OPNET/Riverbed Modeler:
Deploying Standard Applications

Roman Dunaytsev

The Bonch-Bruevich Saint-Petersburg
State University of Telecommunications

roman.dunaytsev@spbgut.ru

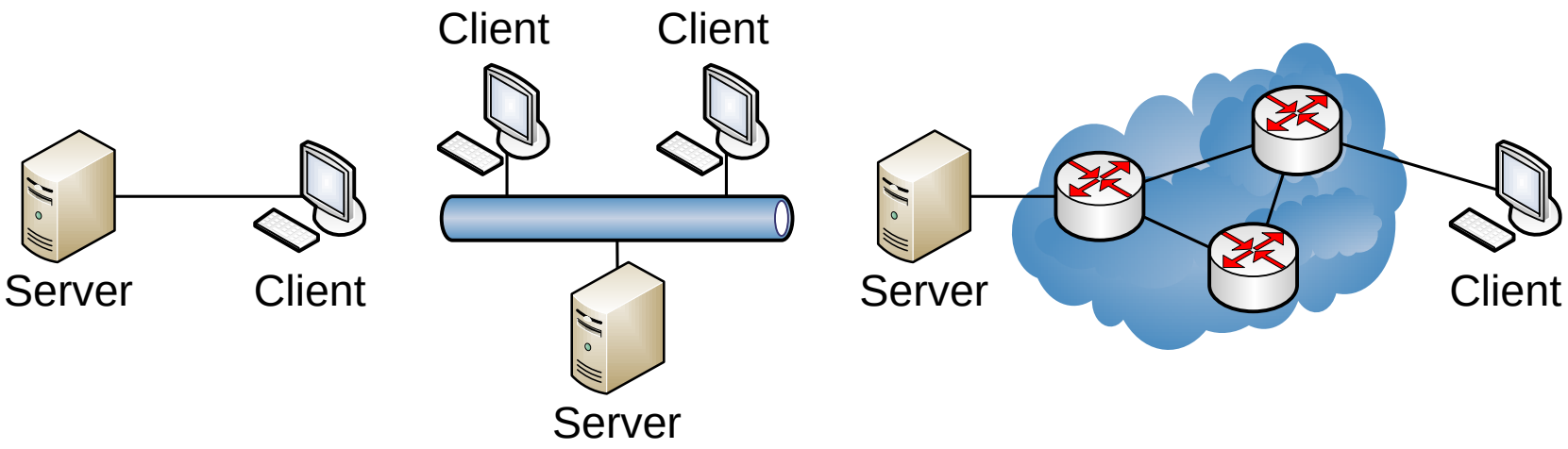
Lecture № 8

Outline

Outline

Application Architectures

- End systems can be made to communicate and share resources according to different interaction models
- **2 fundamental interaction models**:
  - Client/server
  - Peer-to-peer (aka P2P)
- **These models are relevant to end systems only**, regardless of how the end systems are connected to each other



Application Architectures (cont'd)

- In the client/server model, all end systems are divided into clients and servers each designed for specific purposes

- **Clients** – have an **active** role and initiate a communication session by sending requests to servers
  - Clients must have knowledge of the available servers and the services they provide
  - Clients can communicate with servers only; they cannot see each other

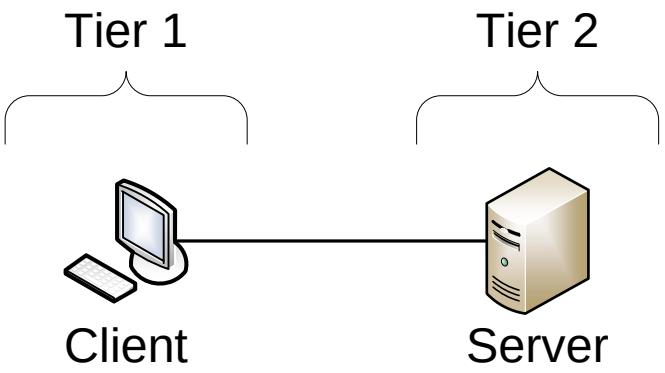
- **Servers** – have a **passive** role and respond to their clients by acting on each request and returning results
  - One server usually supports numerous clients

- The purpose of servers is to provide some service(s) to clients

Application Architectures (cont'd)

