On Hybrid TCP Congestion Control

J.Katto, K.Ogura, T.Fujikawa, K.Kaneko & S.Zhou

Dept. of Computer Science, Waseda University

Outline

- Research Backgrounds & TCP-Fusion
  - TCP Variants
  - TFRC: TCP Friendly Rate Control
  - TCP-Fusion
- Analytical Models & Performance Analysis
  - Ideal Models (Loss、Delay、Hybrid)
  - TCP-Fusion Extensions
  - Delay-based TFRC
- Conclusions

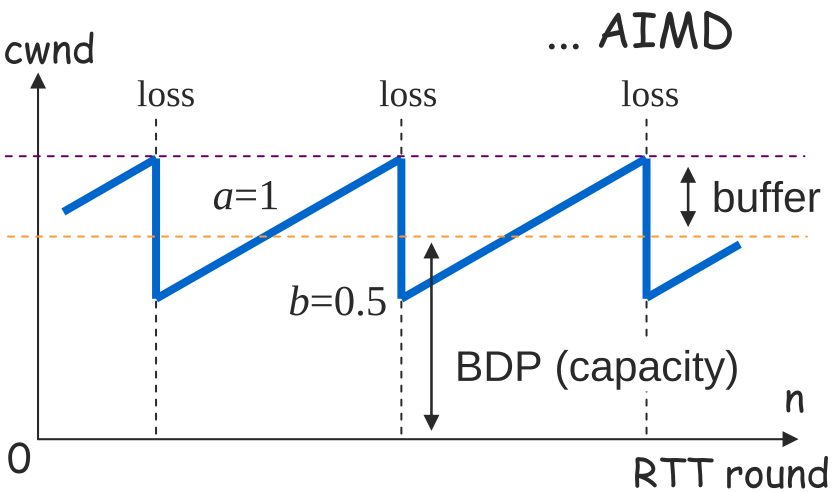
Backgrounds：TCP Variants (1)

- Loss-based (AIMD: Additive Increase Multiplicative Decrease)
  - TCP-Reno / NewReno / SACK
  - High-Speed TCP (IETF RFC 3649, Dec 2003)
  - Scalable TCP (PFLDnet 2003)
  - BIC-TCP / CUBIC-TCP (IEEE INFOCOM 2004, PFLDnet 2005)
  - H-TCP (PFLDnet 2004)
  - TCP-Westwood (ACM MOBICOM 2001)
- Delay-based (RTT Observation)
  - TCP-Vegas (IEEE JSAC, Oct 1995)
  - FAST-TCP (INFOCOM 2004)
- Hybrid (Loss & RTT)
  - Gentle High-Speed TCP (PfHSN 2003)
  - TCP-Africa (IEEE INFOCOM 2005)
  - Compound TCP (PFLDnet 2006)
  - Adaptive Reno (PFLDnet 2006)
  - TCP-Illinois (ValueTools 2006)
  - YeAH-TCP (PFLDnet 2007)
  - TCP-Fusion (PFLDnet 2007)

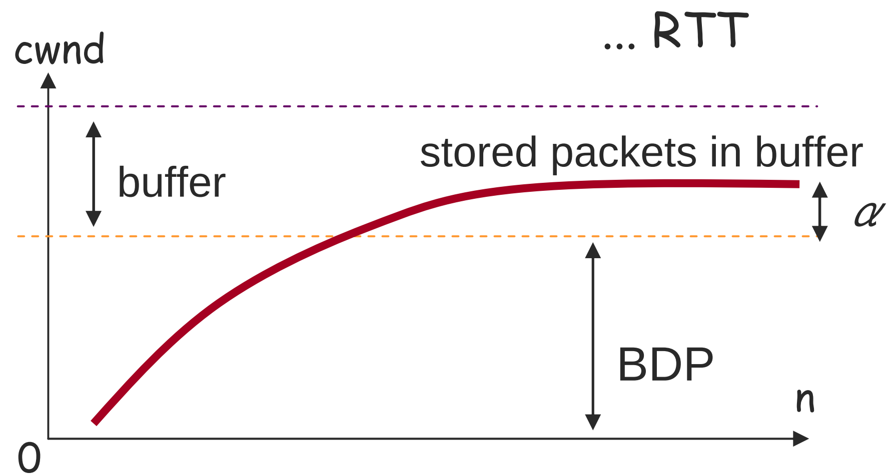
Quite many proposals !

Backgrounds：TCP Variants (2)

- Loss-based

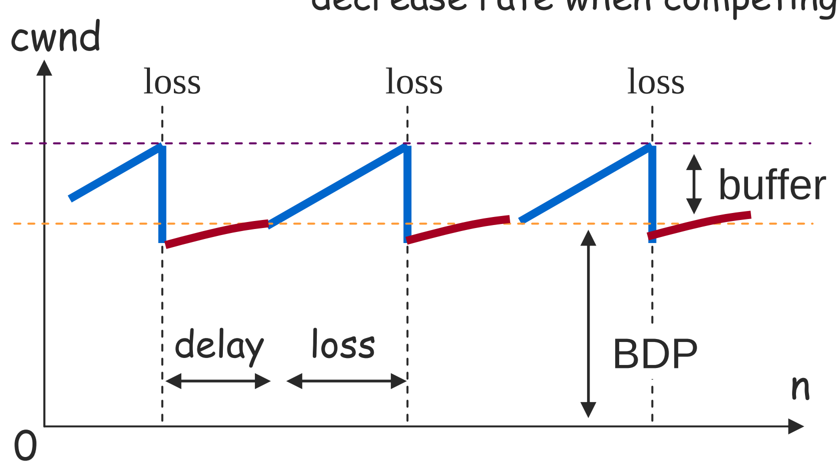


- Delay-based



* decrease rate when competing with AIMD

- Hybrid
  - RTT increase: Loss
  - no RTT increase: Delay



BDP: Bandwidth-Delay Product

Backgrounds：TCP Variants (3)

- Loss-based

Variants	Increase / Update (a)	Decrease (b)
TCP-Reno	1	0.5
HighSpeed TCP (HS-TCP)	$a(w)=\frac{2w^2\cdot b(w)\cdot p(w)}{2-b(w)}$ e.g. 70 (10Gbps, 100ms)	$b(w)=\frac{\log(w)-\log(W_{low})}{\log(W_{high})-\log(W_{low})}(b_{high}-0.5)+0.5$ e.g. 0.1 (10Gbps, 100ms)
Scalable TCP (STCP)	0.01 (per every ACK)	0.875
BIC-TCP	$\begin{cases} additive\ increase\ (fast) \\ binary\ search\ (slow) \\ max\ probing\ (fast) \end{cases}$	0.875
CUBIC-TCP	$w=0.4(t-\sqrt[3]{2W_{\max}})^3+W_{\max}$	0.8
H-TCP	$\alpha \leftarrow 2(1-\beta)\{1+10.5\cdot(t-TH)\}$	$\beta \leftarrow \begin{cases} 0.5 & for\ \left\lvert\frac{B(k+1)-B(k)}{B(k)}\right\rvert>2 \\ \frac{RTT_{\min}}{RTT_{\max}} & otherwise \end{cases}$
TCP-Westwood (TCPW)	1	$\begin{cases} RE*RTT_{\min}/PS & (not\ congested) \\ BE*RTT_{\min}/PS & (congested) \end{cases}$

Backgrounds: TCP Variants (4)

- Delay-based

a b

Variants	Update	Decrease
TCP-Vegas	$w \leftarrow \begin{cases} w+1 & (no\ congestion) \\ w & (stable) \\ w-1 & (early\ congestion) \end{cases}$	0.75
FAST-TCP	$w \leftarrow \min\left\{2w, (1-\gamma)w + \gamma\left(\frac{RTT_{\min}}{RTT} w + \alpha\right)\right\}$	0.5 (?)

Backgrounds：TCP Variants (5)

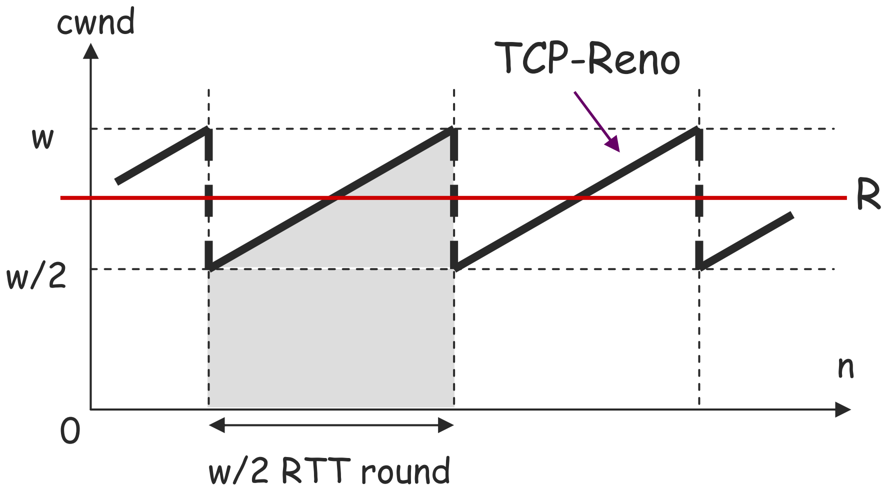
- Hybrid

Variants	Increase a	Decrease b
Gentle HS-TCP	HS-TCP / Reno	HS-TCP
TCP-Africa	HS-TCP / Reno	HS-TCP
Compound TCP (CTCP)	$0.125 \cdot cwnd^{0.75}$ / Reno	0.5
Adaptive Reno (ARENO)	$B/10\text{Mbps}$ / Reno	$\begin{cases} 1 & (non\ congestion\ loss) \\ 0.5 & (congestion\ loss) \end{cases}$
YeAH-TCP	STCP / Reno	$\max\left(\frac{RTT_{\min}}{RTT}, 0.5\right)$
TCP-Fusion	$\frac{B * D_{\min}}{PS}$ / Reno	$\max\left(\frac{RTT_{\min}}{RTT}, 0.5\right)$

Backgrounds：Models & TFRC (1)

