

無線アドホックネットワーク上のビデオ転送における レート制御比較実験

氏川 裕隆[†] 瀧島 要介[†] 野口 和浩[†] 甲藤 二郎[†]

[†] 早稲田大学基幹理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1 55N-06-09

E-mail: [†] {ujikawa, takisima, kazuhilo, katto}@katto.comm.waseda.ac.jp,

あらまし 有線ネットワークにおいては、リアルタイムビデオストリーミングを行う際に有力な手法の一つとして、受信側におけるロス状況などをフィードバック情報として受け取ることで、使用可能な帯域幅を把握し、適切なレートでエンコード・送信を行うことができる TFRC (TCP Friendly Rate Control [1]) が存在する。しかし、ランダムロスの発生する無線ネットワークにおいては単純な TFRC が効果的に働かないため、本稿では無線アドホックネットワーク上でいくつかのレート制御の比較実験を行い、無線アドホックネットワークにおけるビデオ転送に適したレート制御のための検討を行う。

キーワード ビデオ転送、レート制御、TFRC、VTP、アドホック、無線、マルチホップ、実験

A comparative experiment of rate controls for video transmission over wireless ad-hoc networks

Hiroataka UJIKAWA[†] Yohsuke TAKISHIMA[†] Kazuhiro NOGUCHI[†] and Jiro KATTO[†]

[†] Graduate School of Fundamental Science and Engineering, Waseda University,

3-4-1 Ohkubo, Shinjuku-ku, Tokyo, 169-8555 Japan

E-mail: [†] {ujikawa, takisima, kazuhilo, katto}@katto.comm.waseda.ac.jp

Abstract We performed a comparative experiment of some rate controlling protocols for video transmission over wireless ad-hoc networks. One of the protocols is TFRC (TCP Friendly Rate Control [1]) which is an effective protocol for video streaming on wired networks. But it is also known for its ineffectiveness on wireless networks with random losses. By comparing these protocols, we considered the suitable protocol for video transfers on wireless ad-hoc networks.

Keyword Video Transmission, Rate Control, TFRC, VTP, Ad-hoc, Wireless, Multi-hop, Experiment

1. はじめに

現在、PDA (Personal Digital Assistant) や携帯ゲーム機器を始めとする安価な小型無線デバイスの普及が爆発的に広まり、無線アドホックネットワークの使用が広がっている。また同時に、個人向け PC の高性能化・マルチメディアアプリケーションの普及に伴って、ビデオ配信などのマルチメディアコンテンツに対する需要も高まっている。

しかし、実際にこのような小型無線デバイス同士でアドホックネットワークを構築し、ビデオ転送を行う際には、いくつかの注意すべき点が存在する。

まず、安価な小型無線デバイスの特徴として、メモリ容量が小さく、バッファリングに使用できる容量が小さいことが挙げられる。このため、ビデオ送信者は受信者のバッファ容量を超えないようにレート制御を行わなければならない。

また、無線ネットワークにおける通信路は、有線ネットワークに比べて不安定であり、使用可能帯域が時々刻々と変化していくため、ネットワーク状況に対してリアルタイムに応じるレート制御も必要である。特に、ライブストリーミングビデオ配信では、大きな遅延が許されないため、画質の急激な変化を防ぐようにスムーズなレートを保ちながら、低遅延の通信を達成しなければならない。

本稿では、これらの点を考慮した既存のプロトコルの比較実験を行い、それぞれの特徴的な機構がどのような条件下で有効に働き、どのような条件下では有効に働かないかを調べることによって、アドホックネットワークにおける適切なビデオ転送を行うプロトコルに向けた検討・考察を行う。

2. 関連研究

2.1. TFRC

TFRC は、ライブストリーミングビデオのような、完全な信頼性よりもリアルタイム性を重視するアプリケーションに向けたプロトコルであり、UDP 同様、再送制御を行わない。また、混在する TCP[2]フローとの公平な帯域利用を目的としており、TCP の輻輳制御との親和性を持つ。具体的な制御方法としては、受信者からロス率・受信レートなどをフィードバック情報として受け取り、送信者とその情報と測定した RTT を用い、TFRC の方程式(2.1)に基づいてレート推定を行う。

$$X = \frac{s}{R\sqrt{\frac{2bp}{3}} + t_{RTO}(3\sqrt{\frac{3bp}{8}}p(1+32p^2))} \quad (2.1)$$