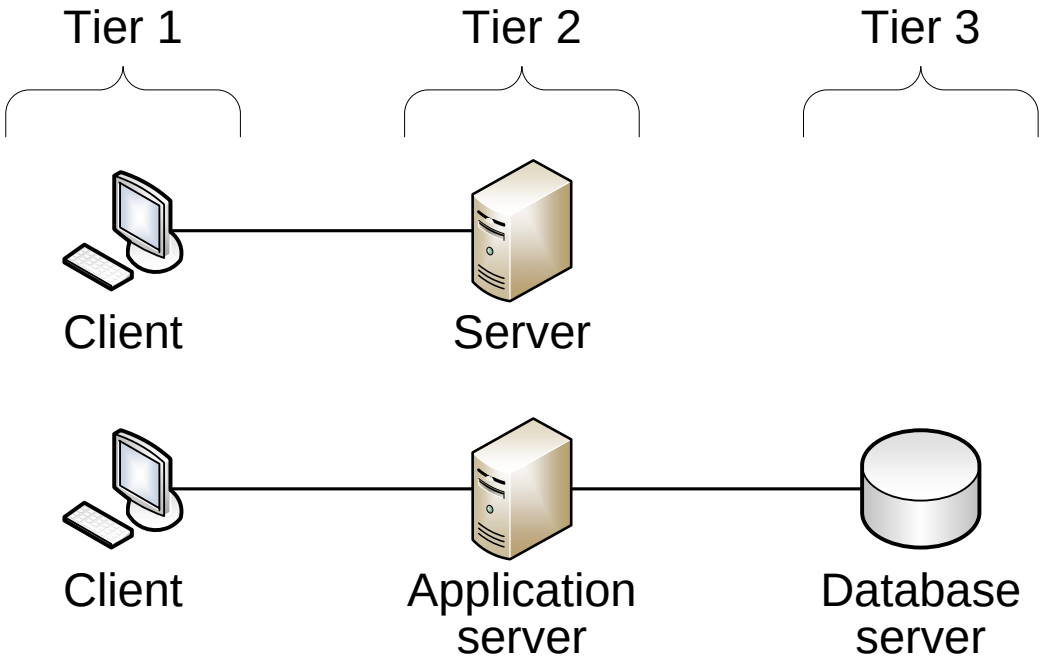
- **1-tier architecture** – used to describe centralized architectures in which all of the processing is done on a single host
  - Users can access such systems (aka **mainframes**) through display terminals (aka **dumb terminals**) but what is displayed and how it appears is controlled by the mainframe

- **2-tier architecture** (aka **flat**) – used to describe client/server application architectures, where clients request resources and servers respond directly to these requests, using there own resources



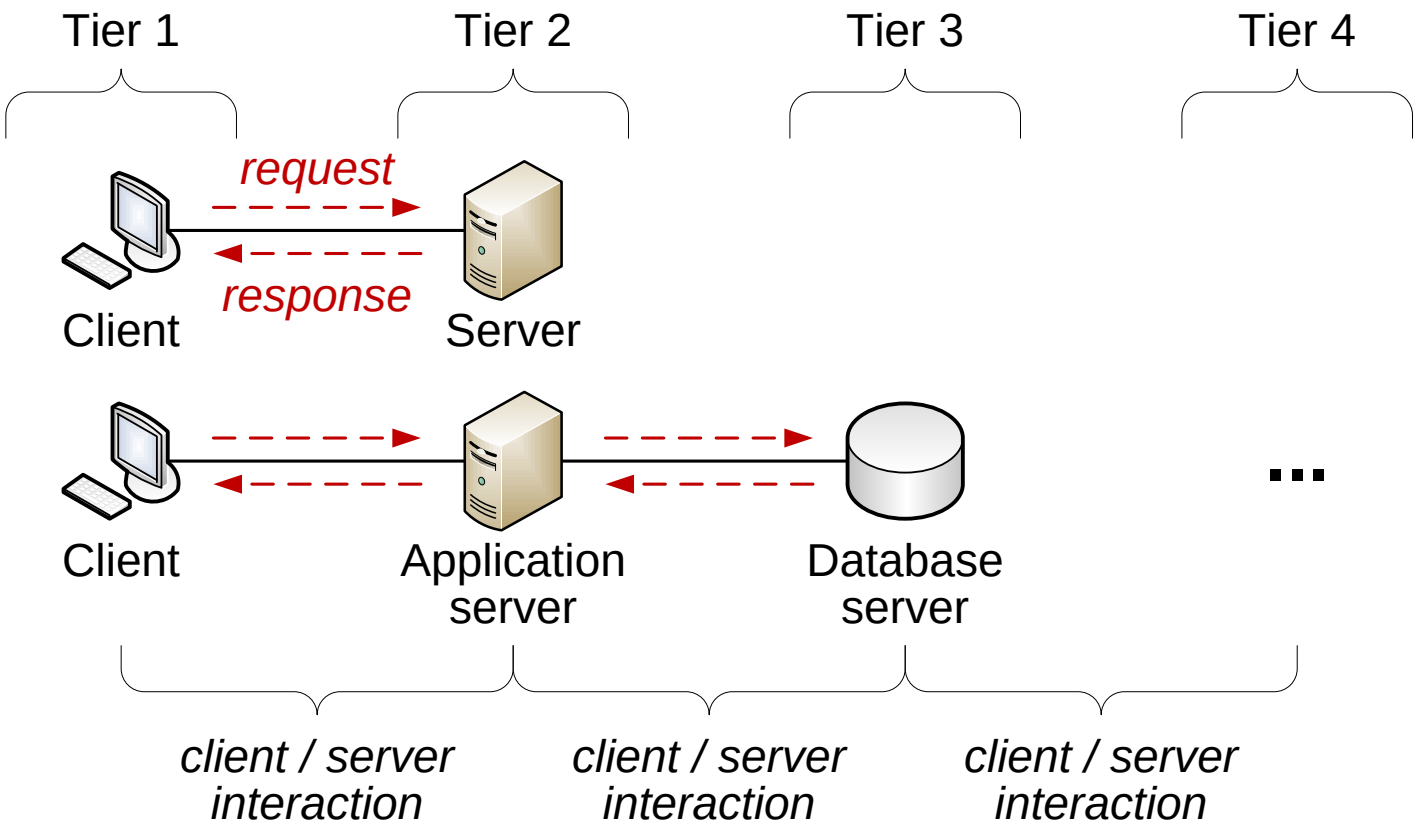
Application Architectures (cont'd)