- TFRC: TCP Friendly Rate Control

* equation based rate control



w: cwnd when packet losses happen
p: packet loss ratio
RTT: round trip time
R: (equivalent) rate
b: delayed ACK counts

$$\Rightarrow \begin{cases} p = \dfrac{8}{3w^2} \\ R = \dfrac{PS}{RTT} \cdot \sqrt{\dfrac{3}{2p}} \end{cases}$$

with timeout consideration

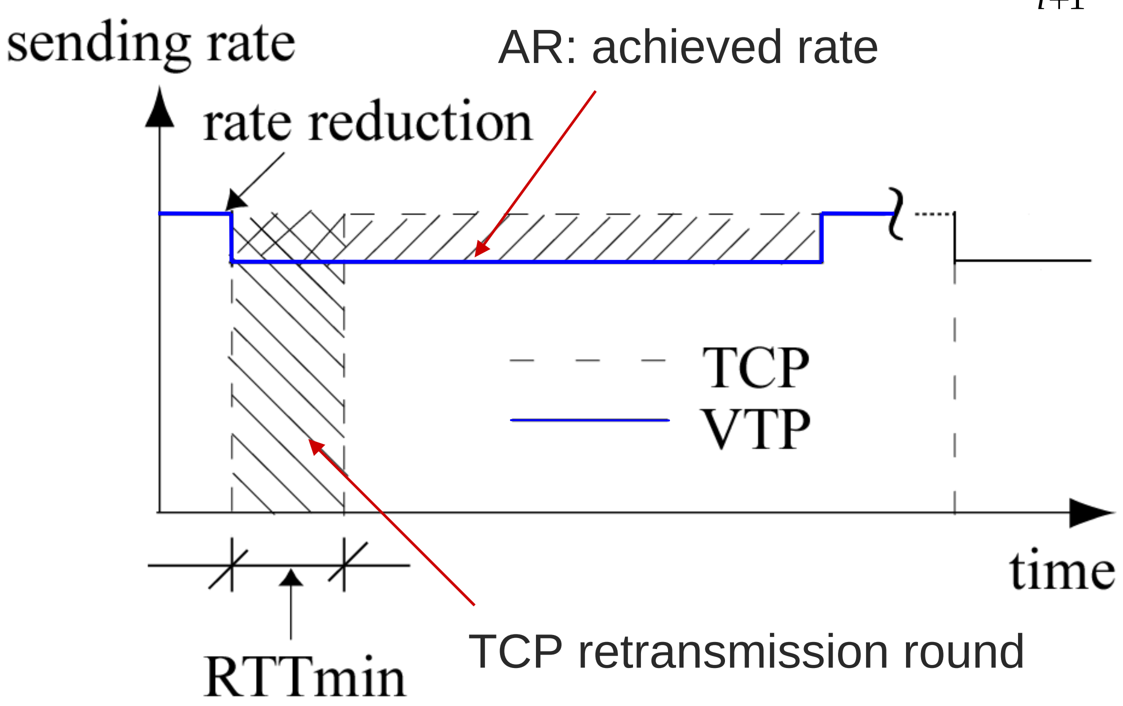
$$\Rightarrow R_{loss} = \frac{PS}{RTT\sqrt{\frac{2bp}{3}} + t_{RTO,loss} \cdot 3\sqrt{\frac{3bp}{8}} \cdot p(1+32p^2)}$$

Backgrounds: Models & TFRC (2)

- VTP: Video Transport Protocol

$$R_{i+1} = \frac{ewnd_{i+1}}{RTT_{i+1}} = \frac{R_i * RTT_i + 1}{RTT_i + \Delta RTT}$$

$$\Delta RTT = RTT_i - RTT_{i-1}$$



RTT_{min}: rate increase

RTT increase: compensate

rate increase by ⊿RTT

similar behavior to TCP-Reno

Problems & Objectives

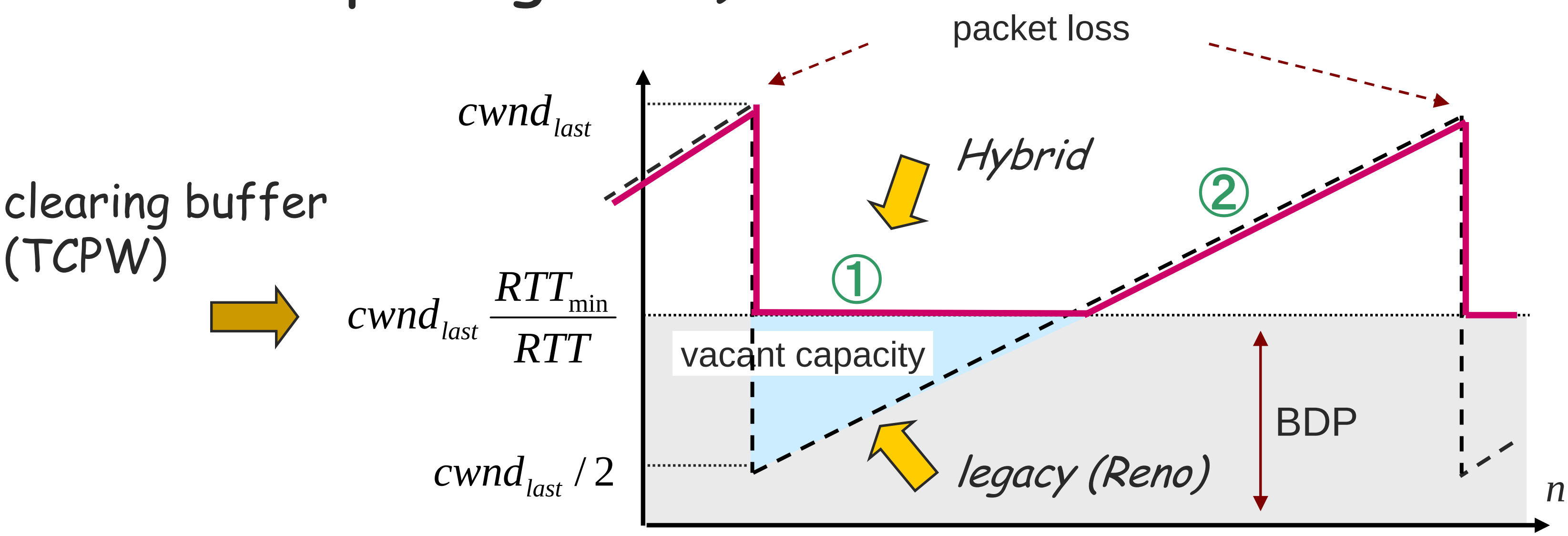
- On Hybrid TCP Congestion Control
  - <u>Theoretical background is not sufficient</u> (best performance is not clear)
  - <u>Too many tuning parameters</u> (we have to tune them by many simulations and implementations)

↓

  - Building an analytical model for hybrid TCP congestion control
  - Theoretically validates its throughput efficiency and friendliness against legacy TCPs
  - Enables analytical parameter tuning
  - Formulates a new TFRC method

TCP-Fusion (1)

- Basic mechanism (single flow case: no competing flow)

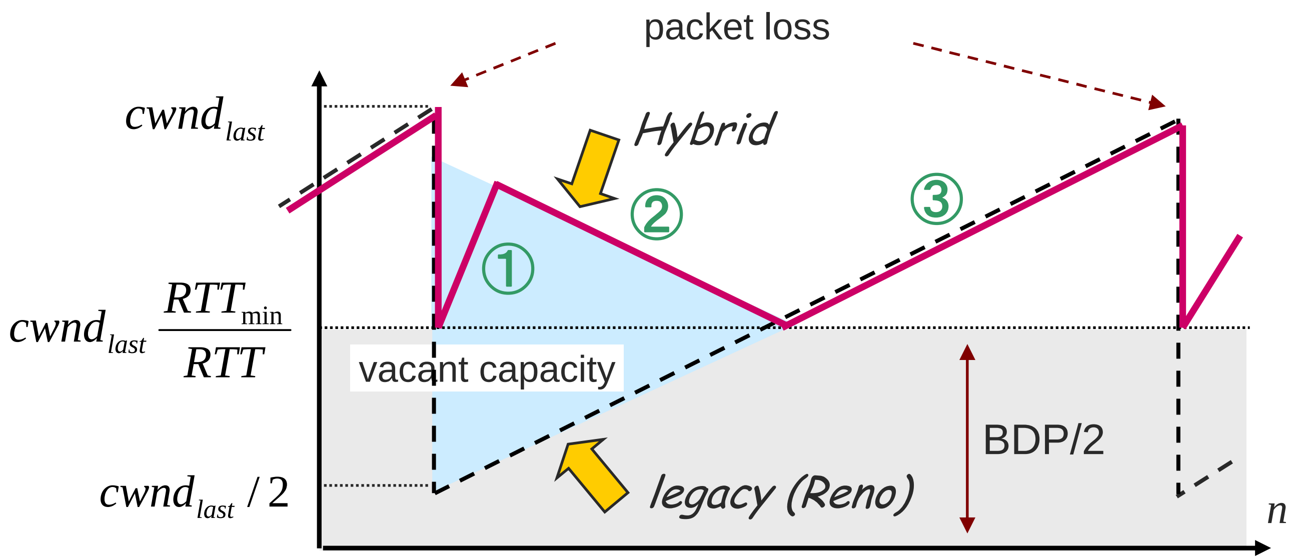


Window control by switching two modes (loss & delay):
① Keeps constant rate until RTT increases (delay mode)
② Performs as TCP-Reno (loss mode)

TCP-Fusion (2)

- Basic mechanism (two flow case: when competing with TCP-Reno)

clearing buffer (TCPW)



Window control by switching two modes (loss & delay):
① Rapid increase of cwnd (delay … throughput efficiency)
② Gradual decrease of cwnd (delay … avoids congestion)
③ Performs as TCP-Reno (loss … friendliness to legacy TCP)

TCP-Fusion (3)

- Actual window control algorithm (later)

$$cwnd_{new} = \begin{cases} cwnd_{last} + W_{inc} / cwnd_{last}, & \text{if } diff < \alpha \\ cwnd_{last} + (-diff + \alpha) / cwnd_{last}, & \text{if } diff > 3 * \alpha \\ cwnd_{last}, & \text{otherwise} \end{cases}$$

→ (2)-① (first case)
→ (2)-② (second case)
→ (1)-① (otherwise)

$$cwnd_{new} = reno\_cwnd, \quad \text{if } cwnd_{new} < reno\_cwnd$$

→ (1)-② (2)-③

W_{inc}: window increase parameter
α: target number of buffered packets
diff: estimated number of buffered packets

Ideal Model (1)

- Ideal Model Definition
  - Loss-based (AIMD: TCP-Reno) :
    - cwnd += 1 (per RTT round)
    - cwnd *= 1/2 (when packet losses happen)
  - Delay-based :
    - always fills a pipe (BDP) without causing RTT increase
  - Hybrid :
    - performs in delay-mode when RTT stays at its minimum
    - performs in loss-mode when RTT increase is observed

Ideal Model (2)

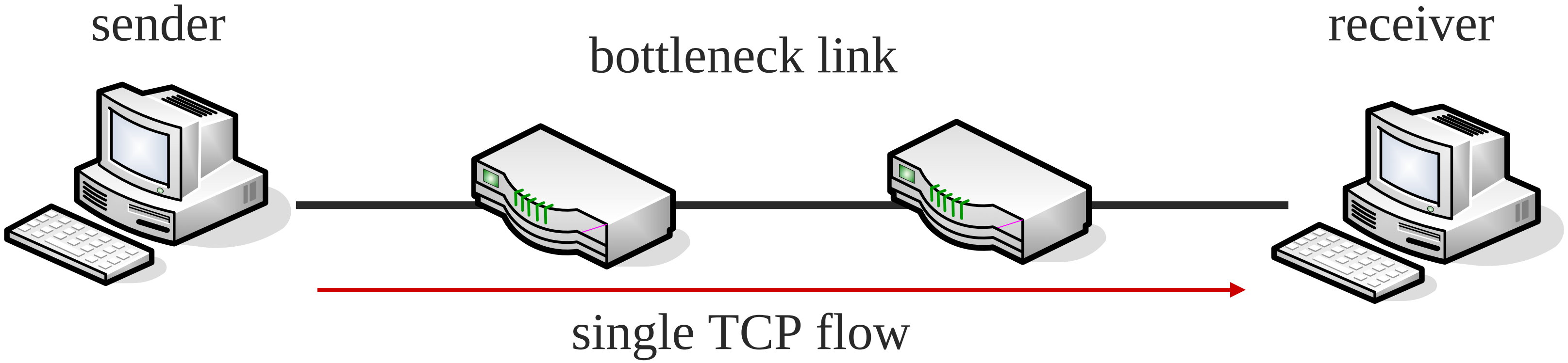
- Parameter definition
  - w : cwnd when packet losses happen
  - W : number of packets corresponding to BDP
  - p : packet loss rate

