Active network measurements: new techniques, analyses and experiments

Ph.D. Candidate: Antonio A. de A. Rocha

Advisors: Rosa Maria Meri Leão
Edmundo A. de Souza e Silva

Federal University of Rio de Janeiro
LAND - Laboratory for modeling, analysis and development of networks and computer systems
COPPE/Prog. of computer science and system
Outline

- Introduction
- Objectives of this work
- Technique to estimate wireless bit rate
  - Motivation
  - Proposal
  - Validation and experiments
  - Conclusions and on-going work
- Technique to estimate OWD
  - Motivation
  - Proposal
  - Extension
  - Validation and experiments
  - Conclusions and on-going work
- Summary of contributions
- Future works
• Measurement is an important area in network;
  - Often used to...
    + understand network characteristics;
    + help the modeling and analysis process;
    + improve the performance of applications;
Introduction

- Measurement is an important area in network;
  - Often used to...
    - understand network characteristics;
    - help the modeling and analysis process;
    - improve the performance of applications;

- Active measurement subarea;
  - Probes are generated;
• Measurement is an important area in network;
  - Often used to...
    + understand network characteristics;
    + help the modeling and analysis process;
    + improve the performance of applications;

- Active measurement subarea;
  + Probes are generated;
  + Metrics are estimated;
Introduction

- Important network metrics are related to:
  - Loss;
    - Loss rate, Consecutive loss rate, ...
  - Delay;
    - One-way delay, Round-trip time, ...
  - Transmission capacity;
    - Bottleneck link capacity, Available bandwidth, ...
• Important network metrics are related to:
  - Loss;
    + Loss rate, Consecutive loss rate, ...
  - Delay;
    + One-way delay, Round-trip time, ...
  - Transmission capacity;
    + Bottleneck link capacity, Available bandwidth, ...

• Although some measures are relatively easy to compute, ...
  - Round trip packet loss and delay;
Introduction

• Important network metrics are related to:
  
  - Loss;
    + Loss rate, Consecutive loss rate, ...
  
  - Delay;
    + One-way delay, Round-trip time, ...
  
  - Transmission capacity;
    + Bottleneck link capacity, Available bandwidth, ...

• Although some measures are relatively easy to compute, ...
  
  - Round trip packet loss and delay;

• ... other network metrics are much harder to obtain.
  
  - one-way measurements, especially with no access to the remote machine;
  - characteristics of the remote access network machine, for example, the transmission rate, when it is a WLAN;
Objectives of this work

- Development of new techniques to estimate important network characteristics
Objectives of this work

- Development of new techniques to estimate important network characteristics

- Two techniques have been proposed
Objectives of this work

- Development of new techniques to estimate important network characteristics

- Two techniques have been proposed
  - End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
Objectives of this work

- Development of new techniques to estimate important network characteristics

- Two techniques have been proposed
  - End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
  - Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;
Objectives of this work

- Development of new techniques to estimate important network characteristics

- Two techniques have been proposed
  - End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
  - Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;

Next, I will describe the motivations, functionalities and results obtained for both techniques.
End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
Main questions

- The main questions of this work are:
  - Is it possible to identify the type of the access network of a remote host?
Main questions

The main questions of this work are:

- Is it possible to identify the type of the access network of a remote host?
- If it is a WLAN, what transmission bit rate is it operating?
For several applications, to identify characteristics of the access network of a remote host can be very useful:

- video streaming;
- overlay networks;
- peer-to-peer;
- topology discovery;
- network management;
Recent researches try to identify the type of connection of a remote user [Cheng et al. 2001], [Wei et al. 2005] and [Baiamonte et al. 2007]

- classify the type of last hop connection:
  - wired: Ethernet, DSL, Cable Modem, dial-up;
  - wireless: 802.11a/b/g;
Motivation

- WLANs based on IEEE 802.11 have become one of the most popular access networks.
  - Significant reduction in the costs of the equipment;
  - High transmission rates (802.11a/g);
- Wireless stations can adapt their transmission rates dynamically, depending on channel conditions
  - rates from 1 up to 54 Mbps;
Motivation

- WLANs based on IEEE 802.11 have become one of the most popular access networks.
  - Significant reduction in the costs of the equipment;
  - High transmission rates (802.11a/g);

- Wireless stations can adapt their transmission rates dynamically, depending on channel conditions
  - rates from 1 up to 54 Mbps;

If the type of access is wireless, we believe that to estimate the actual bit rate of a WLAN device is useful
First main contribution

- End-to-end technique to infer the transmission rate of the access network when it is a WLAN

- Our method uses an extension of the packet pair technique, that I describe next;

- Algorithms such as [Wei et al. 2005] to verify if it is a WLAN, before estimating the transmission rate;
Packet Pair

- Well-known technique
- Extensively used to estimate metrics related to path capacity
  - i.e., Bottleneck link capacity and available bandwidth
• Well-known technique

