Multimedia Transport Technologies over Wired/Wireless Networks

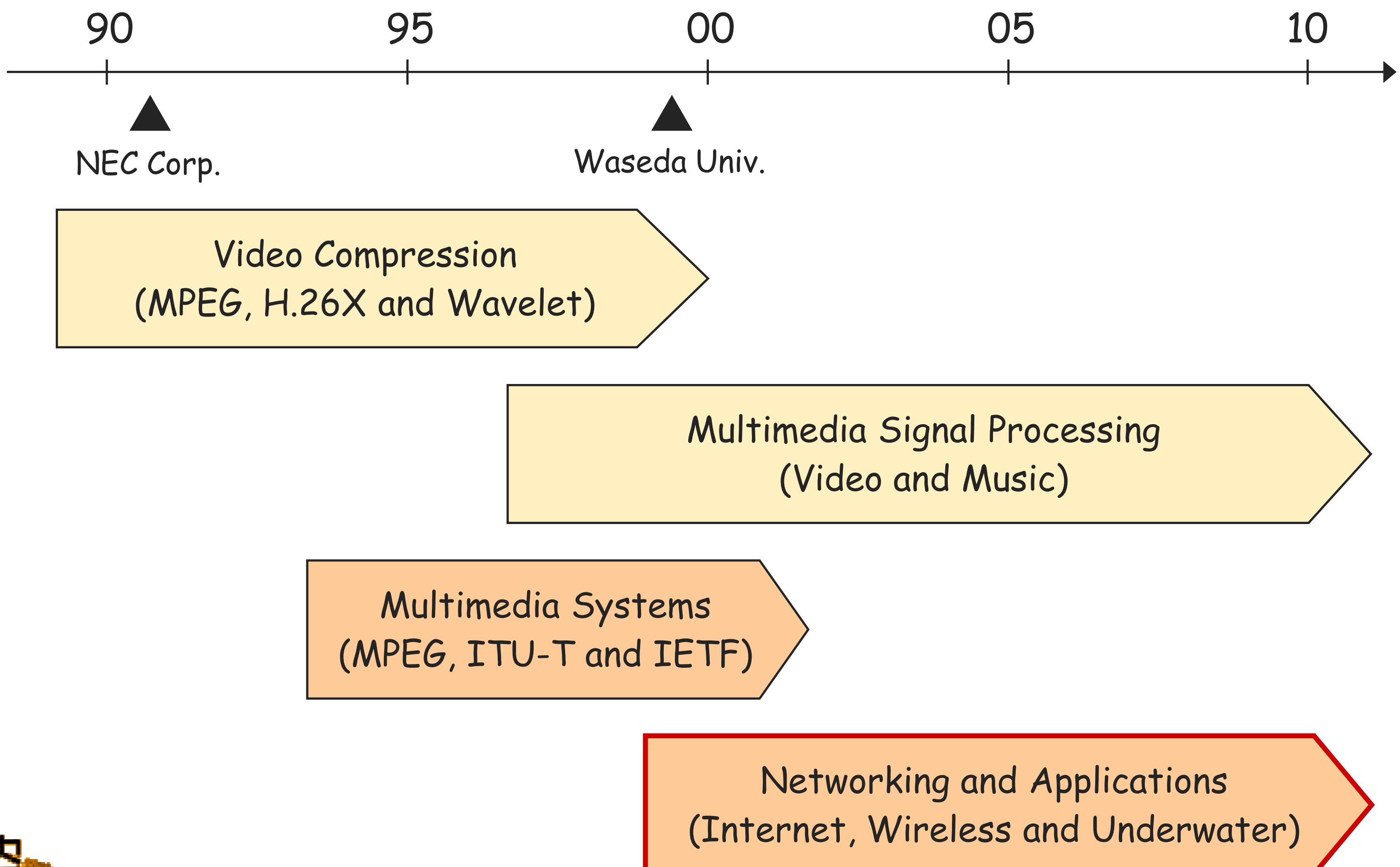
Jiro Katto

Dept. of Computer Science and Engineering,

Waseda University

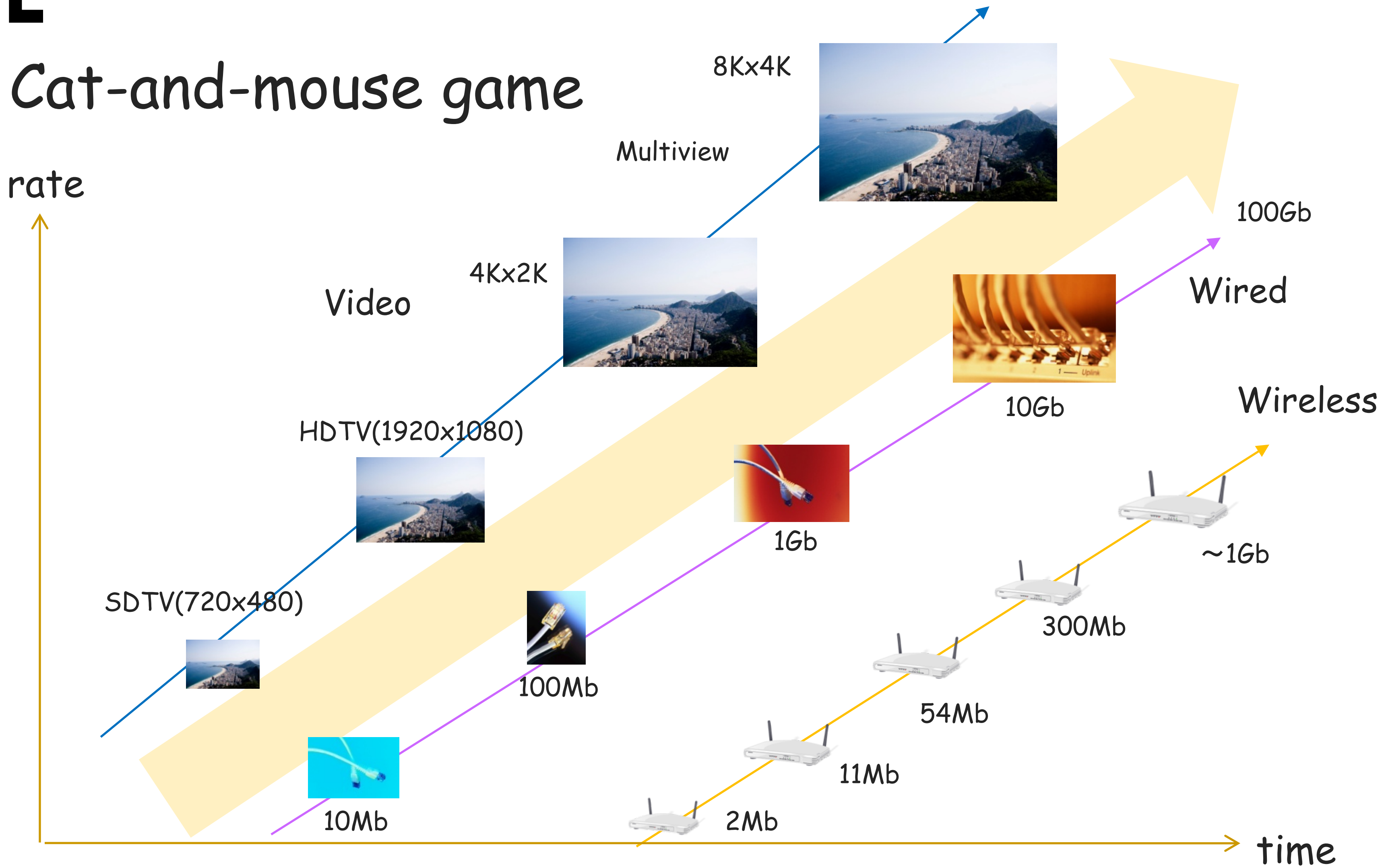
1. Introduction

Self Introduction



Networks and Multimedia

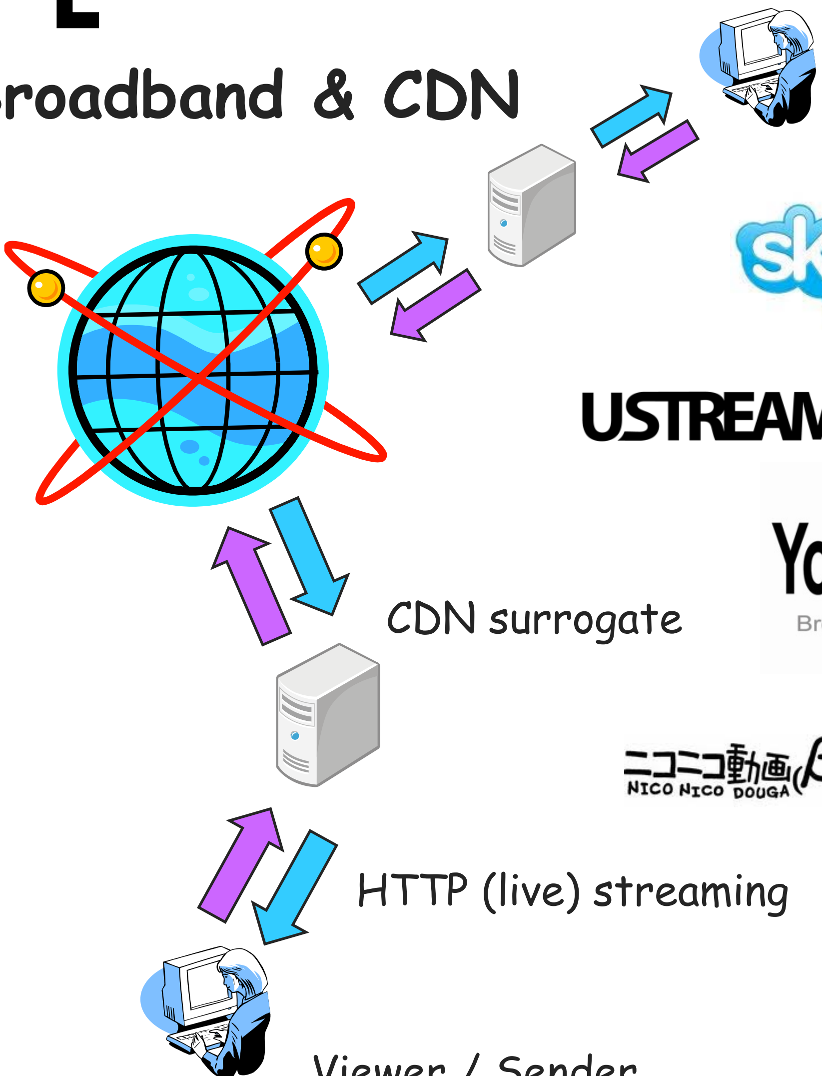
- Cat-and-mouse game



Wired Networks

VoIP, IPTV, Streaming

Broadband & CDN



RTP/UDP & RTSP & TFRC

→ <u>HTTP/TCP streaming</u>

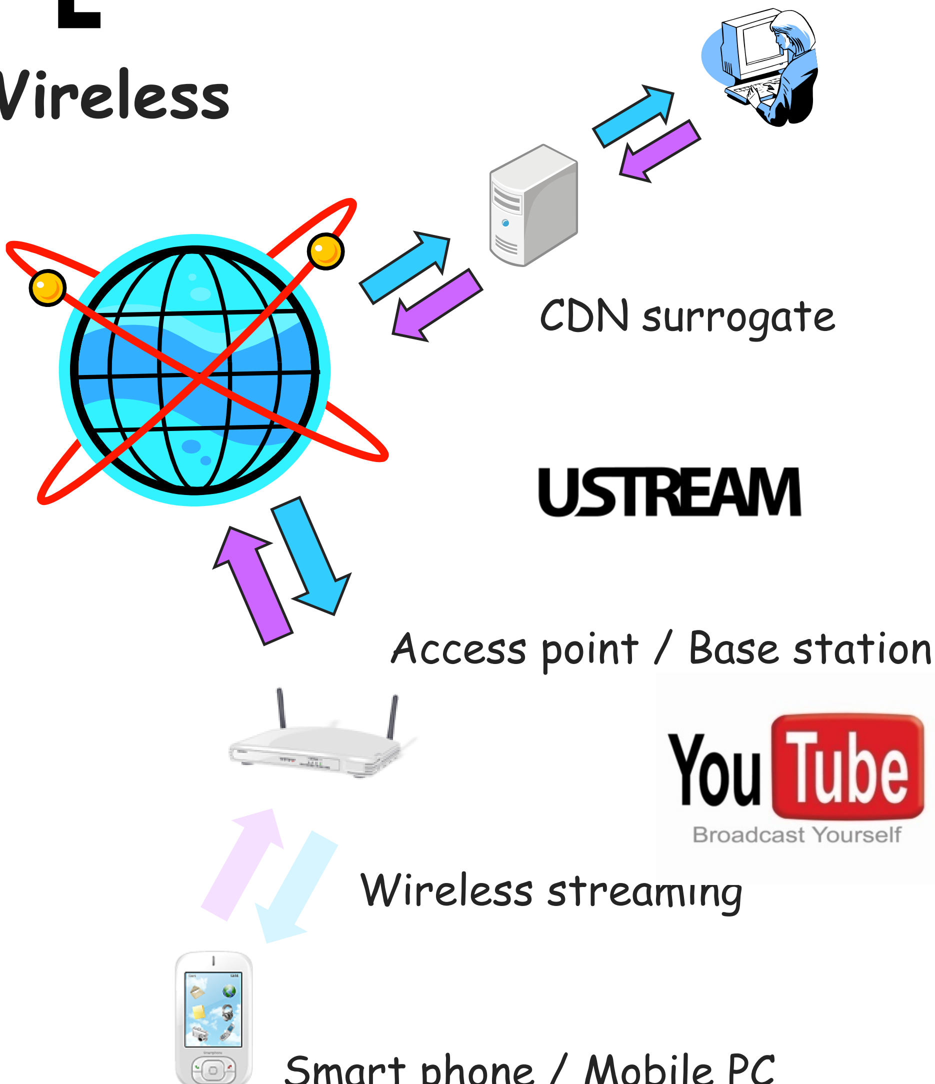
- Broadband
- CDN (Akamai, Lime Networks)
- Firewall (port 80)
- …

One-way (on-demand / live)

Bi-directional (interactive)

Wireless Networks

Wireless



Wireless specific problems

- Wireless LAN
- (Cellular)
- (WiMAX)
- (Home Networks)
- (Satellite)
- ...