- Assumption
  - Next equation holds for random loss case (e.g. RED router and wireless) as well as for buffer overflow case

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

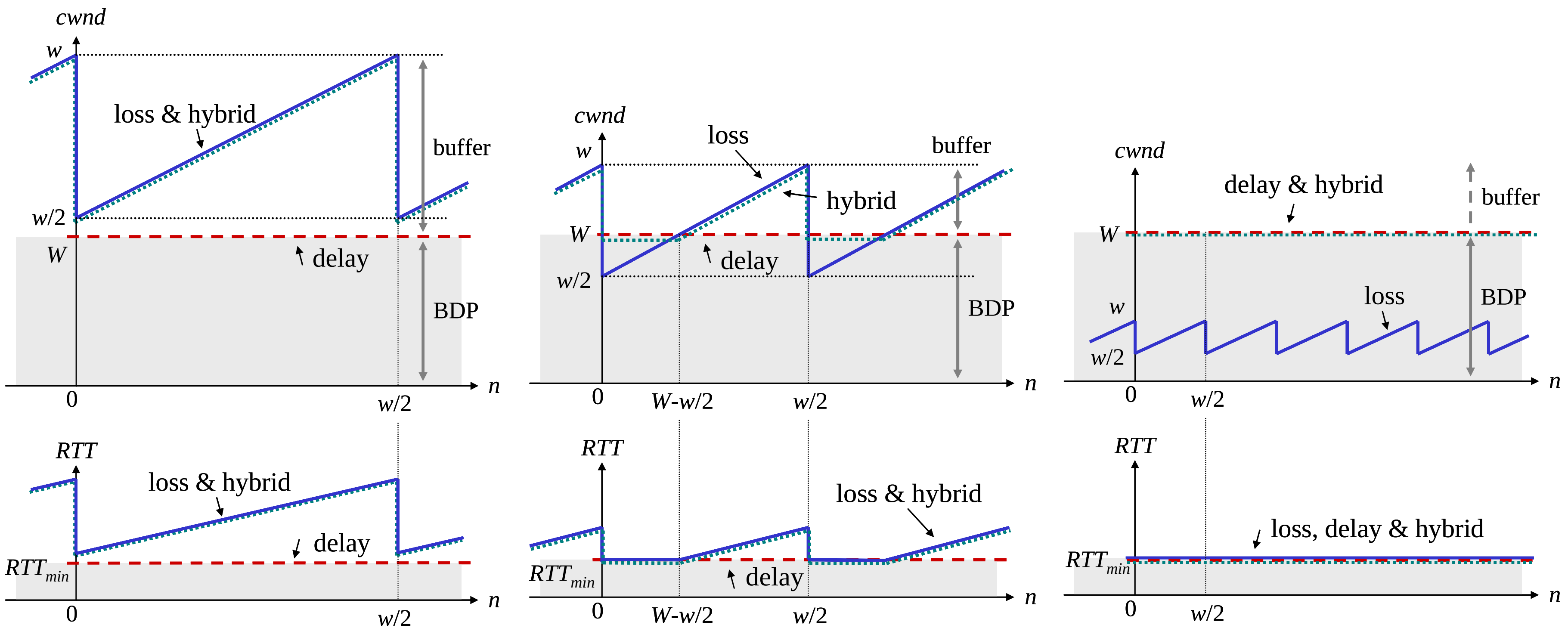
Single Flow Model (1)

- Connection Topology



Single Flow Model (2)

- Classification to three cases



(i) $W < w/2$ (low PER)

always buffering (loss-mode)

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

vacant → buffering (delay → loss)

(iii) $w < W$ (high PER)

always vacant (delay-mode)

Single Flow Model (3)

- Transmitted packets and elapsed time per single congestion avoidance round

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	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{8}(3w^2 - 4wW) \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{2}(w-W)^2 \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min}$

Single Flow Model (4)

- Incorporation of timeout penalty
  - Timeout penalty of TCP-Reno

$$t_{RTO,loss} = T_0 \cdot \left(1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6\right)/(1 - p)$$

  - Ratio of transmitted packets

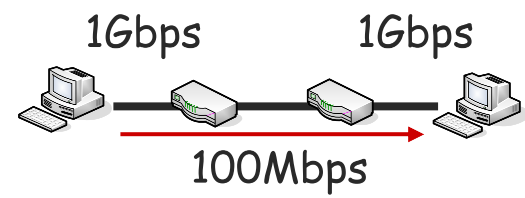
$$t_{RTO,delay} = \frac{K_{delay}}{K_{loss}} \cdot t_{RTO,loss}$$

  - Estimated throughput

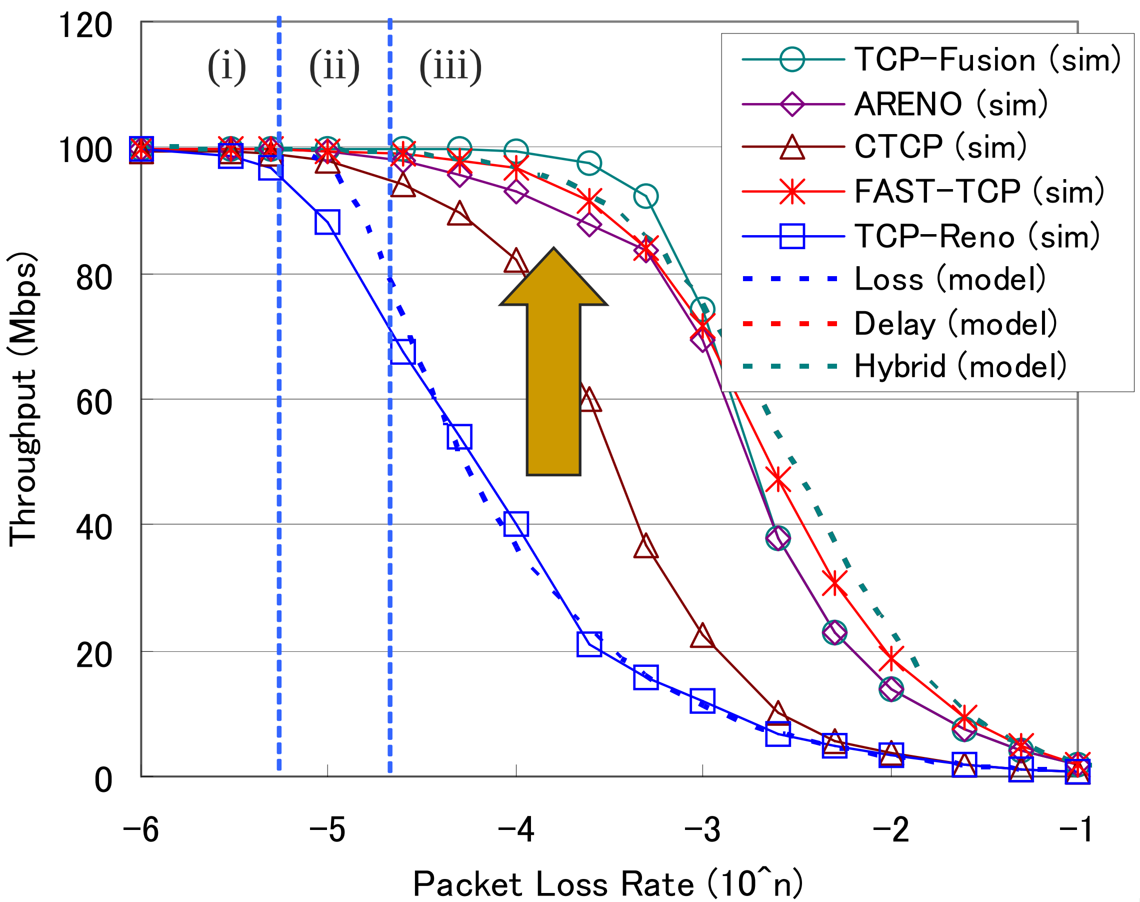
$$\frac{transmitted\ \ packets}{elapsed\ time\ +\ timeout\ \ penalty}$$

Single Flow Model (5)

- Result (1) PLR



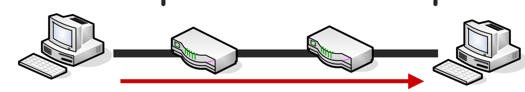
buffer size = BDP (constant)
PLR : variable

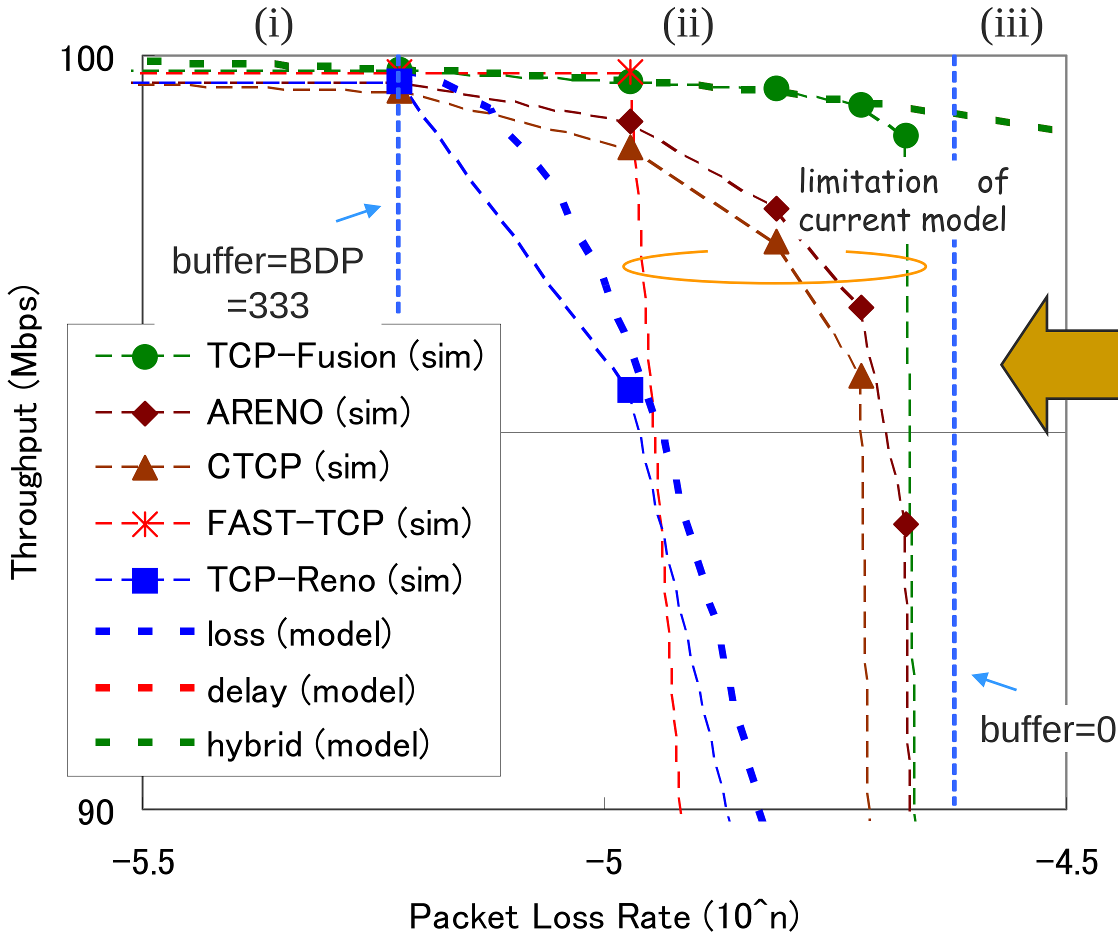


When PLR is large (in regions (ii) & (iii)), delay based and hybrid TCPs drastically outperform loss based TCP.