• Extensively used to estimate metrics related to path capacity
  - i.e., Bottleneck link capacity and available bandwidth

• Consists of two packets of the SAME SIZE sent from the source to the destination
• Well-known technique

• Extensively used to estimate metrics related to path capacity
  – i.e., Bottleneck link capacity and available bandwidth

• Consists of two packets of the SAME SIZE sent from the source to the destination
  – one immediately after the other (initial gap close to zero)
Packet Pair

- Well-known technique
- Extensively used to estimate metrics related to path capacity
  - i.e., Bottleneck link capacity and available bandwidth
- Consists of two packets of the SAME SIZE sent from the source to the destination
  - one immediately after the other (initial gap close to zero)
Packet Pair

- Well-known technique
- Extensively used to estimate metrics related to path capacity
  - i.e., Bottleneck link capacity and available bandwidth
- Consists of two packets of the SAME SIZE sent from the source to the destination
  - one immediately after the other (initial gap close to zero)
  - packets are expanded by the bottleneck link
• Well-known technique

• Extensively used to estimate metrics related to path capacity
  - i.e., Bottleneck link capacity and available bandwidth

• Consists of two packets of the SAME SIZE sent from the source to the destination
  - one immediately after the other (initial gap close to zero)
  - packets are expanded by the bottleneck link
  - this gap is preserved until they arrive at the destination
Packet Pair

- Well-known technique
- Extensively used to estimate metrics related to path capacity
  - i.e., Bottleneck link capacity and available bandwidth
- Consists of two packets of the SAME SIZE sent from the source to the destination
  - one immediately after the other (initial gap close to zero)
  - packets are expanded by the bottleneck link
  - this gap is preserved until they arrive at the destination
Packet Pair

- Well-known technique

- Extensively used to estimate metrics related to path capacity
  - i.e., Bottleneck link capacity and available bandwidth

- Consists of two packets of the SAME SIZE sent from the source to the destination
  - one immediately after the other (initial gap close to zero)
  - packets are expanded by the bottleneck link
  - this gap is preserved until they arrive at the destination

\[
\text{Bottleneck Capacity} = \frac{\text{Size (bits)}}{\text{Gap (seg.)}}
\]
Packet Pair over WLAN

- Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN
  - WLAN may not be the bottleneck

Because of the high transmissions rates of the 802.11 standards, it is possible that the WLAN is not the path bottleneck.
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN:

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN:

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN:

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN:

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN:

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN:

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol
Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN:

- WLAN may not be the bottleneck
- Overhead of the 802.11 protocol

Our technique take into account both issues
Packet Pair over WLAN

- Two main issues have to be addressed for estimating the bit rate of the last hop in a WLAN
  - WLAN may not be the bottleneck
  - Overhead of the 802.11 protocol

Tools like CapProbe and PathRate can not be employed to our problem
Using any existing technique, we verify if the access network link of the remote host is wireless;
Using any existing technique, we verify if the access network link of the remote host is wireless; then, the technique that I describe next is used to estimate the bit transmission rate;
Proposed Technique

- Several packet pairs with different sizes are generated from source to destination
  - size of first packet (P1) of all pairs is 1500 bytes
  - second packet (P2) in a pair has smaller sizes: ...
Several packet pairs with different sizes are generated from source to destination:
- size of first packet (P1) of all pairs is 1500 bytes
- second packet (P2) in a pair has smaller sizes: ... 600 bytes
• Several packet pairs with different sizes are generated from source to destination
  - size of first packet (P1) of all pairs is 1500 bytes
  - second packet (P2) in a pair has smaller sizes: ...
    - 600 bytes
    - 800 bytes
Several packet pairs with different sizes are generated from source to destination
- size of first packet (P1) of all pairs is 1500 bytes
- second packet (P2) in a pair has smaller sizes: ...
  600 bytes
  800 bytes
  1000 bytes
Several packet pairs with different sizes are generated from source to destination:
- size of first packet (P1) of all pairs is 1500 bytes
- second packet (P2) in a pair has smaller sizes: 600 bytes, 800 bytes, 1000 bytes, 1200 bytes

The rationality is that P1 larger "slows down" P2 increasing the likelihood that they arrive back to back at the receiver.
Several packet pairs with different sizes are generated from source to destination:

- Size of first packet (P1) of all pairs is 1500 bytes.
- Second packet (P2) in a pair has smaller sizes: 600 bytes, 800 bytes, 1000 bytes, 1200 bytes.

