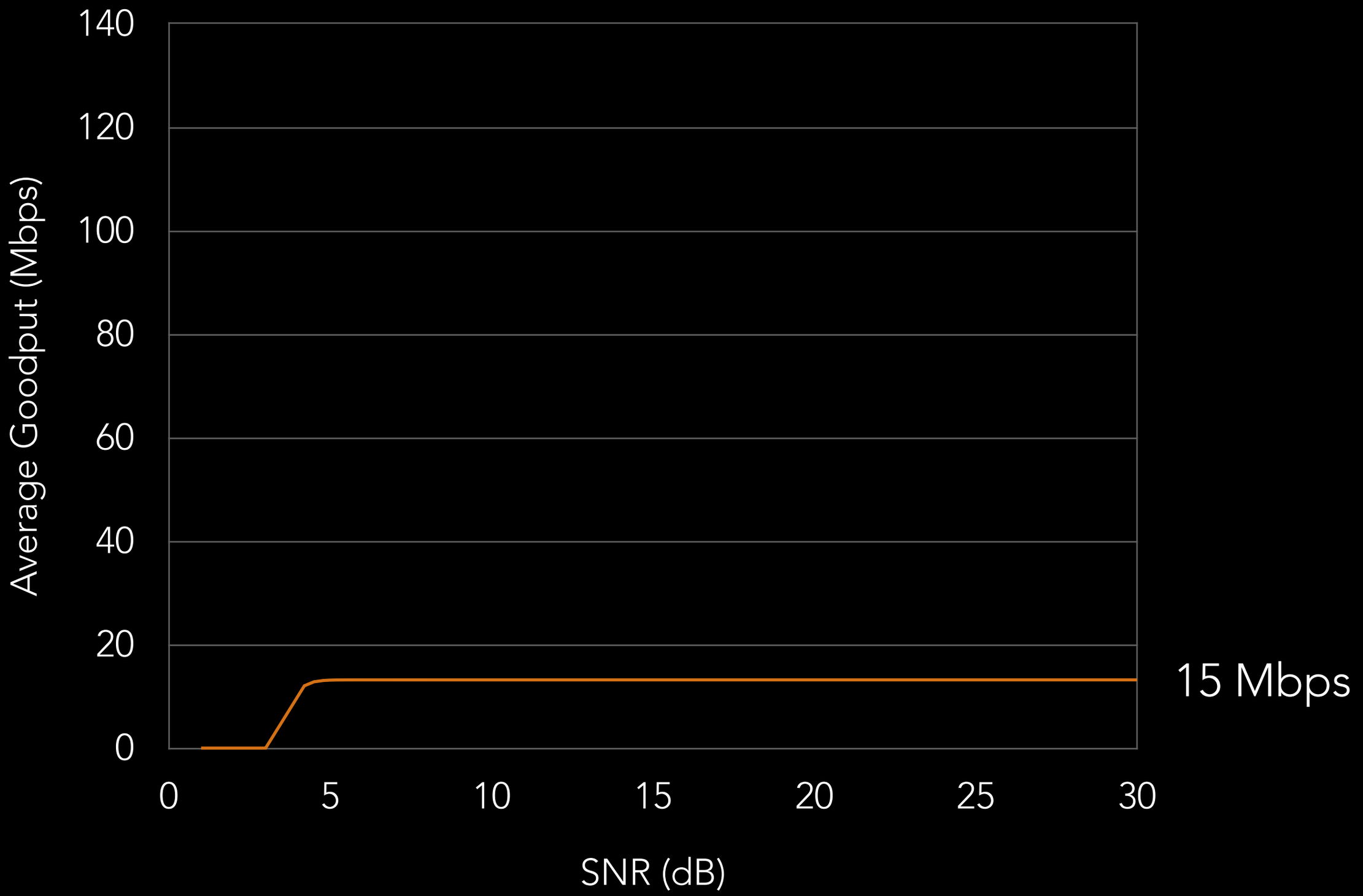


TCP/Wifi TCP/HACK

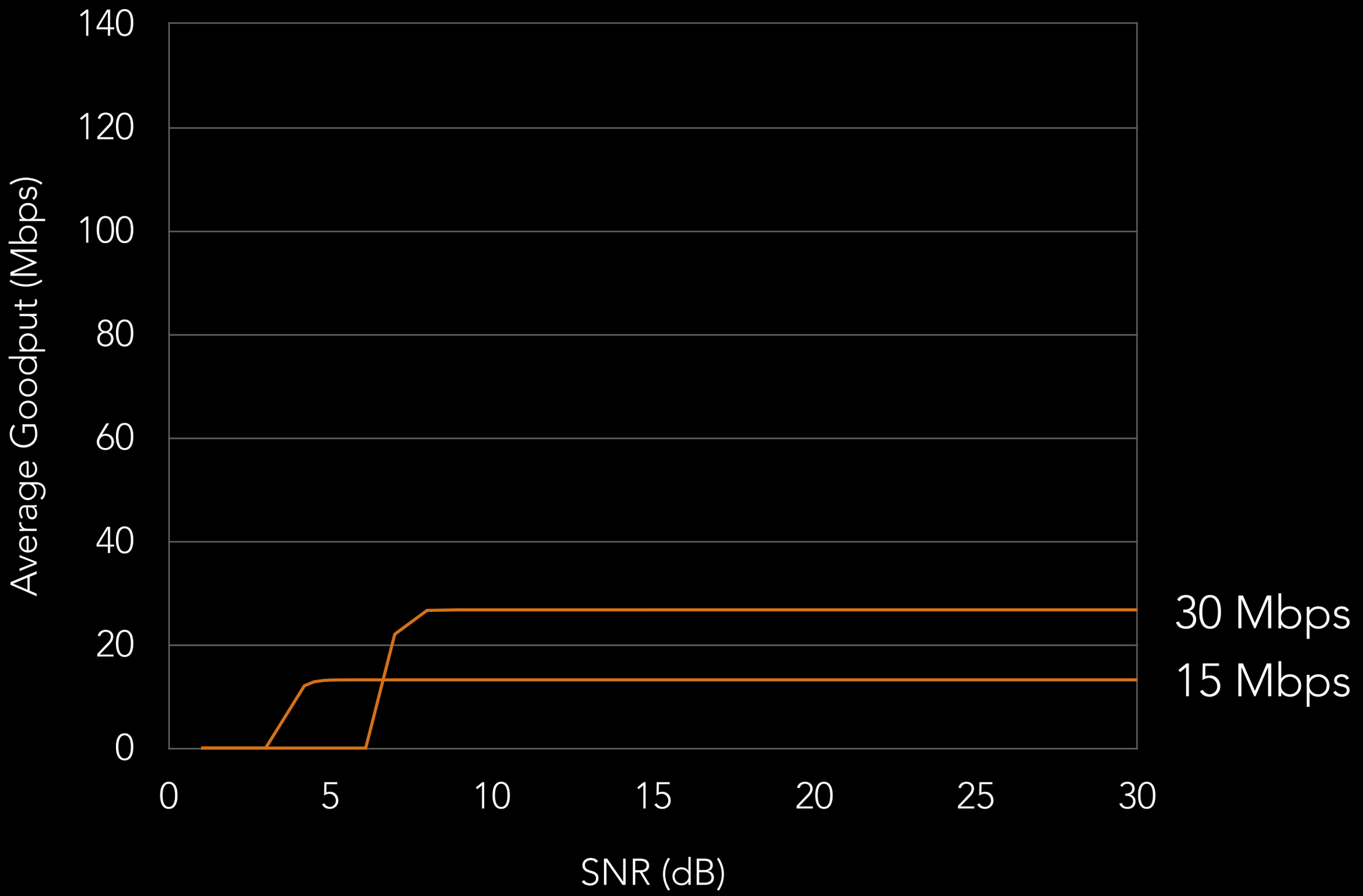
29 - 32 % Improvement