Single Flow Model (6)

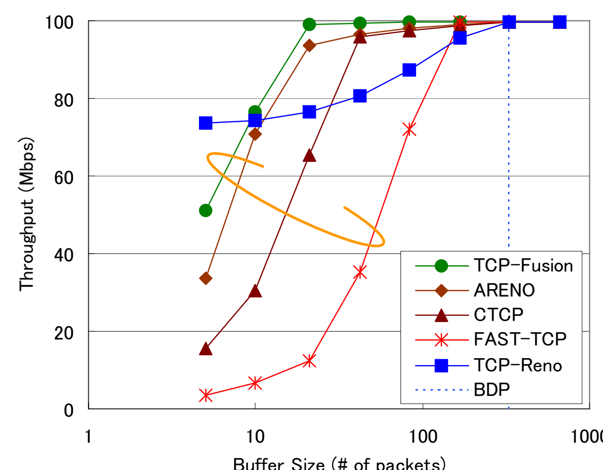
- Result (2) buffer size





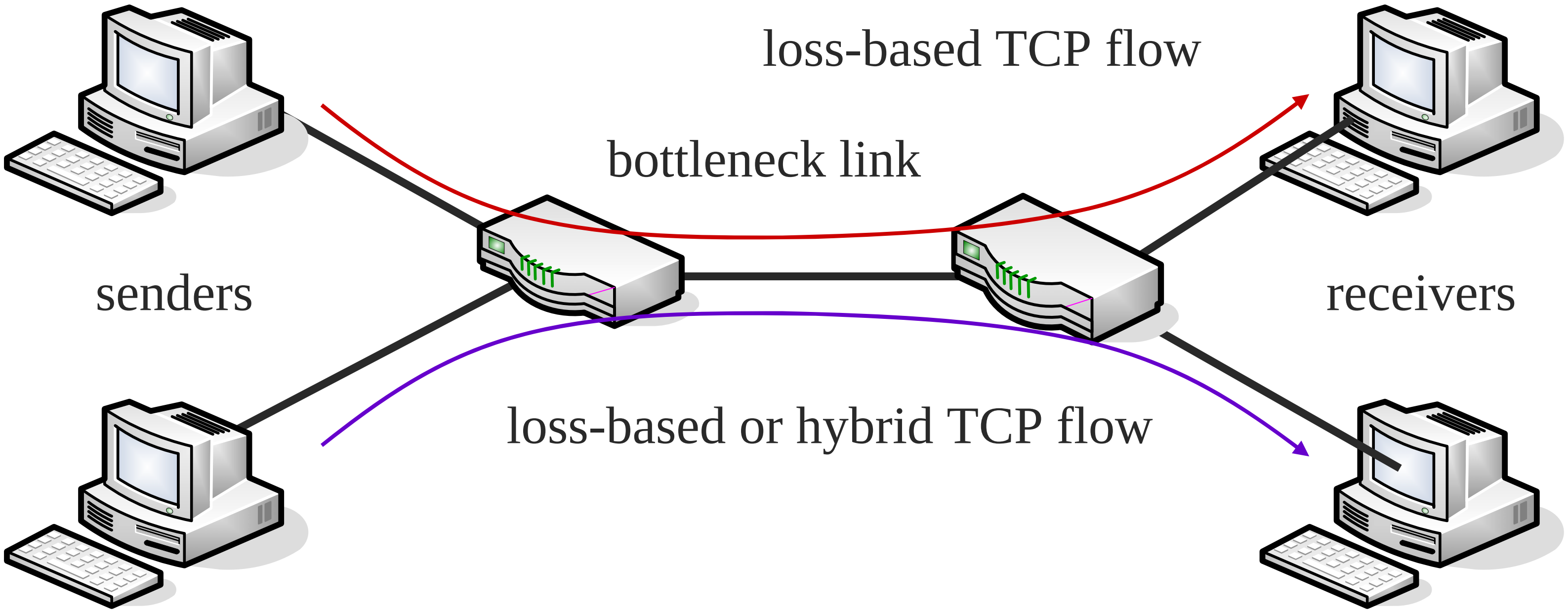
PLR = 10^{-6} (constant)

buffer size : variable



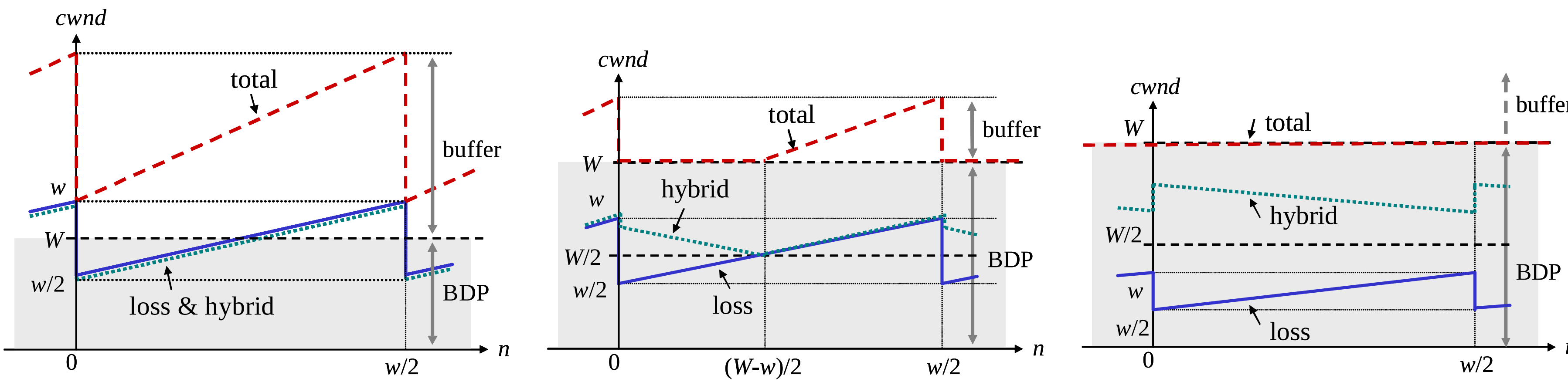
Two Flow Model (1)

- Connection Topology



Two Flow Model (2)

- Classification to three cases



(i) $W < w$ (low PLR)

always buffering
(loss mode)

(ii) $w < W < 2*w$ (medium PLR)

vacant → buffering
(delay → loss)

(iii) $2*w < W$ (high PLR)

always vacant
(delay mode)

(1) w is scaled to half value
(2) delay-based TCP is omitted

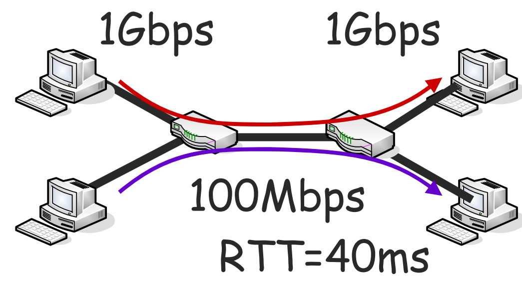
Two Flow Model (3)

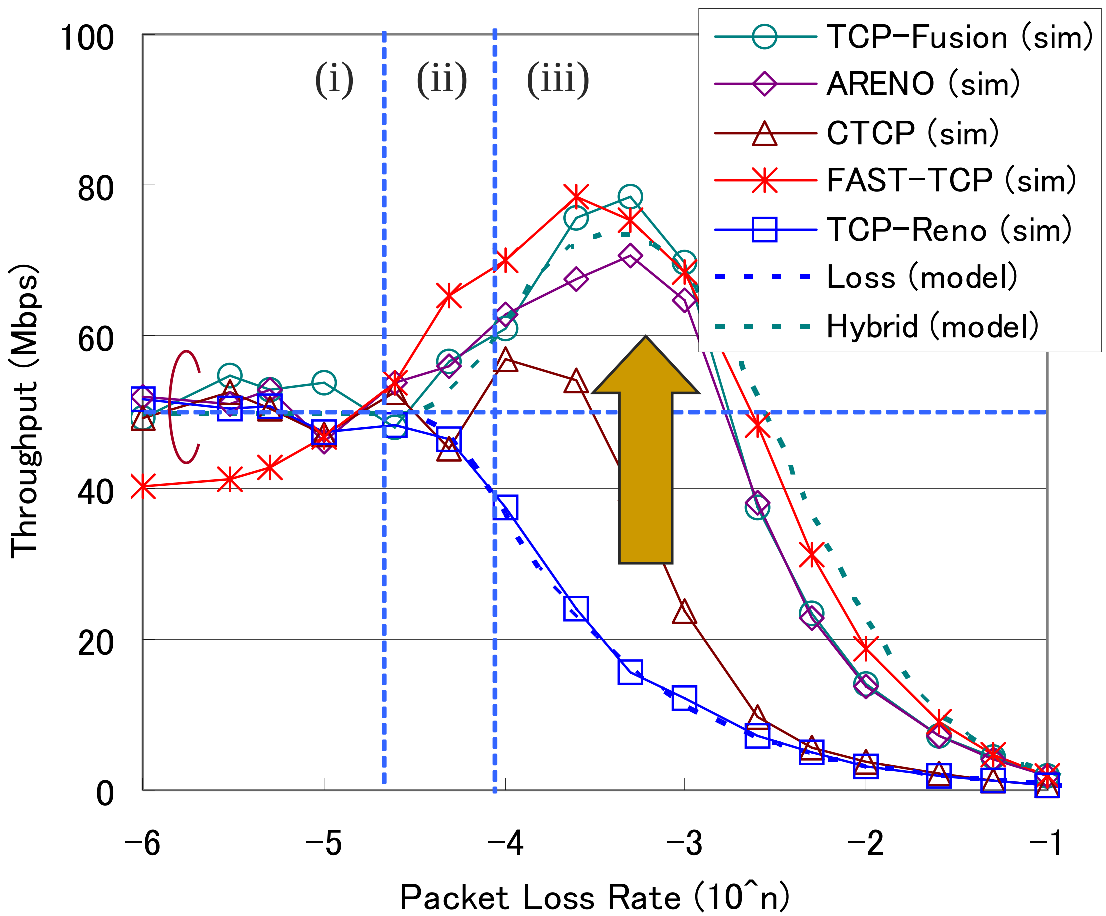
- Transmitted packets and elapsed time per single congestion avoidance round

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	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{4}w(3w-2W) \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min} + \frac{1}{4}(2w-W)^2 \cdot \frac{PS}{B}$	$\frac{1}{2}w \cdot RTT_{\min}$

Two Flow Model (4)

- Result





When PLR is low (region (i)), hybrid TCP shows good friendliness to legacy TCP.