In this way we address the issue 1, in which the bottleneck may NOT be wireless link.
For each received pair, we compute the dispersion at destination

- \( D = P_2 \) arrival time - \( P_1 \) arrival time
For each received pair, we compute the dispersion at destination

- \( D = P_2 \text{ arrival time} - P_1 \text{ arrival time} \)
• Dispersion computed at the receiver
  - results of an experiment where 40 packet pairs were generated
• Dispersion computed at the receiver
  - results of an experiment where 40 packet pairs were generated

![Graph showing dispersion vs. size of the second packet (P2)]

For each size used for P2, we select the smallest dispersion value in the samples.
Proposed Technique

- Dispersion computed at the receiver
  - results of an experiment where 40 packet pairs were generated

![Graph showing dispersion vs. size of the second packet (P2)]

The goal is to reduce the effects of backoff time and cross traffic.
In an idealized scenario without cross traffic and backoff time equal to zero, the dispersion is a function of the following times:

- SIFS and DIFS (which are constants)

\[ D_{j,C_n} = t_{SIFS} + t_{DIFS} + \frac{L_{ACK}}{C_n} + \frac{L_{2,j}}{C_n} \]
In an idealized scenario, without cross traffic and backoff time equal to zero, the dispersion is a function of the following times:

- SIFS and DIFS (which are constants)
- transmission of the ACK and of the second packet in a pair.

\[ D_{j,C_n} = t_{SIFS} + t_{DIFS} + \frac{L_{ACK}}{C_n} + \frac{L_{2,j}}{C_n} \]

Where "C_n" is one of the available bit rates for the IEEE 802.11 standard (1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48 or 54 Mbps)
In an idealized scenario, without cross traffic and backoff time equal to zero, the dispersion is a function of the following times:

- SIFS and DIFS (which are constants)
- transmission of the ACK and of the second packet in a pair.

\[ D_{j,Cn} = t_{SIFS} + t_{DIFS} + \frac{L_{ACK}}{C_n} + \frac{L_{2,j}}{C_n} \]

Where "Cn" is one of the available bit rates for the IEEE 802.11 standard (1, 2, 5.5, 6, 9, 11, 12, 18, 24, 36, 48 or 54 Mbps)

This equation considering 802.11 overhead let us to address the second issue.
Proposed Technique

- Dispersion ($D_{j,c}$) for second packet ($P_2$) values and for each 802.11 capacity.
Proposed Technique

- Dispersion \((D_{j,C})\) for second packet \((P2)\) values and for each 802.11 capacity.

- The final step consists of calculating the MSE between selected samples \((\delta)\) and \(D_{j,C}\) for each value of \(C_n\).
Proposed Technique

- In summary, our technique has five-steps, as described below:

  1. Using any existing technique identify the connection type of the last hop. If it is a WLAN, then continue Steps 2-5;

  2. Generate a sequence of packet pairs and collect it at the receiver;

  3. At the receiver compute the dispersion $\delta_{n,j}^k$ for all packet pairs;

  4. For each size of the second pair, select the smallest dispersion value;

  5. Estimate the $C_{WLAN}$ using the minimum MSE between the select samples and $D_{j,C_n}$ for each value of $C_n;$
Proposed Technique

- Wireless station may change its transmission rate depending on the channel conditions

- The method can also be employed for estimating the transmission rate dynamically
  - We generate continuously the packet pairs
  - For the transmission rate estimation, we use a window of $W$ packet pairs and apply the algorithm
  - For the next estimation, the window slides and the new pair replace and old pair.
Validation

- Validation through experimentation and simulation

- Experimentation for two scenarios
  - WLAN at UFRJ
  - Internet, including paths in which the bottleneck is not the wireless link

- Simulation
  - NS-2
  - Models considering multi-rate adaptation for wireless devices
• Scenario I: Experimentation

- Several runs
  - AP configured to operate at different transmission rates
  - ARC off
Validation

- Scenario I: Experimentation
  - Rate 11 Mbps

![Graph showing dispersion vs size of the second packet](image)

<table>
<thead>
<tr>
<th>Rate</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>48194925</td>
</tr>
<tr>
<td>2</td>
<td>10231681</td>
</tr>
<tr>
<td>5.5</td>
<td>824976</td>
</tr>
<tr>
<td>6</td>
<td>639383</td>
</tr>
<tr>
<td>9</td>
<td>286762</td>
</tr>
<tr>
<td>11</td>
<td>244734</td>
</tr>
<tr>
<td>12</td>
<td>494648</td>
</tr>
<tr>
<td>18</td>
<td>283931</td>
</tr>
<tr>
<td>24</td>
<td>337879</td>
</tr>
<tr>
<td>36</td>
<td>413688</td>
</tr>
<tr>
<td>48</td>
<td>459787</td>
</tr>
<tr>
<td>54</td>
<td>476372</td>
</tr>
</tbody>
</table>

Selected samples
Validation

- Scenario I: Experimentation
  - Rate 54 Mbps

![Graph showing dispersion vs. size of the second packet for different rates: 1, 2, 5.5, and 11 Mbps. The graph includes a table with MSE values for rates 12, 18, 24, 36, 48, and 54 Mbps. The selected sample rate is 54 Mbps, with an MSE value of 16654.]
Scenario II: Experimentation

- Source machine in our Lab and destination is in a home in Rio de Janeiro
- The AP is connected to the Internet via 512 kbps cable modem
- The AP was configured to operate at 2Mbps
- ARC off
Validation

- Scenario II: Experimentation
  - Source machine in our Lab and destination is in a home in Rio de Janeiro
  - The AP is connected to the Internet via 512 kbps cable modem
  - The AP was configured to operate at 2Mbps
  - ARC off

Note that the WLAN is NOT the bottleneck link
Scenario II: Experimentation

- Rate 2 Mbps

![Graph showing Dispersion (µs) vs Size of the second packet for different rates and MSE values.]

