Multimedia Transport Technologies over Wired/Wireless Networks

Jiro Katto

Dept. of Computer Science and Engineering, Waseda University
1. Introduction
Self Introduction

NEC Corp.

Video Compression
(MPEG, H.26X and Wavelet)

Waseda Univ.

Multimedia Signal Processing
(Video and Music)

Multimedia Systems
(MPEG, ITU-T and IETF)

Networking and Applications
(Internet, Wireless and Underwater)
Networks and Multimedia

**Cat-and-mouse game**

- SDTV (720x480)
- HDTV (1920x1080)
- Video 4Kx2K
- Video 8Kx4K
- Multiview
- Wired
- Wireless

Rate:
- 10Mb
- 100Mb
- 1Gb
- 10Gb
- 100Gb
- ~1Gb

Time:
- 11Mb
- 2Mb
- 300Mb
- 54Mb
- 1Gb
- 10Gb
- 100Gb

Katto lab.
Wired Networks

Broadband & CDN

- RTP/UDP & RTSP & TFRC
- → HTTP/TCP streaming
  - Broadband
  - CDN (Akamai, Lime Networks)
  - Firewall (port 80)
  - ...

- One-way (on-demand / live)
- Bi-directional (interactive)

Viewer / Sender

CDN surrogate

HTTP (live) streaming

VoIP, IPTV, Streaming
Wireless Networks

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

AUV (Autonomous Underwater Vehicle)

Oversea experiments

Home aquarium

(TCP and) MAC extensions

Remote control & browsing
## Multimedia Transport

### Protocol Stack

<table>
<thead>
<tr>
<th>Layer</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transport Layer</strong></td>
<td>compression, computer vision, 3D, overlay (CDN, P2P), applications, ...</td>
</tr>
<tr>
<td><strong>Network Layer</strong></td>
<td>RTP/RTCP (synchronization, packet loss detection, congestion control)</td>
</tr>
<tr>
<td><strong>Data Link &amp; Physical Layer</strong></td>
<td>TCP, UDP, TFRC (end-to-end control)</td>
</tr>
<tr>
<td><strong>IP</strong></td>
<td>IP (routing, multicast, mobility)</td>
</tr>
<tr>
<td><strong>Wired</strong></td>
<td>wired (fast and broadband)</td>
</tr>
<tr>
<td><strong>Wireless</strong></td>
<td>wireless (WiFi, multi-hop, underwater)</td>
</tr>
<tr>
<td><strong>MAC</strong></td>
<td>MAC (multiple access, full/half duplex), channel coding, modulation, MIMO, ...</td>
</tr>
</tbody>
</table>

- **Video**
- **Audio**
- **Signaling**
- **Adaptation**
## TCP Variants

<table>
<thead>
<tr>
<th>Wired</th>
<th>Wireless</th>
<th>Satellite</th>
<th>Underwater</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP-Reno/SACK</td>
<td>TCP-Westwood</td>
<td>TCP-Hybla</td>
<td>-</td>
</tr>
<tr>
<td>High-speed TCP</td>
<td>TCP-J</td>
<td>TCP-STAR</td>
<td>-</td>
</tr>
<tr>
<td>Scalable TCP</td>
<td>LDA</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>CUBIC-TCP</td>
<td>TCP-FIT</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>H-TCP</td>
<td>Indirect TCP</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>TCP-Vegas</td>
<td>Snoop TCP</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>FAST-TCP</td>
<td>Freeze TCP</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Compound TCP</td>
<td>Vegas-W</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Adaptive Reno</td>
<td>FeW</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>TCP-Illinois</td>
<td>(cross layer)</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>YeAH-TCP</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>TCP-Fusion</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>TCP</td>
<td>TFRC</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>TFRC/DCCP</td>
<td>TFRC-Wireless</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>RAP</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>TEAR</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>MULTFRC</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>VTP</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Hybrid-TFRC</td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>
# Wired/Wireless Classification

- MAC, hops and transmission media

<table>
<thead>
<tr>
<th></th>
<th>Wired</th>
<th>Wireless LAN</th>
<th>Multihop</th>
<th>Underwater</th>
</tr>
</thead>
<tbody>
<tr>
<td>duplex</td>
<td>Full duplex</td>
<td>Half duplex</td>
<td>Half duplex</td>
<td>Half duplex</td>
</tr>
<tr>
<td>multiple access</td>
<td>Switch</td>
<td>CSMA</td>
<td>CSMA</td>
<td>CSMA (TDMA)</td>
</tr>
<tr>
<td># of hops</td>
<td></td>
<td>Multiple</td>
<td>Multiple</td>
<td></td>
</tr>
<tr>
<td>signal</td>
<td></td>
<td></td>
<td></td>
<td>Acoustic</td>
</tr>
</tbody>
</table>
Can we achieve high-throughput and low-delay simultaneously by TCP for multimedia streaming?

