画像情報特論 (4)
Advanced Image Information (4)

TCP over Wireless

情報理工学専攻　甲藤二郎

E-Mail: katto@waseda.jp

TCP over Wireless

Wireless issues

- error control (L1)
  - BER (bit error rate), PER (packet error rate)
  - error model: AWGN, Two-States Markov
- access control (L2)
  - CSMA/CA (MACA, MACAW):
    - interference, collision
    - hidden terminal, exposed terminal
    - grey zone: receive range & carrier sense range
  - TDMA, FDMA, CDMA
- ad-hoc routing (L3)
  - DSDV, DSR, AODV, OLSR, TORA, AOMDV, ...
- transport protocol (L4)
  - Wireless TCP/TFRC, multi-hop TCP/TFRC
- mobility management (L3 / L7)
  - Mobile IP (L3), SIP mobility (L7)
- energy consumption (all layers)
  - energy model

Discussion

- Wireless LAN
  - CSMA/CA, half-duplex, interferences, random errors, ...
    - cannot send packets when the sender wants to
    - packets are continuously stored into a transmission buffer of the sender
  - NIC buffer size is very large
    - Hybrid TCP always operates in the loss mode only
  - Unfairness between upload and download
    - D.Leith: WiOpt 2005

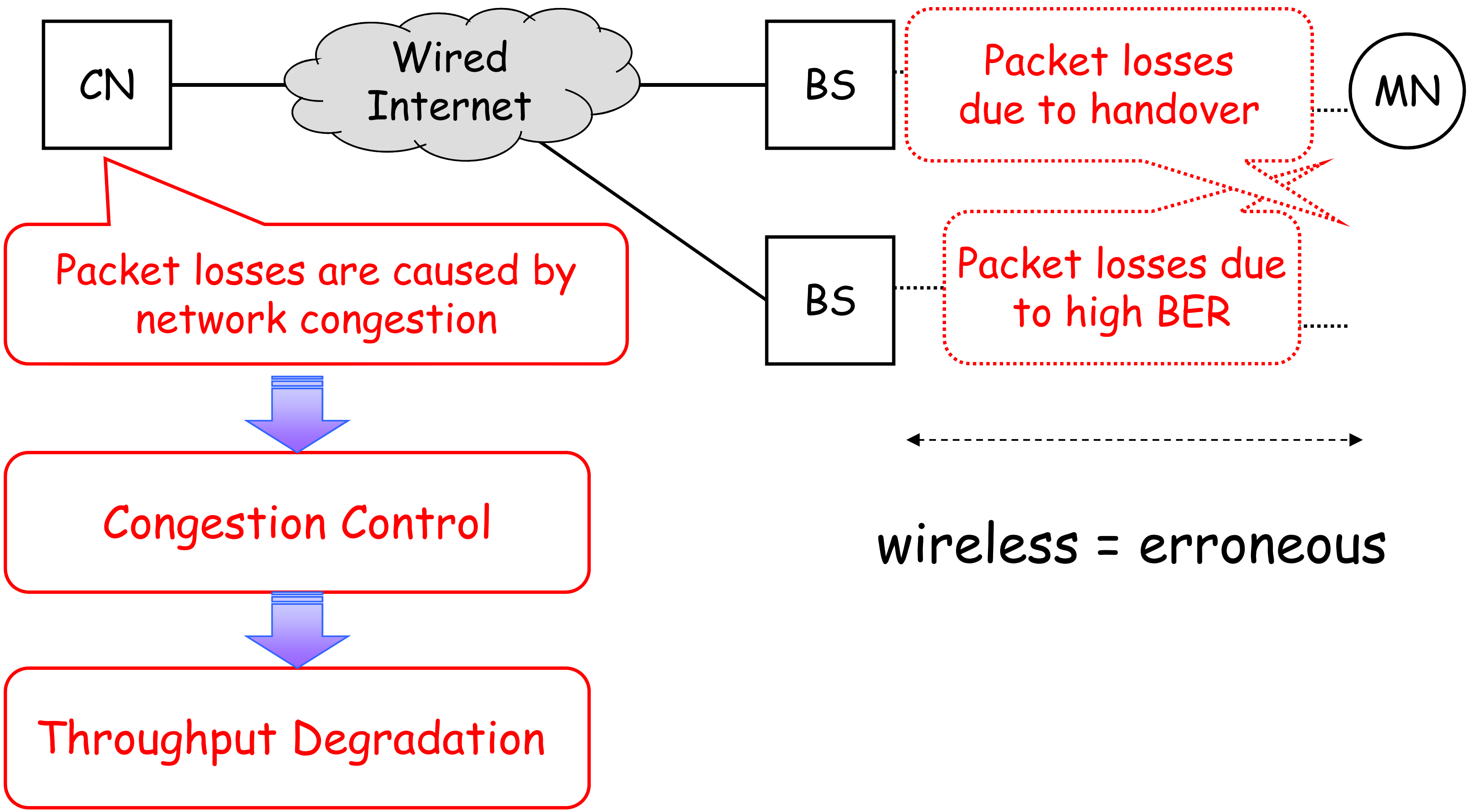
TCP over Wireless
(1) Error Management

Summary

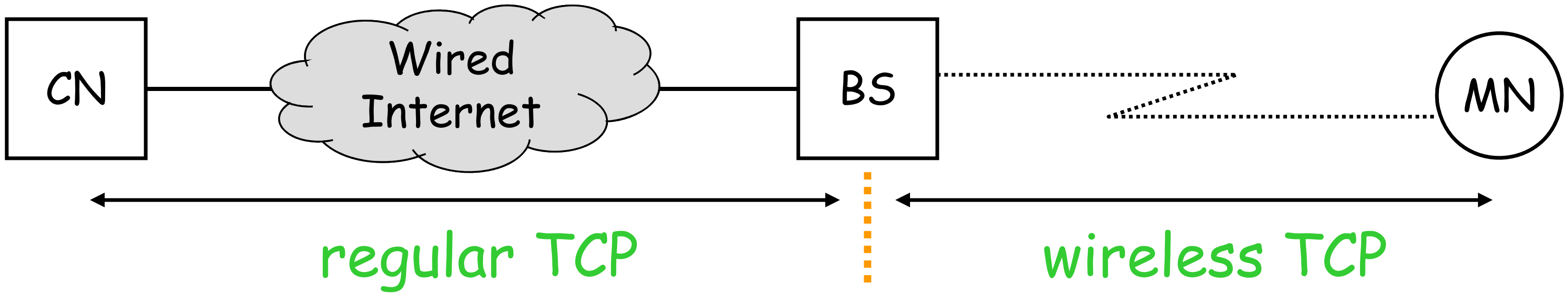
- TCP Extensions for Mobile Networks
  - Split Connection: Indirect TCP
  - Proxy: Snoop TCP
  - End-to-End: Freeze TCP

- L2/L4 collaboration (practical)