<table>
<thead>
<tr>
<th>Rate</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10980400</td>
</tr>
<tr>
<td>2</td>
<td>=&gt;208884&lt;=</td>
</tr>
<tr>
<td>5.5</td>
<td>8106042</td>
</tr>
<tr>
<td>6</td>
<td>8895886</td>
</tr>
<tr>
<td>9</td>
<td>11564614</td>
</tr>
<tr>
<td>11</td>
<td>12469772</td>
</tr>
<tr>
<td>12</td>
<td>13030199</td>
</tr>
<tr>
<td>18</td>
<td>14582867</td>
</tr>
<tr>
<td>24</td>
<td>15392299</td>
</tr>
<tr>
<td>36</td>
<td>16223556</td>
</tr>
<tr>
<td>48</td>
<td>16647337</td>
</tr>
<tr>
<td>54</td>
<td>16789594</td>
</tr>
</tbody>
</table>
Validation

- Model used in NS-2 simulations
Validation

- Model used in NS-2 simulations
Validation

- Model used in NS-2 simulations

source and receiver of the cross traffic
• Model used in NS-2 simulations

The path has three wired links (100Mb, 100Mb and 10Mb) and the last hop is an 802.11
Validation

- Model used in NS-2 simulations

802.11 link configured to operate with multi-rate adaptation
• Simulation result (Estimated Rate and Real Rate)
- Simulation result (Relative Error)

More than 75% of estimates have relative error smaller than 10%
I present a simple and accurate measurement technique to infer the bit rate of an IEEE802.11 device

- Based on the packet pair technique adapted to take into account parameters of the IEEE802.11
- Furthermore, the technique does not require the WLAN to be the bottleneck link in the path

Results from measurements and simulation show the proposal accuracy

- both in scenarios where the WLAN devices can adapt their rates as well as WLANs with fixed transmission rates.

New experiments with other scenarios and different network loads will be performed.
Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;
Motivation

- Not difficult to estimate round-trip measures
  - Several tools;

- Internet forward and backward paths may be asymmetric
  - different capacities in forward and backward paths;
  - different sequence of hops;
  - asymmetric traffic in router queues;
Motivation

- Not difficult to estimate round-trip measures
  - Several tools;

- Internet forward and backward paths may be asymmetric
  - different capacities in forward and backward paths;
  - different sequence of hops;
  - asymmetric traffic in router queues;

Important to perform one-way measurements!
Motivation (Cont.)

- To measure in one-way may be a difficult task;

- Estimating one-way delay, for example, is complex if:
Motivation (Cont.)

- To measure in one-way may be a difficult task;

- Estimating one-way delay, for example, is complex if:
  - the analyst has no access to the remote machine;
  - non-cooperative technique required
Motivation (Cont.)

- To measure in one-way may be a difficult task;

- Estimating one-way delay, for example, is complex if:
  - the analyst has no access to the remote machine; ✗
  + non-cooperative technique required
  - both clocks (source and destiny) are not synchronized;
Motivation (Cont.)

- To measure in one-way may be a difficult task;

![Diagram of Internet with Source, Probes, and Destiny]

- both clocks (source and destiny) are not synchronized;

† Skew and Offset problems
Second main contribution

- Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;
Second main contribution

- Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;

  - Non-cooperative because we do not need no access to the remote machine;
Second main contribution

- Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;
  - Non-cooperative because we do not need access to the remote machine;
  - Clocks do not need to be synchronized;
IPID

- The proposed technique is based on IPID;

- IPID is a identification field of the IP packet;
  - Used by fragmentation process;

- Some O.S´s implement the IPID as a global counter of packet sent;
  - i.e., Windows;

- Other O.S.’s implement a pseudo-random counter or set IPID to zero;
  - i.e., some versions of Linux and FreeBSD;
Several works exploit IPID for estimating network characteristics;

Some measurements in one-way:

- Loss rate [Mahajan et al. 2003];
- Fraction of packets received out of order [Bellardo and Savage 2002];
- One-way delay difference [Chen et al. 2005];

Our proposal is based on [Chen et al. 2005] that I describe next:
In [Chen et al. 2005], hosts A and B send probes to a target machine (D):

- Estimate the difference between the OWD’s ($d_{AD}$ and $d_{BD}$);

- Assume A and B have synchronized clocks;
• Technique of [Chen et al. 2005] for estimating OWD difference:
  - Hosts (A and B) send probes to a target machine (D);
IPID usages in measurement (Cont.)