- **Wireless issues**

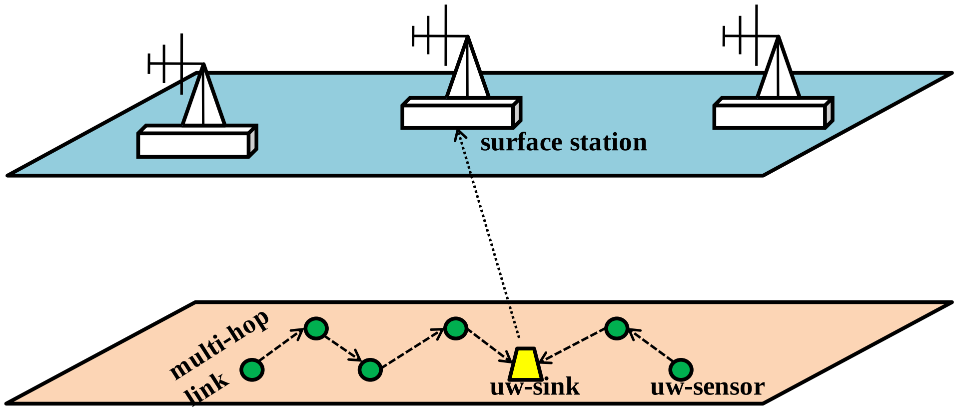
random errors, collisions, interference, delay increase

- **Multi-hop issues**

severe interference, lower throughput and higher delay

Underwater Sensor Networks

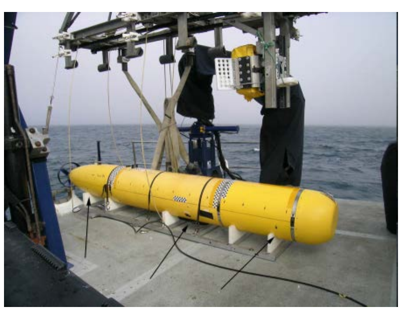
Underwater



Underwater sensor networks

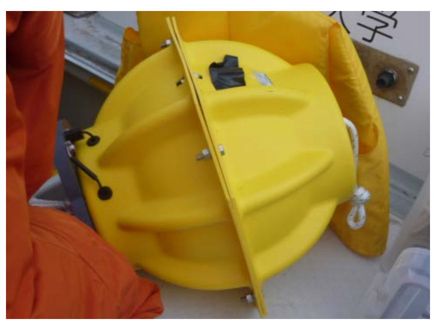
- Acoustic channels (sound speed, narrow band, huge delay)
- Temperature and depth effect
- Vertical or tilted (direction)
- …

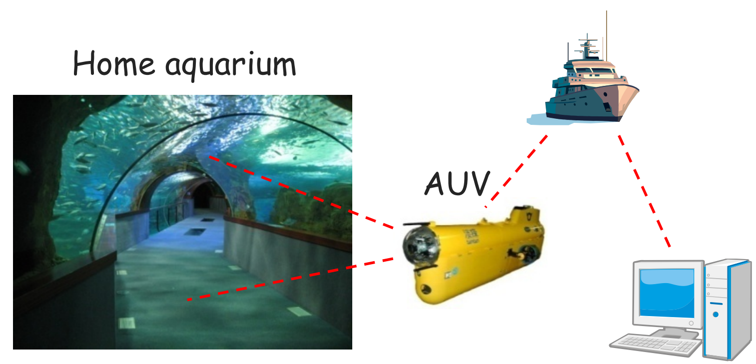
AUV (Autonomous Underwater Vehicle)



http://www.mbari.org/auv/IAUV.htm

Oversea experiments





<u>**(TCP and) MAC extensions**</u>

Multimedia Transport

- Protocol Stack

video	audio	signaling	compression, computer vision, 3D, overlay (CDN, P2P), applications, ...
adaptation		signaling	RTP/RTCP (synchronization, packet loss detection, congestion control)
transport layer			**TCP**, UDP, TFRC (end-to-end control)
network layer			IP (routing, multicast, mobility)
data link & physical layer			**wired** (fast and broadband) **wireless** (WiFi, multi-hop, underwater)
data link & physical layer			MAC (multiple access, full/half duplex), channel coding, modulation, MIMO, ...

TCP Variants

	Wired	Wireless	Satellite	Underwater
TCP	TCP-Reno/SACK High-speed TCP Scalable TCP CUBIC-TCP H-TCP TCP-Vegas FAST-TCP Compound TCP Adaptive Reno TCP-Illinois YeAH-TCP TCP-Fusion	TCP-Westwood TCP-J LDA TCP-FIT *Indirect TCP* *Snoop TCP* *Freeze TCP* Vegas-W FeW (cross layer)	TCP-Hybla TCP-STAR	–
TFRC	TFRC/DCCP RAP TEAR MULTFRC VTP Hybrid-TFRC	TFRC-Wireless → →	–	–

Wired/Wireless Classification

- MAC, hops and transmission media

	Wired	Wireless LAN	Multihop	Underwater
duplex	Full duplex	Half duplex	Half duplex	Half duplex
multiple access	Switch	CSMA	CSMA	CSMA (TDMA)
# of hops			Multiple	Multiple
signal				Acoustic

Outline

- Introduction
- Wired Networks
  - TCP Variants
  - Hybrid TCP
- Wireless and Underwater Networks
  - Extensions for WiFi, Multihop and Underwater Sensor Networks
  - DTN extension
- Conclusions

Can we achieve high-throughput ans low-delay simultaneously by TCP for multimedia streaming ?

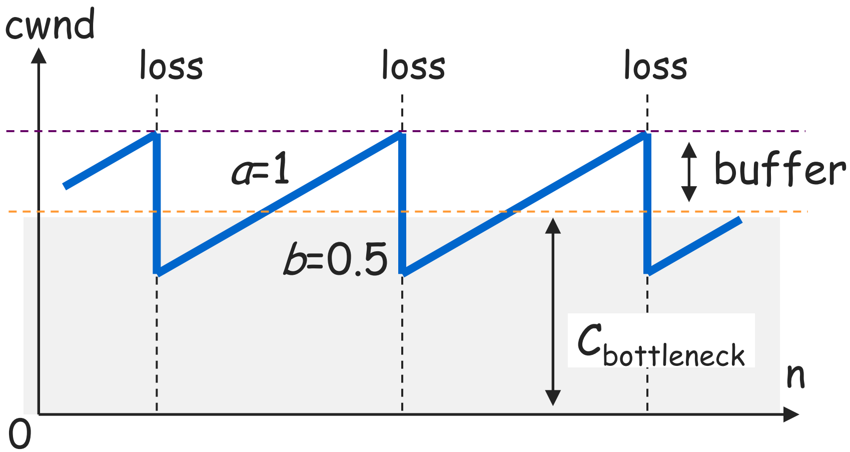
2. Wired (Fast & Broadband) Networks

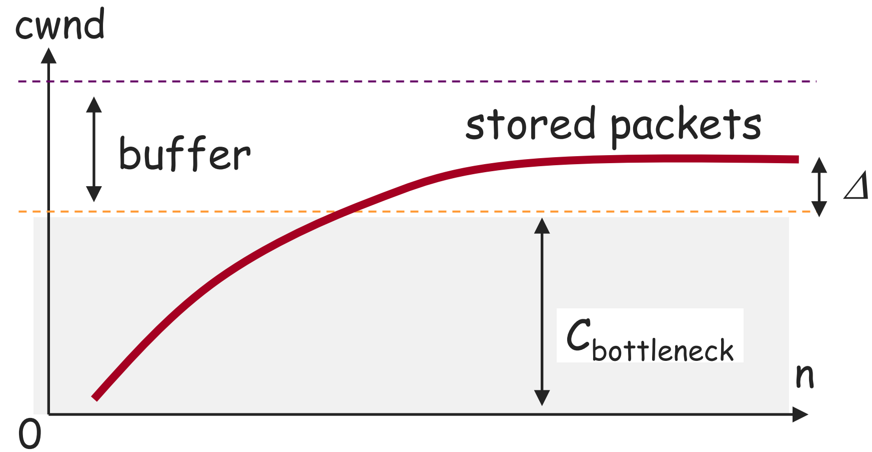
TCP Variants

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

TCP-Reno and Vegas

- TCP-Reno (loss)
- TCP-Vegas (delay)





$$cwnd = \begin{cases} cwnd + 1/cwnd & \text{(no packet loss)} \\ cwnd/2 & \text{(packet loss)} \end{cases}$$

for each ACK

$$\left(cwnd = cwnd + 1 \quad \text{for every RTT} \right)$$

$$cwnd = \begin{cases} cwnd + 1 & (\Delta \le \alpha) \\ cwnd & (\alpha < \Delta < \beta) \\ cwnd - 1 & (\Delta \ge \beta) \end{cases}$$

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

for every RTT

TCP-Vegas is more efficient but is expelled by TCP-Reno.
These are too slow for fast and broadband networks.

TCP Variants

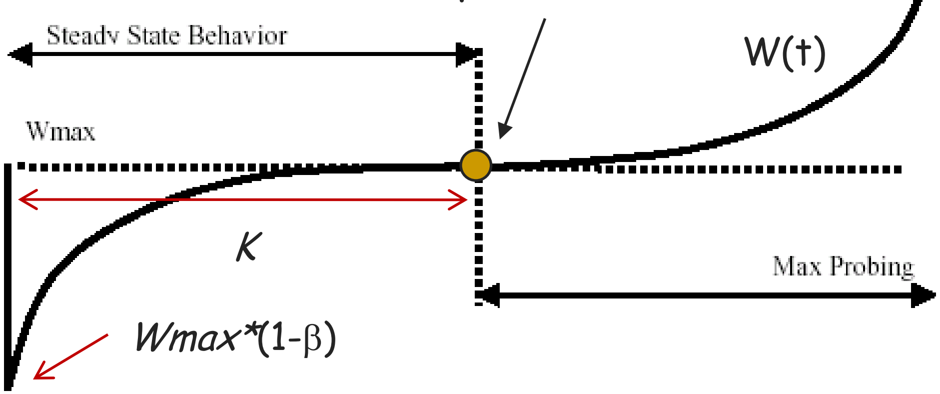
- **loss based (AIMD: additive increase & multiplicative decrease upon packet losses)**
  - TCP-Reno / NewReno / SACK
  - High-Speed TCP (IETF RFC 3649, Dec 2003)
  - Scalable TCP (PFLDnet 2003)
  - BIC / CUBIC-TCP (IEEE INFOCOM 2004, PFLDnet 2005) ... Linux
  - H-TCP (PFLDnet 2004)
  - TCP-Westwood (ACM MOBICOM 2001)
- **delay based (RTT observation)**
  - TCP-Vegas (IEEE JSAC, Oct 1995)
  - FAST-TCP (INFOCOM 2004)
- **hybrid (adaptive selection of loss and delay modes)**
  - Gentle High-Speed TCP (PfHSN 2003)
  - TCP-Africa (IEEE INFOCOM 2005)
  - Compound TCP (PFLDnet 2006) ... Windows
  - Adaptive Reno (PFLDnet 2006)
  - YeAH-TCP (PFLDnet 2007)
  - TCP-Fusion (PFLDnet 2007) ... Our contribution

CUBIC-TCP (1)

- Fast Window Increase

fast increase at first,
gradual increase around the target

"cubic" approximation of window control of BIC-TCP



$$W(t) = C*(t-K)^3 + W_{max}$$

$$K = \sqrt[3]{\frac{W_{max}\beta}{C}}$$

equivalent to Reno

$$W_{tcp}(t) = W_{max}(1-\beta) + 3\frac{\beta}{2-\beta}\frac{t}{RTT}$$

```
/* cubic function */
Winc = W(t+RTT) – cwnd;

cwnd = cwnd + Winc / cwnd;

/* TCP mode */
if ( Wtcp > cwnd )
   cwnd = Wtcp;
```

β: window decrease rate (e.g. 0.2)

C: constant (e.g. 0.4)

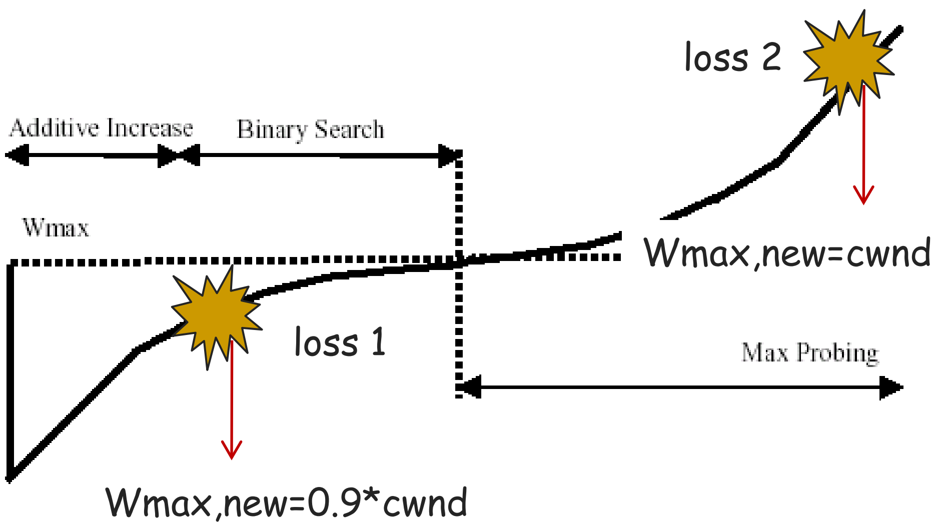
S.Ha et al: "CUBIC: A New TCP Friendly HighSpeed TCP Variant", ACM SIGOPS Review, 2008.