15 - 22 % Improvement

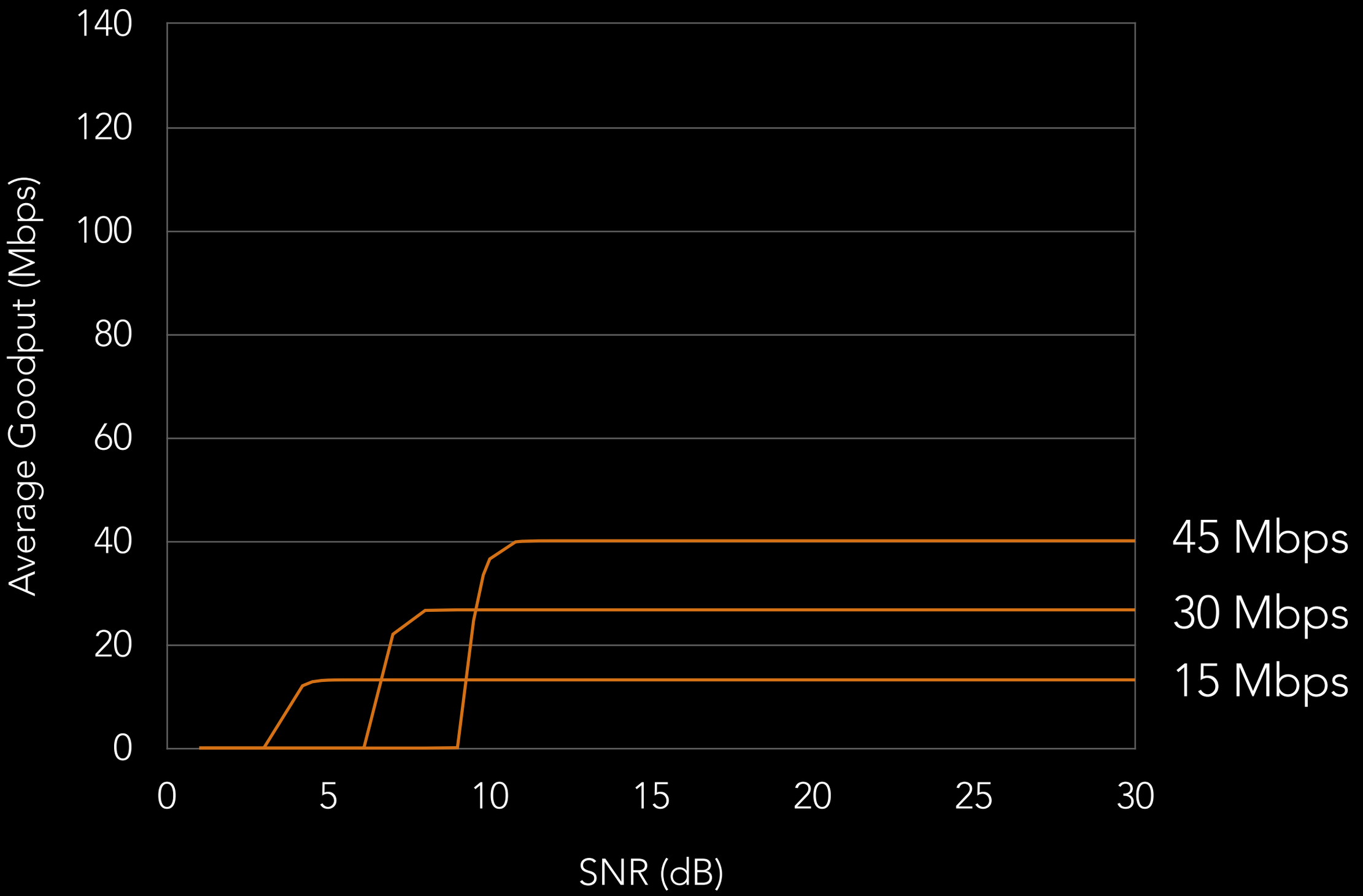
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