ただし

- X : 送信レート [bytes/sec]
- s : パケットサイズ[bytes]
- R : RTT [sec]
- p : ロス率 (ロスイベント/転送パケット)
- b : 1 回の TCP ACK に対するパケット数
- t_{RTO} : TCP 再送タイムアウト値 [sec]

TFRC 自体は有線ネットワーク向けに設計されているプロトコルである。そのため、無線ネットワーク環境のように、ランダムロスが多発する状況においては、性能が落ちることが報告されている[3]。

2.2. LDA[4](Loss Differentiation Algorithm)

無線ネットワーク環境では、様々な要因によりパケットロスが発生する。実際にパケットロスが起きた際に、そのロスが輻輳によって起きたものか、あるいは無線ネットワーク特有のランダムロスによって起きたものかなど、ロスの種類を区別する機構を LDA と呼ぶ。

この機構により、輻輳が要因で起きたロスに応じて送信レートを下げるといった動作をする手法において、ランダムロスによる非合理的なレート減少を防ぐことができる。具体的な例として、TFRC に LDA を組み込む手法などが存在する。

LDA の代表的な手法として、Spike 手法[4]が挙げられる。Spike 手法では、輻輳時に $ROTT$ (Relative One-way Trip Time) が急激に増加(spike)していることを前提としており、測定される $ROTT$ を用いることでロスの区別を行う。具体的には、図 2.2 で示すように、(2.2)式で与えられる閾値を設定し、図中の実線部を輻輳によるロス、点線部をランダムロスとして判断する。

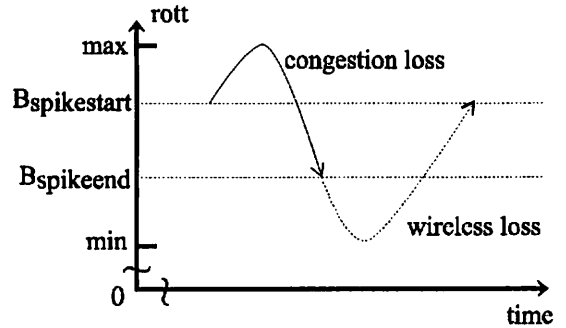


図 2.2 Spike アルゴリズム ([4]より)

$$B_{spikestart} = roth_{min} + \alpha * (roth_{max} - roth_{min}) \quad (2.2)$$

$$B_{spikeend} = roth_{min} + \beta * (roth_{max} - roth_{min})$$

ただし、 $roth_{min}$ および $roth_{max}$ はそれぞれ、付近で観測される $ROTT$ の最小値と最大値であり、 α および β は $0 \leq \beta \leq \alpha \leq 1$ となる任意の値を設定する。

2.3. VTP[5](Video Transport Protocol)

VTP は、ロスの判別を行う LDA、推定レートを用いてレートの下限を保持する AR (Achieved Rate) といった機構を組み込んだ、エンド間でのレート制御方式である。LDA により、ランダムロスに対してロバストであり、AR によって急激なレート変化を防ぐと同時に、帯域の有効活用を行う。また、内部で仮想的に TCP の動作をエミュレートし、仮想 TCP のレート推定に応じたレート制御を行うことで、TCP との親和性を保っている。

VTP の基本的な戦略は図 2.3 に示すように、輻輳によるロスが発生した際に、TCP がレートを急激に落とすのに対し、VTP は算出した推定レート (AR) までしかレートを落とさない。そして、TCP との公平性を保つために、仮想 TCP が VTP に追いつくまでは送信レートを AR のまま維持し、TCP が追いついたら VTP の送信レートを TCP と等しくなるように増加させる。

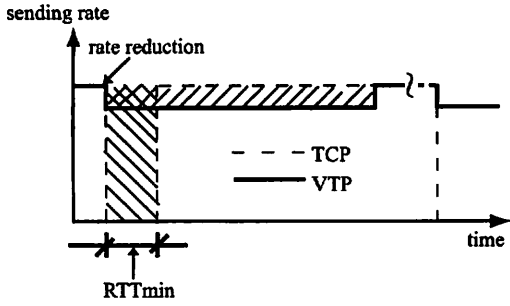


図 2.3.VTP におけるレート制御 ([5]より)