When PLR is high (regions (ii) & (iii)), hybrid TCP shows high throughput efficiency.

TCP-Fusion (3)

- Actual window control algorithm

$$cwnd_{new} =$$
$$\begin{cases} cwnd_{last} + W_{inc} / cwnd_{last}, & \text{if } diff < \alpha \\ cwnd_{last} + (-diff + \alpha) / cwnd_{last}, & \text{if } diff > 3 * \alpha \\ cwnd_{last}, & \text{otherwise} \end{cases}$$

$$cwnd_{new} = reno\_cwnd, \ \text{if } \ cwnd_{new} < reno\_cwnd$$

W_{inc}: window increase parameter
α: target number of buffered packets
$diff$: estimated number of buffered packets

TCP-Fusion (4)

- α and W_{inc} :

D_{min}: minimum time resolution which can be detected by the end host

N : number of flows

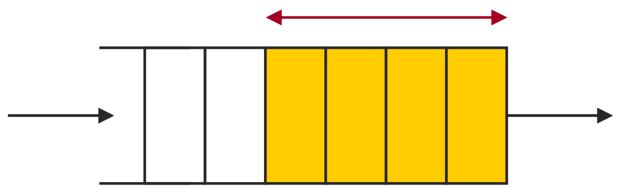
B : link bandwidth estimation

RE : rate estimation

PS : packet size

B and RE can be estimated by using TCP-Westwood mechanism

G: minimum buffer size which can be estimated by end host



$$\alpha = \frac{G}{N} = \frac{(B/N) * D_{\min}}{PS} \approx \frac{RE * D_{\min}}{PS}$$

minimum packets which will not cause buffer overflow and scales to the number of flows.

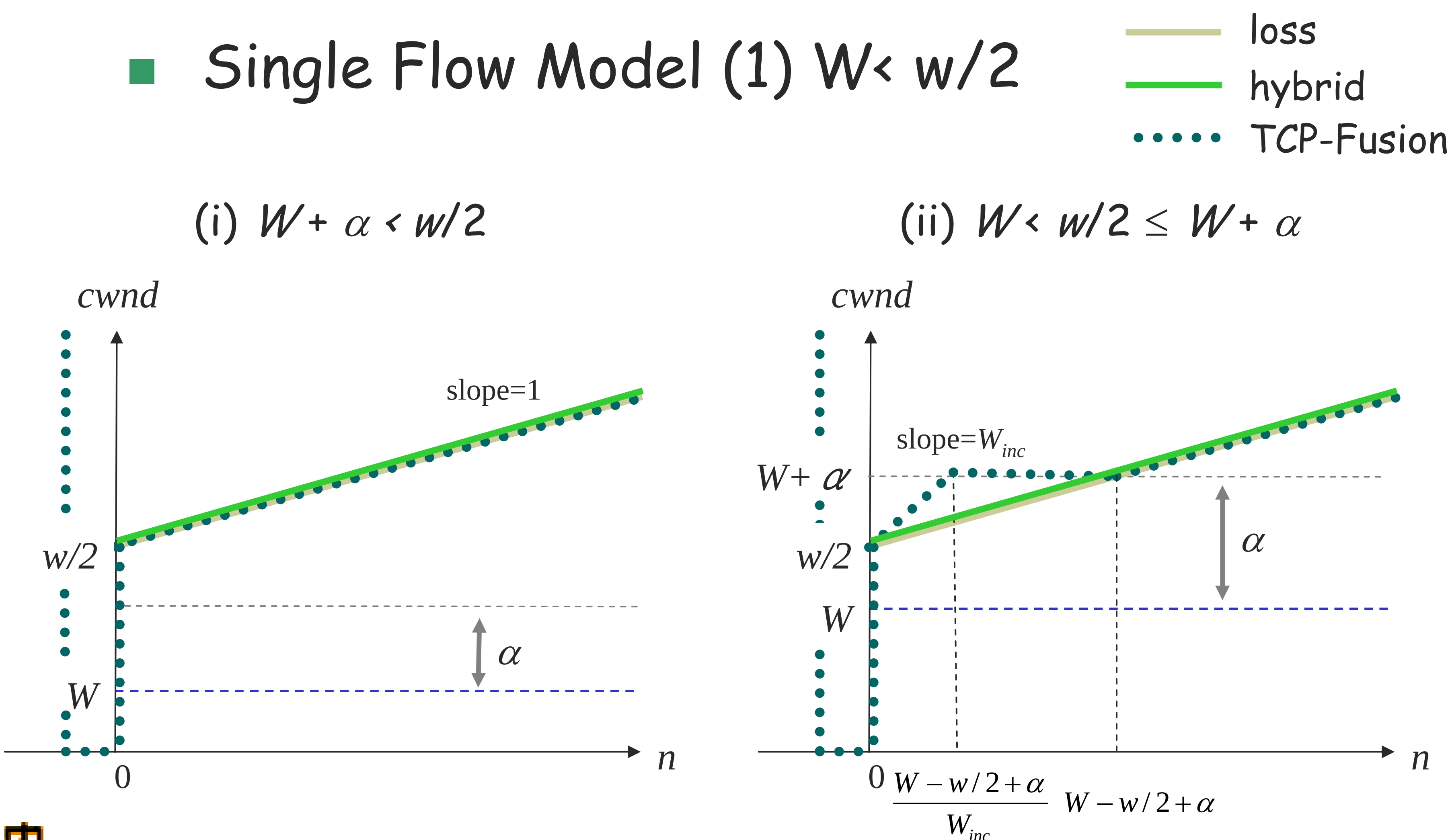
$$W_{inc} \le G = \frac{B * D_{\min}}{PS}$$

the number of packets which can recover to α buffered packet state in a single RTT round.

Model Improvement (1)

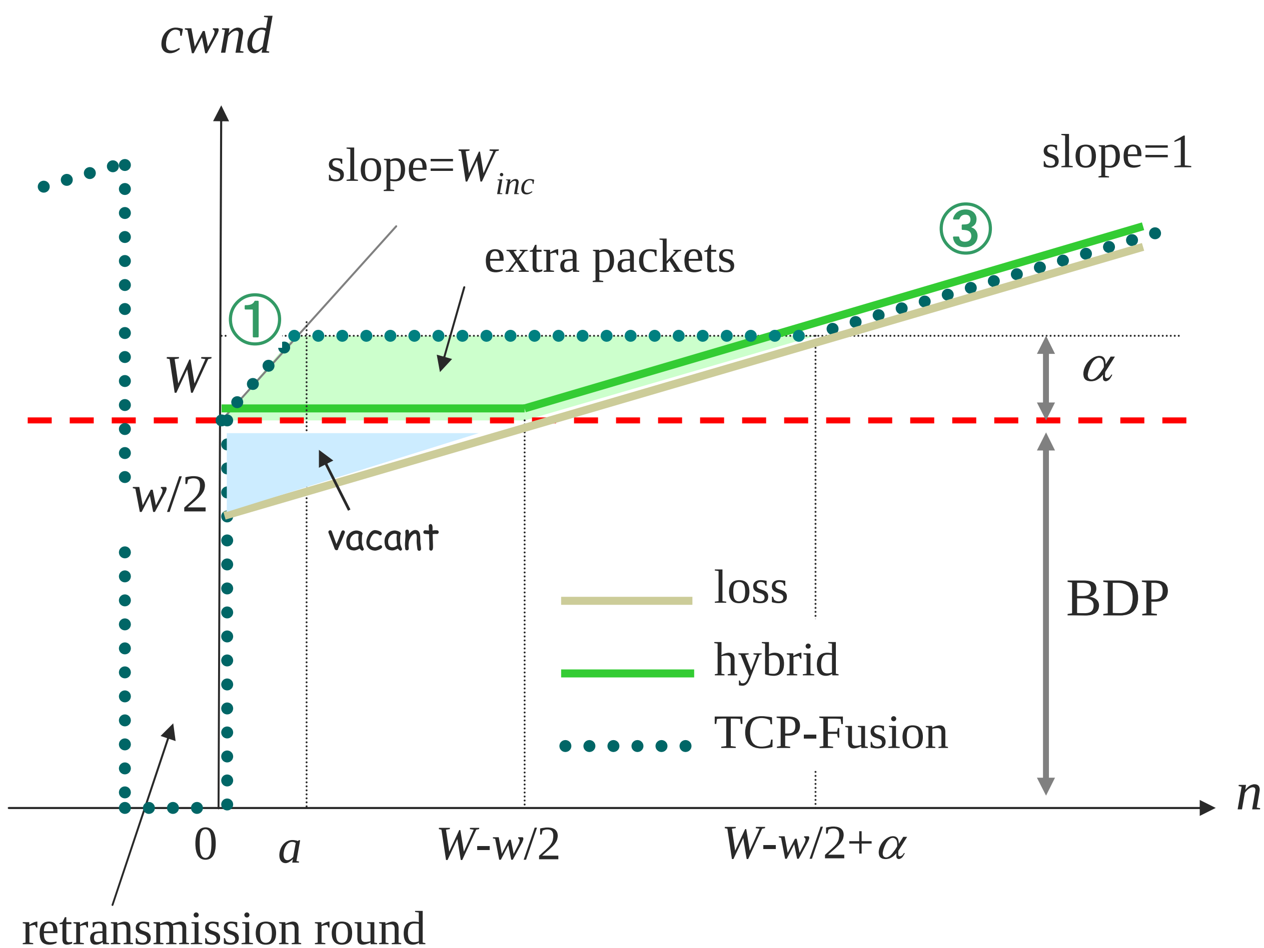
- Single Flow Model (1) $W < w/2$

(i) $W + \alpha < w/2$ (ii) $W < w/2 \le W + \alpha$



Model Improvement (2)