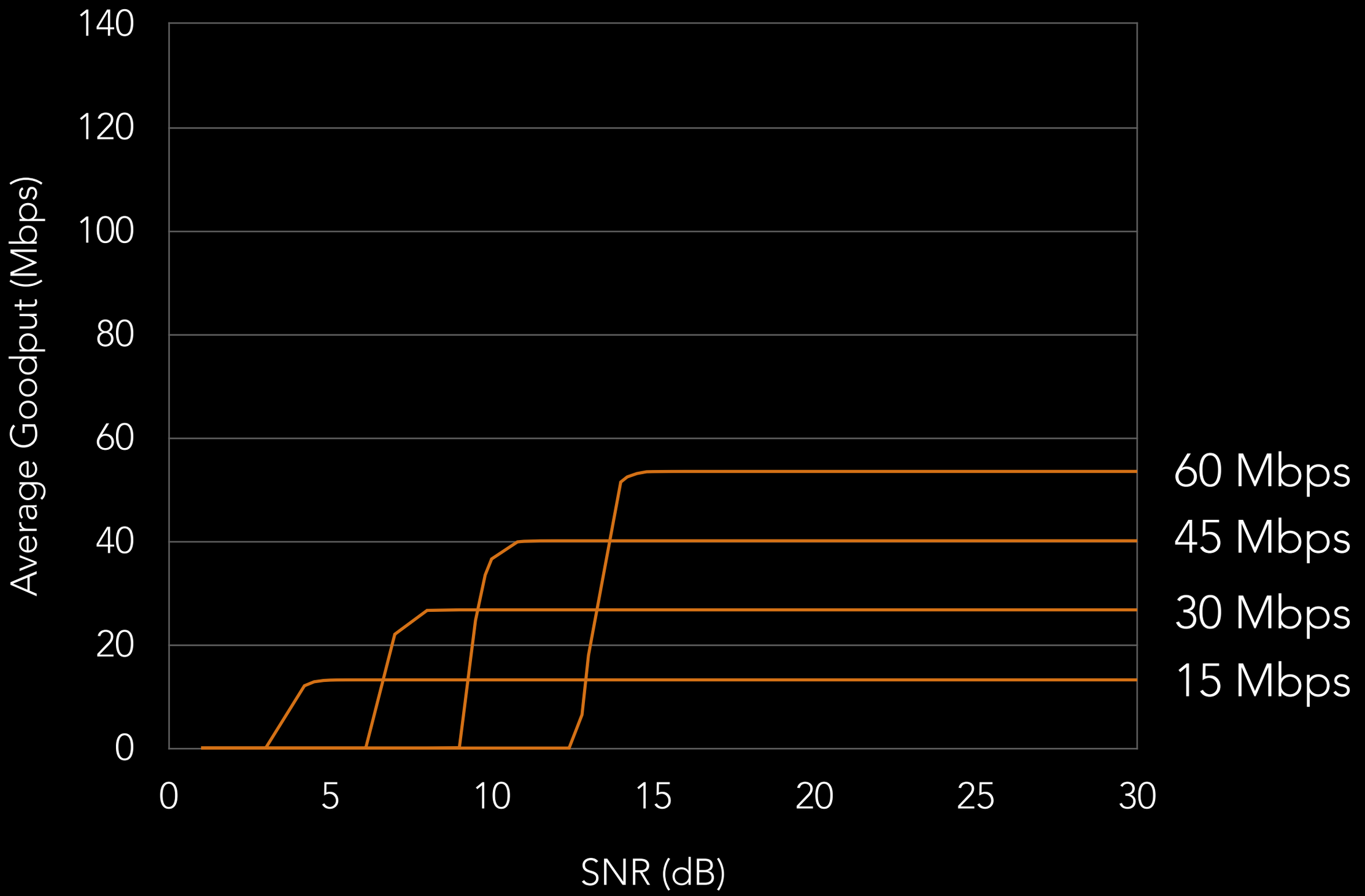
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



