



Università degli Studi di Modena e Reggio Emilia

NS-3 e QoE Monitor

Ing. Daniela Saladino, PhD

3 Giugno 2013

Sommario

1. Introduzione alla simulazione numerica
2. Simulazione di reti di telecomunicazioni
3. Simulatore di rete: NS-3
4. Modulo di valutazione della QoE: QoE Monitor

Sommario

1. Introduzione alla simulazione numerica
2. Simulazione di reti di telecomunicazioni
3. Simulatore di rete: NS-3
4. Modulo di valutazione della QoE: QoE Monitor

Problemi Ingegneristici

- Una particolare classe di problemi in ingegneria è quella dei problemi di decisione, in cui, cioè, è necessario compiere una scelta di una o più tecnologie e/o soluzioni fra varie alternative disponibile (tipicamente molte)
- In linea di principio, i problemi di decisione possono essere risolti “provando” o “sperimentando” le varie alternative e decidendo per la migliore, rispetto ad una data metrica

Metodologie di Risoluzione

- Le principali metodologie utilizzabili per la soluzione di un problema di decisione sono:
 - 1) Utilizzo di un **modello matematico**, costituito da relazioni matematiche che descrivono il funzionamento del sistema oggetto dello studio
 - 2) Utilizzo di un **modello in scala** del sistema considerato, costituito da un insieme limitato di funzionalità, rispetto al sistema originale
 - 3) Utilizzo di un **programma di simulazione**, il quale replica il funzionamento del sistema oggetto dello studio
 - 4) Utilizzo di un **sistema reale**, ovvero di un'istanza completa del sistema studiato

Sommario

1. Introduzione alla simulazione numerica
2. Simulazione di reti di telecomunicazioni
3. Simulatore di rete: NS-3
4. Modulo di valutazione della QoE: QoE Monitor

La Simulazione in Sintesi

- La simulazione numerica consiste nei seguenti punti:
 - **Modellazione del sistema**: estrazione dal sistema considerato dei dettagli di interesse per lo studio → costruzione di un modello del sistema
 - Deve scendere ad un livello di dettaglio adeguato per l'obiettivo dello studio
 - **Realizzazione di un programma di simulazione** capace di riprodurre le caratteristiche e il funzionamento del modello estratto
 - Il programma sarà tanto più complesso quanti più dettagli verranno presi in considerazione nel modello
 - **Esecuzione del programma di simulazione** per mezzo di un computer ed estrazione dei risultati di interesse

Pianificazione della Simulazione

- La simulazione numerica, come qualsiasi altro esperimento, deve essere pianificata accuratamente. È necessario:
 - Individuare in modo chiaro gli **obiettivi** dello studio
 - Individuare le **metriche** di interesse (output)
 - Individuare le **variabili** dello studio (cioè le grandezze la cui variazione caratterizza il contesto di valutazione, es: numero di nodi, tipologia dei servizi, numero dei servizi, tecnologie trasmissive, etc)
 - “**Randomizzare**” e **replicare** l’esperimento
 - Per evitare errori sistematici e per fornire una misura della precisione dell’esperimento (errore sperimentale)
 - Il numero di repliche dipenderà dalla precisione che si desidera per l’esperimento

Esempi di problematiche - Network Engineering 1

- **Pianificazione di una rete wireless**
 - Si consideri un edificio costituito da più piani, all'interno del quale dovranno essere realizzati un numero predefinito di uffici. Si vuole fornire ad ognuno di essi la connettività dati attraverso l'installazione di un certo numero di access point IEEE 802.11
 - Qual'è il range di copertura radio fornito dal singolo access point?
 - Quanti utenti è possibile servire con un singolo access point?
 - Quanti access point devono essere installati in ogni piano?

Esempi di problematiche - Network Engineering 2

- **Dimensionamento della rete**
 - Si consideri una rete data, costituita da una topologia prefissata e supportante un determinato insieme di servizi. La rete deve essere estesa (es: implementazione di un link di connessione fra due sedi distaccate)
 - Quali tecnologie di interconnessione possono essere utilizzate?
 - Qual'è la richiesta di risorsa che deve essere fatta al provider? (es: service-level agreement?)

Esempi di problematiche - Traffic Engineering 1

- **Valutazione di prestazioni in scenari complessi**
 - Si consideri una rete data, con una predefinita topologia e un determinato insieme di servizi che devono essere supportati
 - La topologia considerata è in grado di supportare i servizi richiesti?
 - Entro che margini la rete considerata è in grado di supportare i servizi dati?

Esempi di problematiche - Traffic Engineering 2

- **Valutazione dei bottleneck**
 - Data una rete predefinita (es: rete aziendale) e un insieme di servizi di rete attivi, si vogliono analizzare i colli di bottiglia della rete stessa al fine di individuare i segmenti critici
 - Tali segmenti potranno essere quelli più interessati ad opere di upgrade della rete.