- Single Flow Model (2) W > w/2

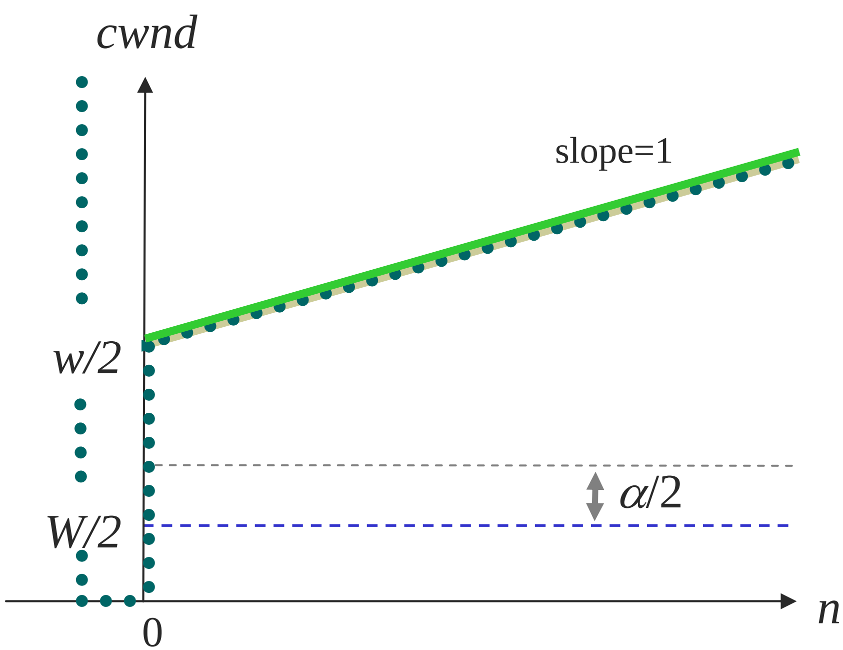


Model Improvement (3)

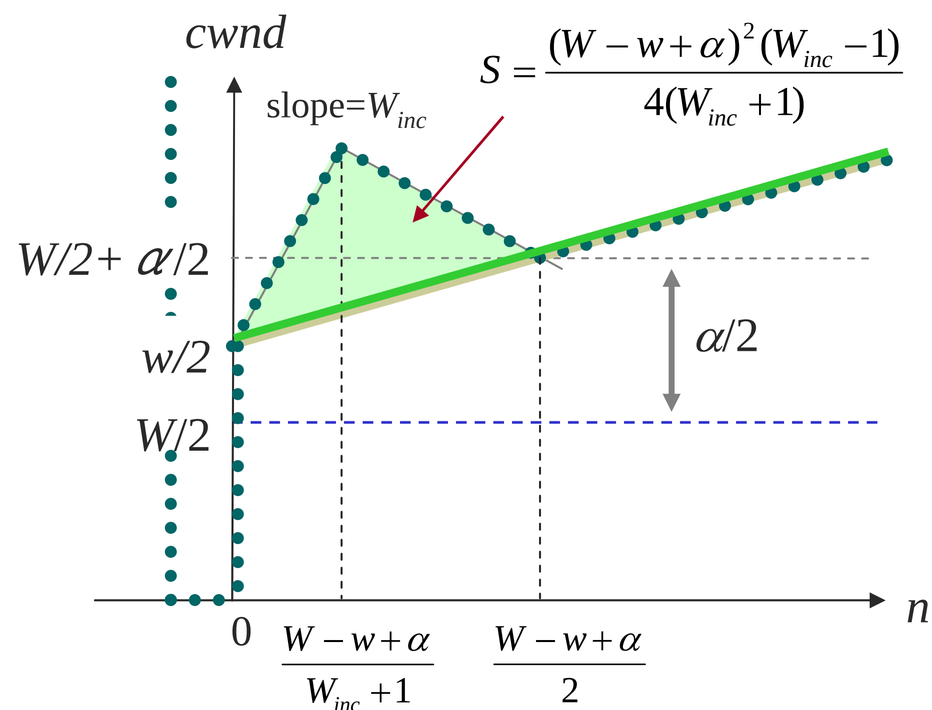
- Two Flow Model (1) low PLR

loss
hybrid
TCP-Fusion

(i) $W + \alpha < w$



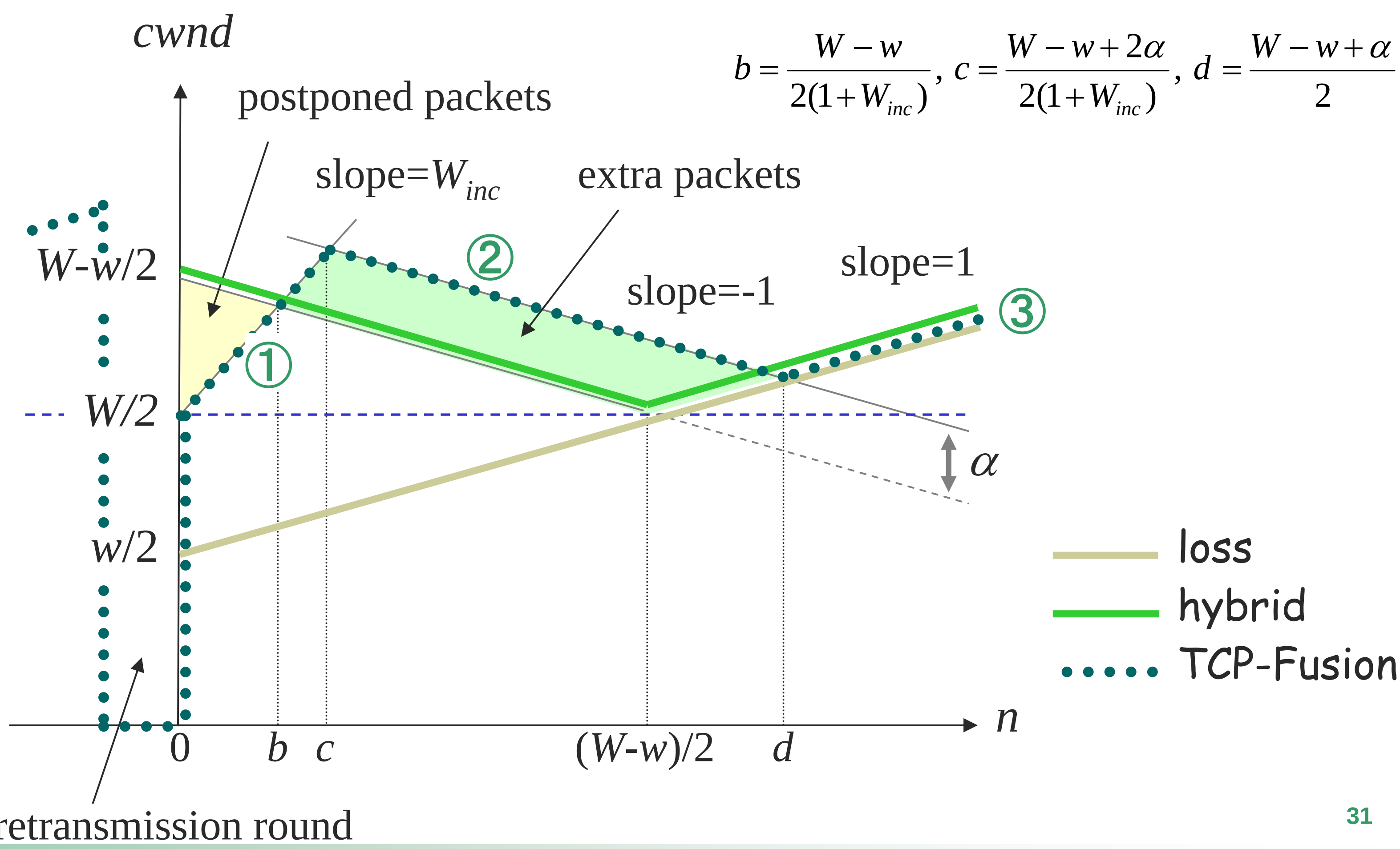
(ii) $W < w \leq W + \alpha$



Model Improvement (4)

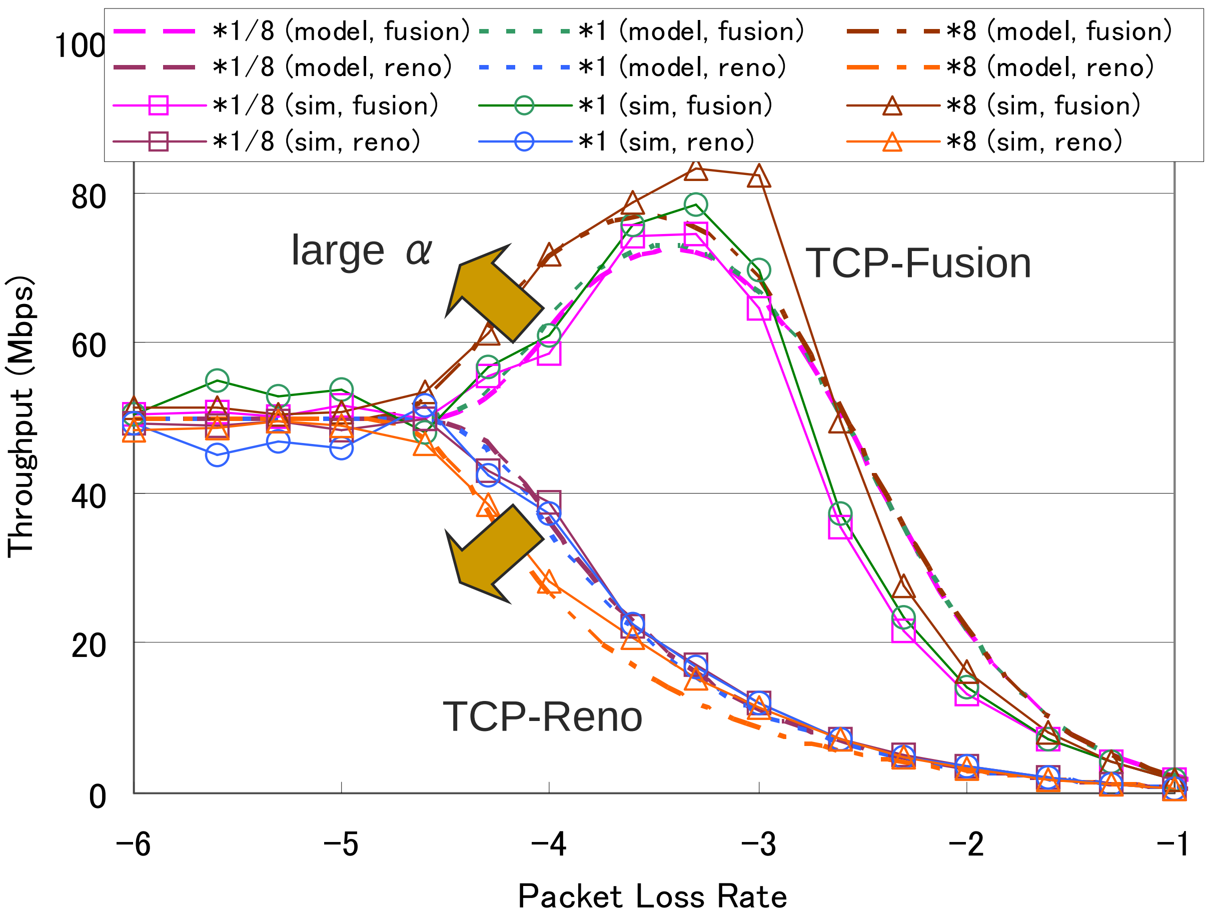
- Two Flow Model (2) high PLR

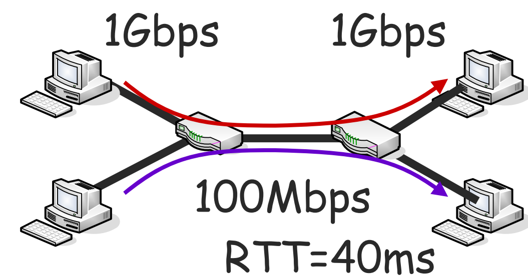
$$b = \frac{W-w}{2(1+W_{inc})},\ c = \frac{W-w+2\alpha}{2(1+W_{inc})},\ d = \frac{W-w+\alpha}{2}$$