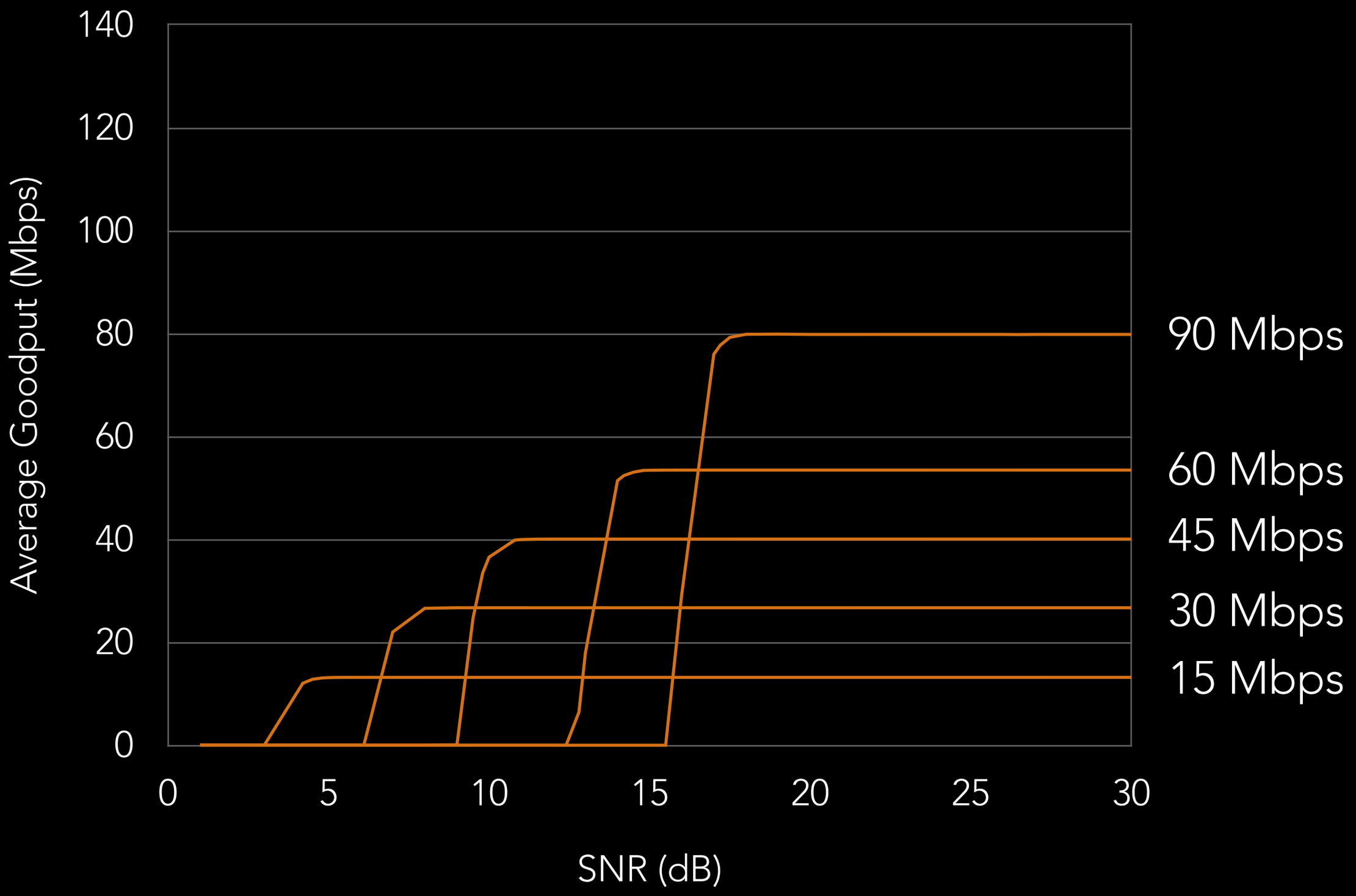
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



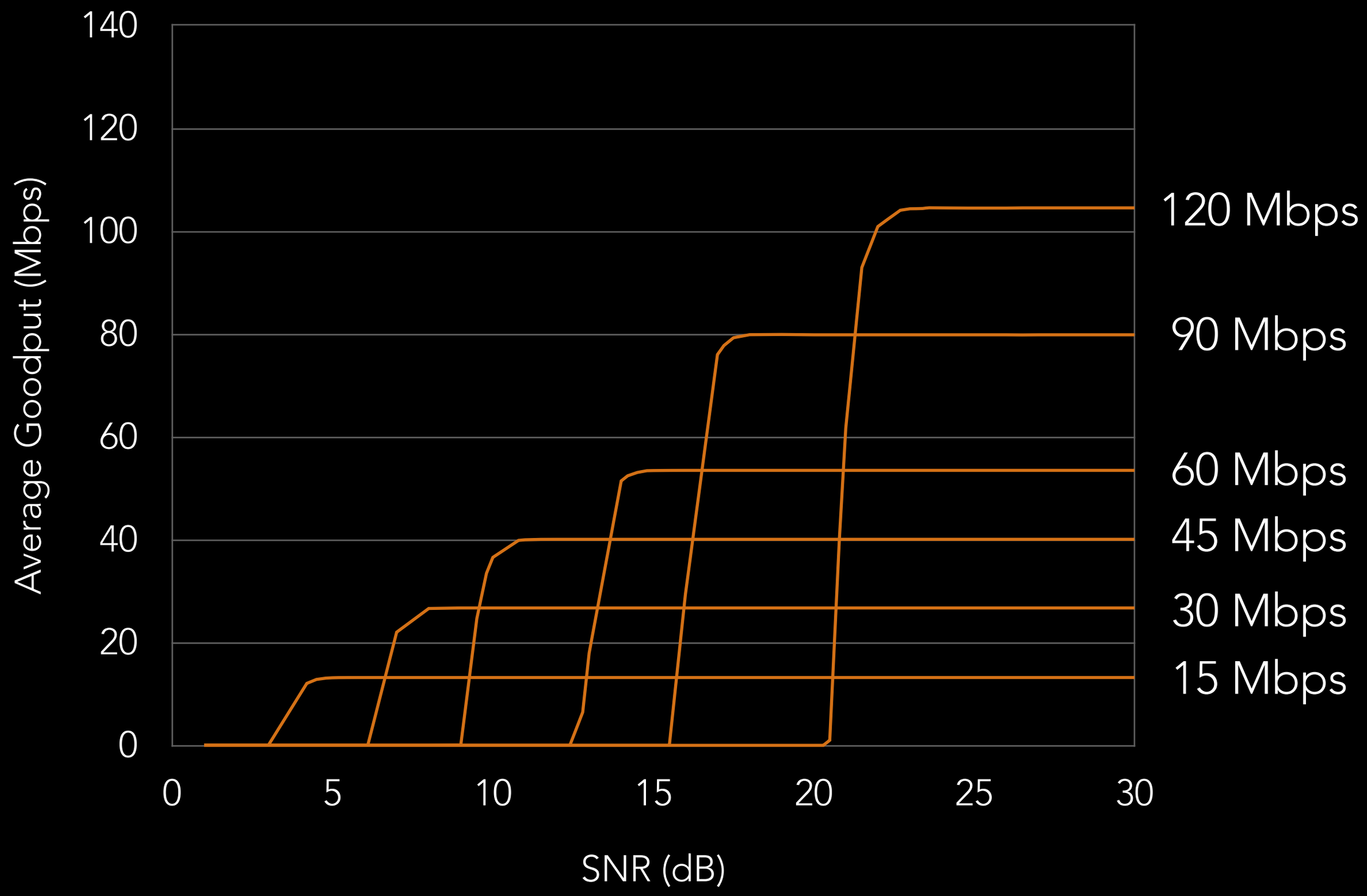
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