Simulatori di rete

- Esistono numerosi simulatori di rete, sia di tipo commerciale, sia di natura freeware e/o open-source. Alcuni dei più famosi sono:
 - OPNET (commerciale)
 - OMNeT++ (versione non commerciale) e OMNEST (versione commerciale)
 - Network Simulator 2 (open source, di largo utilizzo)
 - **Network Simulator 3** (open source, sviluppo iniziato nel 2006)
 - NetSim (commerciale)

Sommario

1. Introduzione alla simulazione numerica
2. Simulazione di reti di telecomunicazioni
3. Simulatore di rete: **NS-3**
4. Modulo di valutazione della QoE: QoE Monitor

NS-3

- “Network Simulator ver. 3” (NS-3) è un simulatore di rete ad eventi discreti, sviluppato a partire dal 2006 come progetto open-source (www.nsnam.org)
- NS-3 è un simulatore nuovo, non compatibile all’indietro con NS-2
- Obiettivo principale: essere il più possibile realistico
- Offre il supporto alla virtualizzazione e alla realizzazione di testbed (emulazione)
 - Il simulatore può essere interfacciato con hardware reale, sia all’interno di un testbed, sia all’interno di un sistema reale
- Semplifica le operazioni di tracing dei risultati

Ambiente di Sviluppo

- Le simulazioni in NS-3 vengono programmate principalmente in linguaggio C++
 - Sono attualmente disponibili bindings per python
- Non è previsto alcun IDE predefinito
 - Ogni script di simulazione può essere scritto tramite un qualsiasi editor di testo
 - È possibile integrare il sistema di compilazione di NS-3 con Eclipse (Eclipse + plugin CDT), per sfruttarne le funzionalità

Come iniziare?

- Il percorso “suggerito” per iniziare a conoscere ed utilizzare NS-3 è:
 - 1) Iniziare con il tutorial
(<http://www.nsnam.org/docs/release/3.14/tutorial/singlehtml/index.html>)
 - 2) Sfruttare il manuale di riferimento (“Reference manual”)
 - 3) Sfruttare la documentazione delle classi, disponibile online tramite Doxygen
 - 4) Sfruttare il gruppo di discussione di NS-3 su “google-groups”

Panoramica Generale

- NS-3 è un simulatore di rete → simula nodi e link
 - La classe che rappresenta un nodo di rete è la classe `Node`
 - Su di ogni nodo è possibile installare una o più applicazioni, che vengono realizzate attraverso la classe `Application`
 - I nodi della rete simulata sono collegati fra loro attraverso opportuni canali di comunicazione, realizzati attraverso la classe `Channel` o sue specializzazioni

Panoramica Generale

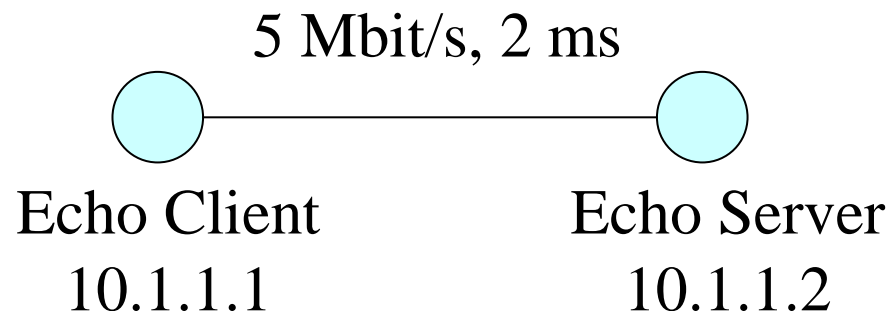
- Per connettere un Node ad un Channel è necessario utilizzare un dispositivo opportuno; in NS-3 esso viene modellato tramite la classe `NetDevice`
 - Un nodo può essere equipaggiato con un numero arbitrario di `NetDevice`, uno per ogni canale al quale esso è connesso
 - Ogni specializzazione di `NetDevice` è specifica per un particolare canale
 - Esempio:
 - `PointToPointNetDevice` è il `NetDevice` specifico per mettere in comunicazione due nodi attraverso un canale `PointToPointChannel`

Panoramica Generale

- NS-3 mette a disposizione numerosi metodi helper, utilizzabili per semplificare la realizzazione di topologie e configurazioni che richiederebbero la riscrittura di numerose linee di codice
 - Esempio:
 - `WifiHelper` è la classe che può essere utilizzata per istanziare un helper utilizzabile per realizzare in modo semplice oggetti di tipo `WifiNetDevice`