CUBIC-TCP (2)

- Small Window Decrease

update of convergence target Wmax and cwnd

β: window decrease rate (e.g. 0.2)




```
if (cwnd < Wmax )
    Wmax,new = cwnd * (2-β) / 2;
else
    Wmax,new = cwnd;

cwnd = cwnd * (1- β);
```

$1-\beta = 0.8$

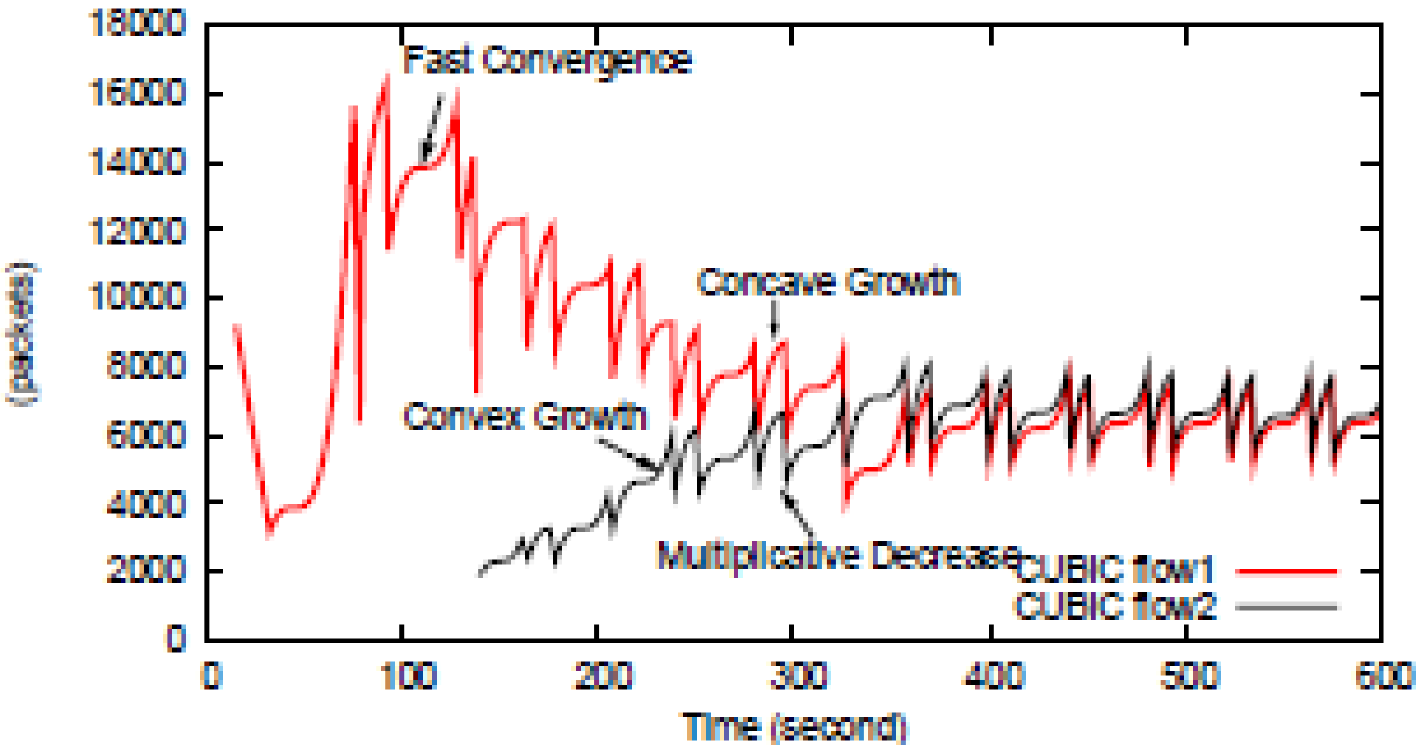
small decrease upon packet losses (less than 1/2)

L.Xu et al: "Binary Increase Congestion Control (BIC) for Fast Long-Distance Networks," IEEE INFOCOM 2004.

CUBIC-TCP (3)

- CUBIC's Cyclic Behavior

repetition of convex & concave shapes



S.Ha et al: "CUBIC: A New TCP Friendly HighSpeed TCP Variant", ACM SIGOPS Review, 2008.

CUBIC-TCP (4)

- Advantages of CUBIC
  - Stability ... packets are always buffered
  - Efficiency ... fast increase, small decrease
  - Friendliness ... by TCP Reno mode
  - Intra-protocol fairness ... gives a chance of bandwidth sharing to newly incoming flows

- Disadvantages of CUBIC
  - Too stable due to heavy packet buffering
    $\Rightarrow$ Delay increase
  - Inter-protocol unfairness ... expels all the other TCP flows, e.g."Linux beats Windows !" (vs. Compound TCP)

K.Munir et al: "Linux beats Windows! or the Worrying Evolution of TCP...", PFLDNet 2007.

TCP Westwood (1)

<u>FSE: Fair Share Estimates</u>

- Duplicate ACKs

$$ssthresh = FSE * RTT_{\min}$$

$$if\ (cwnd > ssthresh)\ cwnd = ssthresh$$

just clear the bottleneck buffer (instead of 1/2)

- Timeout

$$ssthresh = FSE * RTT_{\min}$$

$$cwnd = 1$$

TCP-Reno's case:

$$ssthresh = cwnd / 2$$

- Multiple versions according to FSE estimation methods
  - BSE, RE, ABSE, …

C.Casetti et al: "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", ACM MOBICOM 2001.

TCP Westwood (2)

Fair Share Rate Estimation (TCPW-RE)

similar to TCP-Vegas:

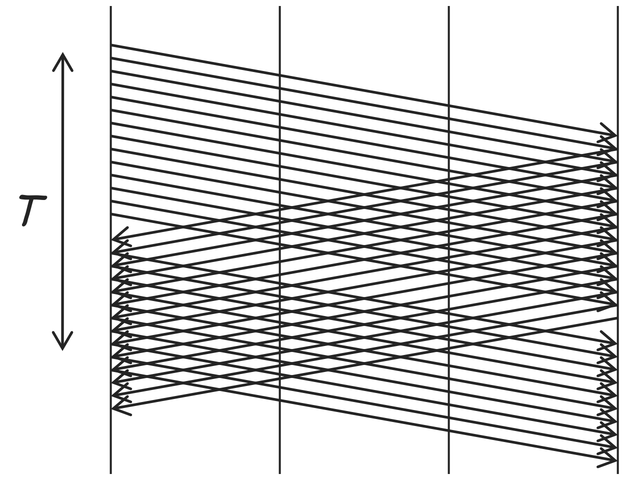
≈ link capacity

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

≈ buffered packets

expect rate

actual rate



TCPW-RE:

$$RE_k = \frac{\sum_{t_j > t_k - T} d_j}{T}$$

cwnd ⇒ byte counts: Σd_k

RTT ⇒ observation time: T= $\Sigma \Delta t_k$

$$\hat{RE}_k \approx \frac{\sum d_k}{\sum \Delta t_k}$$

moving average: $\hat{RE}_k \rightarrow FSE$

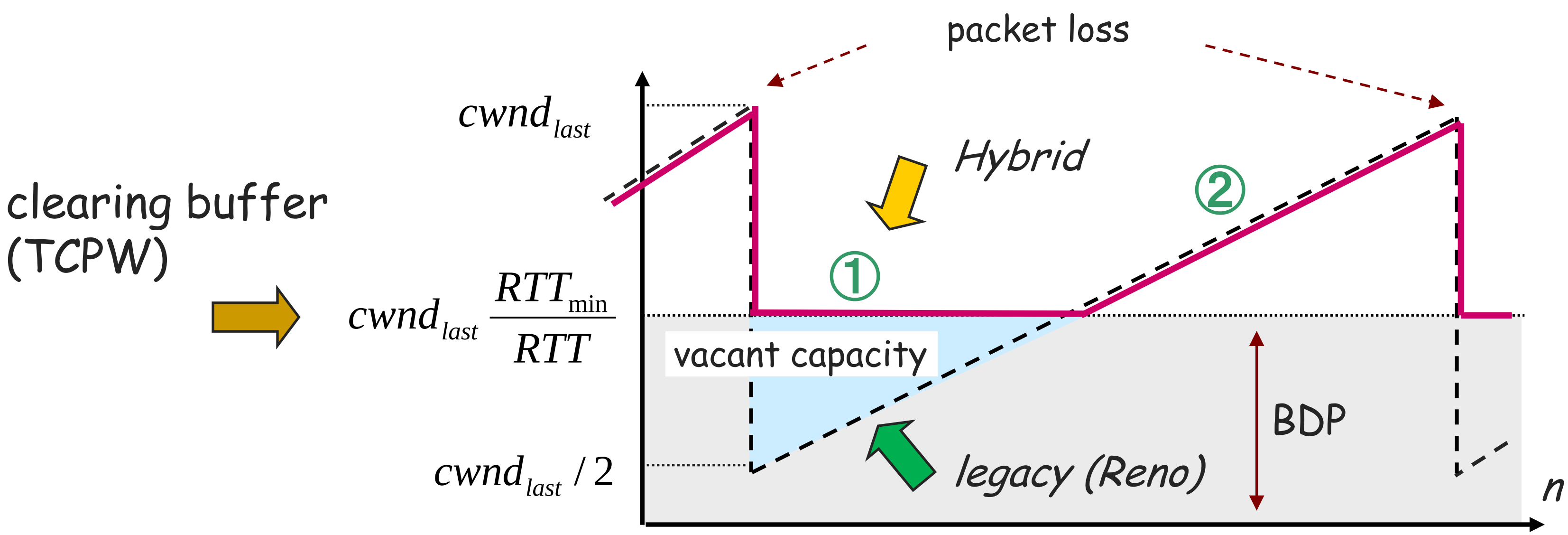
$T = n \cdot RTT$ (e.g. n=4)

M.Gerla et al: "TCP westwood with adaptive bandwidth estimation to improve …", Comp. & Comm., 2004.

Hybrid TCP (1)

Compound-TCP and TCP-Fusion

- Single flow case



Adaptive mode selection between loss & delay modes:

① constant rate until RTT increases (delay mode) : **efficiency**

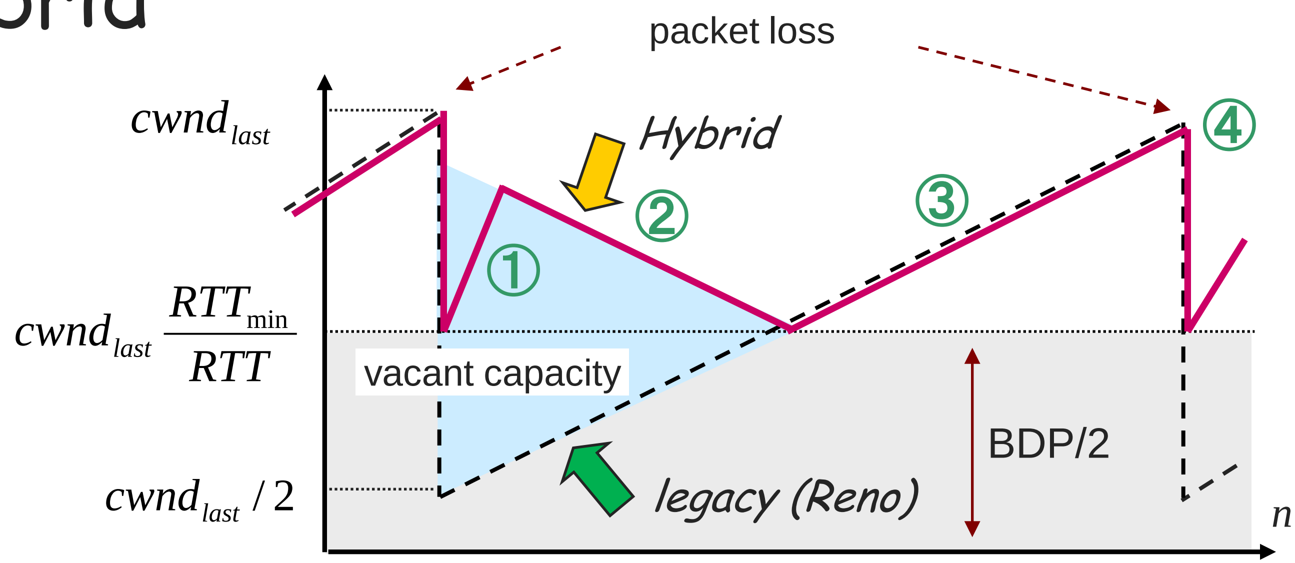
② TCP-Reno when RTT increases (loss mode) : **friendliness**

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

Hybrid TCP (2)

- Competing case between Reno and Hybrid

clearing buffer (TCPW)



Adaptive mode selection between loss & delay modes:

① fast increase of cwnd (delay mode … **efficiency**)

② slow decrease of cwnd (delay mode … **small buffering**)

③ TCP-Reno when RTT increases (loss mode … **friendliness**)

J.Katto et al: "Simple model analysis and performance tuning …", IEEE Globecom 2008.

Hybrid TCP (3)

- Classification

Hybrids	Window increase ①	Window decrease ④
CTCP	$0.125*cwnd^{0.75}$	1/2
ARENO	B/10Mbps	1/2～1
YeAH-TCP	STCP(1.01)	1/2, RTT_{min}/RTT, 7/8
TCP-Fusion	$B*D_{min}/(N*PS)$	RTT_{min}/RTT

D_{min}: timer resolution, N: # of flows

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

Hybrid TCP (4)

- Advantages of Hybrid TCP
  - Efficiency ... fast increase, small decrease (not causing vacant capacity)
  - Friendliness ... loss mode
  - Low delay ... thanks to small buffering when no loss based flows compete

- Disadvantages of Hybrid TCP
  - CUBIC friendliness (CUBIC mode ?)
  - No transition from loss mode to delay mode happens when buffer size > BDP

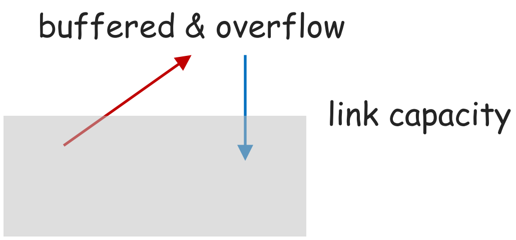
J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

Performance analysis of Hybrid TCP

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

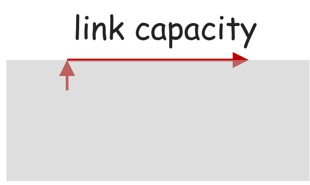
TCP Abstraction (1)

- Definition of abstraction models
  - loss based (TCP-Reno) :
    - cwnd += 1 (per RTT)
    - cwnd *= 1/2 (upon packet losses)



  - delay based :
    - just fills a pipe without RTT increase

(immediately fills the pipe without buffering)



  - hybrid :
    - operates in delay mode when RTT doesn't increase
    - operates in loss mode when RTT increases
    - mode selection: cwnd = max($cwnd_{loss}$, $cwnd_{delay}$)

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

TCP Abstraction(2)

- Parameter definition
  - w : cwnd when packet losses happen
  - W : cwnd which just fill a pipe (i.e. corresponding to BDP)
  - p : packet loss rate (PLR)

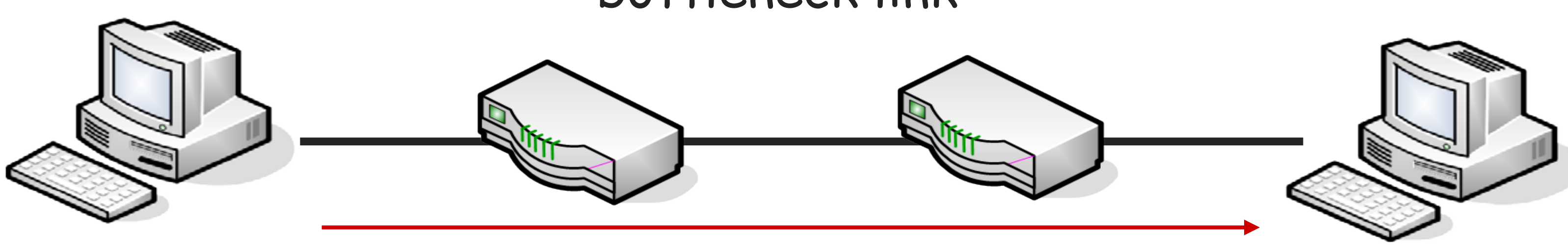
- Assumption
  - Packet losses by buffer overflow is equivalent to those by random errors

$$p = \frac{8}{3w^2}$$ (in case of TCP-Reno)

J.Katto et al: “Simple model analysis and performance tuning …”, IEEE Globecom 2008.