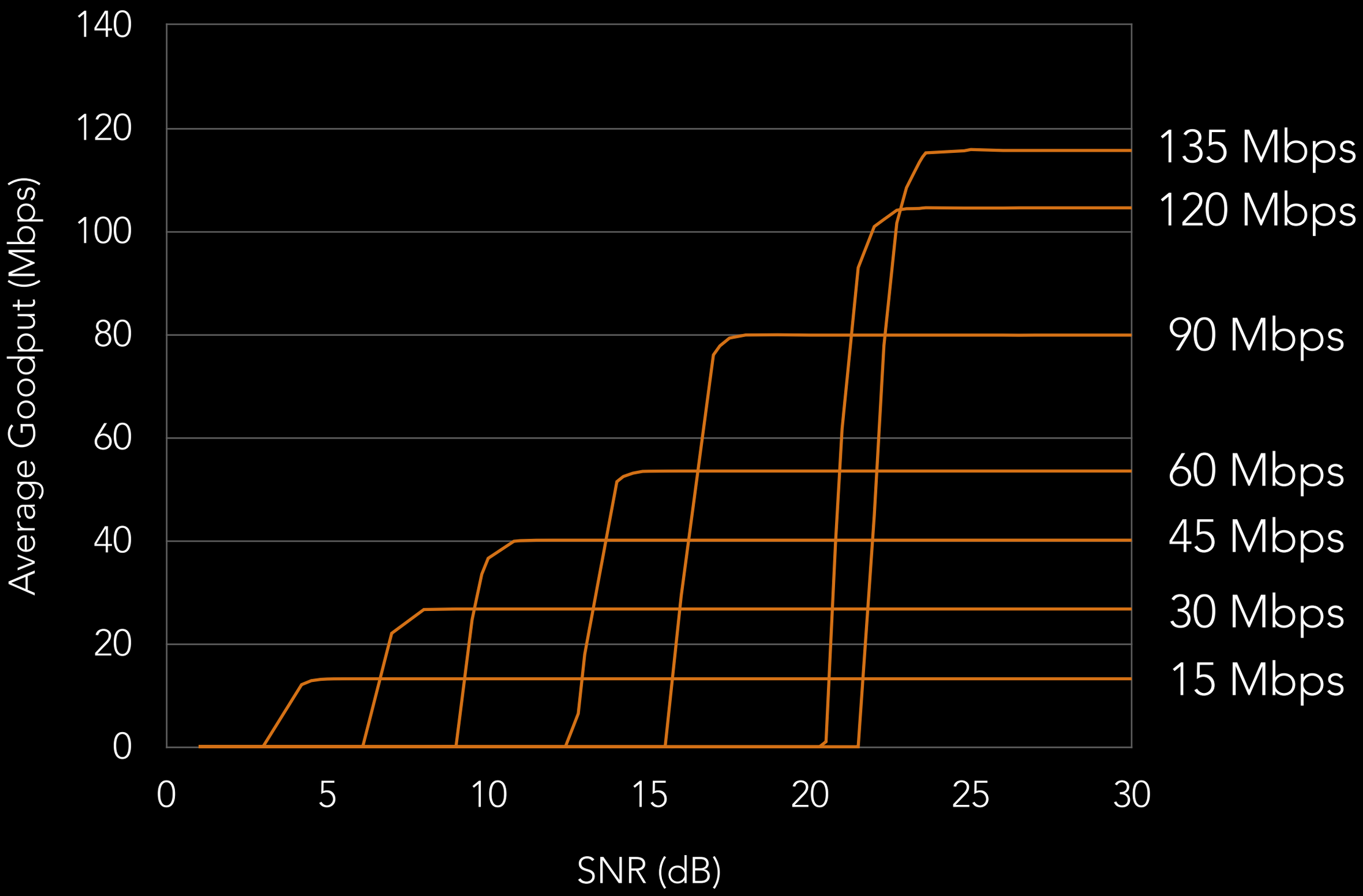
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