- **3-tier architecture** – used to describe client/server application architectures consisting of:
  - **Clients** which request services
  - **Application servers** whose task is to provide the requested resources, but by calling on database servers
  - **Database servers** which provide the application servers with the data they require



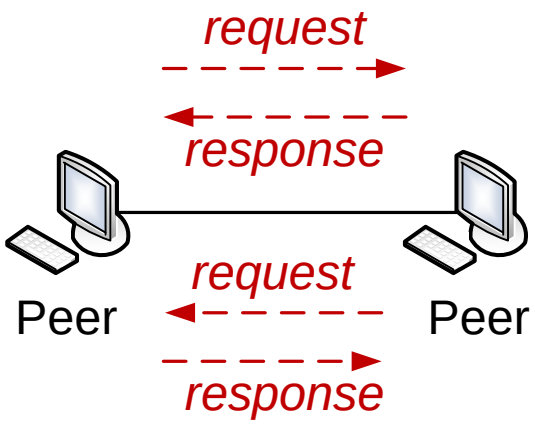
Application Architectures (cont'd)

- **N-tier architecture** – used to describe client/server application architectures consisting of more than 3 tiers



Application Architectures (cont'd)

- In the P2P model, there is minimal (or no) reliance on dedicated servers; instead the application exploits direct communication between pairs of intermittently connected end systems, called **peers**
  - The peers are not owned by the service provider, but are instead desktops and laptops controlled by users, with most of the peers residing in homes, universities, and offices
  - The participants are equal and simultaneously function as both resource providers (servers) and resource requestors (clients)



Outline

Adding Traffic

- Adding traffic – an important step in network modeling and simulation
- Application traffic traveling through the network enables us to observe the behavior and study the performance of various network protocols in their operational environment
- Without it, the network is only populated with the control packets generated by certain protocols upon initialization or periodically

Adding Traffic (cont'd)

- With OPNET/Riverbed, you can model the following types of traffic:
  - Explicit (aka 'discrete') traffic
  - Background (aka 'analytical') traffic
  - Hybrid traffic (explicit + background)

- **Explicit traffic** – provides a very accurate description of traffic behavior, which is achieved by modeling the complete life cycle of every packet generated by the traffic sources
  - I.e., packet created, packet queued, packet transmitted, etc.

- Explicit traffic modeling provides the most accurate results because it simulates all protocol effects

- However, this also results in longer simulation execution time and higher memory usage (because the simulation allocates memory for each individual packet)

Adding Traffic (cont'd)

- **Background traffic** – provides an analytical, and therefore less accurate, representation of data transfer over the network
- The occupancy of the queues at the intermediate nodes is adjusted based on the configuration of the background traffic models, which in turn influences the delays and other performance measures of the corresponding simulated protocols and network devices
- Background traffic modeling does not simulate the life cycle of individual packets; instead, it employs an analytical representation of traffic behavior
- Therefore, a simulation study that includes background traffic models will execute faster and will consume less memory than a simulation study that uses explicit traffic models

Adding Traffic (cont'd)

- **Hybrid traffic** – combines the advantages of both approaches by using explicit models and obtaining an accurate representation of those traffic sources that require a detailed evaluation, while employing background models for the traffic sources that do not require accurate representation

- Hybrid modeling provides greater accuracy over background traffic modeling, while using fewer resources and speeding up execution as compared to explicit traffic modeling

- In practice, simulation studies often rely on a combination of explicit and background models