輻輳回避フェーズにおける仮想 TCP のレート推定には、 $ewnd$ (equivalent window)というウィンドウサイズを用いて算出を行う。

まず、現在のレート $R(i)$ と $RTT(i)$ から

$$ewnd(i) = R(i) \times RTT(i) \quad (2.3)$$

を計算し、これを元に、ウィンドウの増加量を 1、 RTT の増加量を $\Delta RTT(i)$ と仮定すると、レート $R(i+1)$ は

$$R(i+1) = \frac{R(i) \times (RTT(i) + 1)}{RTT(i) + \Delta RTT(i)} \quad (2.4)$$

として求める。

また、レートに対して EWMA (Exponential Weighted Moving Average: 指数加重移動平均) をかけることによって、より滑らかなレートの推移を行っている。

2.4. マルチホップ転送について

マルチホップ転送で通信を行う場合、隠れ端末問題・さらされ端末問題などによって、シングルホップでの通信と比較して、ホップ数が増えるにつれてスループットのキャパシティが低下することが知られている。[6]によれば、図 3.2. のような一列に並んだトポロジにおいて、図 2.4. のように減少すると述べている。

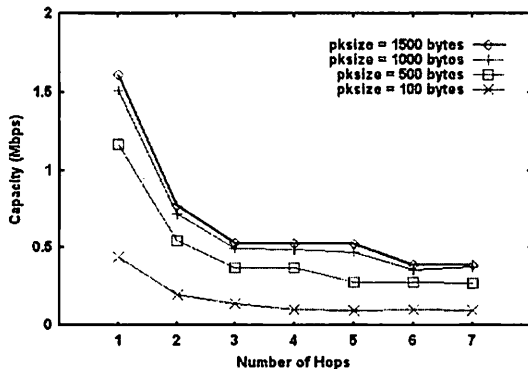


図 2.4. マルチホップにおけるスループット減少 [6]

3. 実験について

3.1. 実験概要

図 3.1 に示すように、フィードバック情報に応じてエンコードレートを適応的に変化させる送信ノードと、受信状況をフィードバック情報として返す受信ノードを用いて、ライブストリーミングビデオの配信を行った。レート制御はビデオ配信アプリケーション内に組み込んでおり、それぞれ TFRC と VTP の動作を行う。

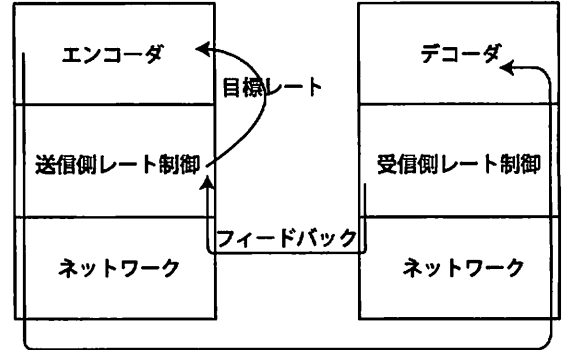


図 3.1 概念図

3.2. 実験環境

実験では図 3.2 に示すように、単純な一列のトポロジを用いて 1 ホップ～3 ホップのマルチホップ通信を行い、ルーティングには AODV [7] を使用した。実装実験で使用した機器や設定に関しては表 3.1 に示す通りになっている。

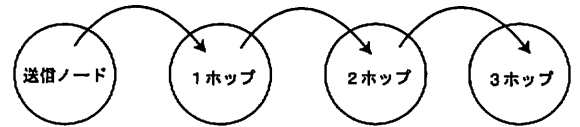


図 3.2 実験トポロジ

表 3.1 実験で使用した機器と設定

受信・送信ノード	Lenovo Thinkpad X60s (Intel Core Duo 1.66GHz)
中継ノード	Thinkpad G50(Celeron M 1.46GHz)
使用 OS	Fedora Core 6.0 kernel-2.6.20-1.2948
入力画像	320*240 / 10fps (Logitech QCam Pro 4000 を使用)
ピクチャ構造	1 ピクチャのみ
初期レート	300kbps
コーデック	H.264/AVC(Intel IPP 5.0[8]を使用)
ルーティング	AODV-UU0.9.3 [9]
無線リンク	802.11b(2Mbps) Ad-hoc デモモード

4. シミュレーション結果

まず、LDA(Spike 手法)の効果を確認するために、NS2[10]によって1ホップから3ホップのアドホックネットワークにおけるTFRCのシミュレーションを行った。図4.1は送信レートの推移を示しており、図4.2にはスループットの推移を示している。シミュレーション条件としては、表4.1に示す通りである。また、Spike手法の(2.2.)式におけるパラメータとして $\alpha=1/2, \beta=1/3$ を用いている。

表4.1.シミュレーション条件

パケットサイズ	1500[bytes]
シミュレーション時間	200[sec]
エラーレート	4%
ルーティング	AODV
帯域幅	2Mbps
MACレイヤ	802.11
キューサイズ	10[packets]