Il Primo Script

- Per iniziare consideriamo lo script “`first.cc`” del tutorial
 - In questo semplice script vengono realizzati due nodi di rete, connessi attraverso un link punto-punto
 - Il link punto-punto è caratterizzato da una bit-rate di livello fisico di 5 Mbps e da un ritardo di 2 ms
 - Il primo nodo ospita un’applicazione di tipo UdpEchoClient, mentre il secondo ospita il corrispondente server



II Primo Script

```
#include "ns3/core-module.h"
#include "ns3/simulator-module.h"
#include "ns3/node-module.h"
#include "ns3/helper-module.h"

using namespace ns3;
NS_LOG_COMPONENT_DEFINE ("FirstScriptExample");

int main (int argc, char *argv[])
{
    LogComponentEnable("UdpEchoClientApplication", LOG_LEVEL_INFO);
    LogComponentEnable("UdpEchoServerApplication", LOG_LEVEL_INFO);

    NodeContainer nodes;
    nodes.Create (2);

    PointToPointHelper pointToPoint;
    pointToPoint.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
    pointToPoint.SetChannelAttribute ("Delay", StringValue ("2ms"));
    ...
}
```

II Primo Script

```
...

NetDeviceContainer devices;
devices = pointToPoint.Install (nodes);

InternetStackHelper stack;
stack.Install (nodes);

Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");

Ipv4InterfaceContainer interfaces = address.Assign (devices);

UdpEchoServerHelper echoServer (9);

ApplicationContainer serverApps = echoServer.Install (nodes.Get (1));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));
...
```

II Primo Script

```
...

UdpEchoClientHelper echoClient (interfaces.GetAddress (1), 9);
echoClient.SetAttribute ("MaxPackets", UIntegerValue (1));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.)));
echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));

ApplicationContainer clientApps = echoClient.Install (nodes.Get (0));
clientApps.Start (Seconds (2.0));
clientApps.Stop (Seconds (10.0));

Simulator::Run ();
Simulator::Destroy ();
return 0;
}
```


Il Primo Script - Spiegazione

- Largo impiego di `container`, ovvero di oggetti che ospitano insiemi di oggetti specifici (es: `NodeContainer`, `ApplicationContainer`)
- Fase di configurazione separata da quella di effettiva implementazione
- Esempio:
 - Configurazione della base degli indirizzi IP con:
`address.SetBase ("10.1.1.0" ,
"255.255.255.0")`
 - Effettiva assegnazione degli indirizzi:
`address.Assign(devices)`

Il Primo Script – Compilazione ed Esecuzione

- La compilazione e la successiva esecuzione dello script vengono effettuate spostandosi, tramite una shell, nella root di NS-3 e lanciando i seguenti comandi:
 - **`./waf`**
 - **`./waf –run scratch/first`**
- Il primo comando effettua la compilazione dell'intero simulatore, compreso lo script “`first.cc`” (opportunamente spostato nella directory `scratch`). Il secondo comando esegue lo script compilato

Il Primo Script – Risultati

- Lo script, tramite le funzionalità di logging attivate ad inizio simulazione, stampa su standard output:
 - **Sent 1024 bytes to 10.1.1.2**
 - **Received 1024 bytes from 10.1.1.1**
 - **Received 1024 bytes from 10.1.1.2**
- Il client (primo nodo) invia 1024 byte verso il server (secondo nodo).
- Il server risponde al client con un pacchetto di echo da 1024 byte

Come continuare?

- 1) Proseguire con gli esempi presenti nel tutorial, reperibile online su www.nsnam.org
- 2) Sfruttare la documentazione reperibile online
 - Doxygen per la documentazione del codice sorgente
 - Manuale di riferimento
 - Gruppo di discussione su google-groups
- 3) Analizzare il codice sorgente fornito con il simulatore
 - Sono presenti numerosi esempi di utilizzo dei vari componenti software che lo compongono

Estensione del Simulatore

- Attualmente il simulatore comprende un buon numero di moduli pronti all'uso
- Può succedere che un modulo necessario per simulare un determinato scenario di rete sia mancante
- Come procedere in questo caso?
 - È possibile estendere le classi già presenti (logica OOP) e realizzare moduli personalizzati
 - Il livello di dettaglio richiesto per ogni modulo dipenderà dagli obiettivi dello studio (es: modellazione a livello di rete, a livello MAC, a livello fisico, etc)

Sommario

1. Introduzione alla simulazione numerica
2. Simulazione di reti di telecomunicazioni
3. Simulatore di rete: NS-3
4. Modulo di valutazione della QoE: **QoE Monitor**

Misura della Qualità Percepita

- Obiettivo:
 - valutazione dell'efficienza di un'eventuale architettura di rete considerata quantificando la qualità audio e video percepita dall'utente finale, detta Quality of Experience (QoE):
 - qualità soggettiva
 - qualità oggettiva