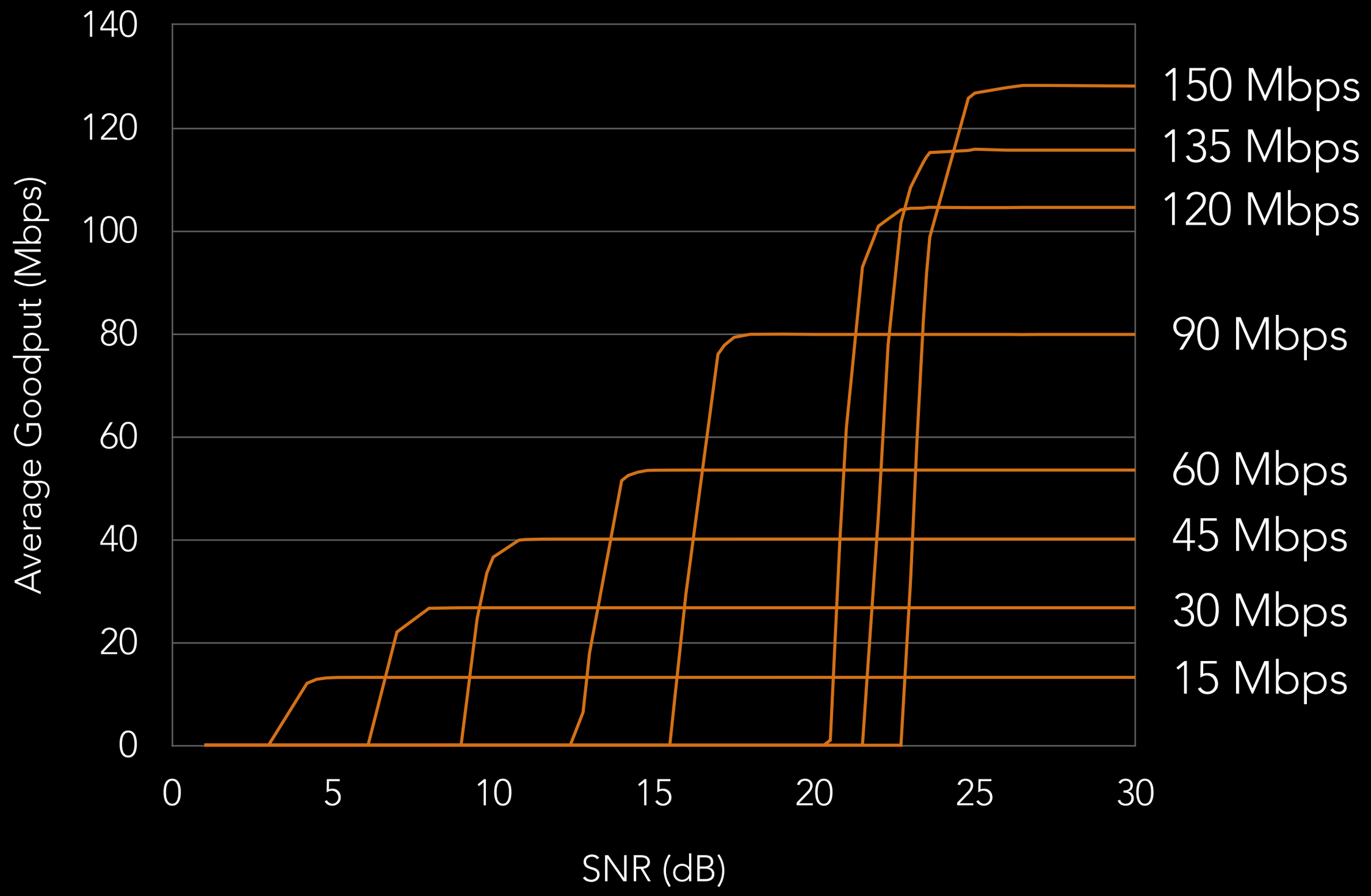
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