Performance Analysis (1)

- Single flow case



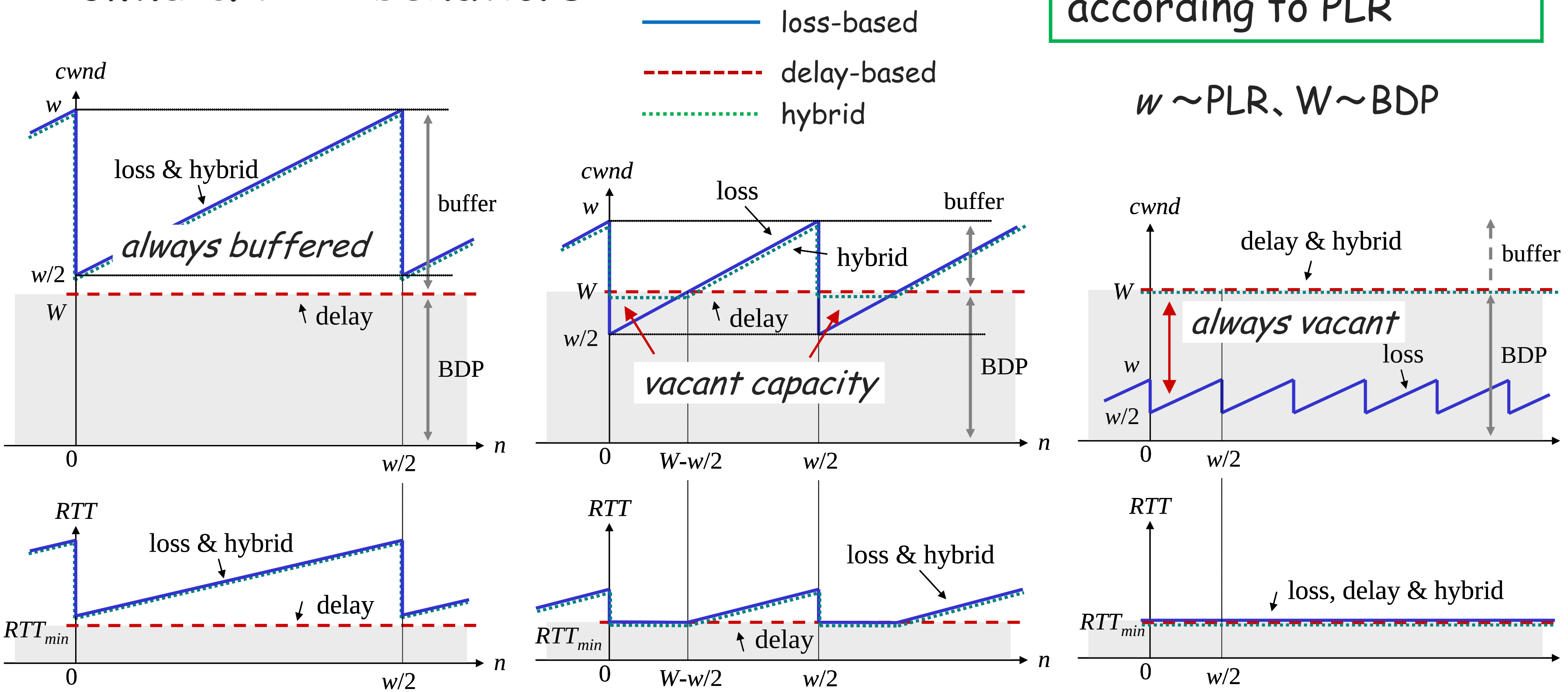
J.Katto et al: “Simple model analysis and performance tuning …”, IEEE Globecom 2008.

Performane Analysis (2)

- **cwnd & RTT behaviors**

Behavior classification according to PLR

w ~PLR、W~BDP



(i) $W < w/2$

(ii) $w/2 < W < w$

(iii) $w < W$

large buffer, small PLR (always loss-mode)

small buffer, medium PLR (delay & loss adaptive)

large PLR, always vacant (always delay-mode)

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

Performance Analysis (3)

- Single flow formulation

TCP	CA round	(i) $W < w/2$	(ii) $w/2 \le W < w$	(iii) $w \le W$
Loss	transmitted packets	$\frac{3}{8}w^2$	$\frac{3}{8}w^2$	$\frac{3}{8}w^2$
	elapsed time	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{8}(3w^2 - 4wW) \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{2}(w-W)^2 \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min}$
Delay	transmitted packets	$\frac{1}{2}w \cdot W$	$\frac{1}{2}w \cdot W$	$\frac{1}{2}w \cdot W$
	elapsed time	$\frac{1}{2}w \cdot RTT_{\min}$	$\frac{1}{2}w \cdot RTT_{\min}$	$\frac{1}{2}w \cdot RTT_{\min}$
Hybrid	transmitted packets	$\frac{3}{8}w^2$	$\frac{1}{2}w \cdot W + \frac{1}{2}(w-W)^2$	$\frac{1}{2}w \cdot W$
	elapsed time	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{8}(3w^2 - 4wW) \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{2}(w-W)^2 \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min}$

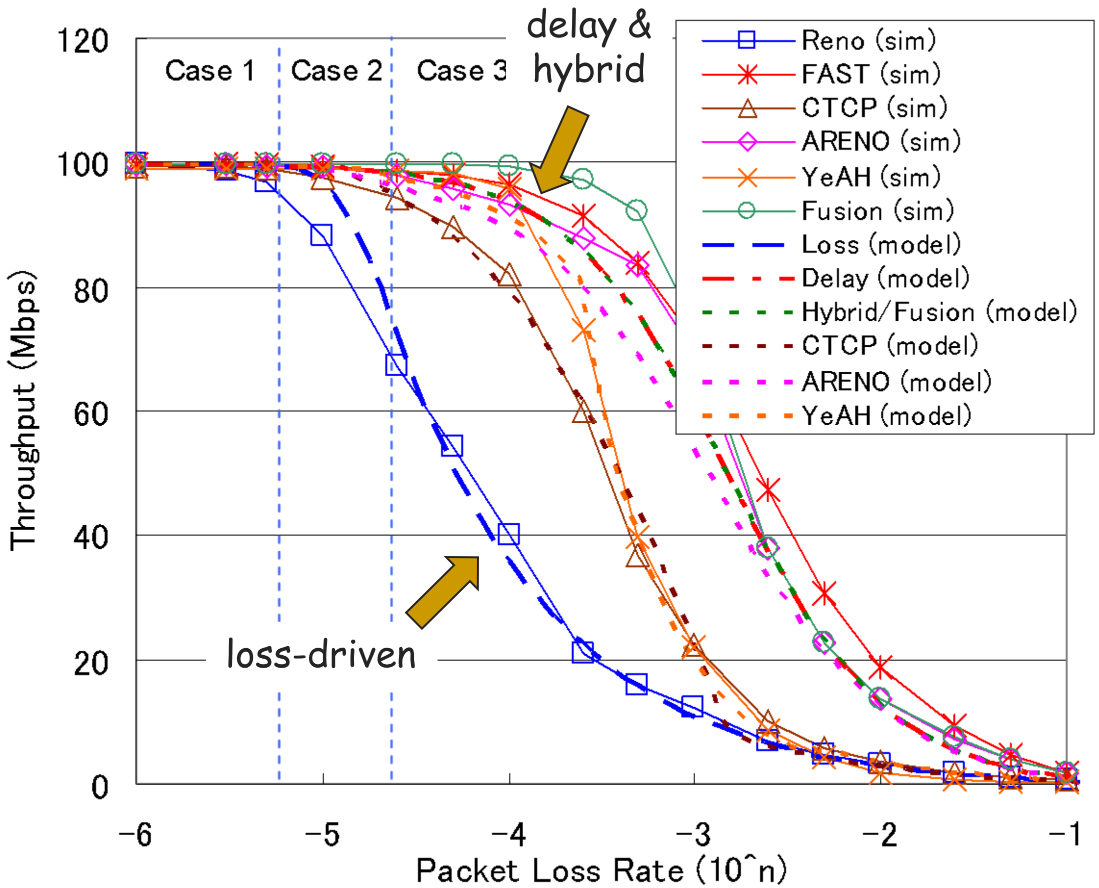
PS: Packet size, B: Link bandwidth

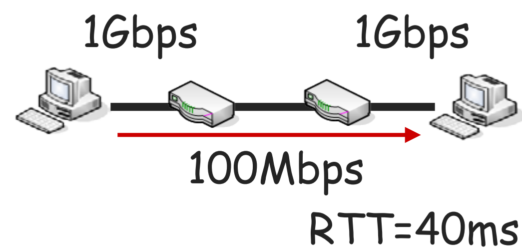
J.Katto et al: "Simple model analysis and performance tuning …", IEEE Globecom 2008.

Performance Analysis (4)

- Analysis & simulations

PER ~ Throughput
(single flow case)





buffer size = BDP (constant)

Packet loss rate : variable

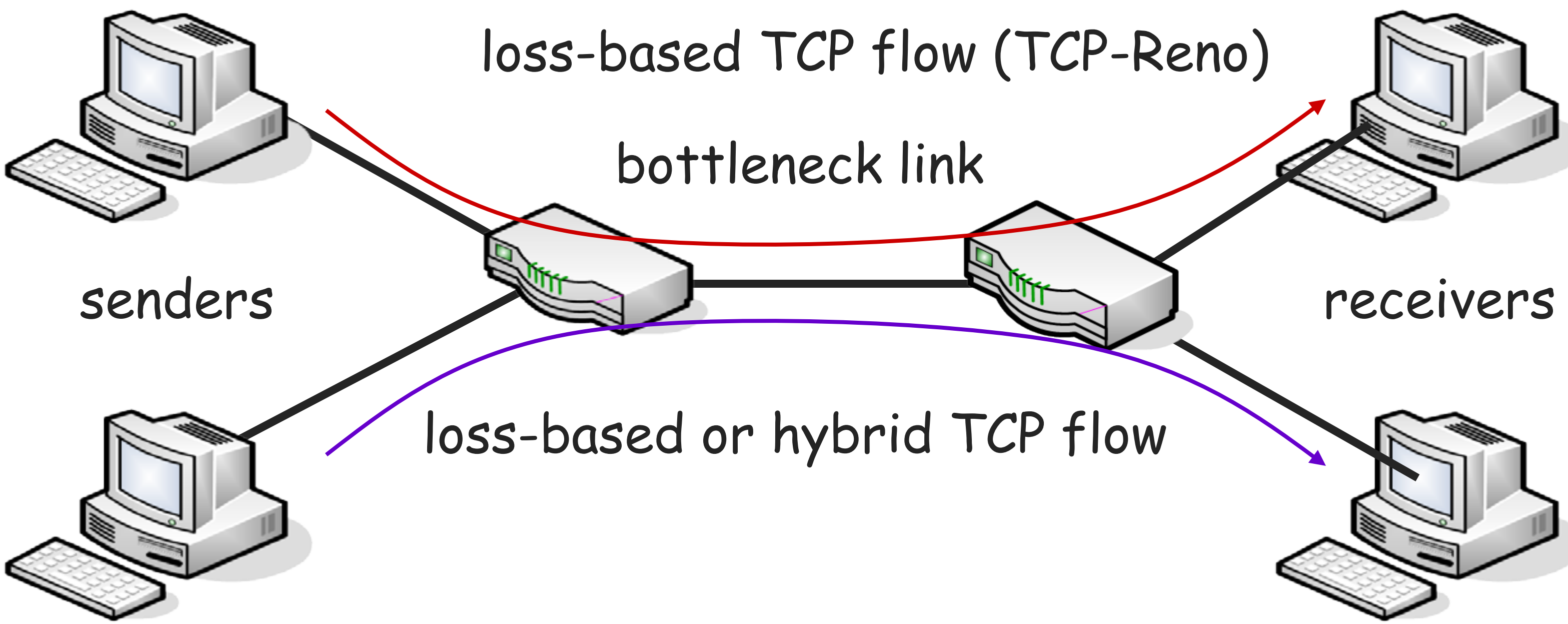
For large PLR (w/2<W), delay & hybrid flows achieve much more throughputs than loss-based one (<u>efficiency</u>)

Compound & YeAH TCPs degrade due to large window decrease rate

J.Katto et al: "Simple model analysis and performance tuning …", IEEE Globecom 2008.