Misura Soggettiva di Qualità

- La misura soggettiva della qualità viene ottenuta facendo una media dei valori associati all'opinione di un gruppo di osservatori/ascoltatori
 - **MOS** (*Mean Opinion Score*)
 - il valore del MOS va da 1 (pessimo) a 5 (eccellente)

Mean Opinion Score (MOS)		
MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Misura Oggettiva di Qualità

- Calcolata automaticamente, si tratta di una misura quantitativa in grado di predire la qualità dell'immagine/voce percepita
- Due metriche per il **video**:
 - Basata sulla differenza tra pixel
 - **PSNR** (*Peak Signal-to-Noise Ratio*)
 - Basata sulla correlazione
 - **SSIM** (*Structural Similarity*)
- Una metrica per l'**audio**:
 - **PESQ** (*Perceptual Evaluation of Speech Quality*)

PSNR

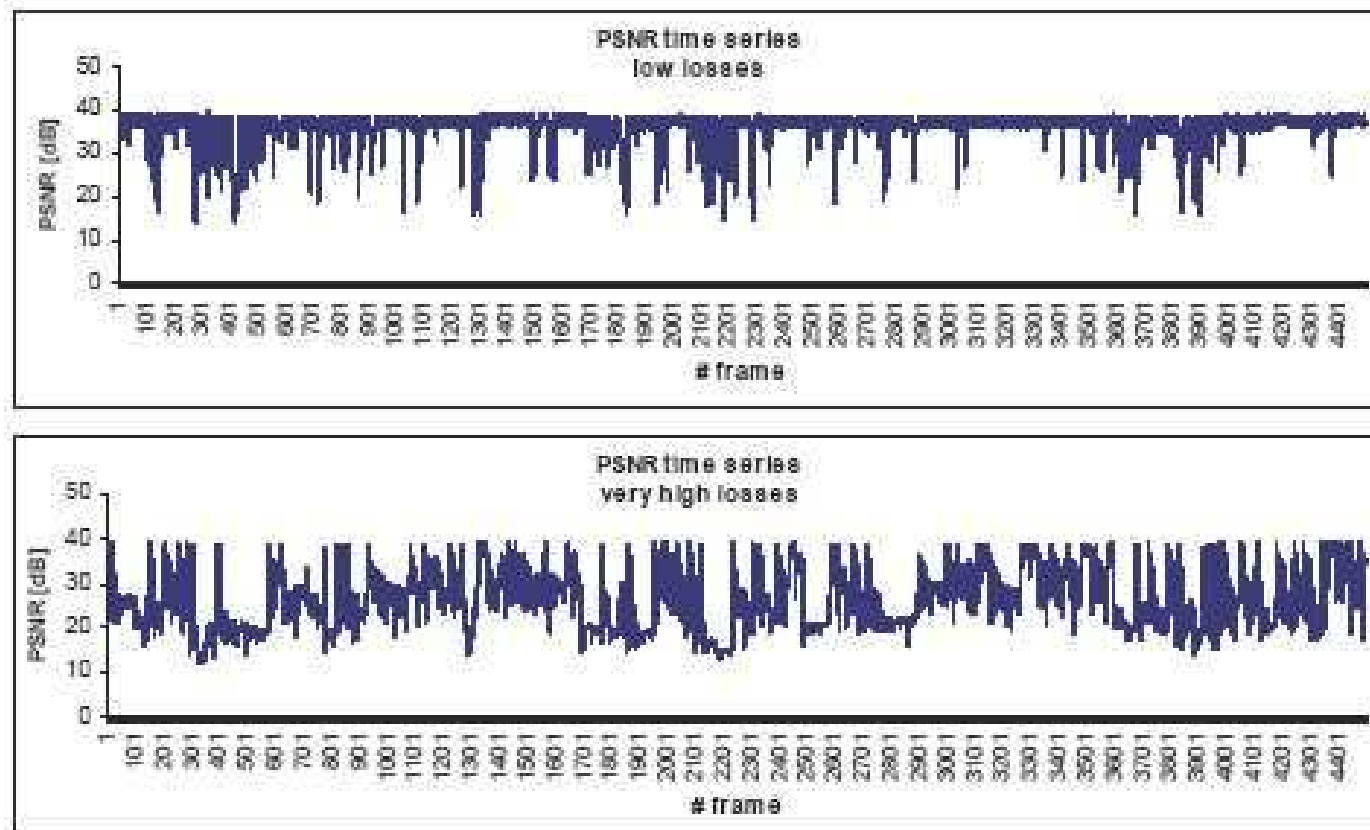
- PSNR è una misura oggettiva di qualità dell'immagine [20dB - 40dB]

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \end{aligned}$$

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

- dove
 - MAX_I è il massimo valore possibile del pixel dell'immagine
 - MSE (Mean Square Error) è l'errore quadratico medio:
 - I è l'immagine originale and K è quella compressa
 - $M \times N$ è la dimensione di entrambe le immagini