TCP over Mobile Networks



(1) Split Connection

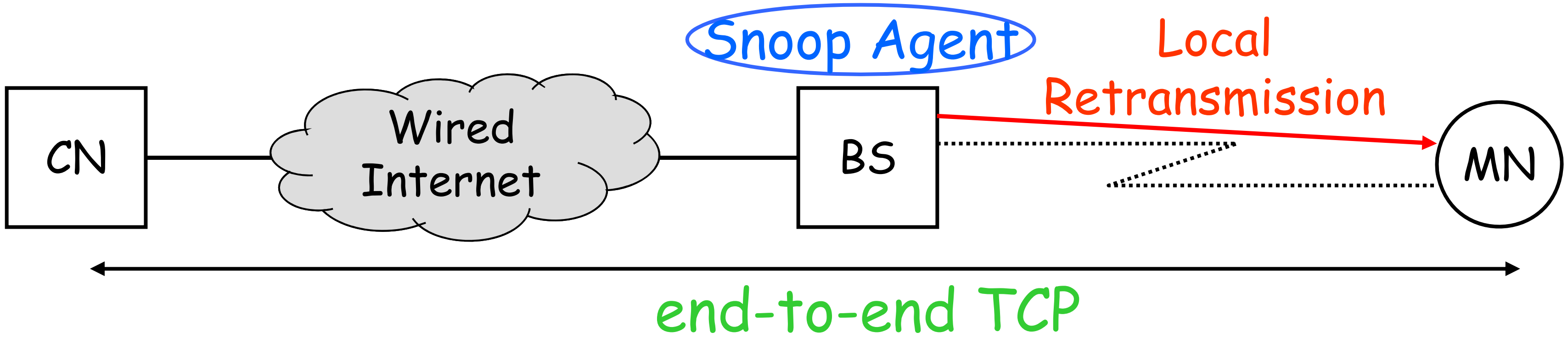


Transmission errors over wireless links are not propagated to wired networks

Forces heavy load on a base station

Breaks end-to-end semantics

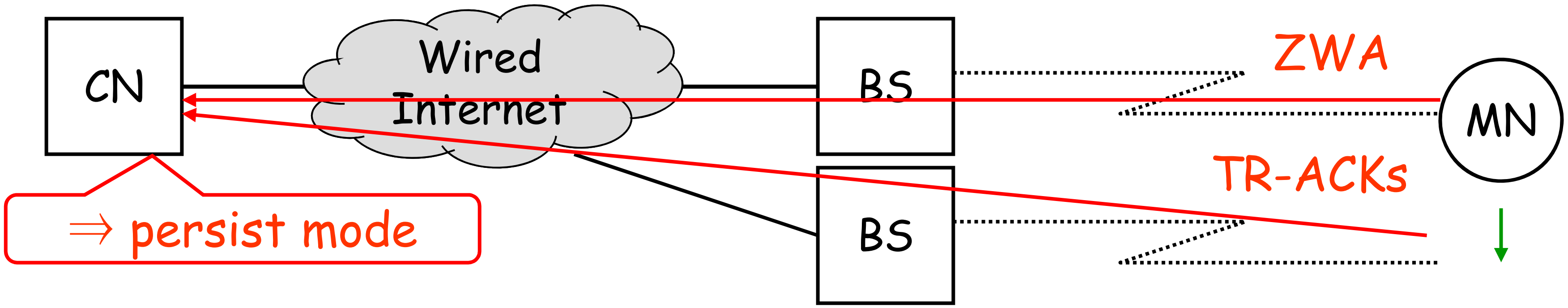
(2) Proxy



Local retransmission over wireless links avoids initiation of unnecessary congestion avoidance

Forces heavy load on a base station

(3) Freeze TCP



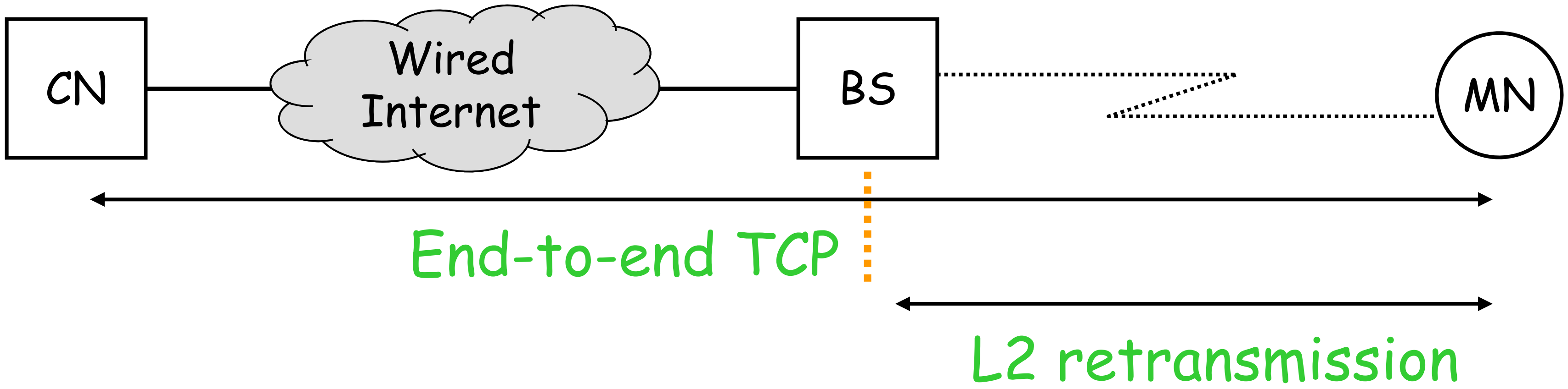
Does not need any base station support

Only TCP of mobile nodes should be modified

A mobile node has to predict a link break precisely before the actual break happens

ZWA: Zero Window Advertisement

(4) L2/L4 Collaboration



Packet losses of wireless links are retransmitted by L2 protocols (e.g. IEEE 802.11, 3G/LTE)

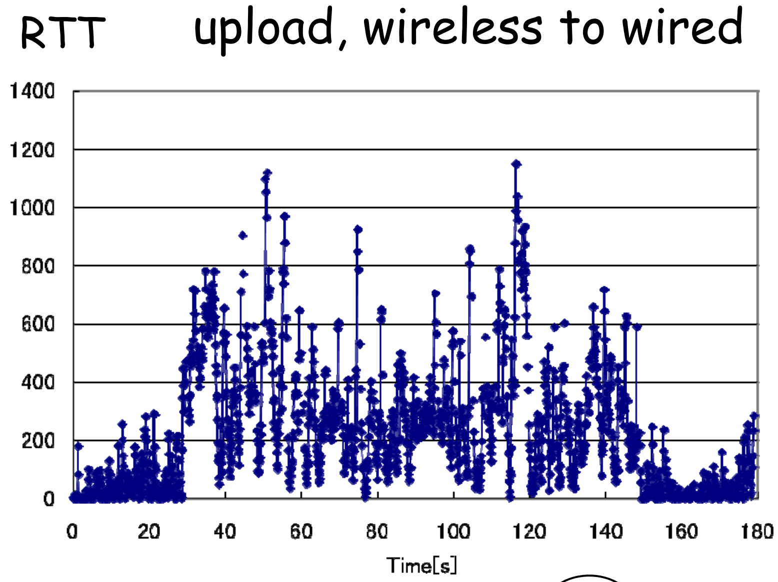
Default retransmission count depends on products: Cisco AP: 32, Buffalo: 4, or adaptive decision

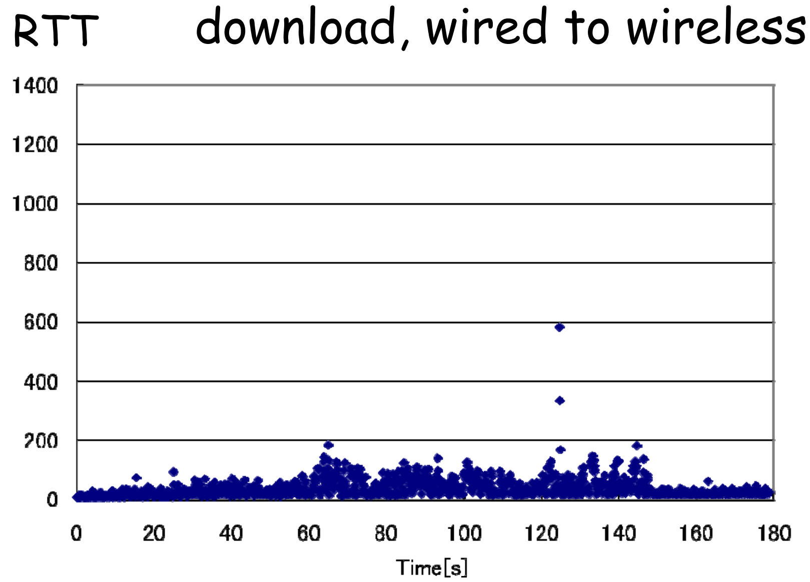
L2 retransmission generally increases latency due to its backoff mechanism