Outline

Application Config

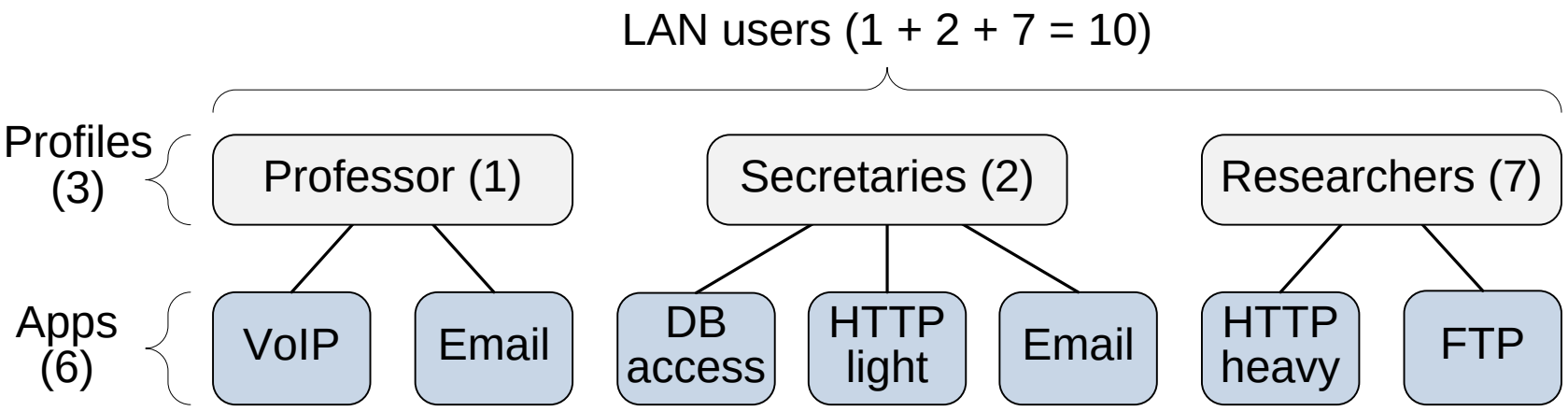
- In OPNET/Riverbed, deploying applications actually means defining user profiles and applications to be simulated

- **Application Config** – specifies **standard** and **custom** applications used in the simulation, including traffic and QoS parameters
  - Standard applications (Light/Medium/Heavy/etc.): Database, Email, FTP, HTTP, Print, Peer-to-peer File Sharing, Remote Login, Video Conferencing, Video Streaming, Voice

- **Profile Config** – specifies the activity patterns of a user or groups of users in terms of the applications used over a period of time
  - When does a user start using applications?
  - What is the duration of his/her activity?
  - What applications does he/she use?
  - How often does he/she use each application?
  - etc.

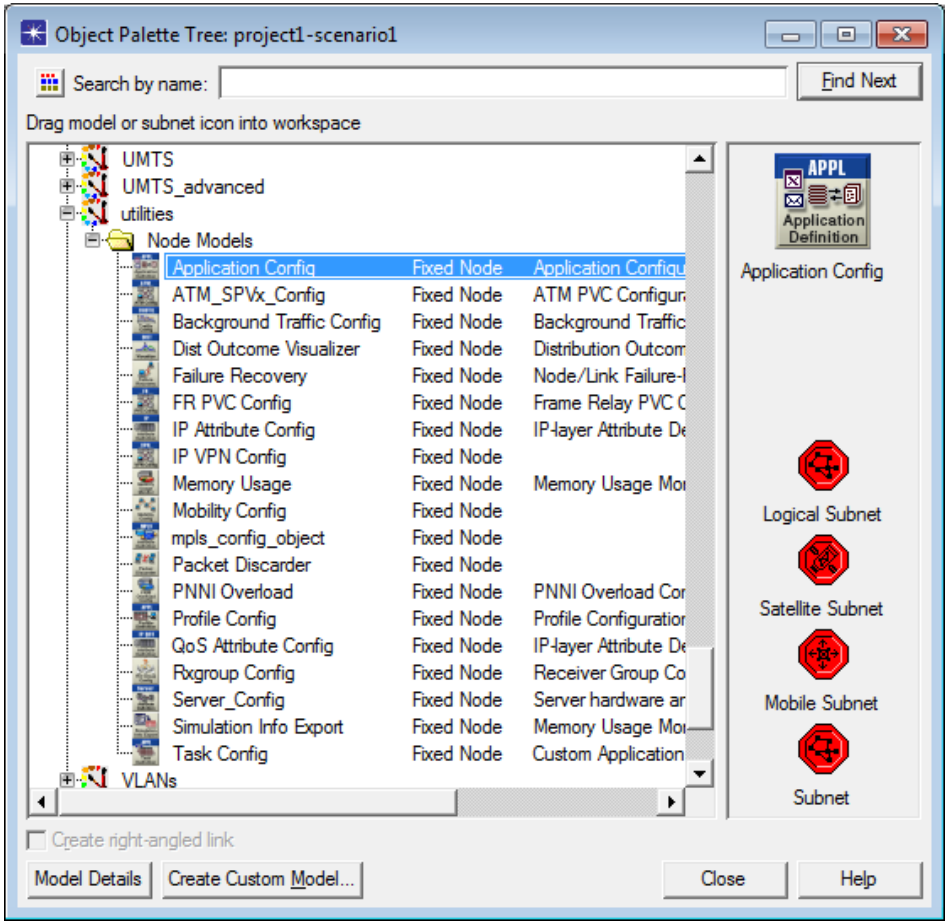
Application Config (cont'd)