PSNR: esempio



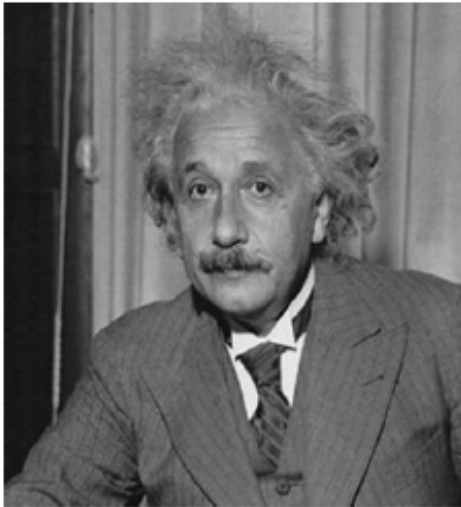
SSIM

- SSIM è una misura oggettiva di qualità dell'immagine [0 - 1]

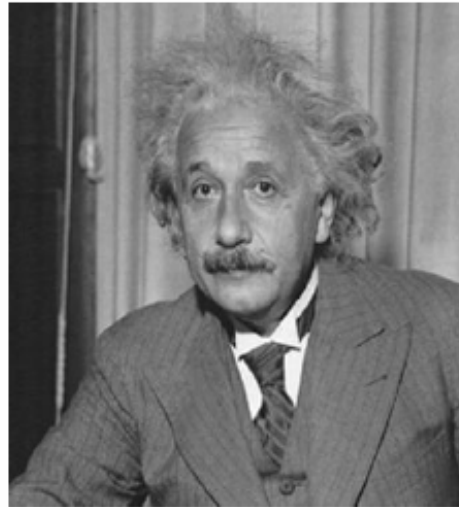
$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

- dove
 - x e y sono due finestre di dimensione NxN (tipicamente 8x8)
 - μ_x e μ_y sono le medie di x e y
 - σ_x^2 e σ_y^2 sono le varianze di x e y
 - σ_{xy} è la covarianza di x e y
 - $c_1=(k_1L)^2$ e $c_2=(k_2L)^2$ sono due variabili che stabilizzano la divisione, con $k_1=0.01$ e $k_2=0.03$ e $L=(2^{\#\text{bits per pixel}} - 1)$