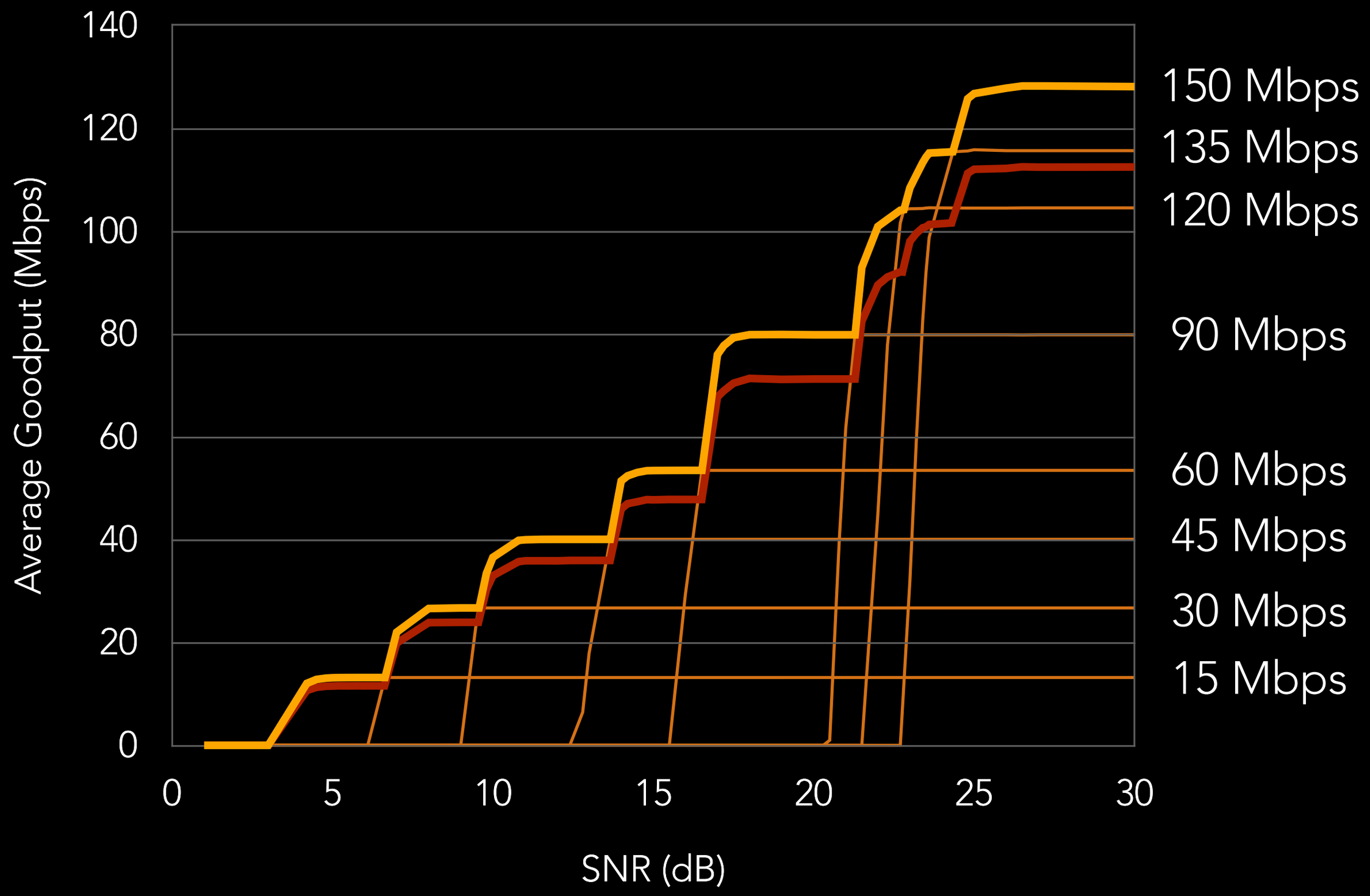
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



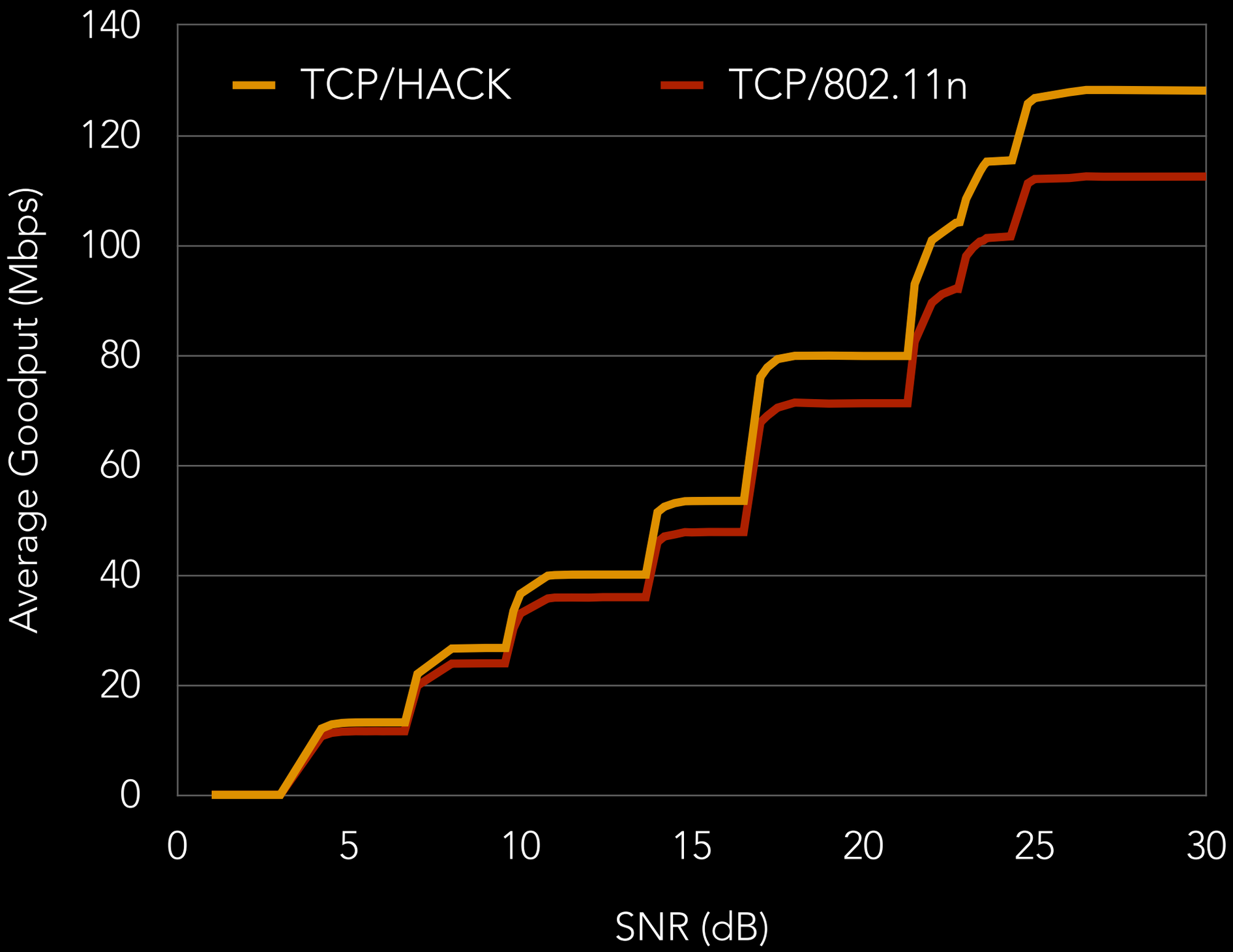
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