Model Improvement (5)

- Result (1) effect of parameter α





Buffered packets:

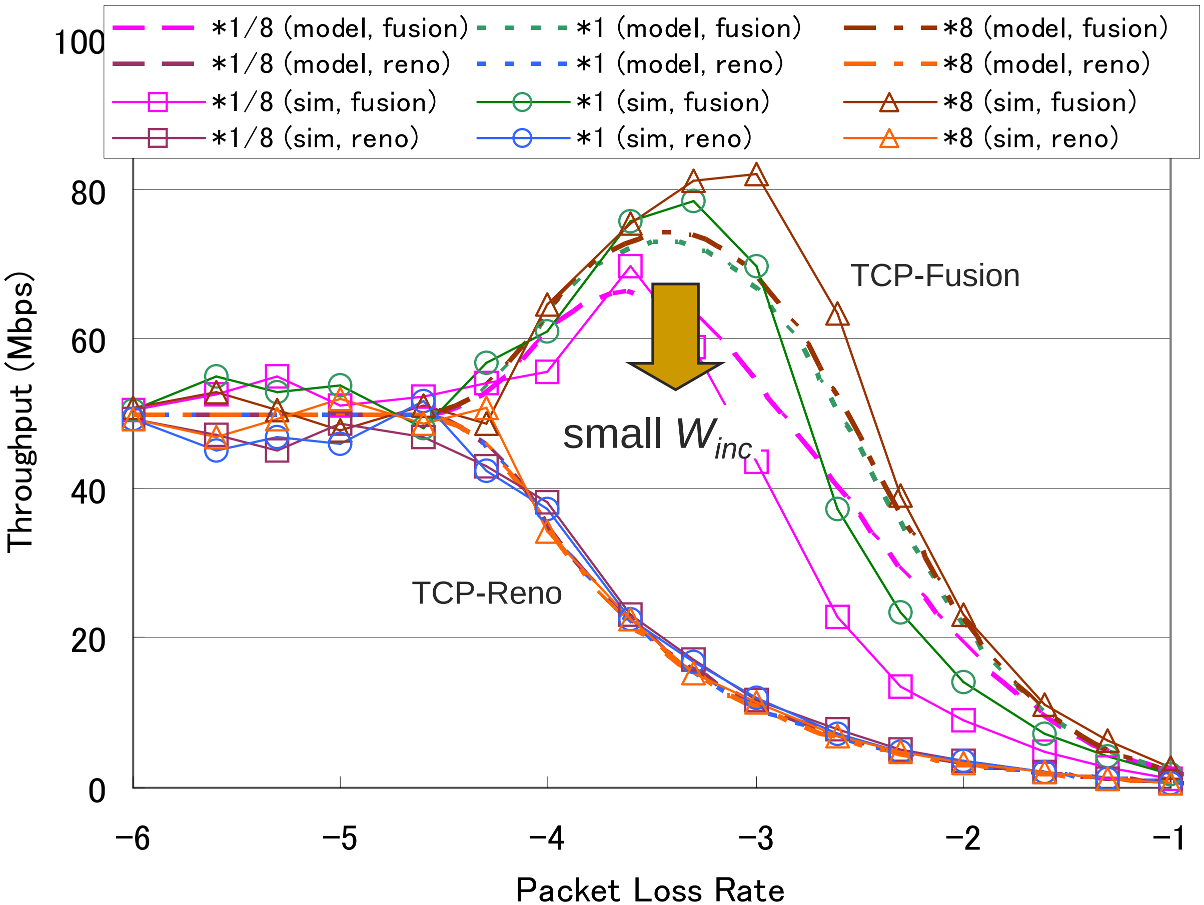
small α ⇒ small change

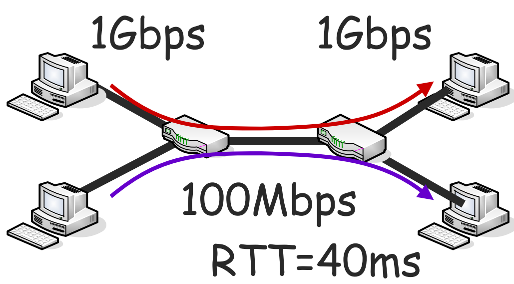
large α ⇒ large change.

TCP-Fusion expels TCP Reno (i.e. unfriendly)

Model Improvement (6)

- Result (2) effect of parameter W_{inc}



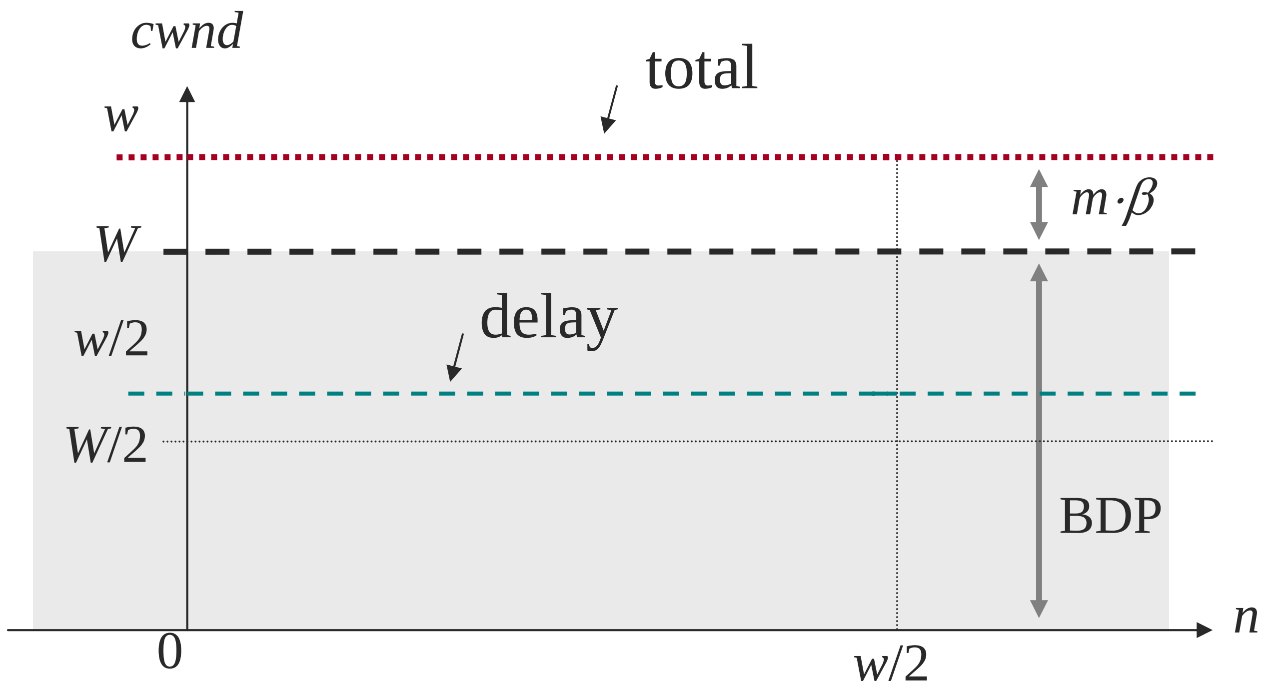


Window increase parameter:

large W_{inc} ⇒ small change
small W_{inc} ⇒ TCP-Fusion decreases but TCP Reno stays the same (inefficient, too conservative cwnd increase)

Delay-based TFRC (1)

- Two flow model for delay-based TCP



Delay-based TFRC (2)

- Throughput derivation

$$R_{delay} = \frac{\frac{w}{2}\left(\frac{W}{M} + \frac{m}{M}\cdot\beta\right)\cdot PS}{\frac{1}{2}w\left(RTT_{\min} + m\cdot D_{\min}\right) + t_{RTO,delay}}$$

$$\begin{cases} R_{delay} = \dfrac{\frac{B}{M}\left(1 + \frac{m}{M}\cdot\frac{D_{\min}}{RTT_{\min}}\right)}{\left(1 + m\cdot\frac{D_{\min}}{RTT_{\min}}\right) + p\cdot\frac{B}{PS}\cdot t_{RTO,loss}} \\ R_{delay} = \dfrac{B/M}{1 + p\cdot B/PS\cdot t_{RTO,loss}} \quad \text{(when } D_{min}\text{=0)} \end{cases}$$

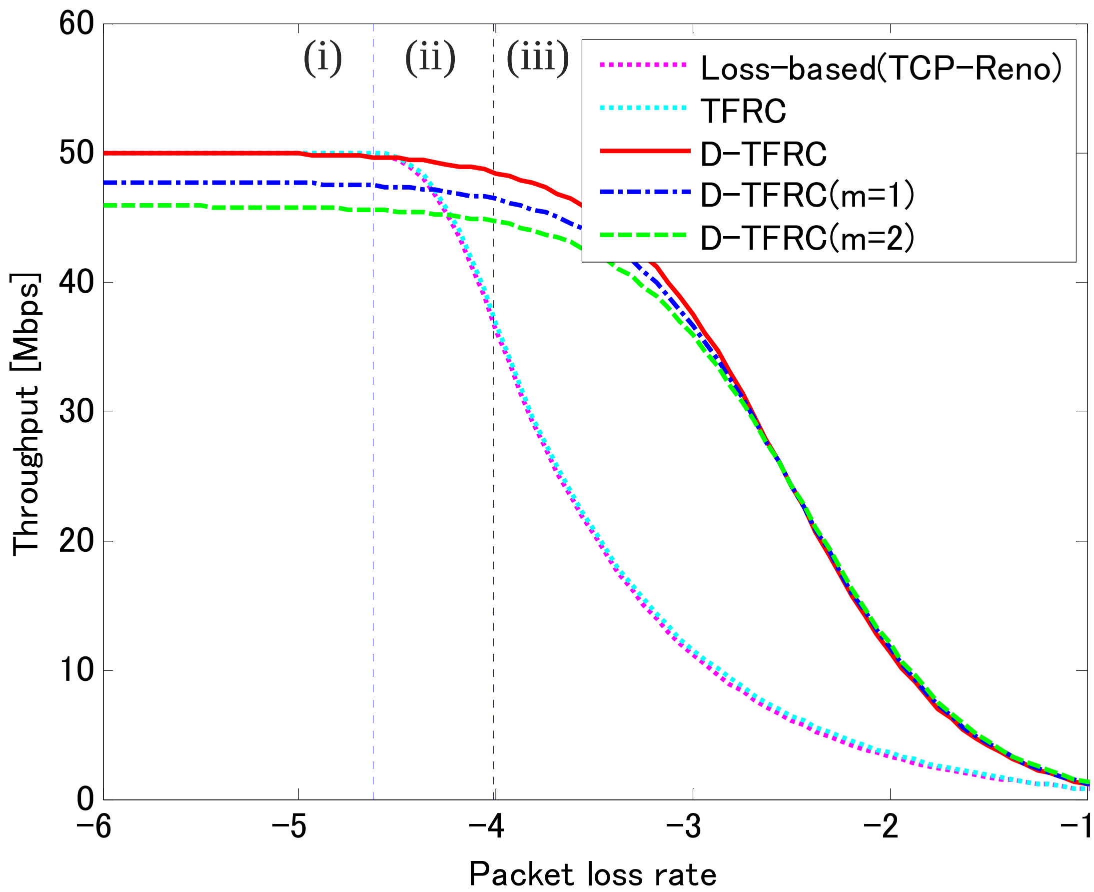
Shares bandwidth B by M flows.
Throughput decreases as p increases.