Outline

- Introduction
- Wired Networks
  - TCP Variants
  - Hybrid TCP
- Wireless and Underwater Networks
  - Extensions for WiFi, Multihop and Underwater Sensor Networks
  - DTN extension
- Conclusions
2. Wired (Fast & Broadband) Networks
TCP Variants
TCP-Reno and Vegas

- **TCP-Reno (loss)**

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

  for each ACK

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

- **TCP-Vegas (delay)**

  - $cwnd = \begin{cases} 
  cwnd + 1 & \text{($\Delta \leq \alpha$)} \\
  cwnd & \text{($\alpha < \Delta < \beta$)} \\
  cwnd - 1 & \text{($\Delta \geq \beta$)}
  \end{cases}$

  for every RTT

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

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

/* 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)

W(t) = C * (t - K)^3 + Wmax
K = 3 * Wmax * β / C
Wtcp(t) = Wmax * (1 - β) + 3 * β * t / (2 - β * RTT)

Small Window Decrease

\[ \beta: \text{window decrease rate (e.g. 0.2)} \]

**Algorithm:**

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

cwnd = cwnd * (1- \beta);
```

\[ 1-\beta = 0.8 \]

update of convergence target Wmax and cwnd

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

CUBIC-TCP (3)

- CUBIC’s Cyclic Behavior

repetition of convex & concave shapes

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
  ⇒ **Delay increase**
- **Inter-protocol unfairness** ... expels all the other TCP flows, e.g. “Linux beats Windows!” (vs. Compound TCP)
TCP Westwood (1)

- **Duplicate ACKs**
  
  \[ ssthresh = FSE \times RTT_{min} \]
  
  if \( cwnd > ssthresh \) \( cwnd = ssthresh \)

- **Timeout**

  \[ ssthresh = FSE \times RTT_{min} \]
  
  \( cwnd = 1 \)

- **Multiple versions according to FSE estimation methods**
  
  - BSE, RE, ABSE, ...
TCP Westwood (2)

Fair Share Rate Estimation (TCPW-RE)

Similar to TCP-Vegas:

\[ \Delta = \left( \frac{cwnd}{RTT_{min}} \right) \cdot RTT \approx \text{link capacity} \]

\[ \approx \text{buffered packets} \]

\[ \text{expect rate} \quad \text{actual rate} \]

TCPW-RE:

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

\[ \text{cwnd} \Rightarrow \text{byte counts: } \Sigma d_k \]

\[ \text{RTT} \Rightarrow \text{observation time: } T = \Sigma \Delta t_k \]

Moving average: \( \hat{RE}_k \rightarrow FSE \)

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

\[ T = n \cdot RTT \quad (\text{e.g. } n=4) \]

Hybrid TCP (1)

- 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**

Hybrid TCP (2)

- Competing case between Reno and Hybrid

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)

### Hybrid TCP (3)

#### Classification

<table>
<thead>
<tr>
<th>Hybrids</th>
<th>Window increase 1</th>
<th>Window decrease 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTCP</td>
<td>$0.125 \times \text{cwnd}^{0.75}$</td>
<td>$1/2$</td>
</tr>
<tr>
<td>ARENO</td>
<td>$B/10\text{Mbps}$</td>
<td>$1/2 \sim 1$</td>
</tr>
<tr>
<td>YeAH-TCP</td>
<td>STCP(1.01)</td>
<td>$1/2, \text{RTT}_{\text{min}}/\text{RTT}, 7/8$</td>
</tr>
<tr>
<td>TCP-Fusion</td>
<td>$B \times D_{\text{min}}/(N \times \text{PS})$</td>
<td>$\text{RTT}_{\text{min}}/\text{RTT}$</td>
</tr>
</tbody>
</table>

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

---

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

Performance analysis of Hybrid TCP
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: $cwnd = \max( cwnd_{\text{loss}}, cwnd_{\text{delay}} )$
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)

Performance Analysis (1)

- Single flow case

Performance Analysis (2)

- cwnd & RTT behaviors

(i) $W < w/2$
- large buffer, small PLR
- (always loss-mode)

(ii) $w/2 < W < w$
- small buffer, medium PLR
- (delay & loss adaptive)

(iii) $w < W$
- large PLR, always vacant