- **Example of 3 user profiles on a small office network:**
  - User profile 'Professor' represents a head of department who only runs VoIP and Email applications
  - User profile 'Secretary' represents a clerk whose duties involve Email, light Web surfing, and database access
  - User profile 'Researcher' represents an employee actively surfing the Web and accessing FTP sites



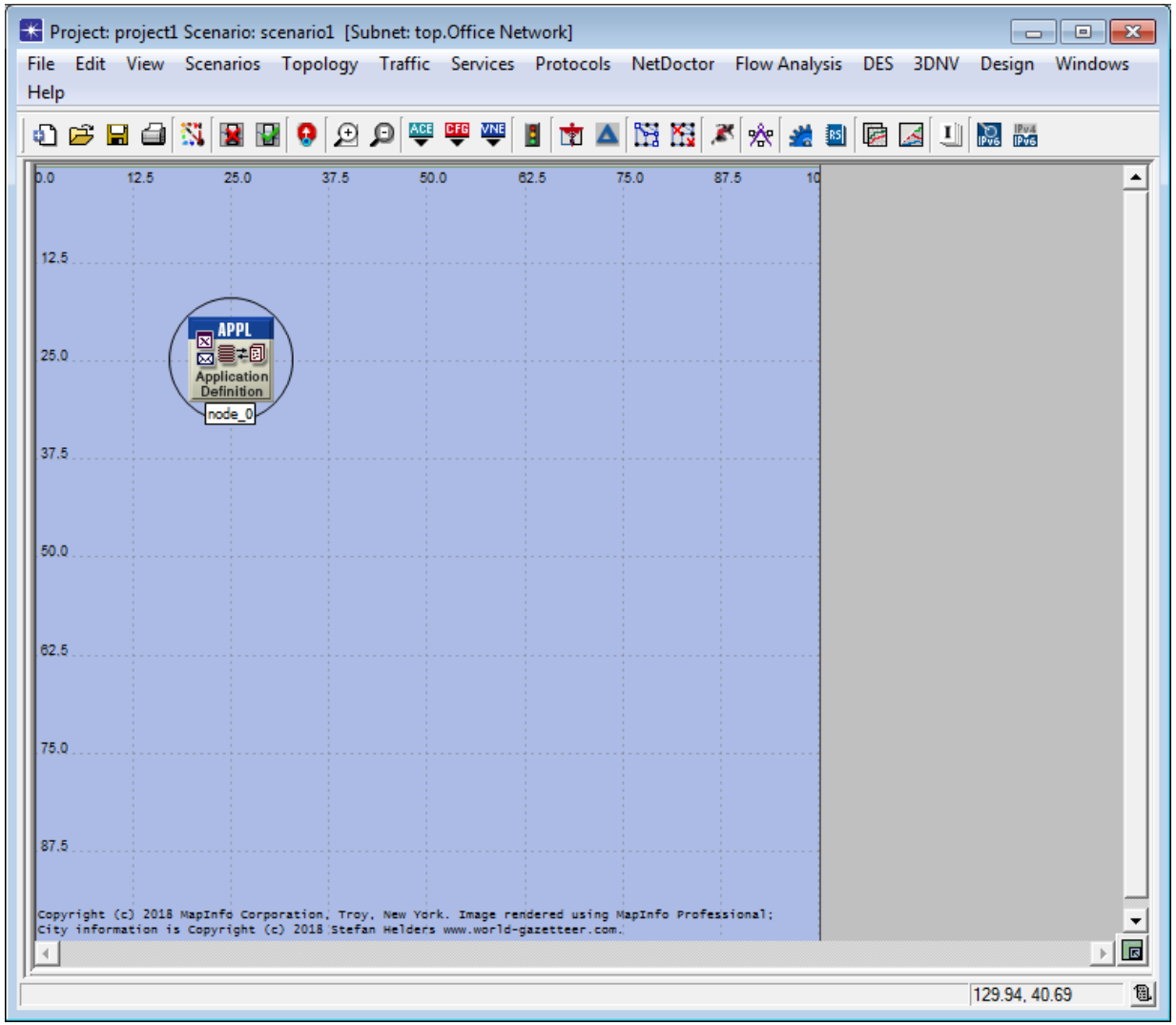
Application Config (cont'd)

- Open Object Palette ⇒
  - ⇒ Shared Object Palettes ⇒ internet_toolbox ⇒ Node Models
  - ⇒ Shared Object Palettes ⇒ applications ⇒ Node Models
  - ⇒ Shared Object Palettes ⇒ utilities ⇒ Node Models