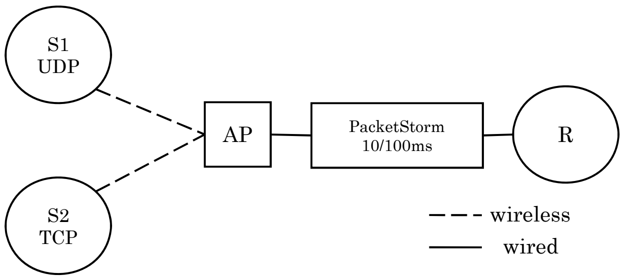
TCP over Wireless Networks
(2) TCP Differentiation

WiFi Example

- RTT instability and unfairness between upload and download





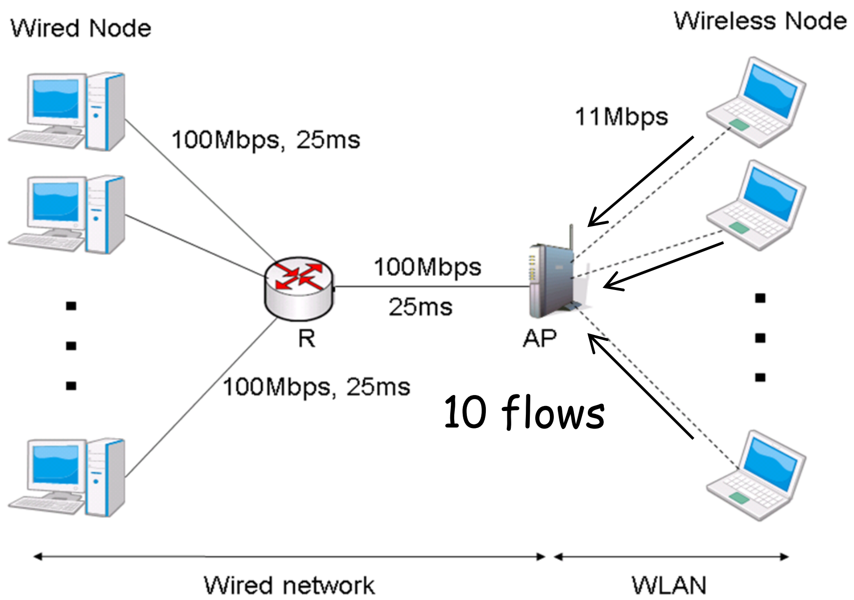


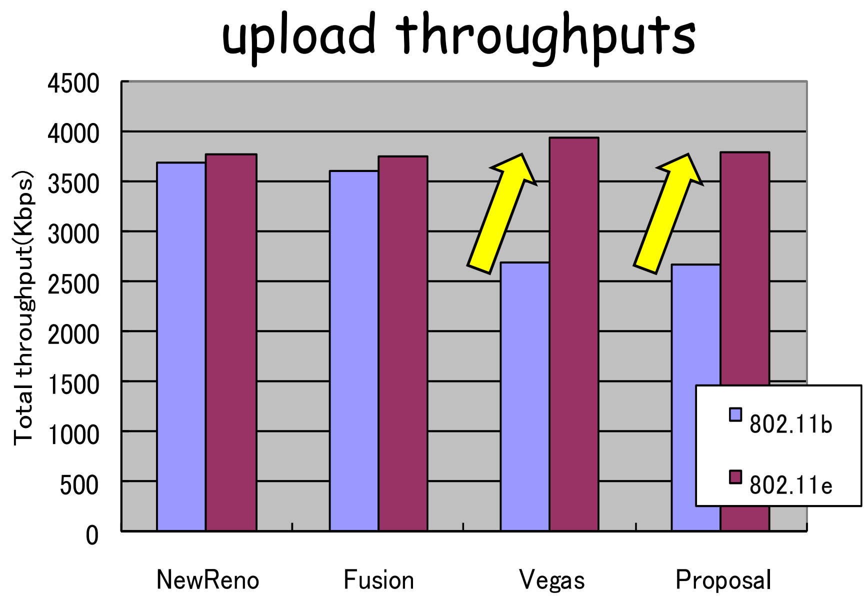
K.Kanai et al: "Performance evaluations of adaptive rate control mechanisms ...", IEICE Tech. Report, 2011.

Wireless LAN (1)

※ ns-2 simulation

- TCPs and throughputs





Apply IEEE 802.11e to alleviate the unfairness problem between upload and download

TCP-Reno: loss based

TCP-Fusion: hybrid

TCP-Vegas: delay based

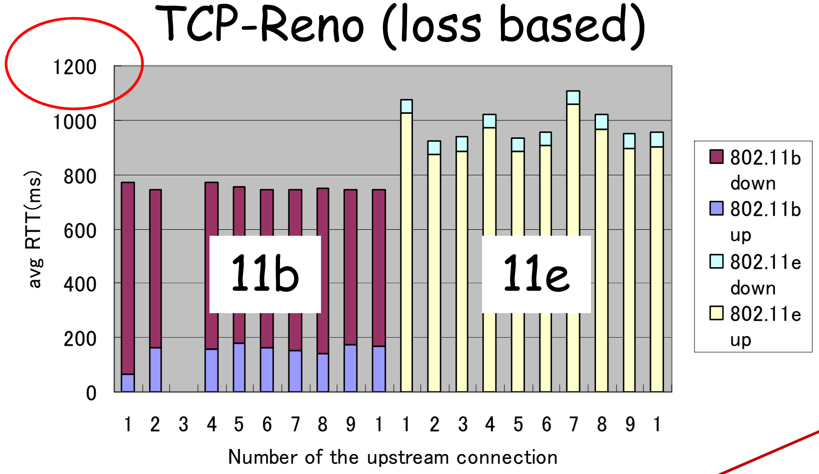
Proposal: Vegas extension

K.Sonoda et al: "TCP Differentiation using Version Identification...", IEEE CCNC 2013.

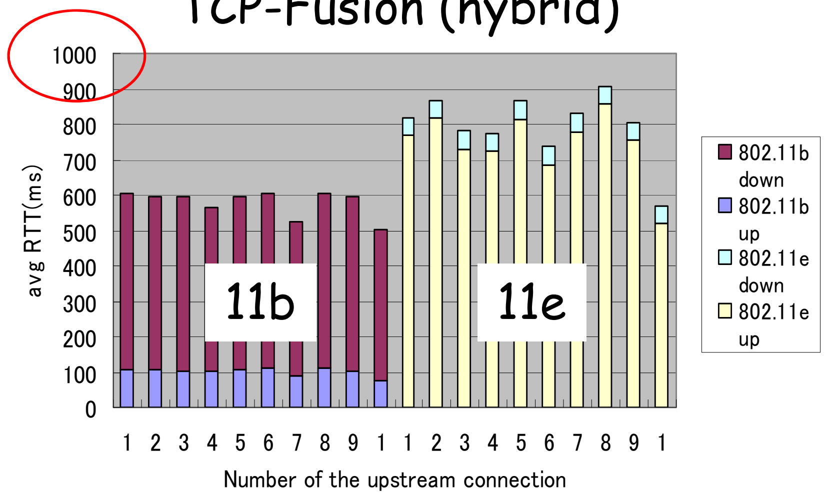
Wireless LAN (2)