- (always delay-mode)

Behavior classification according to PLR:

- $w \sim \text{PLR}$, $W \sim \text{BDP}$

### Performance Analysis (3)

#### Single flow formulation

<table>
<thead>
<tr>
<th>TCP</th>
<th>CA round</th>
<th>(i) $W &lt; w/2$</th>
<th>(ii) $w/2 \leq W &lt; w$</th>
<th>(iii) $w \leq W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>transmitted packets</td>
<td>$\frac{3}{8} w^2$</td>
<td>$\frac{3}{8} w^2$</td>
<td>$\frac{3}{8} w^2$</td>
</tr>
<tr>
<td>elapsed time</td>
<td>$\frac{1}{2} w \cdot RTT_{\min} + \frac{1}{8} (3w^2 - 4wW) \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2} w \cdot RTT_{\min} + \frac{1}{2} (w - W)^2 \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2} w \cdot RTT_{\min}$</td>
<td></td>
</tr>
<tr>
<td>Delay</td>
<td>transmitted packets</td>
<td>$\frac{1}{2} w \cdot W$</td>
<td>$\frac{1}{2} w \cdot W$</td>
<td>$\frac{1}{2} w \cdot W$</td>
</tr>
<tr>
<td>elapsed time</td>
<td>$\frac{1}{2} w \cdot RTT_{\min}$</td>
<td>$\frac{1}{2} w \cdot RTT_{\min}$</td>
<td>$\frac{1}{2} w \cdot RTT_{\min}$</td>
<td></td>
</tr>
<tr>
<td>Hybrid</td>
<td>transmitted packets</td>
<td>$\frac{3}{8} w^2$</td>
<td>$\frac{1}{2} w \cdot W + \frac{1}{2} (w - W)^2$</td>
<td>$\frac{1}{2} w \cdot W$</td>
</tr>
<tr>
<td>elapsed time</td>
<td>$\frac{1}{2} w \cdot RTT_{\min} + \frac{1}{8} (3w^2 - 4wW) \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2} w \cdot RTT_{\min} + \frac{1}{2} (w - W)^2 \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2} w \cdot RTT_{\min}$</td>
<td></td>
</tr>
</tbody>
</table>

**PS:** Packet size, **B:** Link bandwidth

Performance Analysis (4)

- Analysis & simulations

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

Compound & YeAH TCPs degrade due to large window decrease rate

buffer size = BDP (constant)

Packet loss rate : variable

1Gbps 1Gbps
100Mbps
RTT=40ms

PER ~ Throughput (single flow case)
Performance Analysis (5)

- Two competing flows case

![Diagram showing a network with two senders and two receivers, with a bottleneck link and two types of TCP flows: loss-based (TCP-Reno) and loss-based or hybrid TCP flows.]

Performance Analysis (6)

- cwnd behaviors

(i) $W < w$ (low PLR)  
always buffered  
(loss mode)

(ii) $w < W < 2w$ (medium PLR)  
vacant $\rightarrow$ buffered  
(delay $\rightarrow$ loss)

(iii) $2w < W$ (high PLR)  
always vacant  
(delay mode)

large buffer, small PLR  
small buffer, medium PLR  
large PLR, always vacant

$w \sim \text{PLR}, W \sim \text{BDP}$

Performance Analysis (7)

Two flow formulation

<table>
<thead>
<tr>
<th>TCP</th>
<th>CA round</th>
<th>(i) $W &lt; w$</th>
<th>(ii) $w \leq W &lt; 2w$</th>
<th>(iii) $2w \leq W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss</td>
<td>transmitted packets</td>
<td>$\frac{3}{8}w^2$</td>
<td>$\frac{3}{8}w^2$</td>
<td>$\frac{3}{8}w^2$</td>
</tr>
<tr>
<td>Hybrid</td>
<td>transmitted packets</td>
<td>$\frac{3}{8}w^2$</td>
<td>$\frac{3}{8}w^2 + \frac{1}{4}(W-w)^2$</td>
<td>$\frac{1}{2}w \cdot W - \frac{3}{8}w^2$</td>
</tr>
<tr>
<td>(common)</td>
<td>elapsed time</td>
<td>$\frac{1}{2}w \cdot RTT_{\text{min}} + \frac{1}{4}w(3w-2W) \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2}w \cdot RTT_{\text{min}} + \frac{1}{4}(2w-W)^2 \cdot \frac{PS}{B}$</td>
<td>$\frac{1}{2}w \cdot RTT_{\text{min}}$</td>
</tr>
</tbody>
</table>

PS: Packet size, B: Link bandwidth

Performance Analysis (8)

- Analysis & simulations

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

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)
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

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

Wireless LAN (1)

- TCPs and throughputs

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

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