Performance Analysis (5)

- Two competing flows case



J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

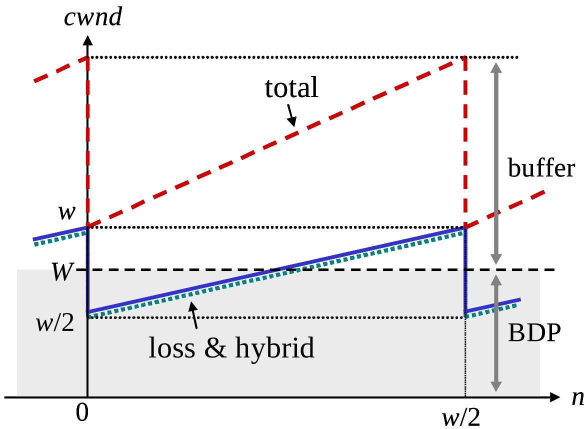
Performance Analysis (6)

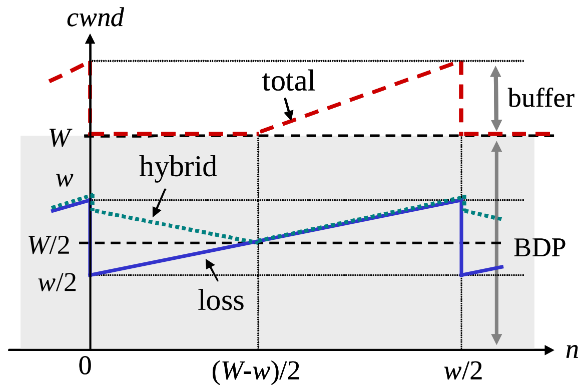
- **cwnd behaviors**

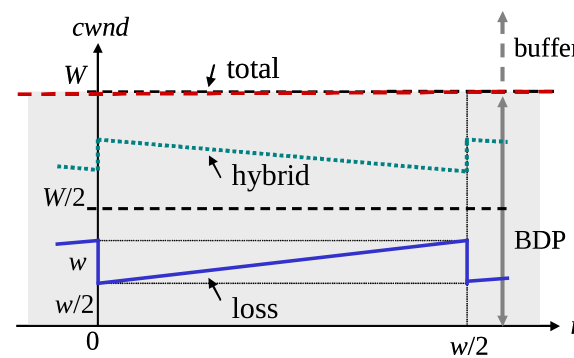
loss-driven
hybrid
total (loss + hybrid)

cwnd behavior classification according to PLR

w ~PLR、W~BDP







(i) $W < w$ (low PLR)	(ii) $w < W < 2^*w$ (medium PLR)	(iii) $2^*w < W$ (high PLR)
always buffered (loss mode)	vacant → buffered (delay → loss)	always vacant (delay mode)
large buffer, small PLR	small buffer, medium PLR	large PLR, always vacant

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

Performance Analysis (7)

- Two flow formulation

TCP	CA round	(i) $W < w$	(ii) $w \le W < 2w$	(iii) $2w \le W$
Loss	transmitted packets	$\frac{3}{8}w^2$	$\frac{3}{8}w^2$	$\frac{3}{8}w^2$
Hybrid	transmitted packets	$\frac{3}{8}w^2$	$\frac{3}{8}w^2 + \frac{1}{4}(W-w)^2$	$\frac{1}{2}w \cdot W - \frac{3}{8}w^2$
(common)	elapsed time	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{4}w(3w-2W) \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{4}(2w-W)^2 \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min}$

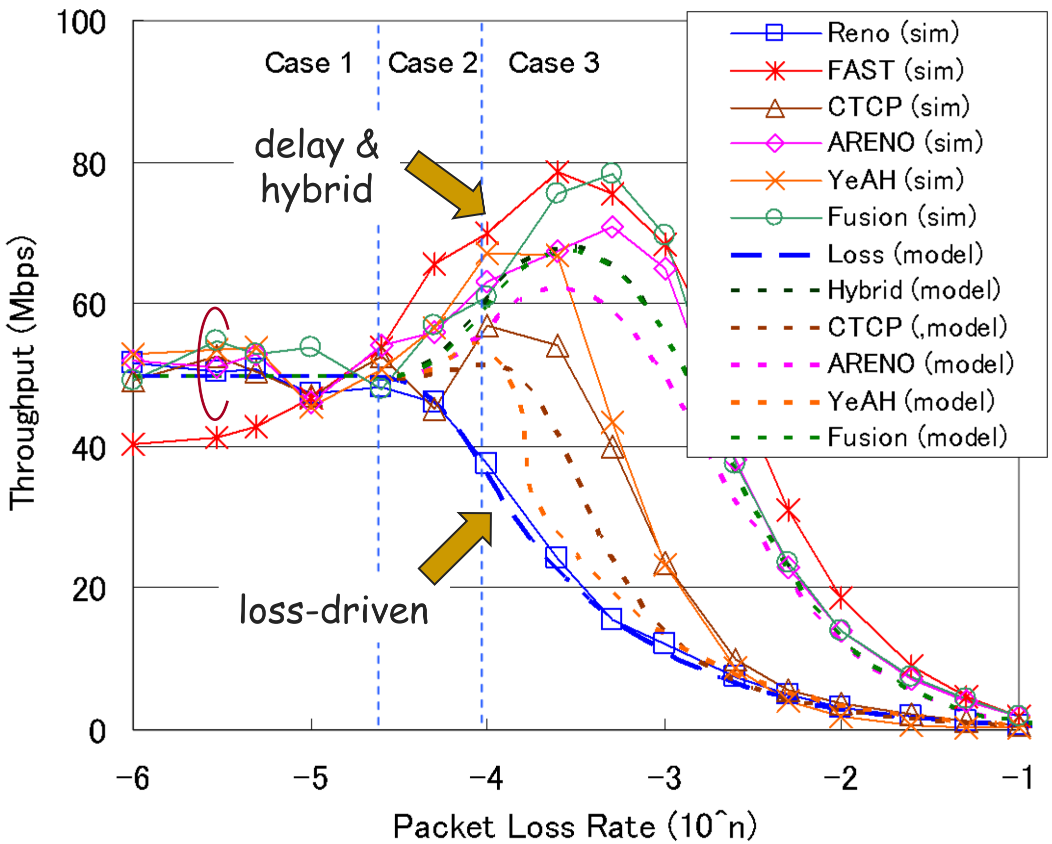
PS: Packet size, B: Link bandwidth

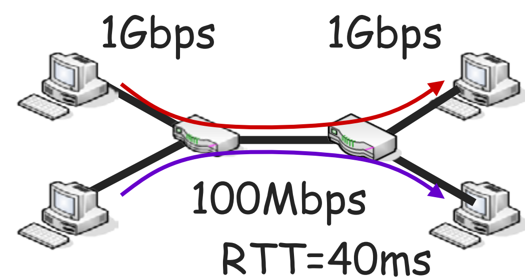
J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

Performance Analysis (8)

- Analysis & simulations

PER ~ Throughput
(two flow case)





buffer size = BDP (constant)

Packet loss rate : variable

For large PLR (w<W), delay & hybrid flows achieve more throughputs than loss-based one (efficiency)

For small PLR (w>W), hybrid behaves as loss-based (friendliness)

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

Performance Analysis (9)

- Hybrid TCP can achieve
  - throughput efficiency and low delay as delay-mode when vacant capacity exists on a link, and
  - TCP-Reno friendliness as loss-mode when packets are buffered at a router

- For wired networks,
  - models, simulations and implementations (though omitted here) perform almost as expected

J.Katto et al: "Simple model analysis and performance tuning ...", IEEE Globecom 2008.

URL

- http://www.katto.comm.waseda.ac.jp/TCP-Fusion
    - MATLAB code for performance analysis
    - ns-2 simulation code
    - Linux implementation code

    - You can enjoy if you have an interest

3. Wireless Networks

Wireless LAN

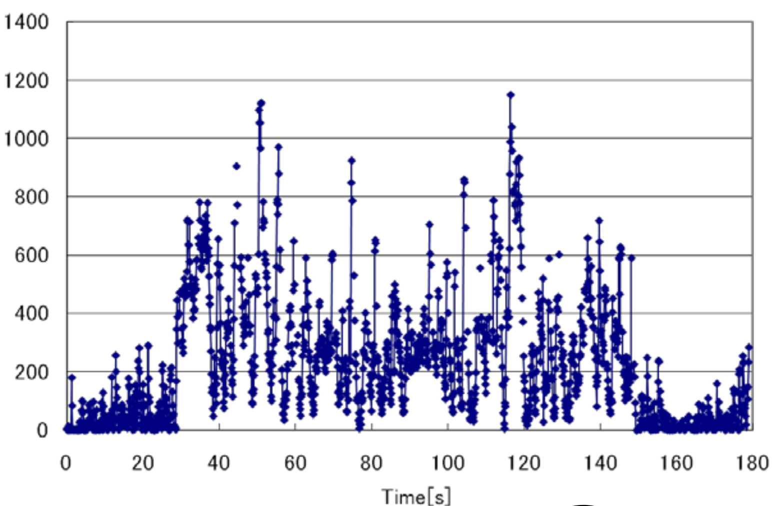
Discussion

- 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

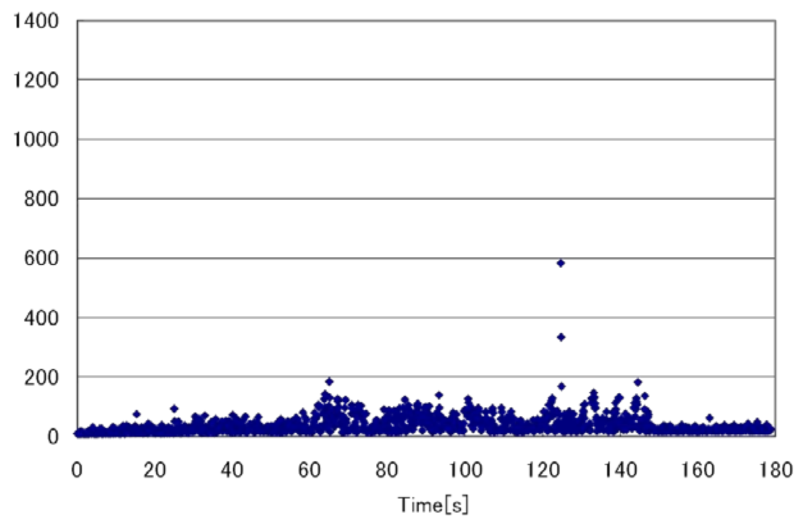
WiFi Example

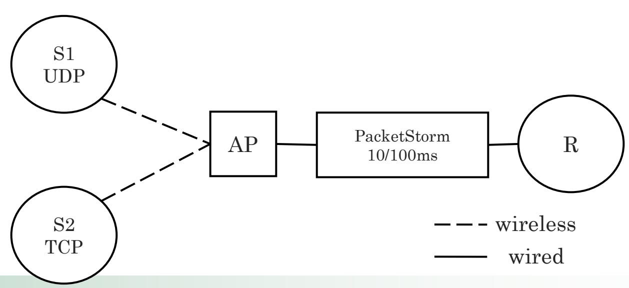
- RTT instability and unfairness between upload and download

RTT upload, wireless to wired



RTT download, wired to wireless



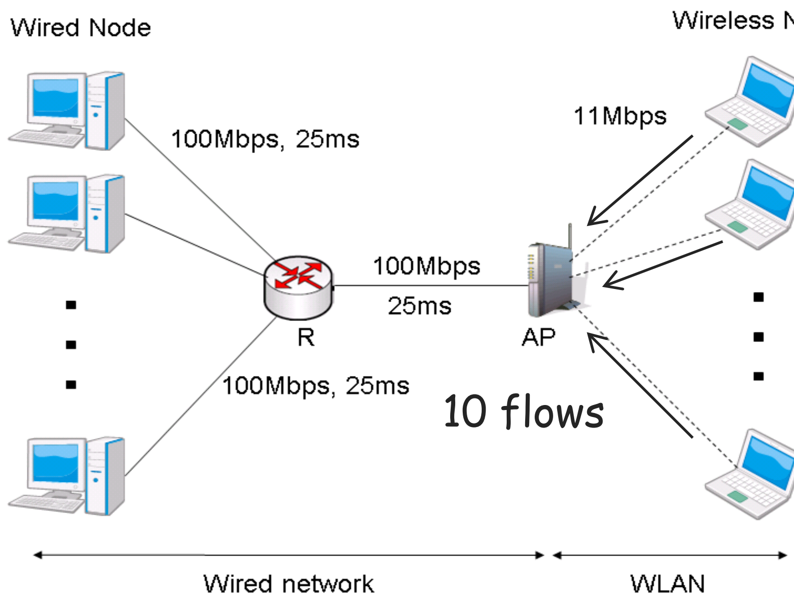


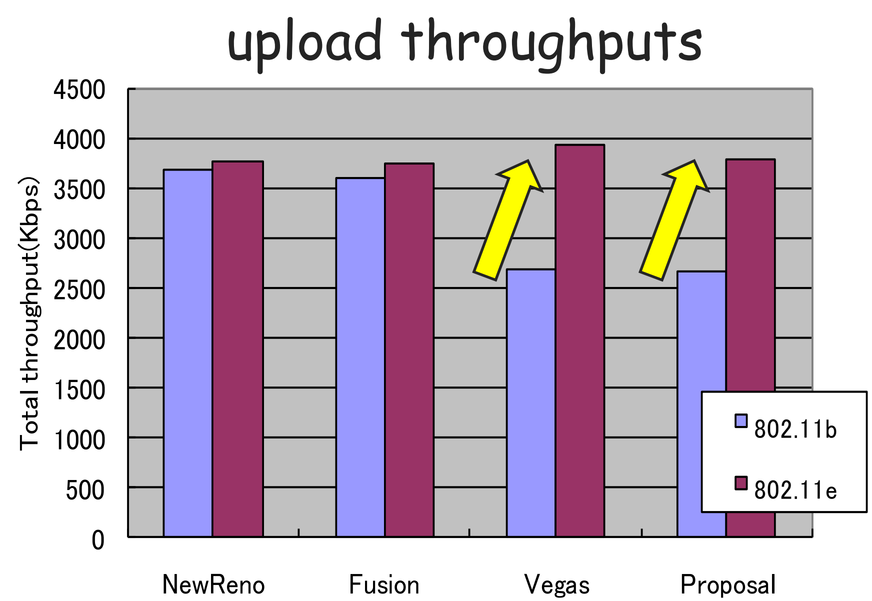
K.Kanai et al: "Performance evaluations of adaptive rate control mechanisms ...", IEICE Tech. Report, 2011.

Wireless LAN (1)

※ ns-2 simulation

- TCPs and throughputs





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

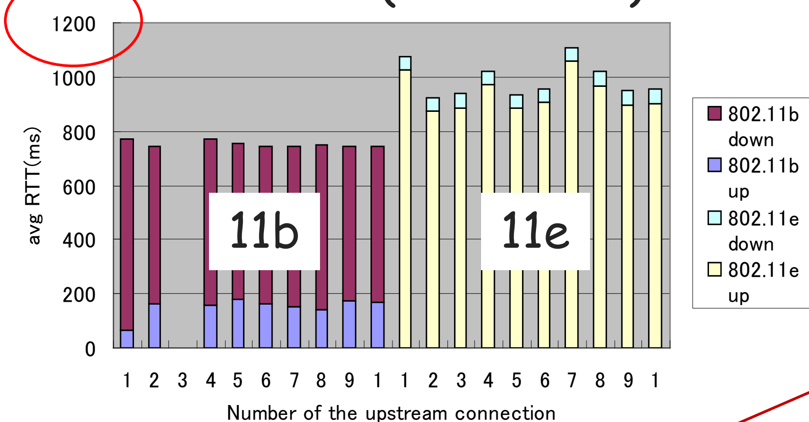
TCP-Reno: loss based
TCP-Fusion: hybrid
TCP-Vegas: delay based
Proposal: Vegas extension

K.Sonoda et al: "Performance Improvement of TCP-Vegas ...", IEICE Tech. Report, 2010.