- TCPs and delays

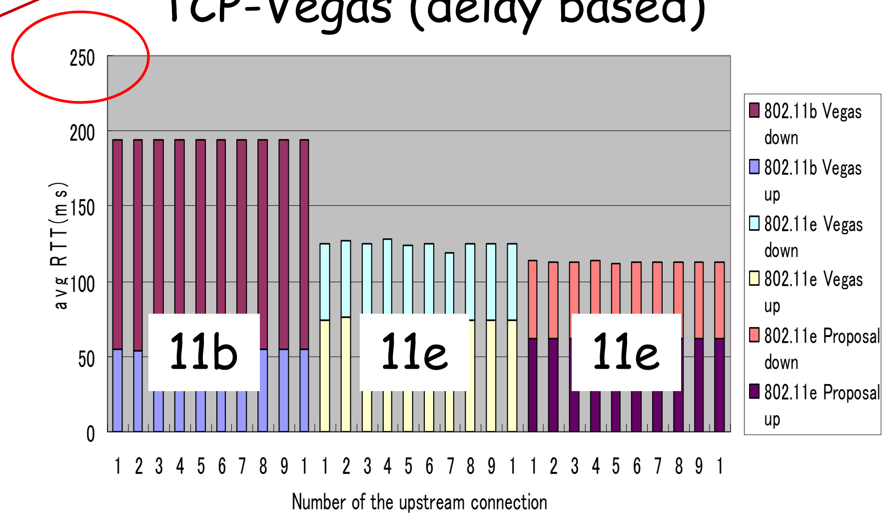
TCP-Reno (loss based)



TCP-Fusion (hybrid)



TCP-Vegas (delay based)



Reno, Fusion: though unfairness was alleviated, delay increases (esp. upload)
Vegas & Proposal: unfairness and delay are decreased (compare vertical axis)

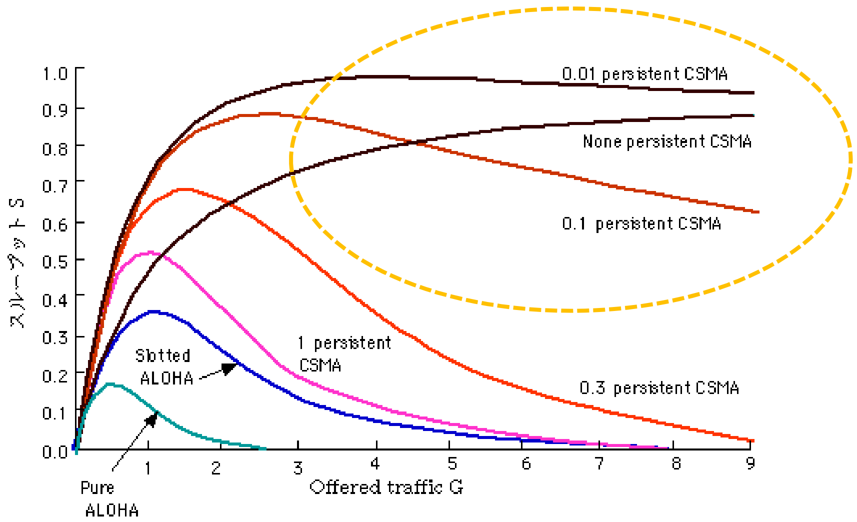
→ Hybrid TCP works in loss mode only

K.Sonoda et al: "TCP Differentiation using Version Identification...", IEEE CCNC 2013.

Wireless LAN (3)

- Common to wired
  - Delay based TCP design is effective if we require low delay transmission (but, it is expelled by loss based flows)
- Differences to wired
  - Hybrid does not operate in “hybrid” (delay mode) due to huge transmission buffer
  - Too many packet insertion causes huge delay due to multiple access mechanism (i.e. CSMA)

Critical throughput-delay tradeoff due to CSMA/CA



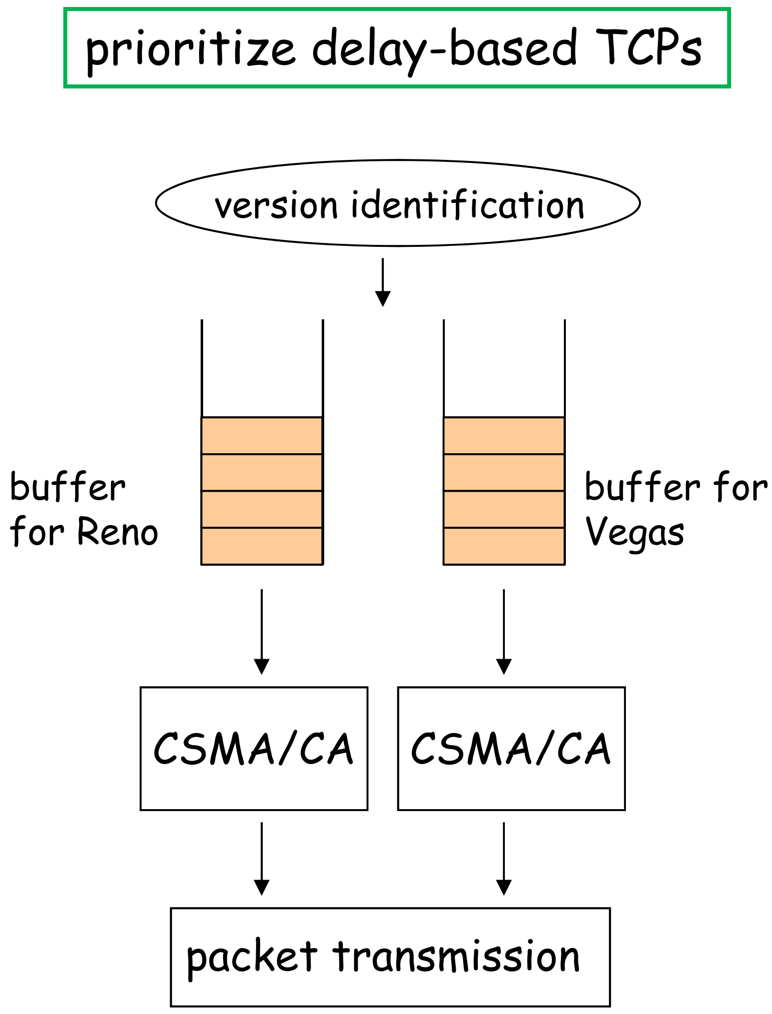
TCP Version Differentiation (1)