- Technique of [Chen et al. 2005] for estimating OWD difference:
  - Hosts (A and B) send probes to a target machine (D);
  - D send probes back to A and B, including IPID of D;

![Diagram showing the process of sending probes and measuring OWD difference](image)
• Technique of [Chen et al. 2005] for estimating OWD difference:
  - Hosts (A and B) send probes to a target machine (D);
  - D send probes back to A and B, including IPID of D;
  - IPID values indicate if probes from A and B reached D “nearly the same instant”;

![Diagram showing the technique of [Chen et al. 2005] for estimating OWD difference.](image)
• Technique of [Chen et al. 2005] for estimating OWD difference:
  - Hosts (A and B) send probes to a target machine (D);
  - D send probes back to A and B, including IPID of D;
  - IPID values indicate if probes from A and B reached D "nearly the same instant";

\[
d_{AD} - d_{BD} \approx \tau_B - \tau_A + n_B \delta_B - n_A \delta_A.
\]
Proposed technique

- The proposal is based on the idea of [Chen et al. 2005];

- It estimates the average of the one-way delay and NOT the OWD difference, as in [Chen et al. 2005];

- The proposal also computes the variance of the OWD;

- Does NOT require clock synchronized;
Problem definition (Cont.):

Using the IPIID technique we can construct the following system of equations:

\[
\begin{align*}
    d_{AD} + d_{DA} &= RTT_{ADA} \\
    d_{BD} + d_{DB} &= RTT_{BDB} \\
    d_{AD} - d_{BD} &= \Psi_{AD-BD} \\
    d_{DA} - d_{DB} &= \Psi_{DA-DB}
\end{align*}
\]
Proposed technique (Cont.)

- Problem definition (Cont.):

  - Using the IPIID technique we can construct the following system of equations:

  \[
  \begin{align*}
  d_{AD} + d_{DA} &= RTT_{ADA} \\
  d_{BD} + d_{DB} &= RTT_{BDB} \\
  d_{AD} - d_{BD} &= \Psi_{AD-BD} \\
  d_{DA} - d_{DB} &= \Psi_{DA-DB}
  \end{align*}
  \]

  - Equations are linearly dependent
  - Extra information to obtain a unique solution
Problem definition (Cont.):

- Using the IPID technique we can construct the following system of equations:

\[
\begin{align*}
    d_{AD} + d_{DA} &= RTT_{ADA} \\
    d_{BD} + d_{DB} &= RTT_{BDB} \\
    d_{AD} - d_{BD} &= \Psi_{AD-BD} \\
    d_{DA} - d_{DB} &= \Psi_{DA-DB}
\end{align*}
\]

- Equations are linearly dependent

- Extra information to obtain a unique solution

I address this issue in what follows...
The OWD is equal to the sum of four terms:
- Transmission, propagation, processing and queue times;
- Assuming, processing time negligible:
  \[ d_{AD} = T_{prop}^AD + T_{tx}^AD + T_{queue}^AD \]

Sending "n" probes of same size:
- Same transmission and propagation times;
- If \( T_{prop} \) and \( T_{tx} \) are known, previews system can be rewritten;
The OWD is equal to the sum of four terms:

- Transmission, propagation, processing and queue times;

- Assuming, processing time negligible:

\[ d_{AD} = T_{prop} + T_{tx} + T_{queue} \]

Sending "n" probes of same size:

- Same transmission and propagation times;

- If \( T_{prop} \) and \( T_{tx} \) are known, previews system can be rewritten;

How to estimate \( T_{prop} \) and \( T_{tx} \) for all directions?
• Using different forward and backward packet sizes, we can estimate $T_{prop}^m$ and $T_{tx}^m$;

• Assuming the smallest probe delays have zero queueing time, common assumption for many works;

• If I get to send probes with size X and receive probes with size Y, we can estimate $T_{prop}^m$ and $T_{tx}^m$ with the following system:

\[
\begin{align*}
T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m,ADA}^{50-50} \\
10T_{AD}^{tx} + 10T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m,ADA}^{500-500} \\
10T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m,ADA}^{500-50}
\end{align*}
\]

(i.e. $X=500$ bytes and $Y=50$ bytes)
Proposed technique (Cont.)