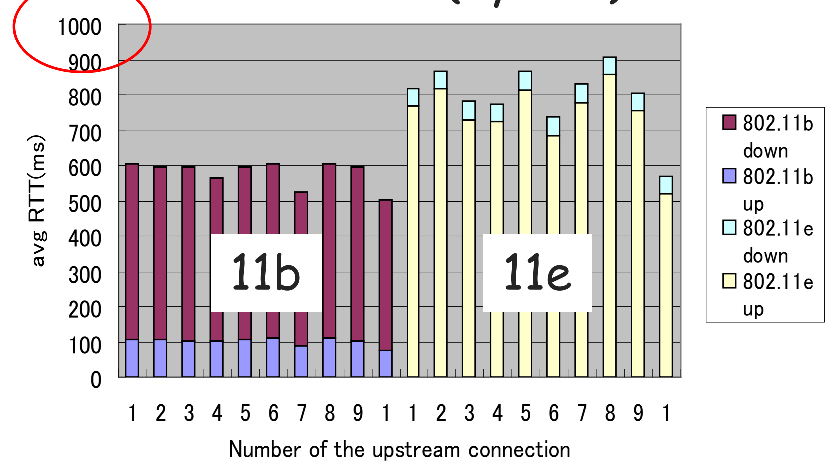
Wireless LAN (2)

- TCPs and delays

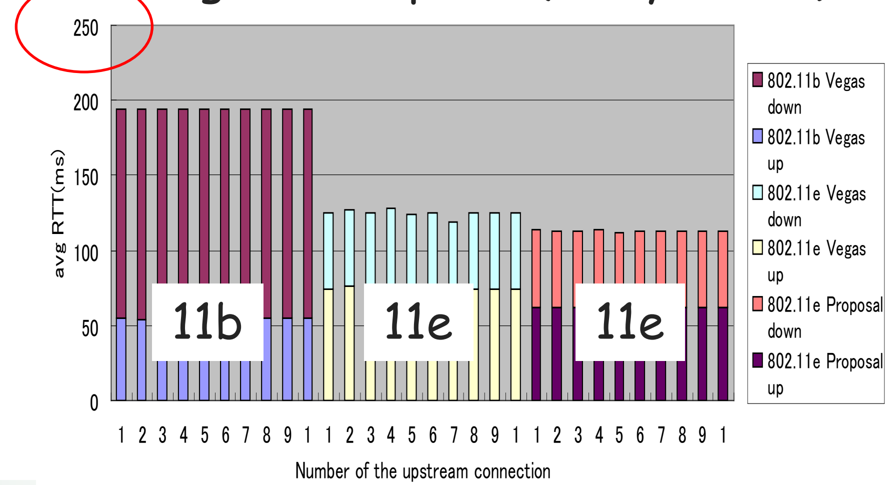
TCP-Reno (loss based)



TCP-Fusion (hybrid)



TCP-Vegas & Proposal (delay based)



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

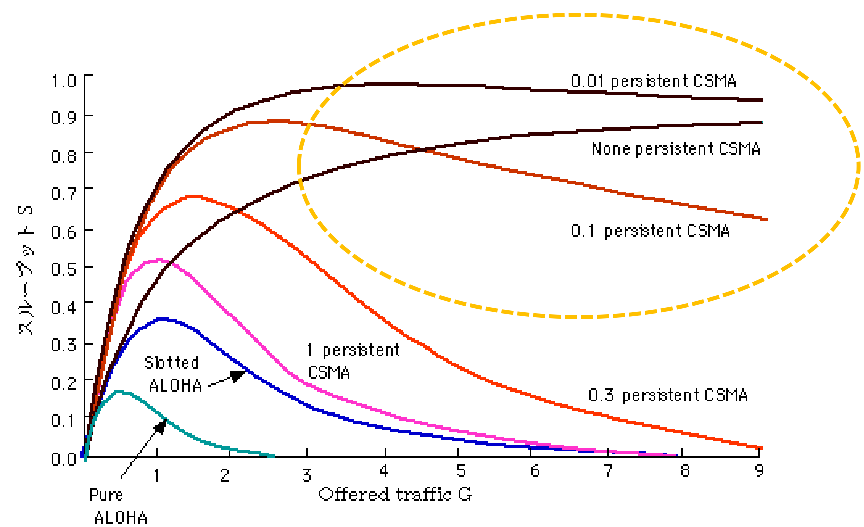
→ Hybrid TCP works in loss mode only

K.Sonoda et al: “Performance Improvement of TCP-Vegas …”, IEICE Tech. Report, 2010.

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



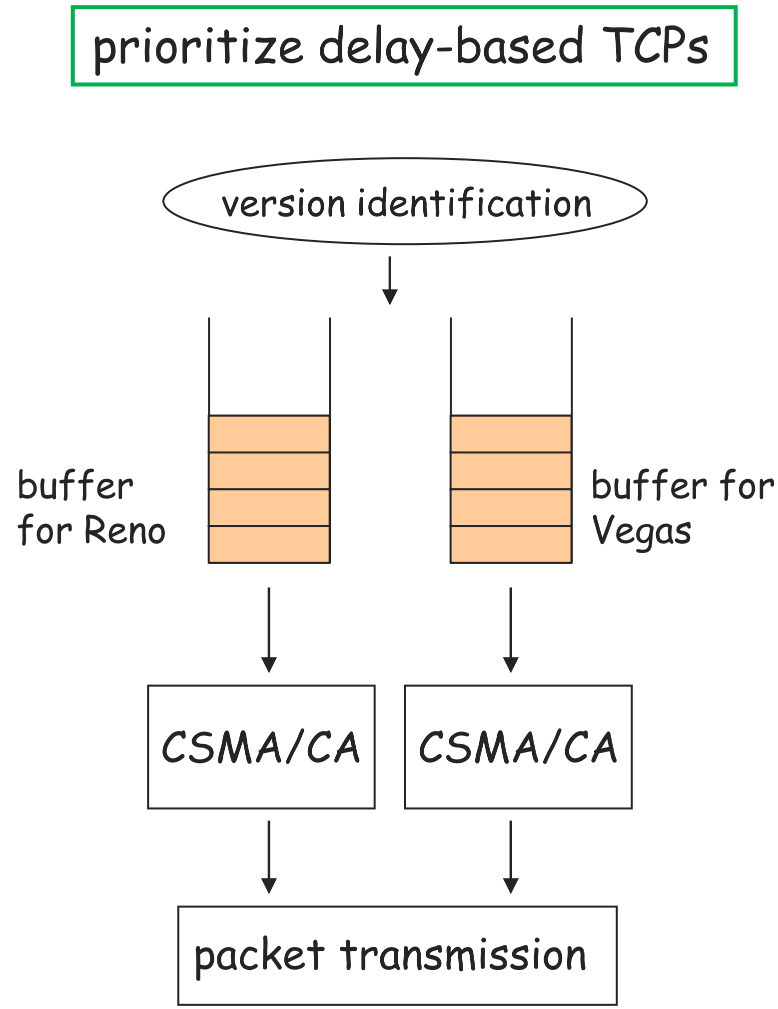
TCP Version Differentiation (1)

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

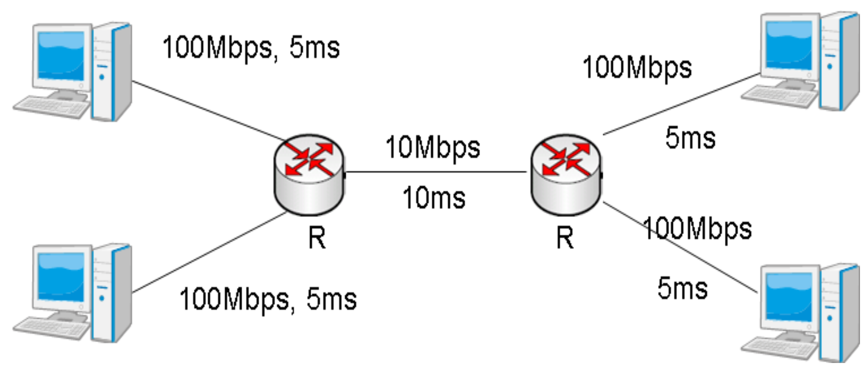


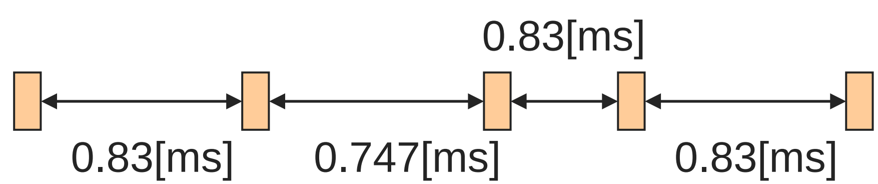
K.Sonoda et al: “A Method of TCP Version Identification ...”, IEICE Tech. Report, 2011.

TCP Version Differentiation (2)

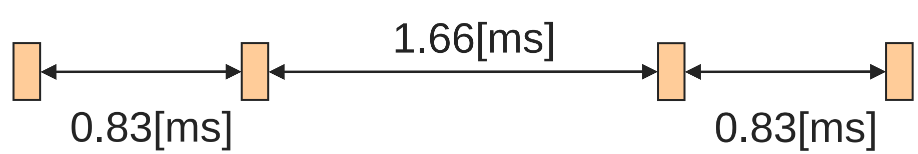
- RTT estimation for delay based flow
  - When cwnd increases by one, two consecutive packets are transmitted
  - When cwnd decreases by one, no packets are transmitted for the last ACK
- cwnd estimation
  - Access points let the number of arrived packets per RTT be "cwnd"

TCP behavior estimation at AP





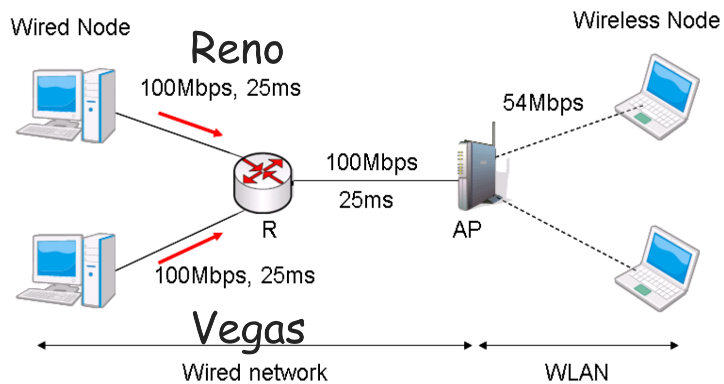
when cwnd increases by 1



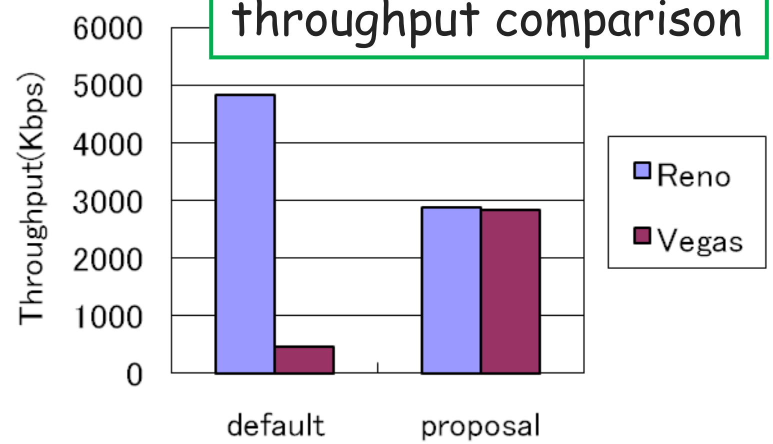
when cwnd decreases by 1

K.Sonoda et al: "A Method of TCP Version Identification ...", IEICE Tech. Report, 2011.

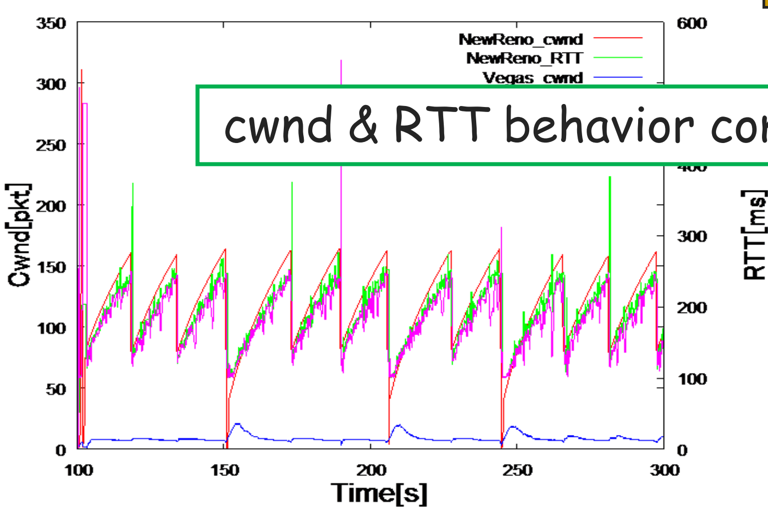
TCP Version Differentiation (3)



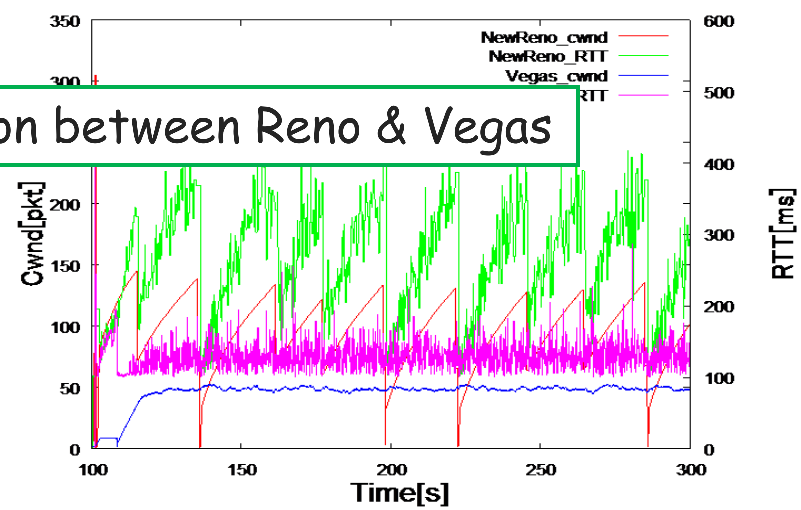
throughput comparison



cwnd & RTT behavior comparison between Reno & Vegas



without differentiation



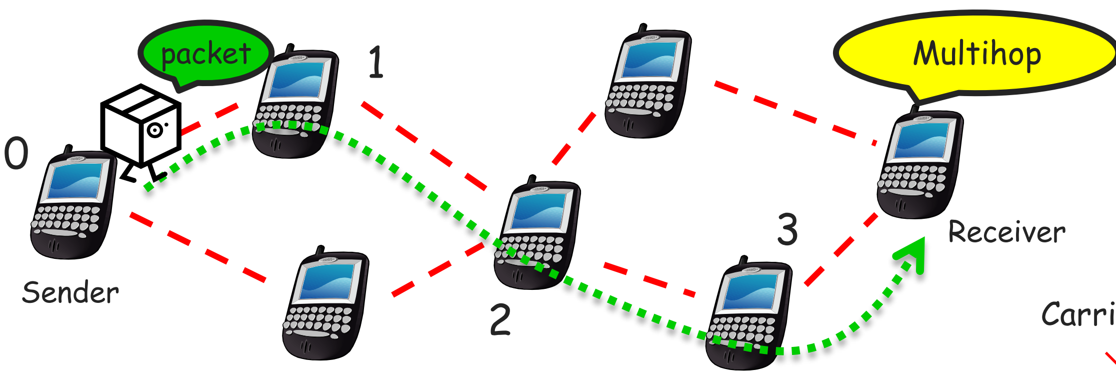
with differentiation

K.Sonoda et al: "A Method of TCP Version Identification ...", IEICE Tech. Report, 2011.