TCP version identification and differentiation

1. Access points identify TCP versions using RTT/cwnd estimation

2. Access points separate different TCP versions into different buffers

3. Prioritize delay based TCP flows by tuning CSMA/CA parameters of IEEE 802.11e

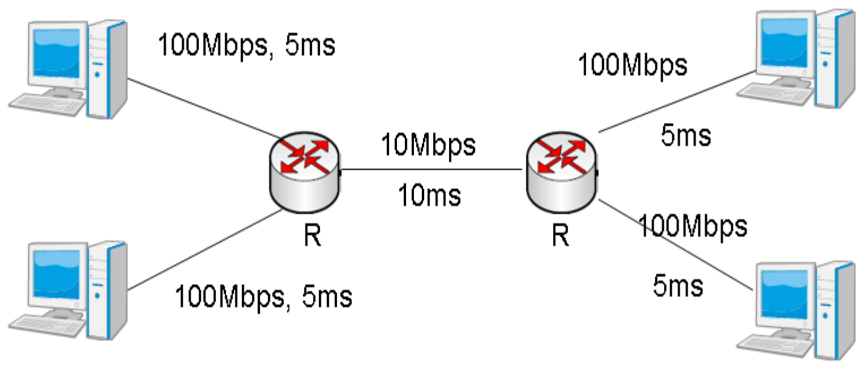


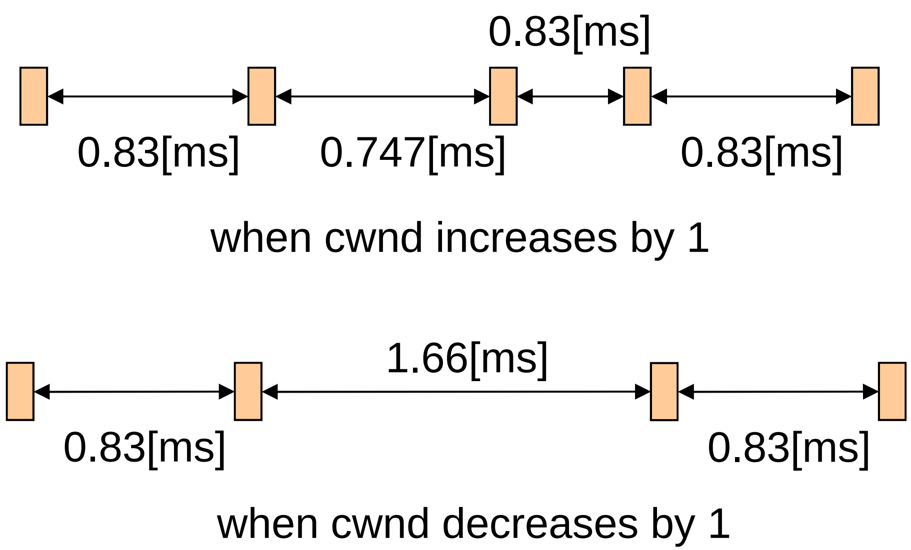
K.Sonoda et al: "TCP Differentiation using Version Identification…", IEEE CCNC 2013.

TCP Version Differentiation (2)

- RTT estimation for delay based flow
  - When cwnd increases by one, two consecutive packets are transmitted
  - When cwnd decreases by one, no packets are transmitted for the last ACK
- cwnd estimation
  - Access points let the number of arrived packets per RTT be "cwnd"

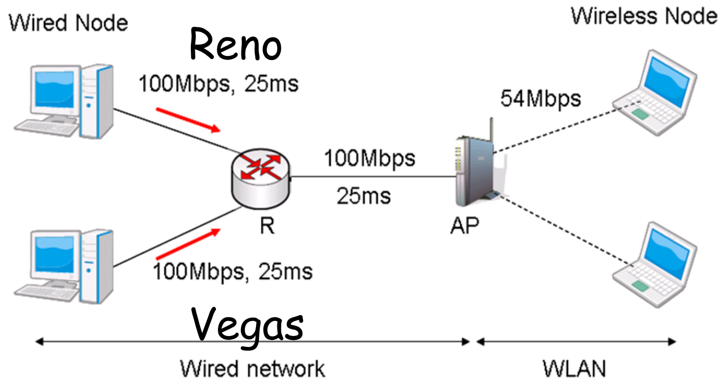
TCP behavior estimation at AP



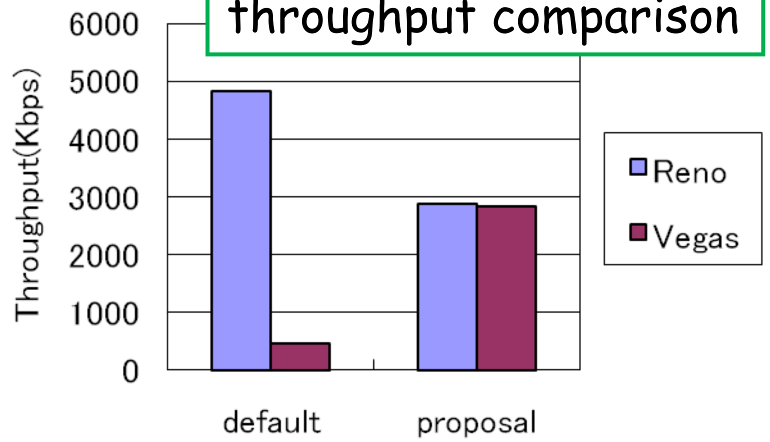


K.Sonoda et al: "TCP Differentiation using Version Identification...", IEEE CCNC 2013.

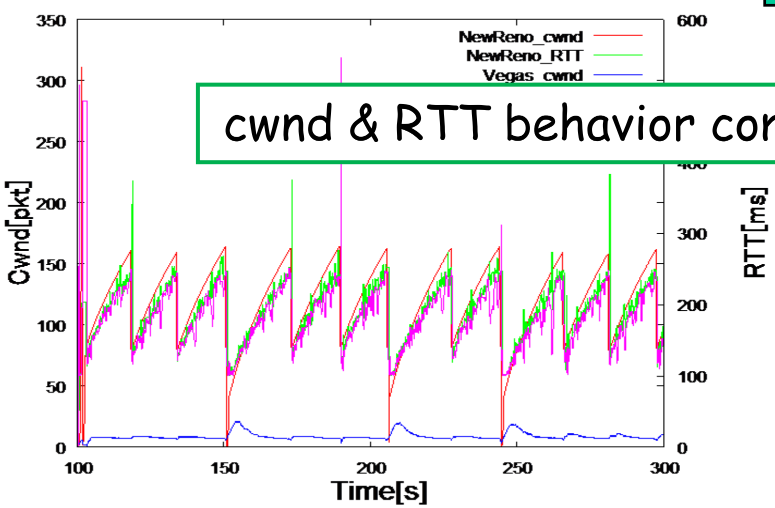
TCP Version Differentiation (3)