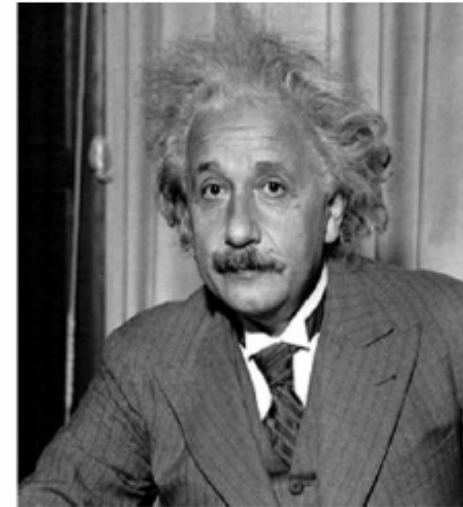
SSIM: esempio



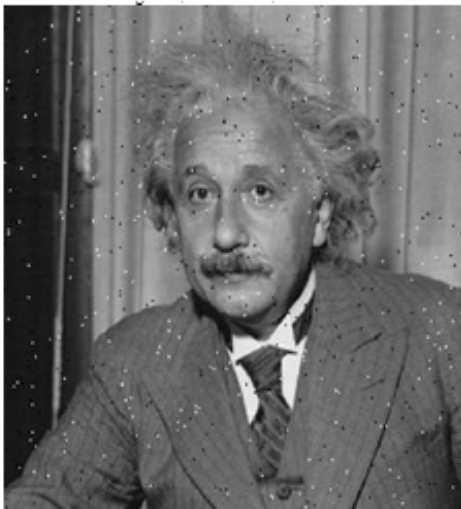
Original, MSE = 0; SSIM = 1



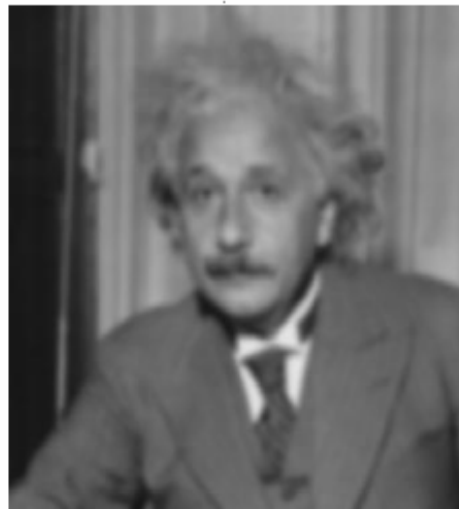
MSE = 144, SSIM = 0.988



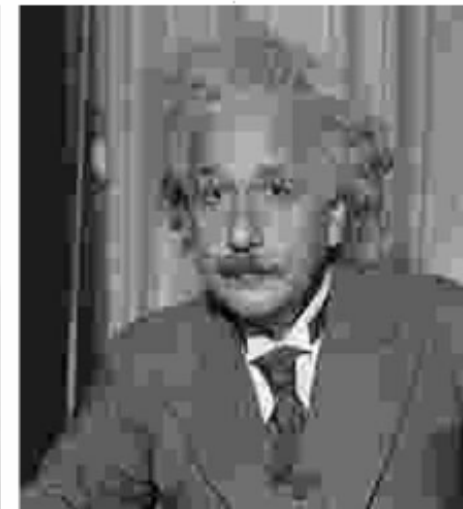
MSE = 144, SSIM = 0.913



MSE = 144, SSIM = 0.840



MSE = 144, SSIM = 0.694

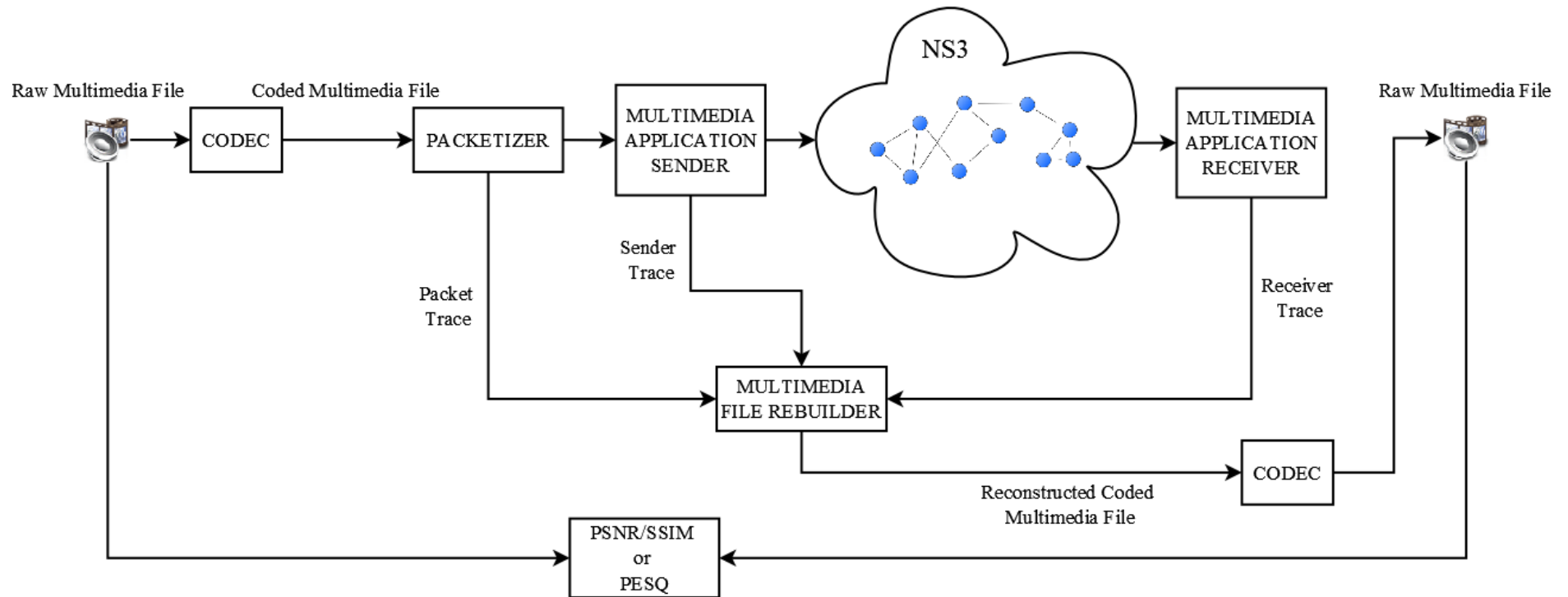


MSE = 142, SSIM = 0.662

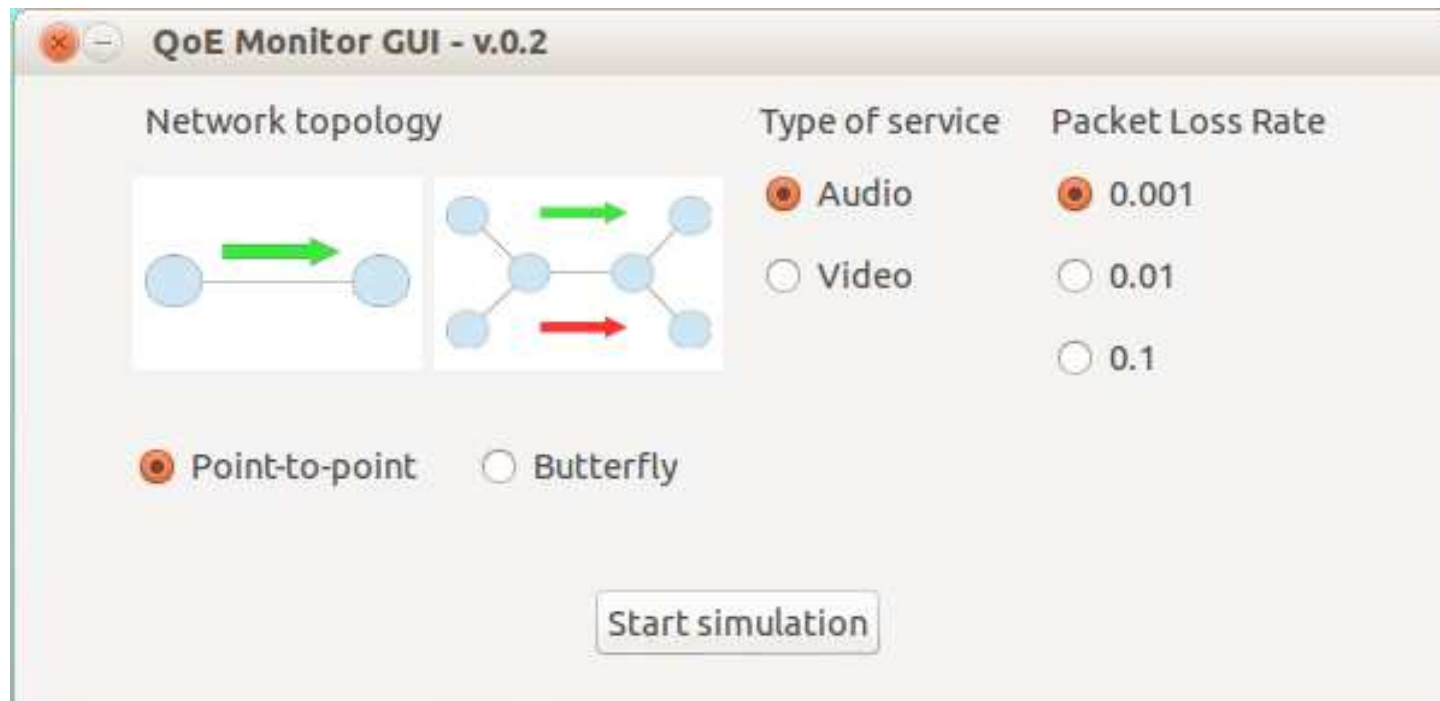
PESQ

- PESQ è una misura oggettiva di qualità della voce [0 – 5]