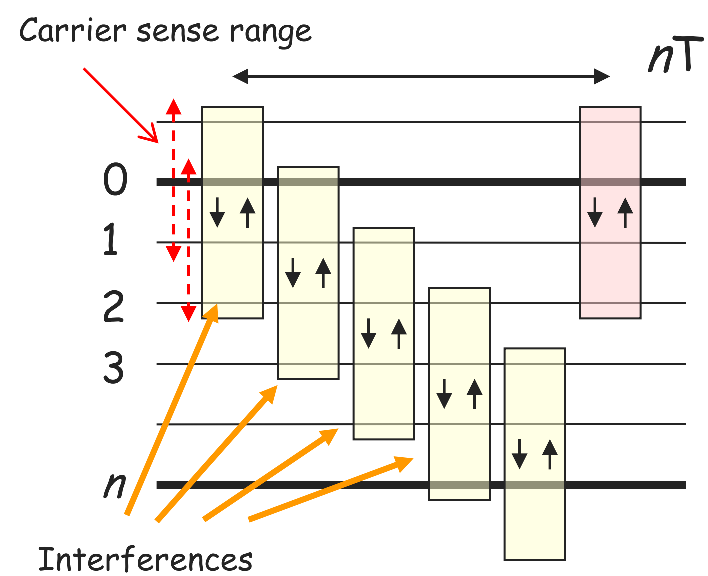
Wireless Multihop Networks

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)

Wireless Multihop Networks (2)

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.

Wireless Multihop Networks (3)

for multihop & USN

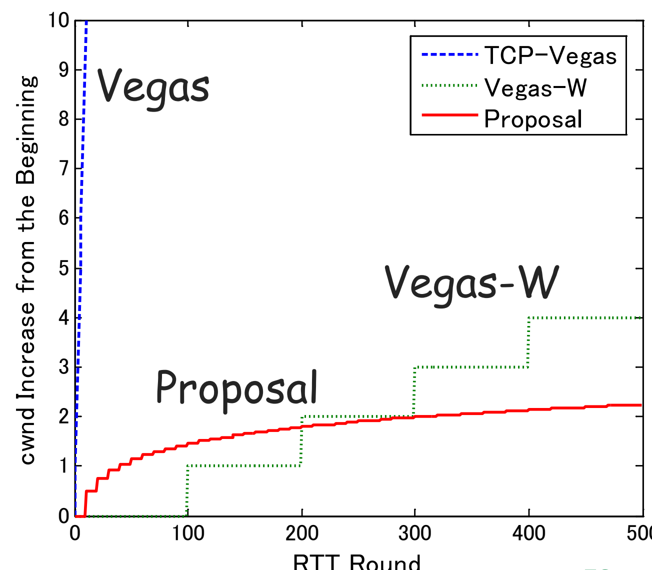
- Our proposal [IEICE, 2009]
  - **Exponential decrease of window increase**

$$cwnd = \begin{cases} cwnd + 1/(cwnd \times 2 \times count) & (\Delta < \alpha \ \&\& \ succ > N) \\ cwnd & (\alpha \le \Delta < \beta \text{ or } \Delta < \alpha \ \&\& \ succ \le N) \\ cwnd - 1/cwnd & (\Delta \ge \beta) \end{cases}$$

succ: # of states consecutively entering into $\Delta < \alpha \ \&\& \ succ \le N$

count: suppression parameter to be incremented

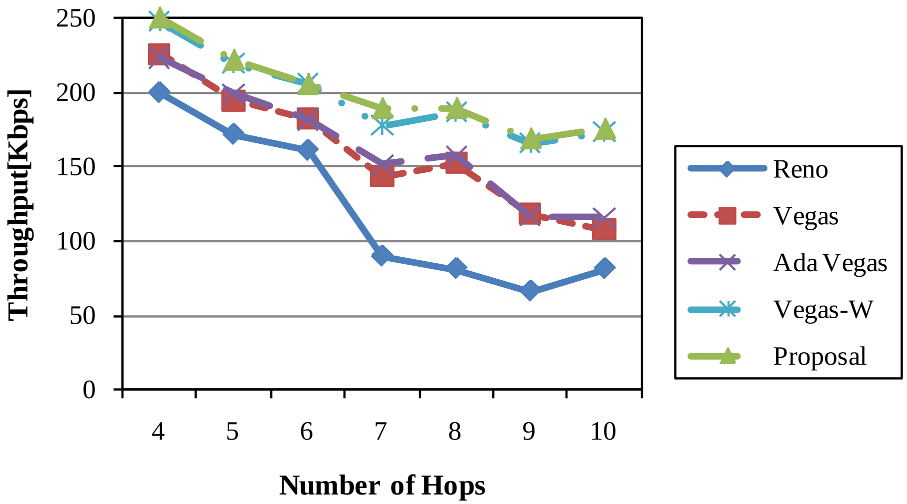
N: succ maximum (e.g. 10)



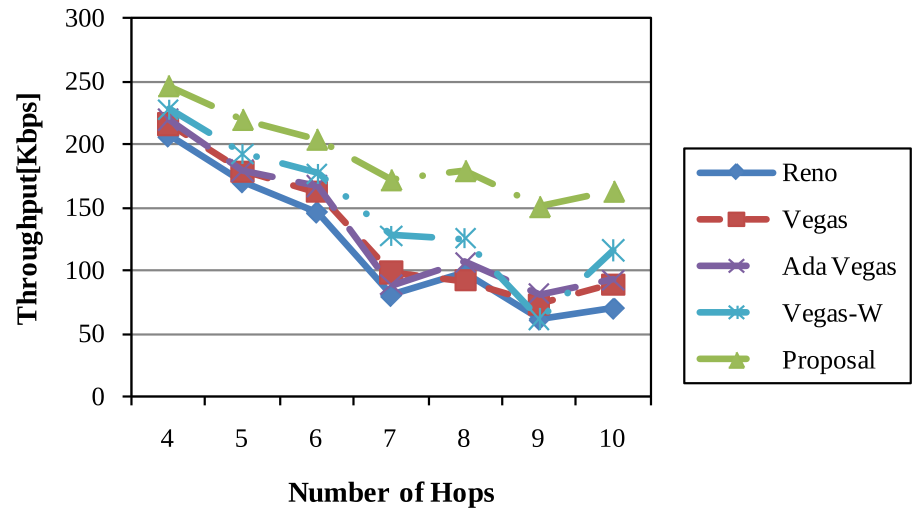
N.Iikubo et al: IEICE Tech. Report, 2009.

Wireless Multihop Networks (4)

※ ns-2 simulations



two flows



four flows

802.11, 2Mbps

effective in slow link capacity,
but might be heurictic

N TCP Flow

0 **1** **2** **R**

Sender **Receiver**

200m

N.Iikubo et al: IEICE Tech. Report, 2009.

Wireless Multihop Networks (6)

- 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

Underwater Sensor Network

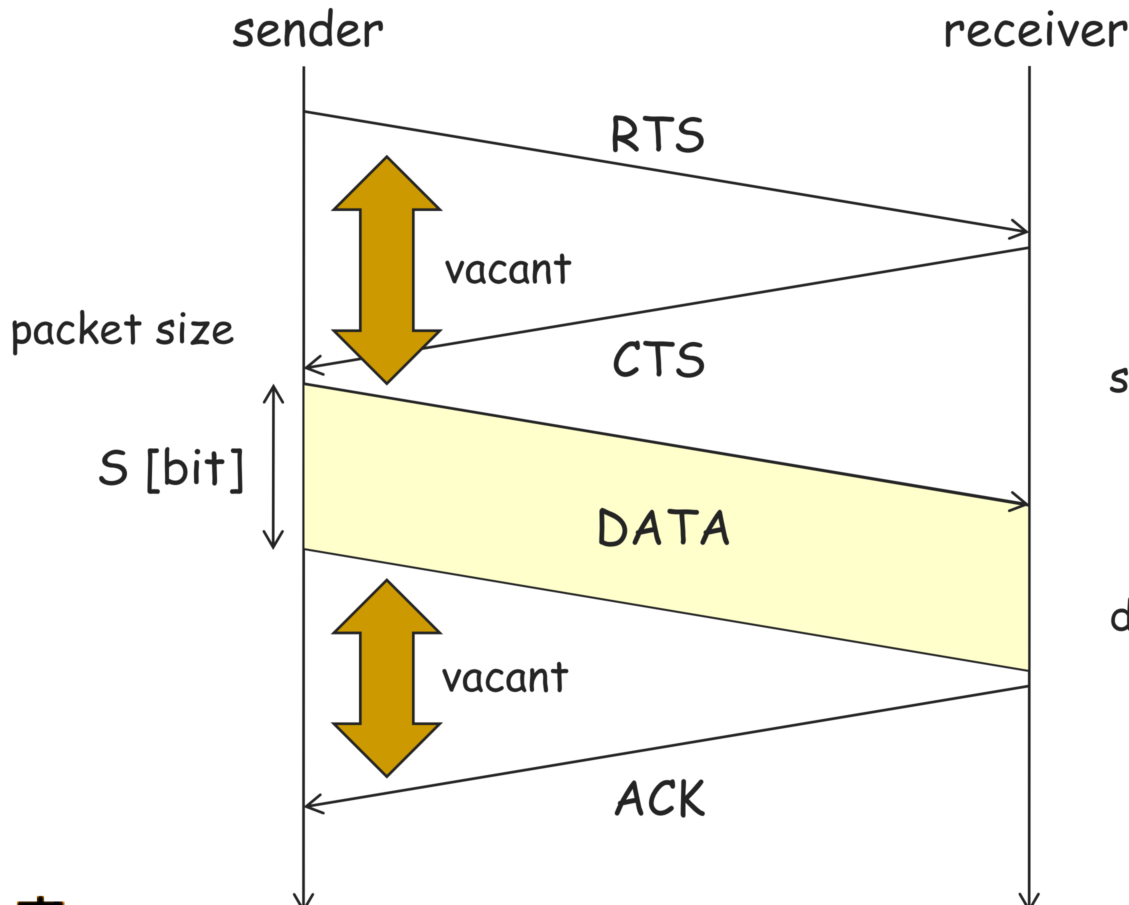
Discussion

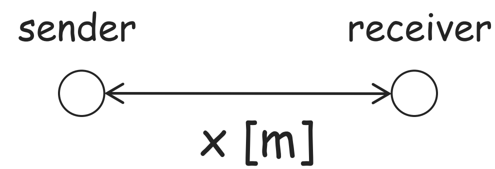
- Uniqueness of underwater sensor networks
  - use acoustic signals instead of electric wave
    - speed of light: 300 000 000 m/s
    - speed of sound (underwater): 1500m/s
  - link utilization ratio decreases as the distance increases
    - due to huge delay
  - interferences and collisions are similar

Underwater Sensor Networks (1)

- Link Utilization (1)

Link utilization decrease due to slow sound speed





speed of sound $v = 1500$ [m/s]

bitrate C [bit/s]