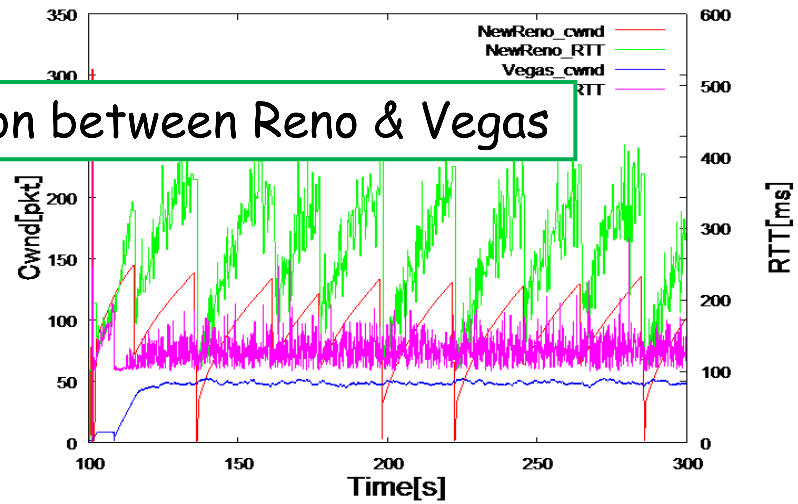
throughput comparison



cwnd & RTT behavior comparison between Reno & Vegas



without differentiation



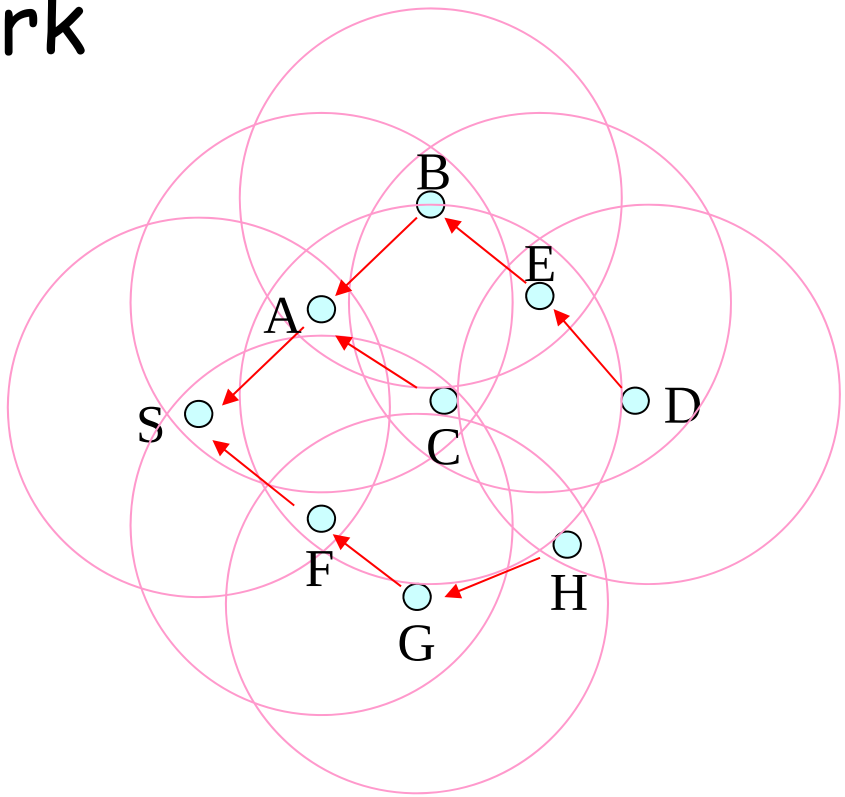
with differentiation

K.Sonoda et al: "TCP Differentiation using Version Identification…", IEEE CCNC 2013.

TCP over Wireless Networks
(3) Multi-hop

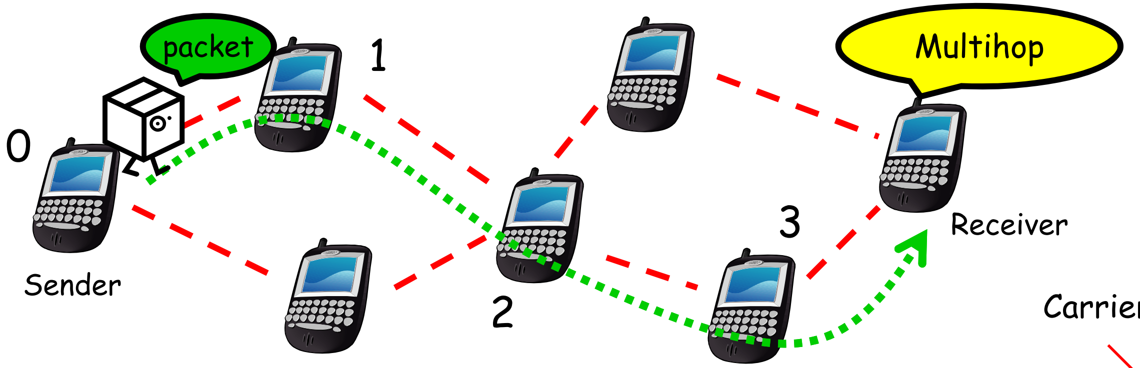
Wireless Multihop Networks

- ad-hoc network
- sensor network

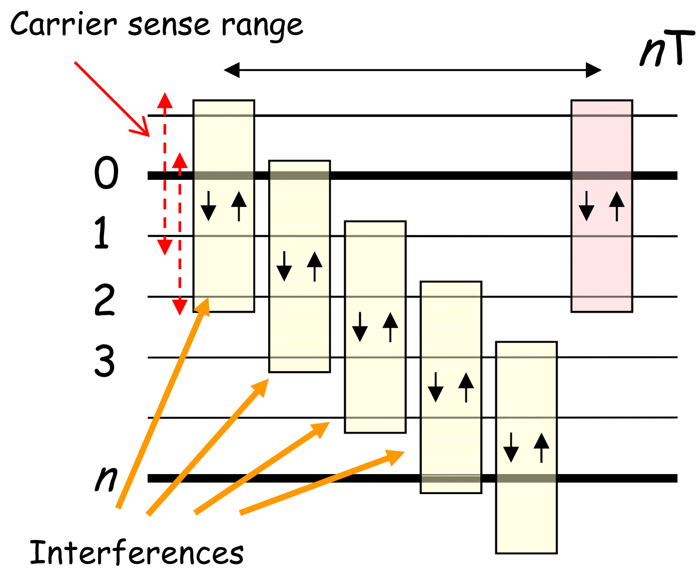


Wireless Multihop Networks (1)

- Single Radio Multi-hop Transmission



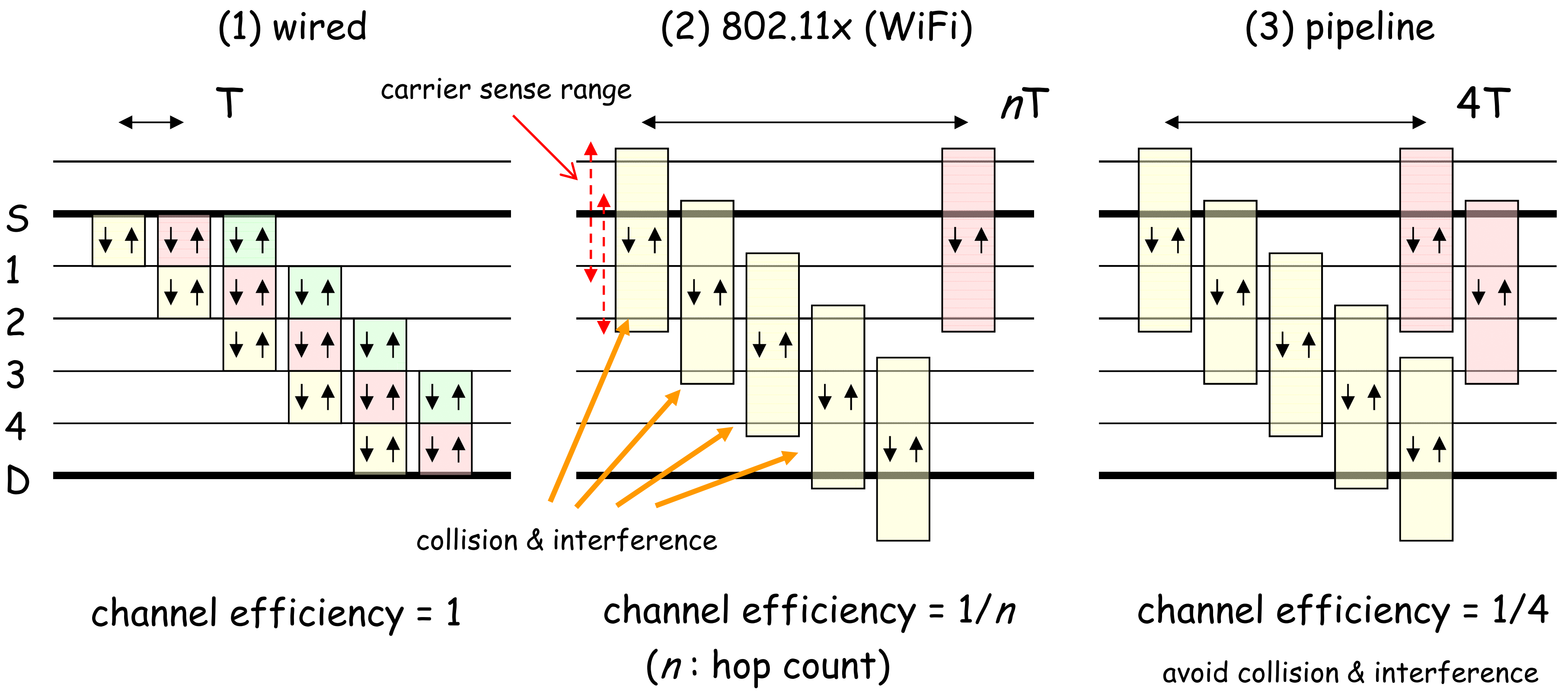
Decrease of link utilization due to radio interferences