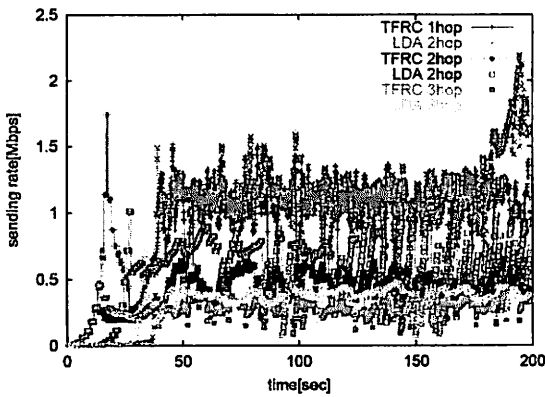


図4.1.TFRCの送信レート推移

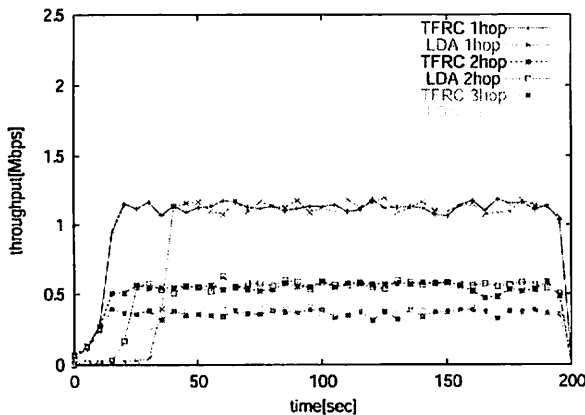


図4.2.TFRCのスループット推移

1ホップから3ホップのそれぞれにおいて、LDAを適用することで一部のロスを見逃すことができるため、TFRC方程式のパラメータであるロス率が下がり、LDAを適用していないTFRCの推定レートに比べて高くなっている。図4.2においては、LDAを適用したTFRCとそうでないTFRCのスループット差が全く出ていないが、これは2.4で述べたように、それぞれのホップ数におけるスループットが通信路のキャパシティに達しているためであり、LDAを適用したTFRCが得た利得は結果として遅延を引き起こしてしまっている。このように、LDAは設定するパラメータの値によって、悪影響を及ぼす場合がある。

次に、図4.3、図4.4にVTPのシミュレーション結果を示す。ARを維持することによって、急激なレートの減少を抑え、輻輳によるロスに対してもロバストであるため、レートを高いまま安定した状態に維持することができる。

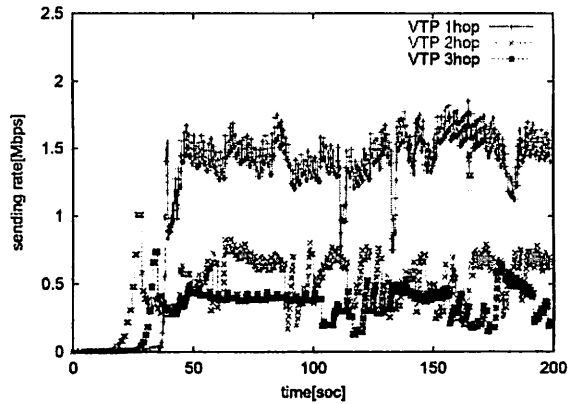


図4.3 VTPの送信レート推移

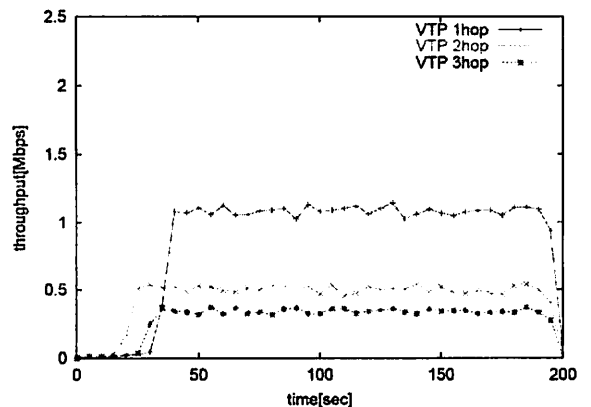


図4.4. VTPのスループット推移

5. 実装実験

5.1. TFRC の実装について

本稿で実装を行ったレート制御は、制御の計算によって得られたレートをエンコードの対象レートに適用するものである。TFRC では、スロースタートフェーズから開始し、バッファオーバーフローが起きるまでレートの増加を行い、ロスが起きた後は TFRC 方程式に基づいて推定レートの算出を行う。