Delay-based TFRC (3)

- Result

$$R_{loss} = \frac{PS}{RTT\sqrt{\frac{2bp}{3}} + t_{RTO,loss} \cdot 3\sqrt{\frac{3bp}{8}} \cdot p(1+32p^2)}$$

$$R_{delay} = \frac{B/M}{1 + p \cdot B/PS \cdot t_{RTO,loss}}$$



(ongoing)

Conclusions (1)

- Performance analysis of hybrid TCP
  - Ideal model (from Ref[10]):
    - Three models : loss-based, delay-based and hybrid
    - Single flow model and two flow (competing flow) model
    - Evaluations by analysis and simulations
  - Model improvement (specific to TCP-Fusion):
    - Incorporation of control parameters of TCP-Fusion : number of buffered packets α and window increase parameter W_{inc}
    - Evaluations by analysis and simulations
  - Delay-based TFRC:
    - Derivation of TFRC based on delay-based TCP
    - Comparison with loss-based TCP & TFRC (only analysis)

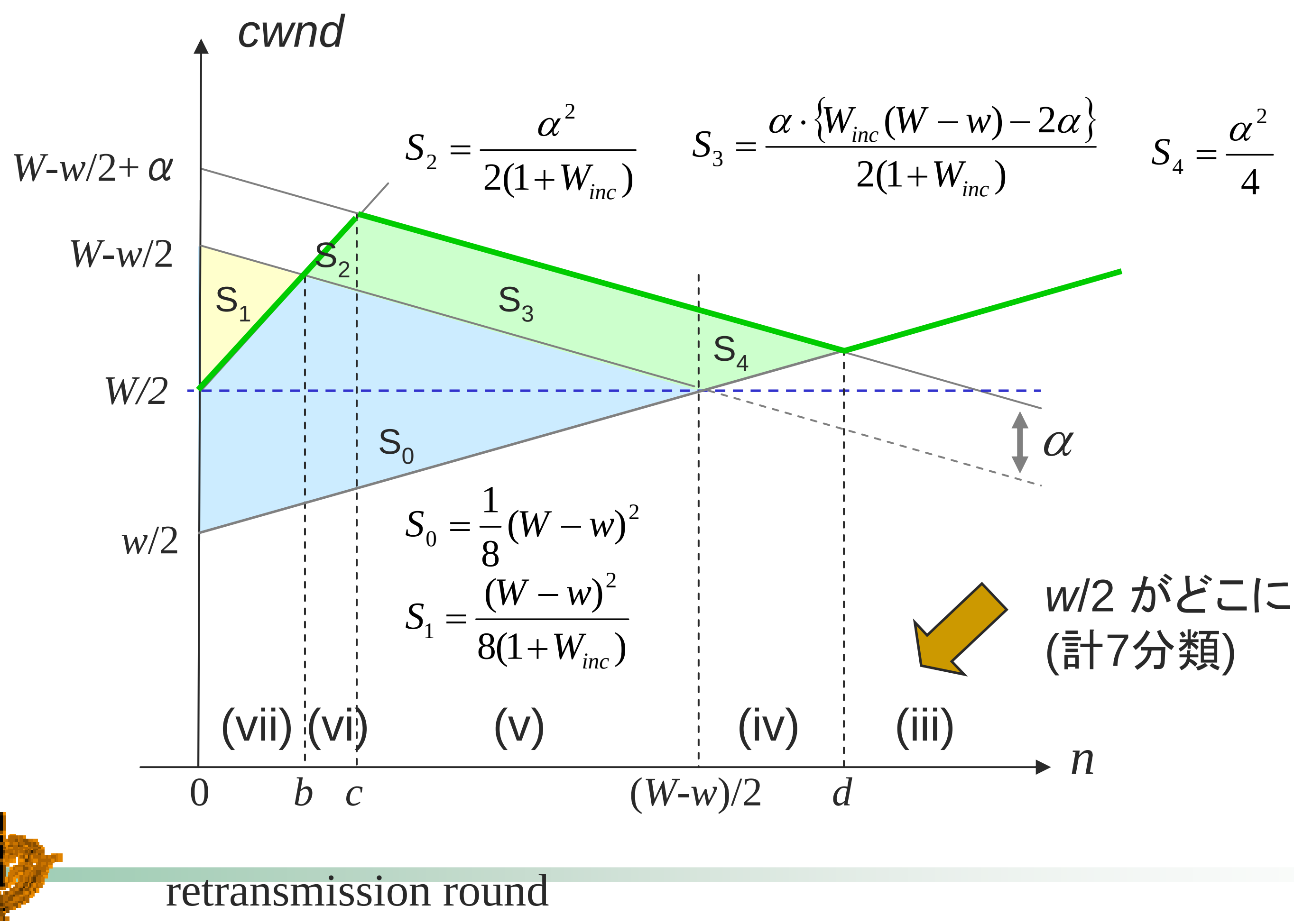
Conclusions (2)

- Future work
  - Extension to many flows
  - Extension to wireless environment (e.g. incorporation of layer 2 retransmission)
  - Incorporation of recent loss-based TCPs (e.g. CUBIC and H-TCP)
  - Evaluations by implementations over actual Internet

  - Model sophistication
  - Extension to RTT fairness
  - Evaluations of reverse traffic and short-lived flow

特性解析: TCP-Fusion 拡張 (5)

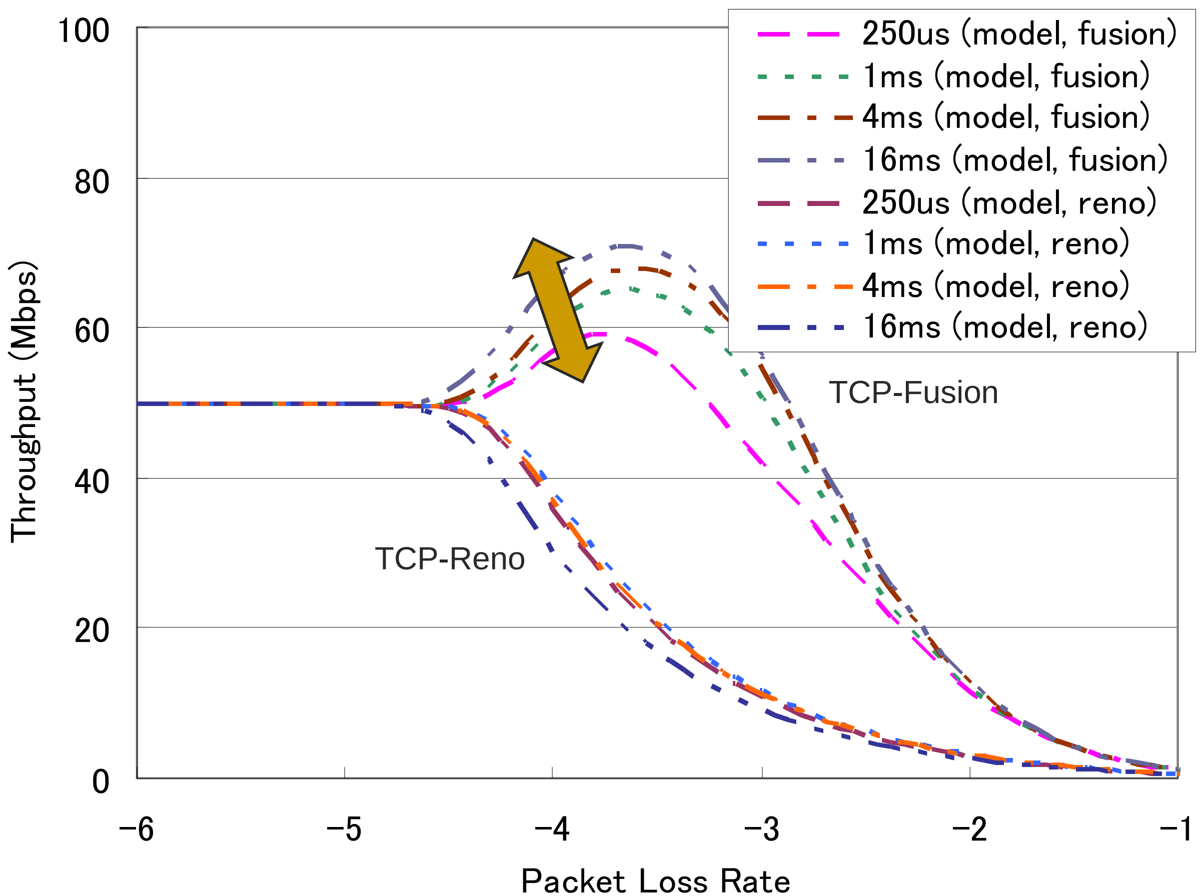
- 競合フローモデル (3) 高廃棄率(続き)

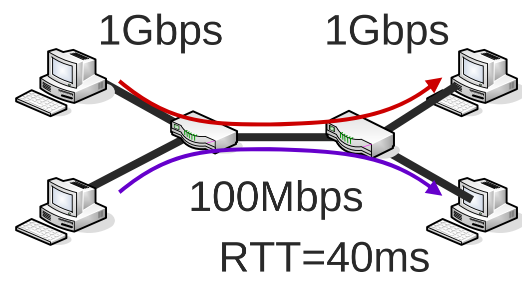


w/2 がどこに来るかで5分類
(計7分類)

特性解析: TCP-Fusion 拡張 (8)

- D_{min}の影響





TCPタイマ解像度

D_{min}小 ⇒ W_{inc} 小 ⇒ Fusion 減少、Reno ほとんど変わらず (inefficient)

D_{min}大 ⇒ α大 ⇒ Fusion 増加、Reno 減少 (unfriendly)

W_{inc} の決め方は要検討