Link utilization ratio can be at most 1/4 (or 1/*n* without pipelining, where *n* = # of hops) (J.Li et al.: *ACM Mobicom 2001*)

Small packet buffering at the intermediate nodes (Z.Hu et al: IEEE INFOCOM 2003)

Multihop Capacity (1)

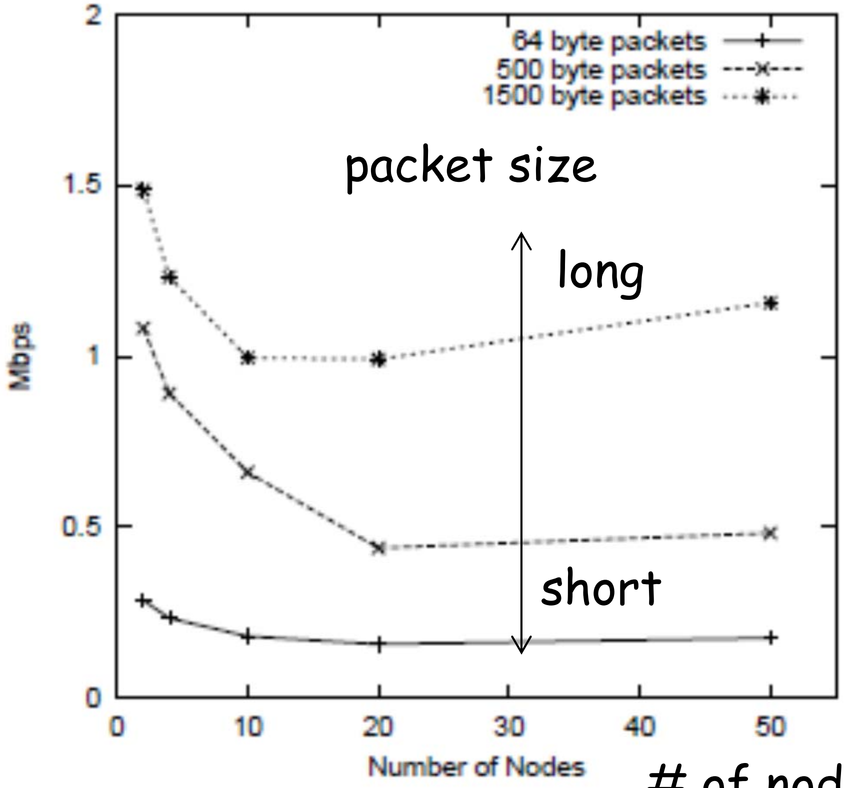


J.Li et al: "Capacity of Wireless Ad Hoc Networks", ACM Mobicom 2001.

Multihop Capacity (2)

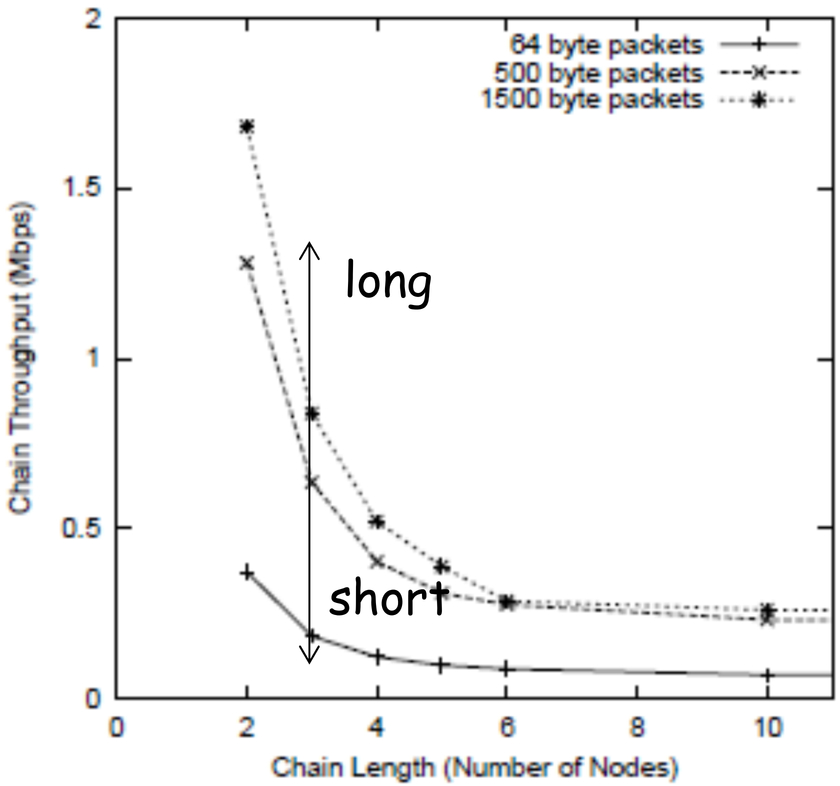
throughput

2Mb/s link



# of nodes

Cell (2D)



# of hops

Chain (1D)

J.Li et al: "Capacity of Wireless Ad Hoc Networks", ACM Mobicom 2001.

Link RED & Adaptive Pacing

- Wireless capacity is limited by # of hops (1/4 is the theoretical maximum channel efficiency for chain topology)

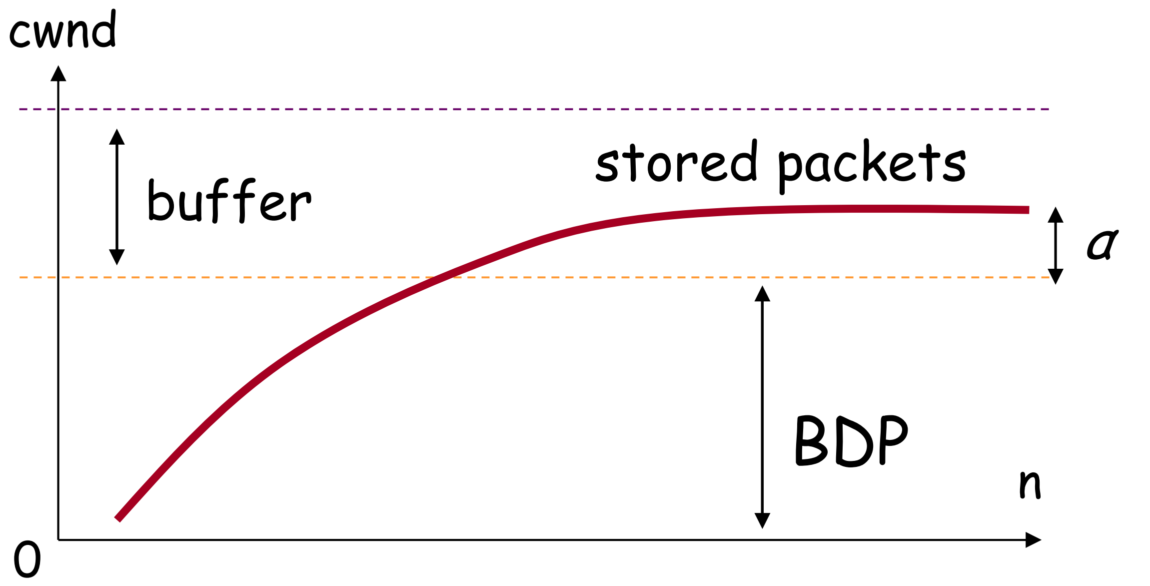
- Distributed Link RED: drops packets randomly at the link level when link load increases (analogous to random early detection)
- Adaptive Pacing: controls packet transmission scheduling in order to approach 1/4 (spatial channel reuse)

	TCP NewReno	LRED+
flow 1	244 Kbps	166 Kbps
flow 2	0 Kbps	153 Kbps
Aggregate	244 Kbps	319 Kbps
Fairness	0.5	0.9983

	TCP NewReno w/standard LL	TCP NewReno w/LL+LRED+PACING
flow 1	532 Kbps	85512 Kbps
flow 2	126229 Kbps	90459 Kbps
flow 3	115554 Kbps	70334 Kbps
flow 4	1608 Kbps	47946 Kbps
Aggregate	242923	294251
Fairness	0.51	0.95

Z.Hu et al: “The Impact of Multihop Wireless Channel on TCP Throughput and Loss”, IEEE INFOCOM 2003.