- E' basato su un modello percettivo-cognitivo che cerca di riprodurre il sistema uditivo reale
- Permette la valutazione della qualità sia in full-reference (confronto tra due file audio) che in no-reference (un solo file)
- **ITU.T P.862** implementa il PESQ: metodo full-reference, cioè confronto tra il file audio di riferimento (trasmesso) e il file audio degradato (ricevuto)

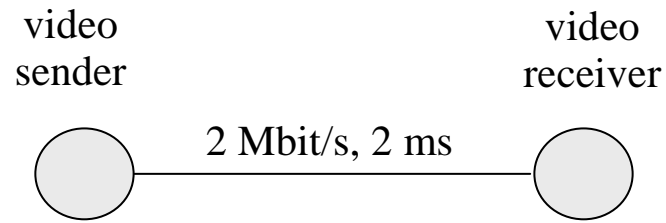
QoE Monitor: Working Principle



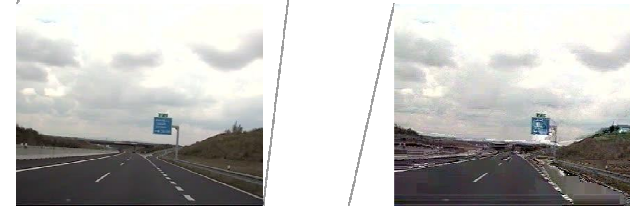
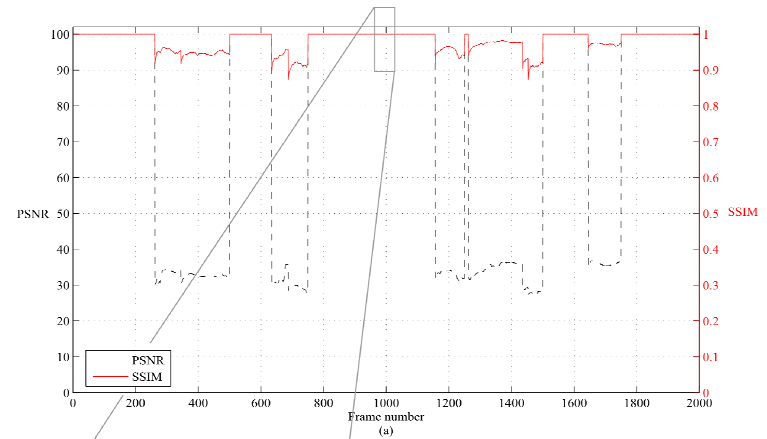
QoE Monitor: GUI