- Using different forward and backward packet sizes, we can estimate $T_{prop}$ and $T_{tx}$;

- Assuming the smallest probe delays have zero queueing time, common assumption for many works;

- If I get to send probes with size $X$ and receive probes with size $Y$, we can estimate $T_{prop}$ and $T_{tx}$ with the following system:

\[
\begin{align*}
T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m, ADA}^{50-50} \\
10T_{AD}^{tx} + 10T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m, ADA}^{500-500} \\
10T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m, ADA}^{500-50} 
\end{align*}
\]

(i.e. $X=500$ bytes and $Y=50$ bytes)

Sending and receiving packets with same size is easy using ICMP Echo Request and Echo Reply messages
Using different forward and backward packet sizes, we can estimate $T_{prop}$ and $T_{tx}$;

Assuming the smallest probe delays have zero queueing time, common assumption for many works;

If I get to send probes with size $X$ and receive probes with size $Y$, we can estimate $T_{prop}$ and $T_{tx}$ with the following system:

$$
\begin{align*}
T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m,ADA}^{50-50} \\
10T_{AD}^{tx} + 10T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m,ADA}^{500-500} \\
10T_{AD}^{tx} + T_{DA}^{tx} + 2T_{AD}^{prop} &= RTT_{m,ADA}^{500-50} 
\end{align*}
$$

(i.e. $X=500$ bytes and $Y=50$ bytes)

The problem now is:
how can I send a probe of certain size, expecting to receive a response of different size?
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a “fake” ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a "fake" ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a "fake" ECHO REPLY;

- Only the second (smaller) packet is replyed;
Proposed technique (Cont.)

- The solution was implemented in an ICMP probe generator:
  - For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a "fake" ECHO REPLY;
  - Only the second (smaller) packet is replyed;

![Diagram of network communication](image)
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a "fake" ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a "fake" ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a “fake” ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a “fake” ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a “fake” ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a “fake” ECHO REPLY;

- Only the second (smaller) packet is replyed;
The solution was implemented in an ICMP probe generator:

- For sending packet of 500 bytes and receiving packet of 50 bytes, we use a packet pair technique where the first packet is a "fake" ECHO REPLY;

- Only the second (smaller) packet is replyed;

\[
\text{\textit{RTT}}_{m,\text{ADA}}^{500-50} = \text{T}_{tx,\text{AD}}^{500} + \text{T}_{\text{prop},\text{AD}} + \text{T}_{tx,\text{AD}}^{50} + \text{T}_{\text{prop},\text{AD}}
\]
Thus, as I know $T_{AD}^{\text{prop}}$ and $T_{AD}^{\text{tx}}$,
if $T_{AD}^{\text{queue}} = 0$, it is possible to estimate OWD $d_{BD}$ and $d_{DB}$;
Technique extension

- Described technique assumes A and B clocks are synchronized;

- We can adapt the technique for dealing with Skew and Offset problems;
Several algorithm to remove Skew and Offset [Paxon et al. 98, Moon et al. 99, Zhang et al. 02, Tsuru et al. 02];

- Probes are generated to/from involved machines

![Diagram showing synchronization issues between two hosts over the internet.](image-url)
Several algorithms to remove Skew and Offset
[Paxon et al. 98, Moon et al. 99, Zhang et al. 02, Tsuru et al. 02];
- Probes are generated to/from involved machines

We do NOT want to send extra probes between source machines (A and B)
It is possible to adapt the algorithms to estimate the Skew, considering the sum of $d_{BD}$ and $d_{DA}$.
It is possible to adapt the algorithms to estimate the Skew, considering the sum of $d_{BD}$ and $d_{DA}$.

- Lower bound given by transmission and propagation time (B->D + D->A) summed to Skew and Offset;
It is possible to adapt the algorithms to remove the Skew, considering the sum of $d_{BD}$ and $d_{DA}$. Skew may be removed from all samples, using an algorithm to estimate the "Lower Bound" [Zhang et al. 02].
After removing the Skew, we can estimate and remove the Offset

- From the minimum value of $RTT_{BDB}$ and $d_{BD-DA}$ among all $i$ samples, it is possible to estimate the Offset

$$O_{AB} = (T_{DB}^{tx} + T_{DB}^{prop}) - (T_{DA}^{tx} + T_{DA}^{prop}) - (RTT_{m,BDB} - d_{m,BD-DA}^{s})$$
After removing the Skew, we can estimate and remove the Offset.

- From the minimum value of $\text{RTT}_{BDB} e d_{BD-DA}$ among all $i$ samples, it is possible to estimate the Offset.

\[
O_{AB} = (T_{txDB} + T_{propDB}) - (T_{txDA} + T_{propDA}) - (\text{RTT}_{m,BDB} - d_{m,BD-DA})
\]
Simulations and experiments were performed to validate the technique;

- Simulation with Tangram-II (www.land.ufrj.br)
- Experiments over Internet among UFRJ, Unifacs e UMass
- Experiments with several points using PlanetLAB
Several runs, varying link utilization;
Several runs, varying link utilization;
Some runs, non-synchronized clocks
- Links utilization 30% to 50%, Scenario 1

<table>
<thead>
<tr>
<th>Path</th>
<th>Average</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Estim./Actual/Error</td>
<td>Estim./Actual/Error</td>
</tr>
<tr>
<td>A-D</td>
<td>1366 / 1338 / 0.020</td>
<td>129 / 137 / 0.058</td>
</tr>
<tr>
<td>D-A</td>
<td>942 / 929 / 0.013</td>
<td>173 / 175 / 0.011</td>
</tr>
<tr>
<td>B-D</td>
<td>765 / 755 / 0.013</td>
<td>124 / 143 / 0.132</td>
</tr>
<tr>
<td>D-B</td>
<td>1707 / 1712 / 0.002</td>
<td>155 / 150 / 0.033</td>
</tr>
</tbody>
</table>