5.2. VTP の実装について

AR には受信側の達成スループットを用いており、AR の平滑化を行う際に EWMA のパラメータ α には 0.9 を使用している。

$$AR_{i+1} = \alpha \times AR_i + (1 - \alpha) AR_{sample}$$

ただし、 AR_i は現在の平滑化された受信レートを、

AR_{sample} は現在の受信レートの瞬時値をあらわす。

5.3. 実験結果

まず、TFRC の実験結果を図 5.1. に示す。

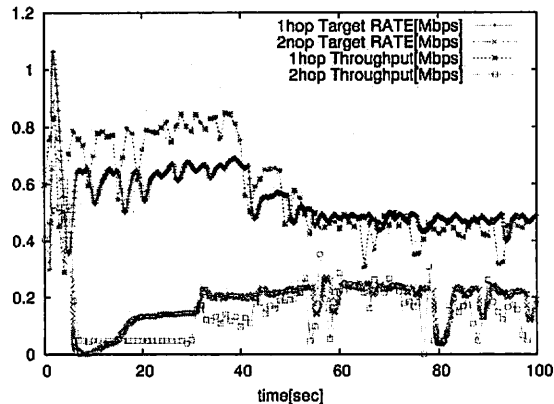


図 5.1. TFRC の実装実験結果

図 4.2. で示したシミュレーション結果とは異なり、スループットのキャパシティを完全に使い切るような活発な帯域の利用は見られなかった。しかし、無線環境で著しくスループットが低下するという [3] の結果と比較すると、比較的高いスループットを示しており、TFRC に無線環境用の対策を取り入れることで、無線環境においても有効なレート制御として用いることも考えられる。

次に、VTP の実験結果として、1 ホップでの送信レート (目標エンコードレート) の推移を図 5.2 に示す。VTP の実装実験では、図 5.2 のように輻輳回避フェーズにおいて、輻輳によるロスが起きるとレートを AR に設定し、仮想的な TCP が追いつくまでそのレートを維持する。そして、TCP が追いつくと (2.4) 式によって、ロスが起きるまで仮想 TCP と同様にレートを上げていく、という処理を繰り返している。

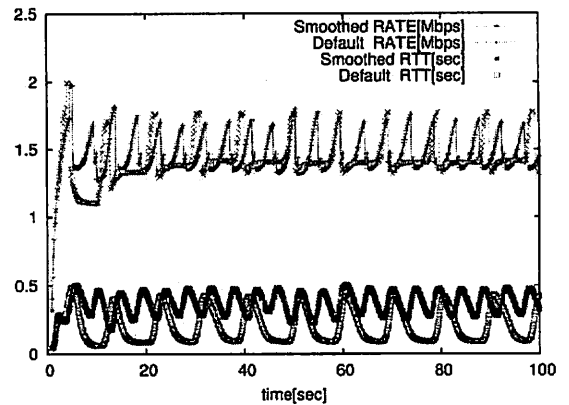


図 5.2. VTP と平滑化を行った VTP (1hop)

これを単純化すると図 5.2 の実線のようになるが、この VTP で算出しているレートは、トランスポート層における実際の送信レートとしてではなく、エンコードの目標レートとして設定しているため、画質の変化を避けるためにはレートの変化を緩めることが望ましい。そこで、図 5.2 の破線 (平滑化レート) のように平滑化を行うことを考える。

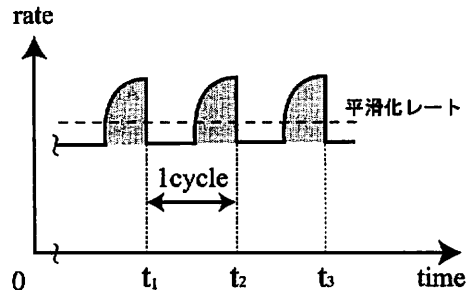


図 5.3. VTP レート模式図.