TCP-Vegas (revisited)



e.g. α=1, β=3

$$diff = \left(\frac{cwnd}{RTT_{\min}} - \frac{cwnd}{RTT}\right) \cdot RTT_{\min}$$

stored packets in a buffer

increase: $$cwnd = \begin{cases} cwnd + 1 & diff < \alpha \\ cwnd & otherwise \\ cwnd - 1 & diff > \beta \end{cases}$$

decrease: $$cwnd = cwnd * 0.75$$

Vegas-W (1)

for wireless multihop

- Vegas-W [Ding, C&C 2008]
  - **Slower window increase than TCP-Vegas**

$$cwnd = \begin{cases} cwnd + 1/cwnd & (\Delta < \alpha \ \& \ n_{CA} > N_{CA}) \\ cwnd & (\alpha \le \Delta \le \beta \text{ or } \Delta \le \alpha \ \& \ n_{CA} \le N_{CA}) \\ cwnd - 1/cwnd & (\Delta > \beta) \end{cases}$$

n_{CA}: # of consecutive states entering into

$(\alpha \le \Delta \le \beta \text{ or } \Delta \le \alpha \ \& \ n_{CA} \le N_{CA})$

N_{CA}: threshold (e.g. 100)

→ much slower than TCP-Vegas

L.Ding et al.: "Improve throughput of TCP-Vegas in multihop ad hoc networks", Computer Communications, Jun.2008.

Vegas-W (2)

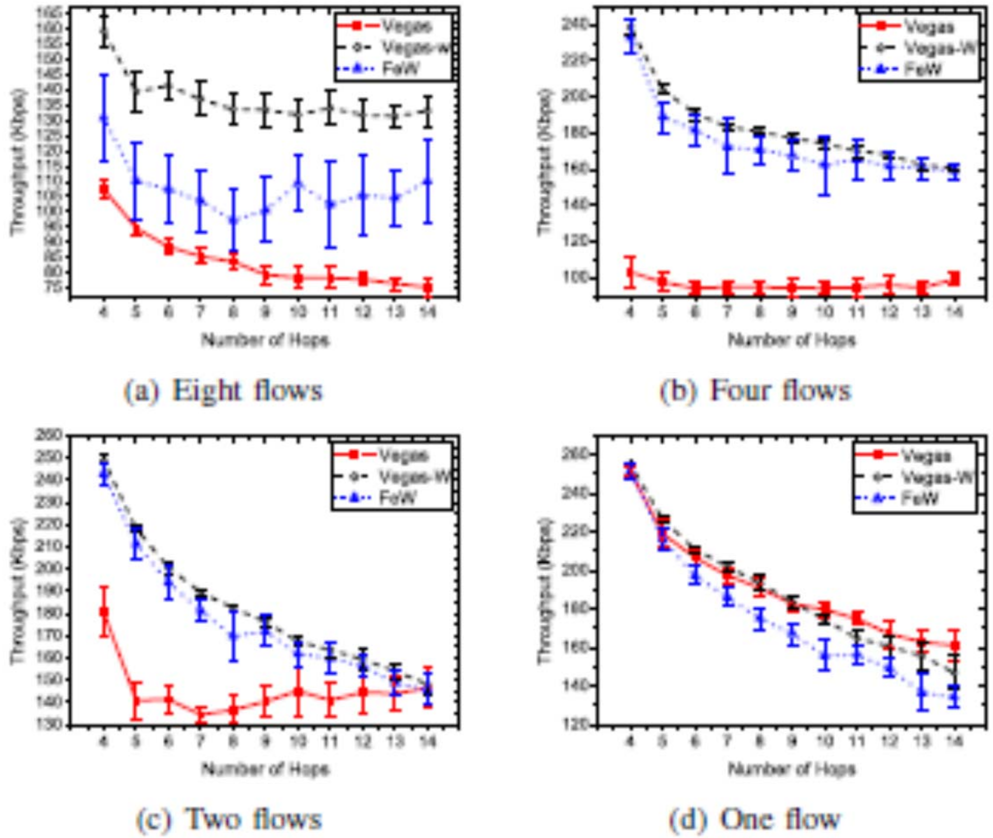


Fig. 4: Throughput comparison over chain topology with DSR and 95% confidence interval.

FeW: Fractional Window Increment (ACM Mobihoc 2005)

Vegas degraded as # of flows increases

Vegas-W improves as # of flows increases

L.Ding et al.: "Improve throughput of TCP-Vegas in multihop ad hoc networks", Computer Communications, Jun.2008.

Summary of Wireless Multihop Networks

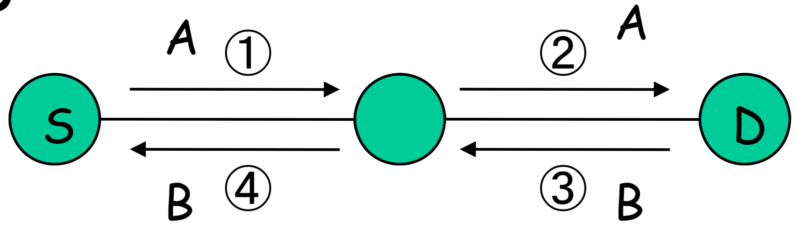
- Common to wired & wireless LAN
  - delay-based TCP is effective as long as no competing loss-based flows exist
- Gap to the wired case
  - wired case: faster window increase "immediately" fills a pipe
  - multi-hop case: slower window increase "safely" fills a pipe

A,B: symbol
a,b: signal

(ref.) Wireless Network Coding

- Multihop

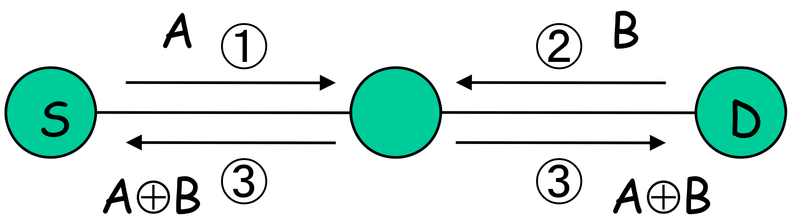
# of time-slots which is necessary to transmit packets A&B between source and destination



4 (channel efficiency1/4)

- Network Coding (in Wireless)

⊕: XOR

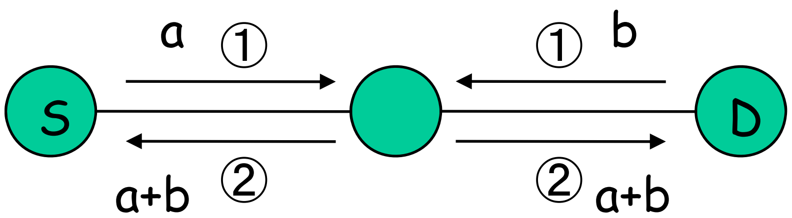


3 (channel efficiency 1/3)

B=A⊕(A⊕B)

A=B⊕(A⊕B)

- Physical-Layer Network Coding



2 (channel efficiency 1/2)

synchronization is the key point

S.Zhang: "Physical-Layer Network Coding", ACM Mobicom 2006.