Relative Error
- Links utilization 65% a 80%, Scenario 2 (nom-synchronized clocks)

<table>
<thead>
<tr>
<th>Path</th>
<th>Average Estim./Actual/Error</th>
<th>Variance Estim./Actual/Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-D</td>
<td>6372 / 6533 / 0.025</td>
<td>19309807 / 19289802 / 0.001</td>
</tr>
<tr>
<td>D-A</td>
<td>2057 / 2242 / 0.082</td>
<td>5233686 / 7395982 / 0.290</td>
</tr>
<tr>
<td>B-D</td>
<td>2086 / 1974 / 0.057</td>
<td>6572138 / 5384199 / 0.220</td>
</tr>
<tr>
<td>D-B</td>
<td>6142 / 6552 / 0.062</td>
<td>28476154 / 30912220 / 0.078</td>
</tr>
</tbody>
</table>
• Links utilization 65% to 80%, Scenario 2 (nom-synchronized clocks)

---

**Average Path DA**

![Graph showing average path delay over simulation time](image)

- **Estimated** path delay
- **Actual** path delay

Simulation time (sec.)

Simulation data points:
- 0 10000 20000 30000 40000 50000 60000 70000 80000 90000 100000
• Links utilization 65% a 80%, Scenario 2 (nom-synchronized clocks)
Several configurations:
• Several configurations:
  
  • From UMass and UNifacs to a target machine at UFRJ
Several configurations:

- From UMass and UNifacs to a target machine at UFRJ
- From UMass and UFRJ to a target machine at UNifacs
Experiments UFRJ/Unifacs/UMass

- Several configurations:
  - From UMass and UNifacs to a target machine at UFRJ
  - From UMass and UFRJ to a target machine at UNifacs
  - From UNifacs and UFRJ to a target machine at UMass
Several configurations:

- From UMass and UNifacs to a target machine at UFRJ
- From UMass and UFRJ to a target machine at UNifacs
- From UNifacs and UFRJ to a target machine at UMass
- Just for validation purpose clock machines are synchronized by GPS;
Experiments UFRJ/Unifacs/UMass

- Several configurations:
  - From UMass and UNifacs to a target machine at UFRJ
  - From UMass and UFRJ to a target machine at UNifacs
  - From UNifacs and UFRJ to a target machine at UMass
  - Just for validation purpose clock machines are synchronized by GPS;
  - Also for validation, target machine compute passively sending and receiving probes;
Several configurations:

- From UMass and UNifacs to a target machine at UFRJ
- From UMass and UFRJ to a target machine at UNifacs
- From UNifacs and UFRJ to a target machine at UMass

Just for validation purpose clock machines are synchronized by GPS;
Also for validation, target machine compute passively sending and receiving probes;

30 min. and aprox. 440 kbps;
Several configurations:

- From UMass and UFRJ to a target machine at Unifacs

<table>
<thead>
<tr>
<th>Path</th>
<th>Relative Error</th>
<th>Average / variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFRJ-Unifacs</td>
<td>0.009 / 0.152</td>
<td></td>
</tr>
<tr>
<td>Unifacs-UFRJ</td>
<td>0.009 / 0.038</td>
<td></td>
</tr>
<tr>
<td>Umass-Unifacs</td>
<td>0.001 / 0.015</td>
<td></td>
</tr>
<tr>
<td>Unifacs-Umass</td>
<td>0.001 / 0.099</td>
<td></td>
</tr>
</tbody>
</table>
Several configurations:

- From UFRJ and Unifacs to a target machine at UMass

<table>
<thead>
<tr>
<th>Path</th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average / variance</td>
</tr>
<tr>
<td>UFRJ-UMass</td>
<td>0.004 / 0.626</td>
</tr>
<tr>
<td>UMass-UFRJ</td>
<td>0.005 / 0.022</td>
</tr>
<tr>
<td>Unifacs-UMass</td>
<td>0.016 / 0.710</td>
</tr>
<tr>
<td>UMass-Unifacs</td>
<td>0.015 / 0.087</td>
</tr>
</tbody>
</table>
• Different nodes on PlanetLAB;

• Several experiment configurations;
• First, two sources and one destiny:

• First 5 minutes of each hour (24 hours);
First, two sources and one destiny:

![Graph showing the comparison between estimated and actual average (d_DB) values over experiment hours. The graph has a y-axis labeled 'Average (d_DB) - µs' ranging from 0 to 140000, and an x-axis labeled 'Experiment Hour' ranging from 0 to 25. The graph includes two lines: one for estimated values marked 'Estimated' with crosses, and one for actual values marked 'Actual' with red lines.](image-url)
First, two sources and one destiny:
• Second, two sources and one destiny:

• First 60 seconds of each hour (10 hours);

• Estimating confidence interval;
- Confidence interval of Kaist->Seattle
Third, several sources and same destiny:

- 60 seconds of measurements;
Third, several sources and same destiny:

**Average**

<table>
<thead>
<tr>
<th>Path</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Estim./Actual/Error</td>
</tr>
<tr>
<td>Texas-UMass</td>
<td>26091 / 25852 / 0.009</td>
</tr>
<tr>
<td>Standford-UMass</td>
<td>35097 / 35562 / 0.013</td>
</tr>
<tr>
<td>U.K.-UMass</td>
<td>43777 / 43948 / 0.003</td>
</tr>
<tr>
<td>Berkeley-UMass</td>
<td>40680 / 40602 / 0.001</td>
</tr>
<tr>
<td>Hong Kong-UMass</td>
<td>19321457 / 20427774 / 0.057</td>
</tr>
<tr>
<td>Israel-UMass</td>
<td>85975 / 85607 / 0.004</td>
</tr>
<tr>
<td>Kaist-UMass</td>
<td>107122 / 106971 / 0.001</td>
</tr>
<tr>
<td>Unifacs-UMass</td>
<td>86716 / 86425 / 0.003</td>
</tr>
<tr>
<td>France-UMass</td>
<td>48513 / 48338 / 0.003</td>
</tr>
</tbody>
</table>
Third, several sources and same destiny:

Variance

<table>
<thead>
<tr>
<th>Path</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Estim./Actual/Error</td>
</tr>
<tr>
<td>Texas-UMass</td>
<td>150976 / 227899 / 0.509</td>
</tr>
<tr>
<td>Standford-UMass</td>
<td>256008 / 261035 / 0.019</td>
</tr>
<tr>
<td>U.K.-UMass</td>
<td>140461 / 199699 / 0.421</td>
</tr>
<tr>
<td>Berkeley-UMass</td>
<td>19321457 / 20427774 / 0.057</td>
</tr>
<tr>
<td>Hong Kong-UMass</td>
<td>178582 / 263283 / 0.474</td>
</tr>
<tr>
<td>Israel-UMass</td>
<td>570297 / 653080 / 0.145</td>
</tr>
<tr>
<td>Kaist-UMass</td>
<td>219852 / 292970 / 0.332</td>
</tr>
<tr>
<td>Unifacs-UMass</td>
<td>982904 / 227814 / 0.768</td>
</tr>
<tr>
<td>France-UMass</td>
<td>207378 / 260358 / 0.255</td>
</tr>
</tbody>
</table>
Partial conclusions

- Presented a non-cooperative active measurement technique for estimating the average and variance of the one-way delay;

- Non-cooperative -> It is not necessary access to remote machine;

- Clocks does NOT need to be synchronized in the source machines;

- Experiments and simulations were performed for validating;
  - Only simulation were performed with non-synchronized clocks
On-going work

- Now I am performing experiments with non-synchronized clocks
  - It is NOT a trivial experimentation, because we need extra machines operating passively

- New simulation runs have been performed to investigate the estimates for several network loads
Two techniques have been proposed and validated;
Two techniques have been proposed and validated;

- End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
Two techniques have been proposed and validated;

- End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;

Several experiments and simulations have been performed;
Two techniques have been proposed and validated;

- End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;

Several experiments and simulations have been performed;

Papers:
- WPerformance 2006
- IEEE/ICC 2007
• Two techniques have been proposed and validated;
  - End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
  - Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;
Two techniques have been proposed and validated;

- End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
- Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;

Several experiments and simulations have been performed;

Papers:
- SBRC 2006 (Best Paper)
- IFIP-Networking 2007
Two techniques have been proposed and validated;
  - End-to-end technique to infer the transmission rate of the access network, when it is a WLAN;
  - Non-cooperative active measurement technique for estimating the average and variance of the one-way delay;

Several experiments and simulations have been performed;

Papers:
- SBRC 2006 (Best Paper)
- IFIP-Networking 2007

Journal paper with new results;
Future works

- Extend the non-cooperative active measurement technique for estimating other network metrics;
  - i.e., Available bandwidth;

- Develop a new probe generation approach to avoid depending on global IPID at the target machines;

- Extend the end-to-end technique to infer not only the transmission rate, but also to detect if the access type is a WLAN;

- Investigate the possibility of creating new techniques to estimate other important network characteristics;

- New studies during Ph.D. Internship;
Thanks!
Questions and suggestions?

Ph.D. Candidate: Antonio A. de A. Rocha

Advisors: Rosa Maria Meri Leão
Edmundo A. de Souza e Silva

Federal University of Rio de Janeiro

LAND - Laboratory for modeling, analysis and development of networks and computer systems

COPPE/Prog. of computer science and system