Wireless LAN (2)

**TCPs and delays**

- **TCP-Reno (loss based)**
  - Avg. RTT (ms)
  - Numbers of the upstream connection

- **TCP-Fusion (hybrid)**
  - Avg. RTT (ms)
  - Numbers of the upstream connection

- **TCP-Vegas & Proposal (delay based)**
  - Avg. RTT (ms)
  - Numbers of the upstream connection

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

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
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

- 0.83[ms]
- 0.747[ms]
- 0.83[ms]

When cwnd decreases by 1

- 1.66[ms]
- 0.83[ms]
- 0.83[ms]
TCP Version Differentiation (3)

Wireless Multihop Networks
Wireless Multihop Networks (1)

- **Single Radio Multi-hop Transmission**

  ![Diagram of a multihop network with a sender, intermediate nodes, and a receiver.]

  - 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)

Wireless Multihop Networks (2)

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

\[
cwnd = \begin{cases} 
  \frac{cwnd + 1}{cwnd} & (\Delta < \alpha \& n_{CA} > N_{CA}) \\
  cwnd & (\alpha \leq \Delta \leq \beta \text{ or } \Delta \leq \alpha \& n_{CA} \leq N_{CA}) \\
  \frac{cwnd - 1}{cwnd} & (\Delta > \beta)
\end{cases}
\]

\[n_{CA}: \text{# of consecutive states entering into}\]
\[(\alpha \leq \Delta \leq \beta \text{ or } \Delta \leq \alpha \& n_{CA} \leq N_{CA})\]
\[N_{CA}: \text{threshold (e.g. 100)}\]

much slower than TCP-Vegas

for wireless multihop

Our proposal [IEICE, 2009]

- Exponential decrease of window increase

\[
cwnd = \begin{cases} 
    cwnd + \frac{1}{cwnd \times 2 \times \text{count}} & (\Delta < \alpha \ & \& \ succ > N) \\
    cwnd & (\alpha \leq \Delta < \beta \ \text{or} \ \Delta < \alpha \ & \& \ succ \leq N) \\
    cwnd - \frac{1}{cwnd} & (\Delta \geq \beta)
\end{cases}
\]

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

count: suppression parameter to be incremented

N: succ maximum (e.g. 10)
Wireless Multihop Networks (4)

Throughput [Kbps] vs Number of Hops

- Two flows
- Effective in slow link capacity, but might be heuristic

Number of TCP Flows: 2
Sender: 0
Receiver: R
Distance: 200m

NS-2 simulations

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
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

  \[ T = \frac{x}{v} \times 2 + \frac{S}{C} + \frac{x}{v} \times 2 \]

  - \( x \) [m] speed of sound
  - \( v = 1500 \) [m/s]
  - \( C \) [bit/s] bitrate
  - \( S \) [bit] packet size
  - \( T \) data transmission time

  Diagram:
  - RTS
  - CTS
  - DATA
  - ACK
  - sender
  - receiver
  - vacant
  - packet size
  - speed of sound
Underwater Sensor Networks (2)

Link Utilization (2)

- Channel efficiency: \( \frac{S}{C} / T \)
- Relationship between distance and link utilization

![Diagram showing link utilization and efficiency](image)

Sender

RTS

CTS

DATA

ACK

Vacant capacity

Effect of propagation delay (can be ignored in radio case)
MAC for USN (1)

Selective ARQ

- Delayed NACK & ACK
- used in Seaweb prototype

JSW ARQ

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

MAC for USN (2)

- **500m case**

![Graph showing channel efficiency vs BER for different schemes: Selective+JSW, Selective, JSW, and S&W.]

- **Tx** to **Rx** over 500-1000m

- **Stop & wait (1 DATA / 1 ACK)** doesn’t work well due to slow sound speed.

- **Proposal (Selective ARQ + JSW)** works the best.

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
- $\rho$: $\text{RTT}/\text{RTT}_0$
  - $\text{RTT}$: round trip time
  - $\text{RTT}_0$: reference RTT (0.025[s])
- $SS$: slow start
- $CA$: congestion avoidance
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)
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 approaches
  - 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 assistance
      - differentiation