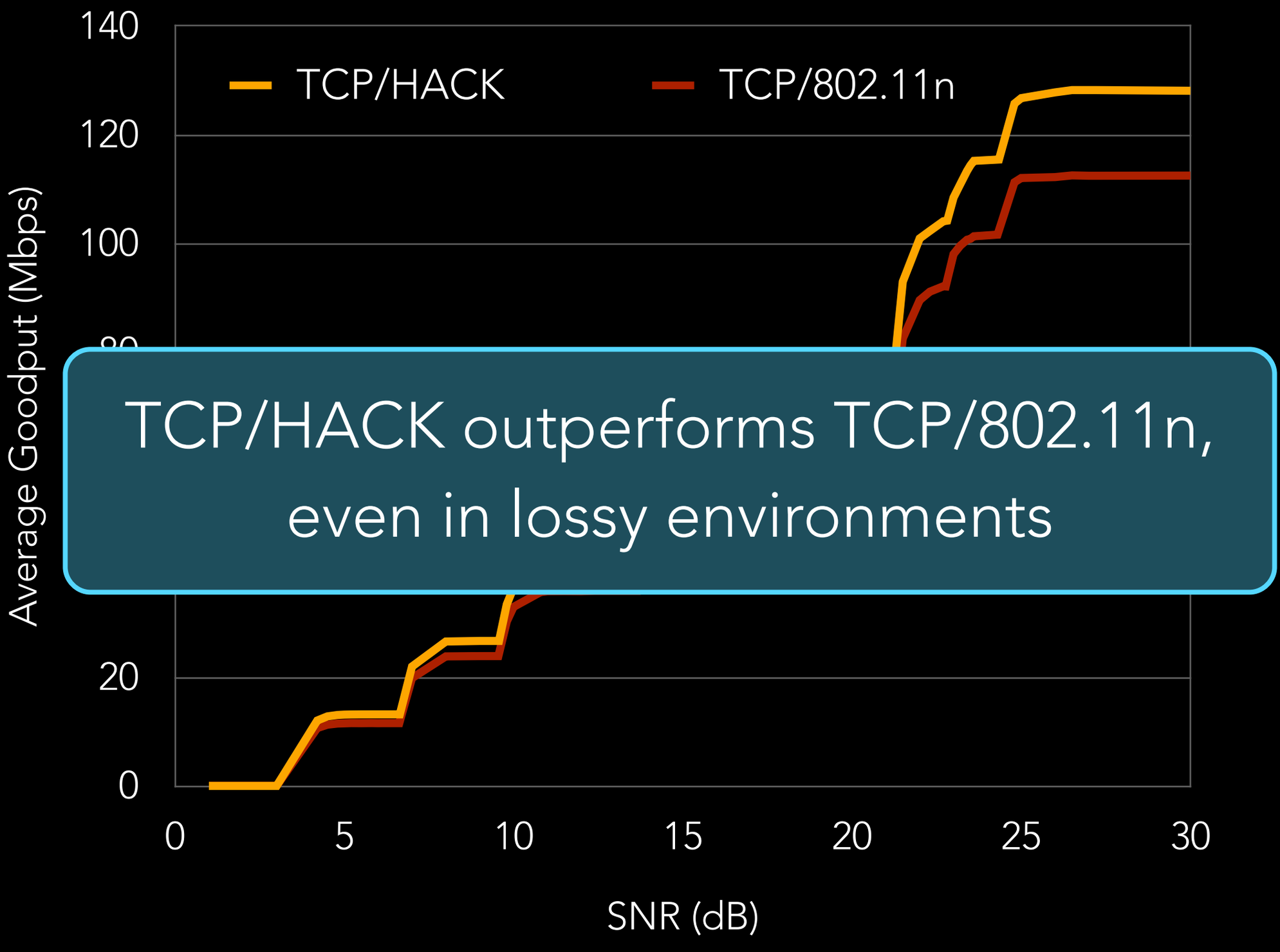
HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



HACK IMPROVES GOODPUT ACROSS FULL RANGE OF LINK RATES



TCP/HACK outperforms TCP/802.11n, even in lossy environments

CONCLUSIONS

- TCP/HACK reduces medium acquisitions.
- Increases goodput by:
  - ~29 - 32% with no aggregation
  - ~11 - 20% with aggregation
- Practical to deploy on real NICs.

http://www0.cs.ucl.ac.uk/staff/A.Zhushi/hack/index.html