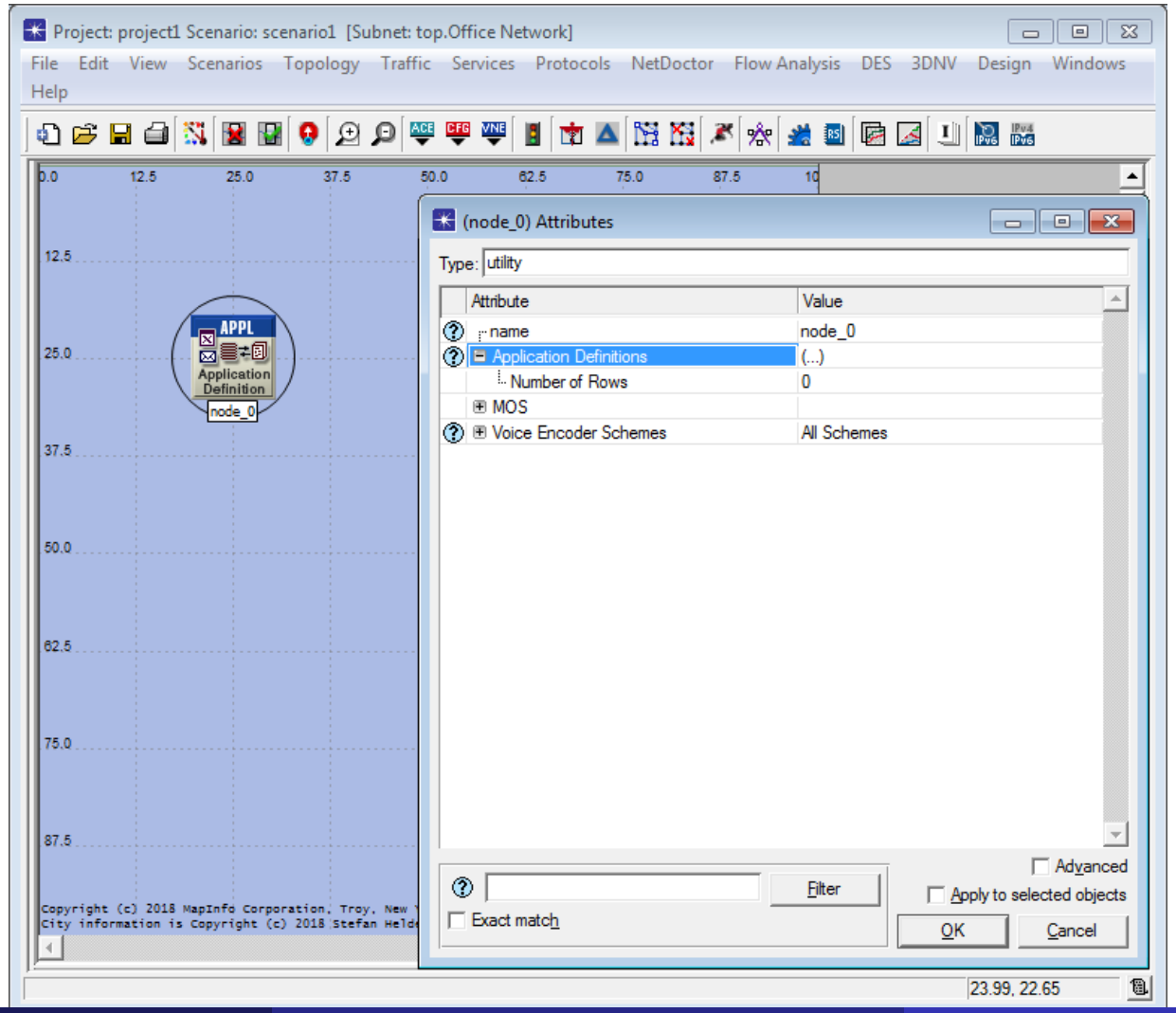
Application Config (cont'd)

- A simulated scenario should contain only 1 **Application Config** object



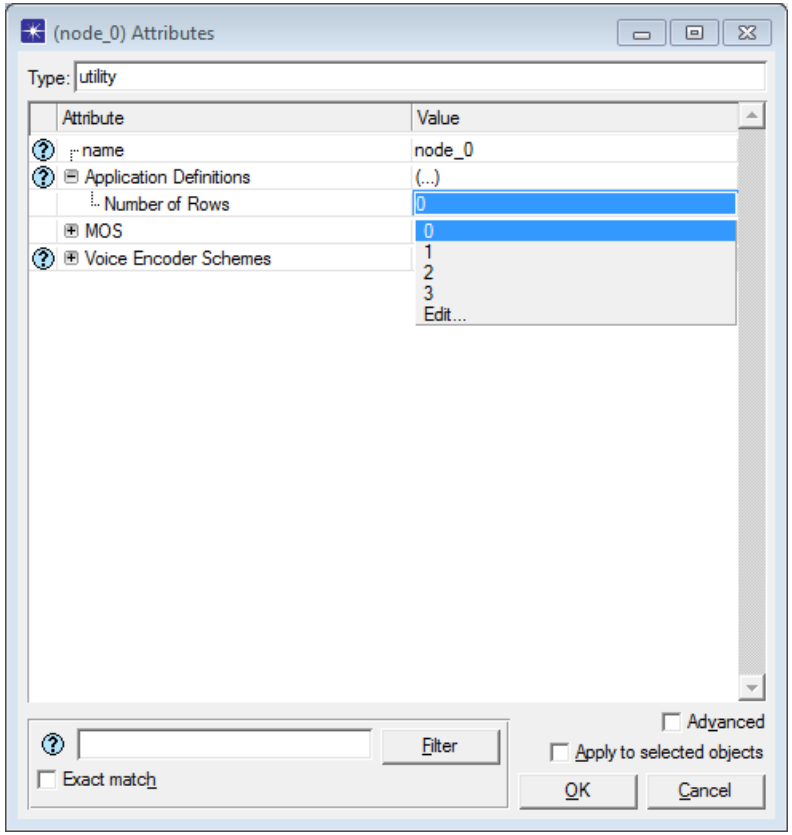
Application Config (cont'd)