図中の灰色に塗りつぶされた部分は、AR と送信レートの差から求められる。また、AR の維持開始から輻輳回避におけるまでを 1 サイクルとし、現在のサイクルと次のサイクルの長さの差を、無視できる小さな値と仮定すると、次のサイクルの平滑化レートが直ちに算出できる。

しかし、この平滑化レートを保持しているだけでは、帯域が空いているか判断できないため、帯域が空いている場合にその有効活用を行うことができない。そのため、頻度を少なくしても、(2.4)式のようにレートを上げて、余剰帯域があるかどうかを調べる必要がある。

ここで、帯域に余裕が無いときには RTT が増加し、逆に帯域に余裕が出てきたときに RTT が減少するということを基に、EWMA をかけて平滑化している現サイクルの RTT が、1 サイクル前の最小 RTT よりも確実に減少している場合に、次のサイクルで(2.4)式によりレートを増加させ、帯域を調べるようにした。

この平滑化を行った VTP の結果は 5.1. に示している通りであり、1 ホップの環境下で平滑化を行うことができた。しかし、2 ホップで実験を行ったところ、平滑化を行っていない VTP でも終始平坦なレートを保ち続けたため、平滑化の効果は得られなかった。

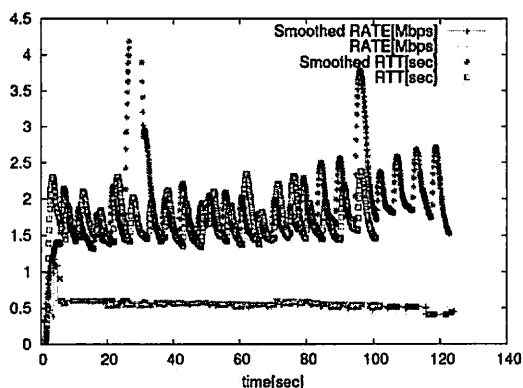


図 5.3.VTP と平滑化を行った VTP(2hop)

6. 結論

本稿では、TFRC におけるレート推定、ロスに対してロバストな LDA、また VTP における AR の維持といった手法について、それぞれの効果の確認を行った。

TFRC のレート推定方程式では、シミュレーション上においては帯域のキャパシティを使い切るほど帯域を有効に使えており、実装実験においても効果的に動作していた。

LDA を使用した場合には、シミュレーションでは大幅な改善は見られなかったものの、ランダムロスによるレート減少を防ぐ効果と同時にパラメータによる悪影響についても確認することができた。

また、VTP においては、シミュレーションおよび実装実験の両方で高いレートを安定して維持しており、帯域の有効活用と急激なレート変化防止の効果を確認した。更に、動画伝送に向けて、よりレートを平滑化する試みを行った。

7. 今後の予定

本稿では、TFRC や VTP などの代表的なレート制御について、シミュレーションと実装によって追試を行い、その効果を確認した。

今後は、これらの手法の優位性を取り入れながら、クロストラヒックの考慮やビデオの特徴を利用したレート制御の提案、更にホップ数を増やした環境での実験を行い、アドホックネットワークにおけるビデオ伝送に適したレート制御の実現に向けて検討を行ってきたい。

文 献

- [1] M. Handley, S. Floyd, J. Padhye and J. Widmer, "TCP Friendly Rate Control(TFRC)", RFC3448, Jan.2003.
- [2] M.Allman, V. Paxson and W. Stevens, "TCP Congestion Control", RFC2581, April 1999.
- [3] Guang Yang, Ling-Jyh Chen, Tony Sun, Mario Gerla and M. Y. Sandidi, "Real-time Streaming over Wireless Links: A Comparative Study", Proceeding, IEEE Symposium on Computers and Communications 2005, pp.249-254, June 2005.
- [4] Song Cen, Pamela C. Cosman and Geoffrey M. Voelker, "End-to-End Differentiation of Congestion and Wireless Losses", IEEE ACM Transactions on Networking, vol.11, no.5, pp.703-717, October 2003.
- [5] Guang Yang, Tony Sun, Mario Gerla, M.Y.Sandidi and Ling-Jyh Chen, "Smooth and efficient real-time video transport in the presence of wireless errors", ACM Transactions on Multimedia Computing, Communications and Applications, vol.2, issue 2, May 2006.
- [6] Ling-Jyh Chen, Tony Sun, Guang Yang, M. Y. Sandidi and Mario Gerla, "Ad Hoc Probe: Path Capacity Probing in Wireless Ad Hoc Networks", WICON'05, pp.156-163, 2005
- [7] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC3561, July 2003.
- [8] Intel, Integrated Performance Primitives 5.0 for Linux, <http://www.intel.com/cd/software/products/ijkk/jpn/perflib/index.htm>
- [9] AODV-UU0.9.3, <http://core.it.uu.se/core/index.php/AODV-UU/>
- [10] The Network Simulator ns-2 v2.28, <http://www.isi.edu/nsnam/ns/>