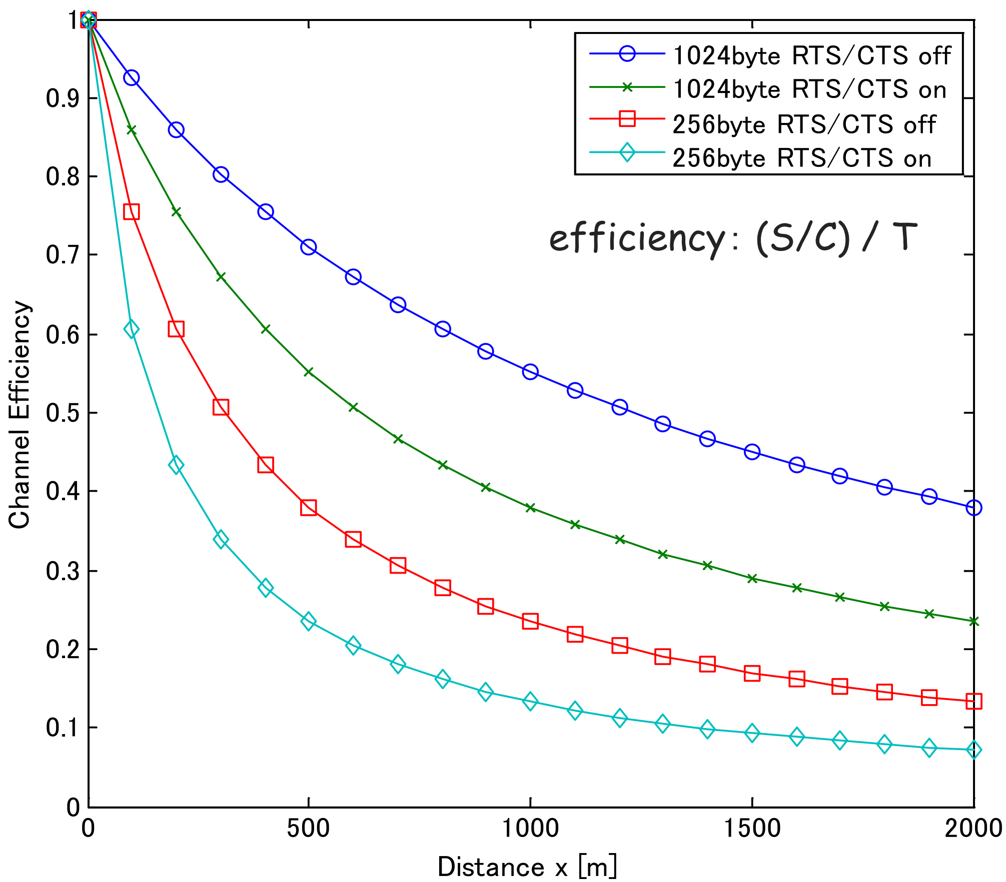
data transmission time

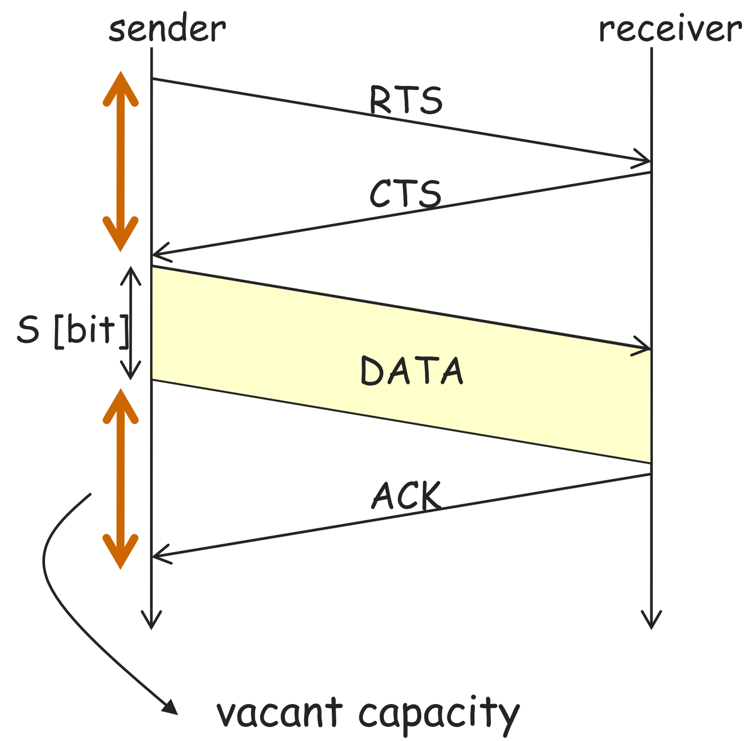
$$T = \underbrace{x/v * 2}_{\text{RTS/CTS}} + S/C + \underbrace{x/v * 2}_{\text{DATA/ACK}}$$

Underwater Sensor Networks (2)

- Link Utilization (2)

relationship between distance and link utilization



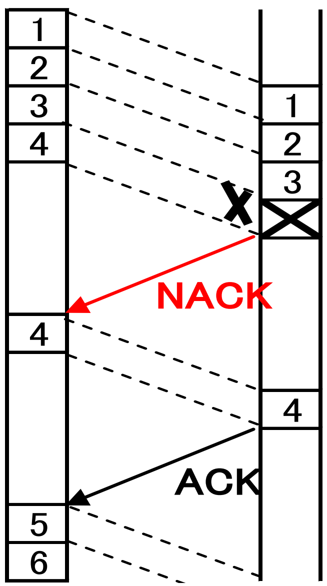


effect of propagation delay
(can be ignored in radio case)

MAC for USN (1)

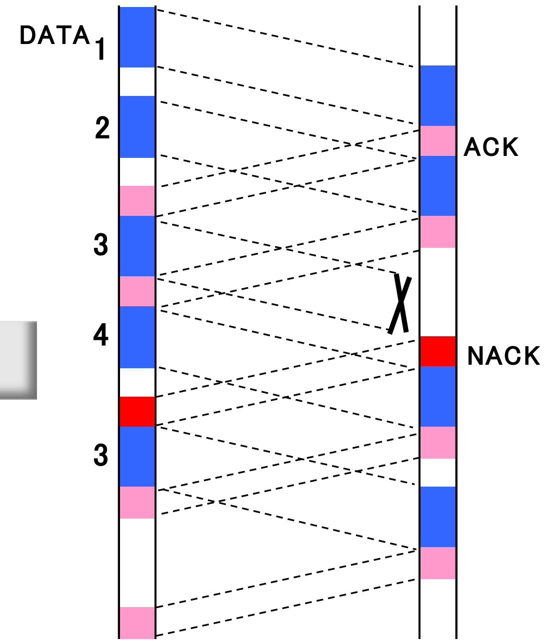
Selective ARQ

J.Rice: ACM WuWNet 2007



JSW ARQ

M.Gao et al., IEEE ICC 2009



combination

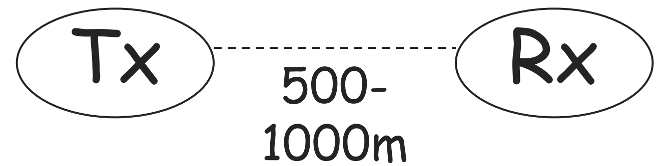
- Delayed NACK & ACK
- used in Seaweb prototype

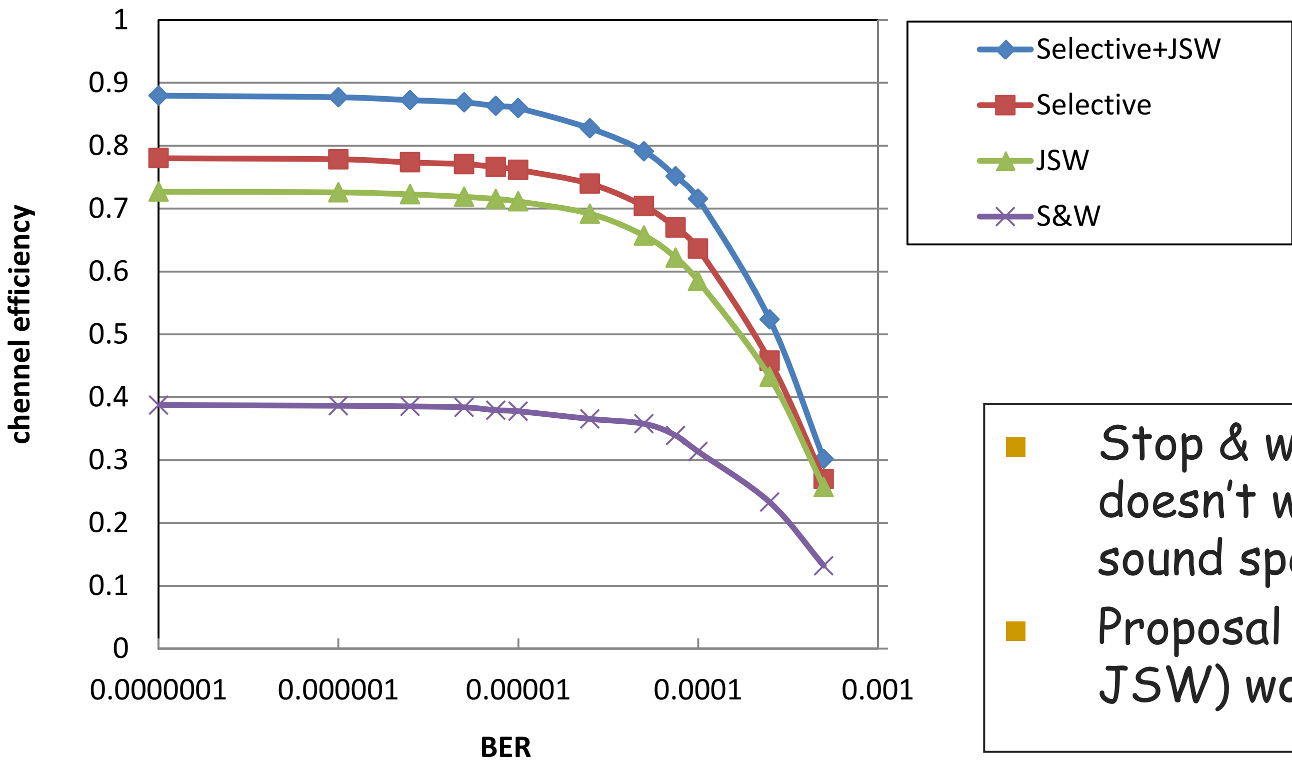
- 1 ACK for 1 DATA
- deliver multiple packets before ACK arrival
- need node synchronization

M.Yoshinaga et al: IEICE Tech. Report, 2011.

MAC for USN (2)

- 500m case





- Stop & wait (1 DATA / 1 ACK) doesn't work well due to slow sound speed
- Proposal (Selective ARQ + JSW) works the best

M.Yoshinaga et al: IEICE Tech. Report, 2011.

TCP for USN (1)

- TCP Hybla
  - TCP for satellite links having large RTT

$$W_{i+1}^{H}=\begin{cases}W_i^H+2^{\rho}-1 & (SS)\\ W_i^H+\rho^2/W_i^H & (CA)\end{cases}$$

W^H: congestion window size

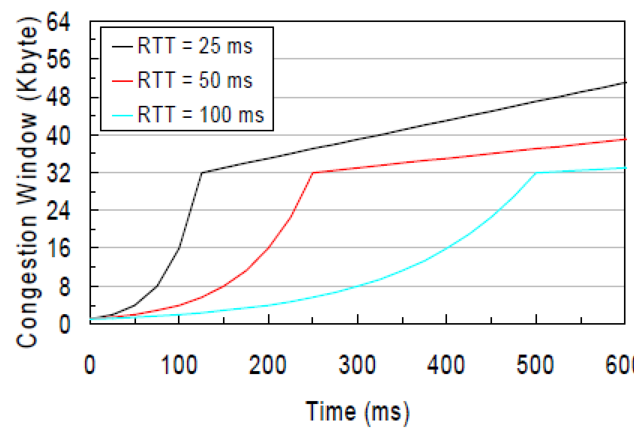
ρ: RTT/RTT_0

RTT: round trip time

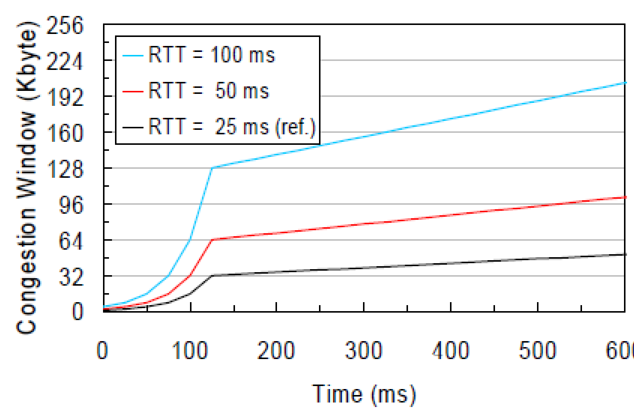
RTT_0: reference RTT (0.025[s])

SS: slow start

CA: congestion avoidance



Reno

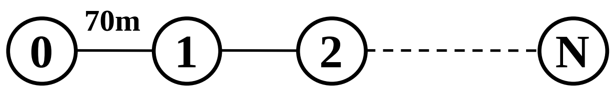


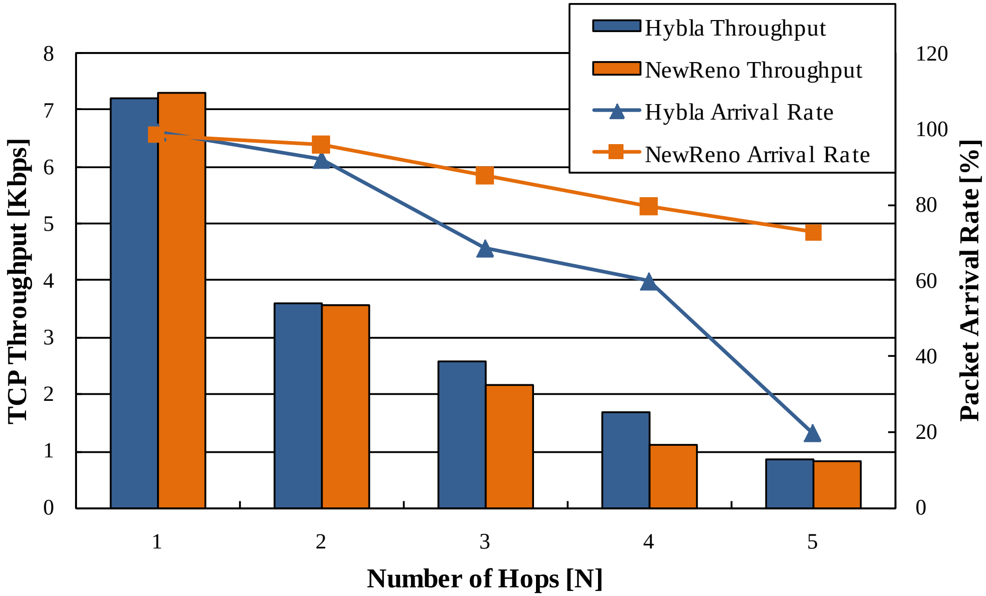
Hybla

C.Caini et al.: EAS ASMS 2003, Jul.2003.

TCP for USN (2)

- TCP Hybla usage for USN





Hybla provides better throughput than Reno, but its PLR is very high

Due to sudden RTT increase, Hybla's congestion window becomes extremely huge

This motivate us to consider lower window increase for multihop & USN (much lower packet loss & low delay)

N.Iikubo et al: IEICE Tech. Report, 2009.

4. Conclusions

Conclusions (1)

- Transport protocol for efficient & low-delay multimedia transmission
  - Hybrid TCP (for wired)
    - efficiency, friendliness, low-delay
  - Wireless extensions
    - TCP differentiation (for WiFi)
    - slow window increase (for multihop & USN)
- Other appoaches
  - Hybrid MAC for USN
  - DTN for Wireless & USN

Conclusions (2)

- Transport protocol integration from wired to wireless/multihop/USN
  - high-efficiency and low-delay
    - without network assistance
      - how to reach target rate
      - how to estimate target rate
    - with network assistane
      - differentiation