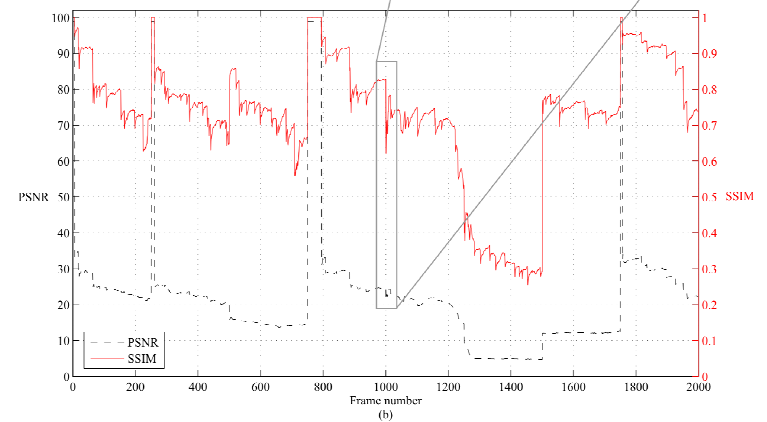
Qualche Risultato Numerico: Video



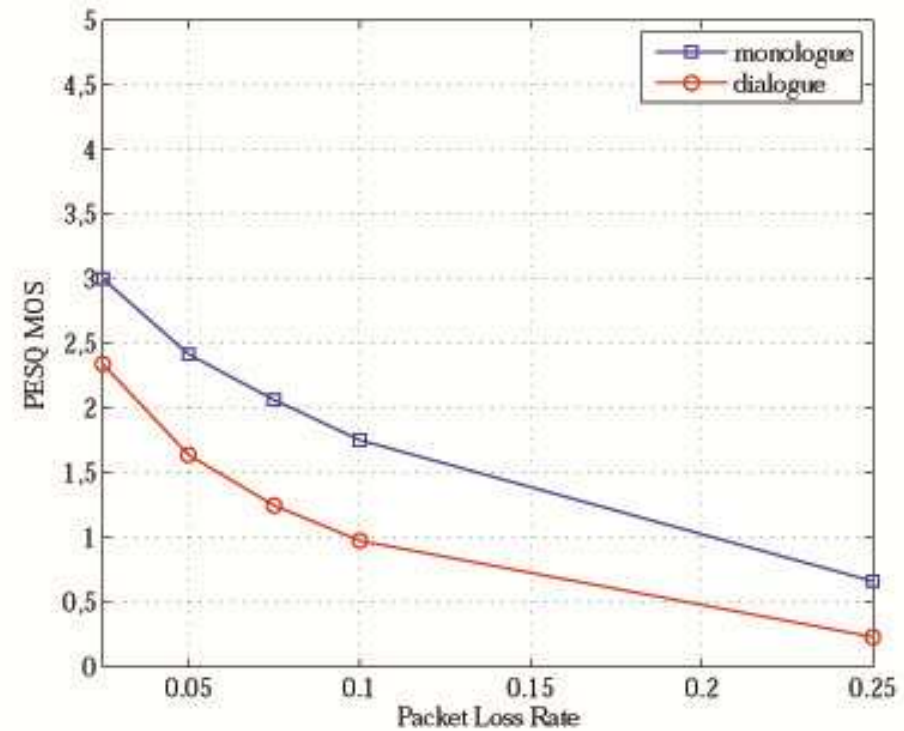
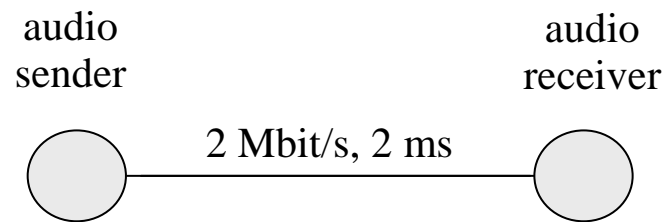
PER= 10^{-3}



PER= 10^{-2}



Qualche Risultato Numerico: Audio



Domande?

Daniela Saladino

email: daniela.saladino@unimore.it

Paper title: “A Tool for Multimedia Quality Assessment in NS3: QoEMonitor”, submitted to the *International Journal of Simulation Modelling Practice and Theory* (SIMPAT), Elsevier.