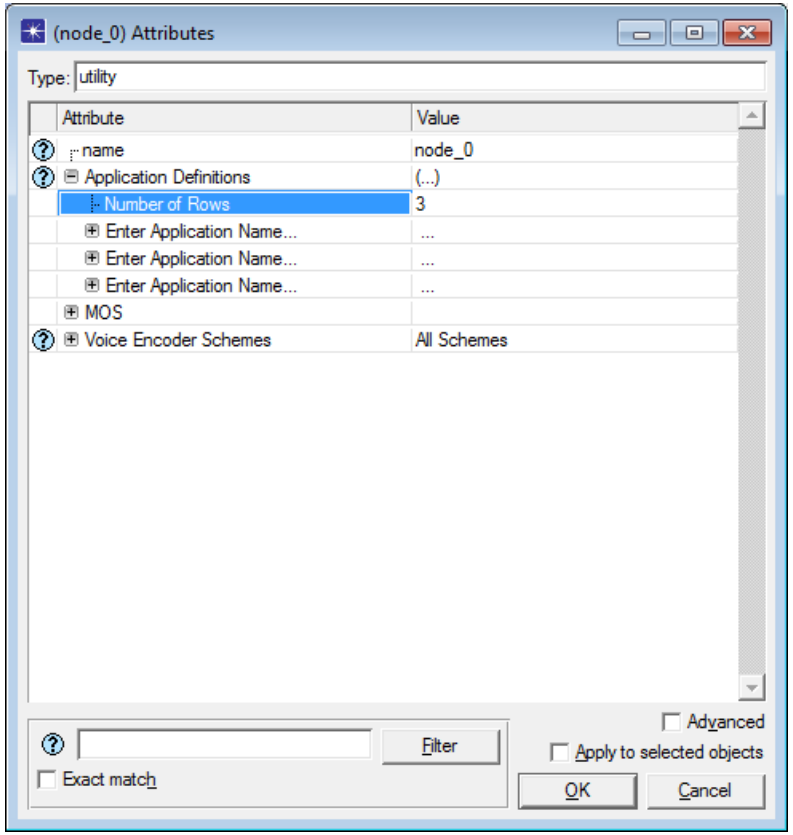
- By default, Number of Rows = 0
  - I.e., no applications configured in the current scenario



Application Config (cont'd)

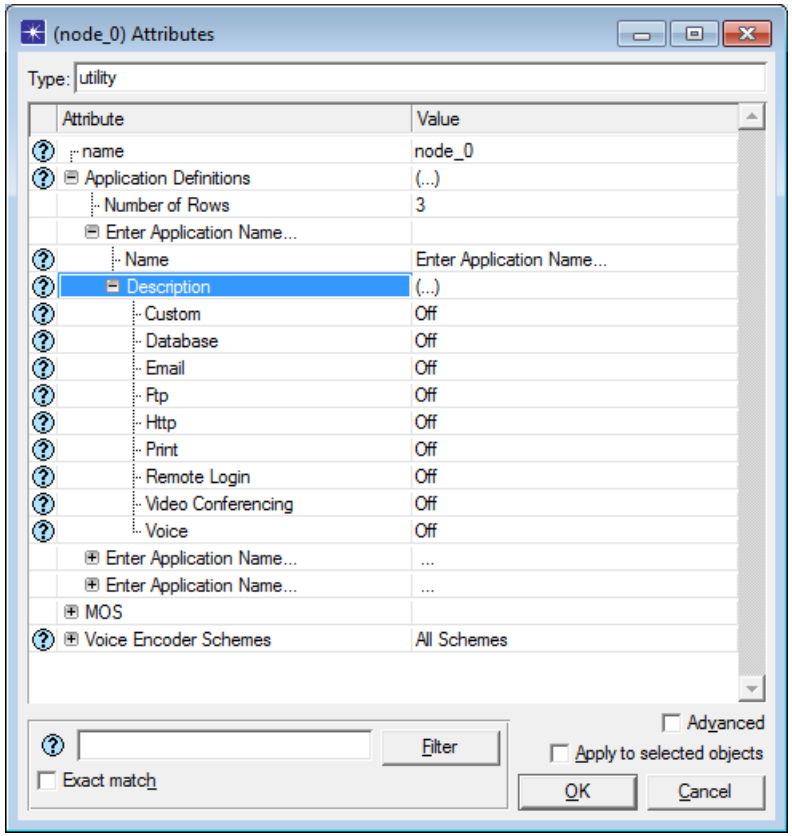
- Number of Rows = N results in N application definition entries

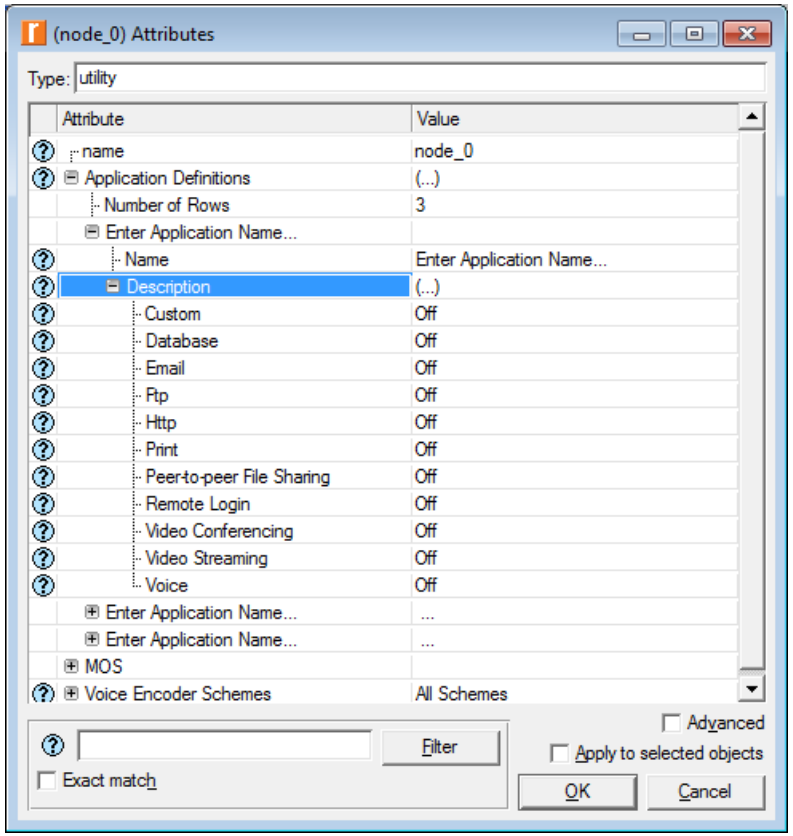




Application Config (cont'd)

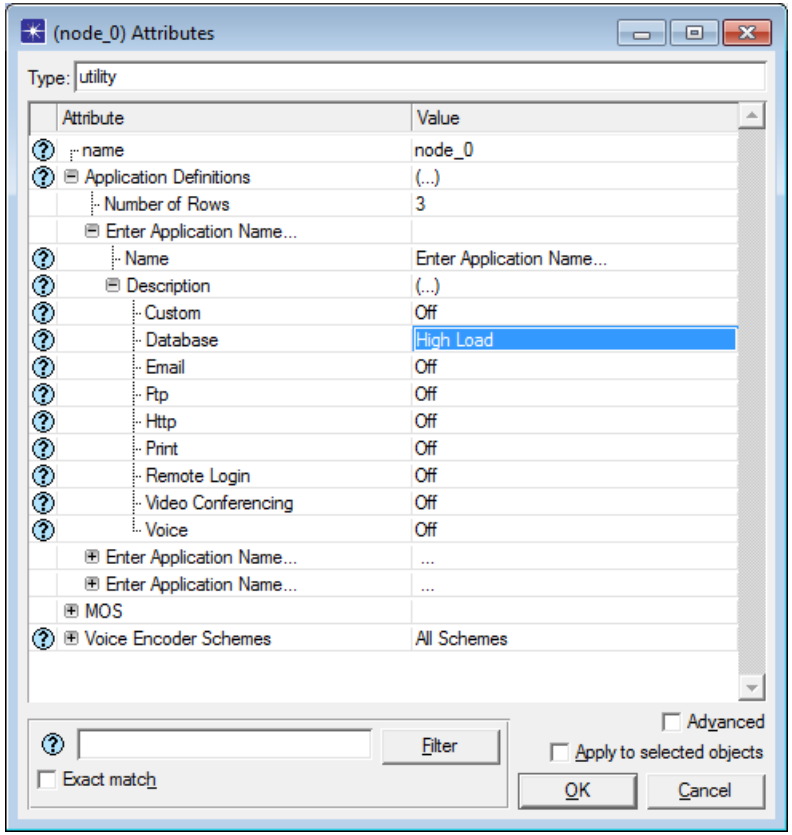
- **OPNET Modeler 14.5** vs. **Riverbed Modeler Academic Edition**
  - + Peer-to-peer File Sharing
  - + Video Streaming

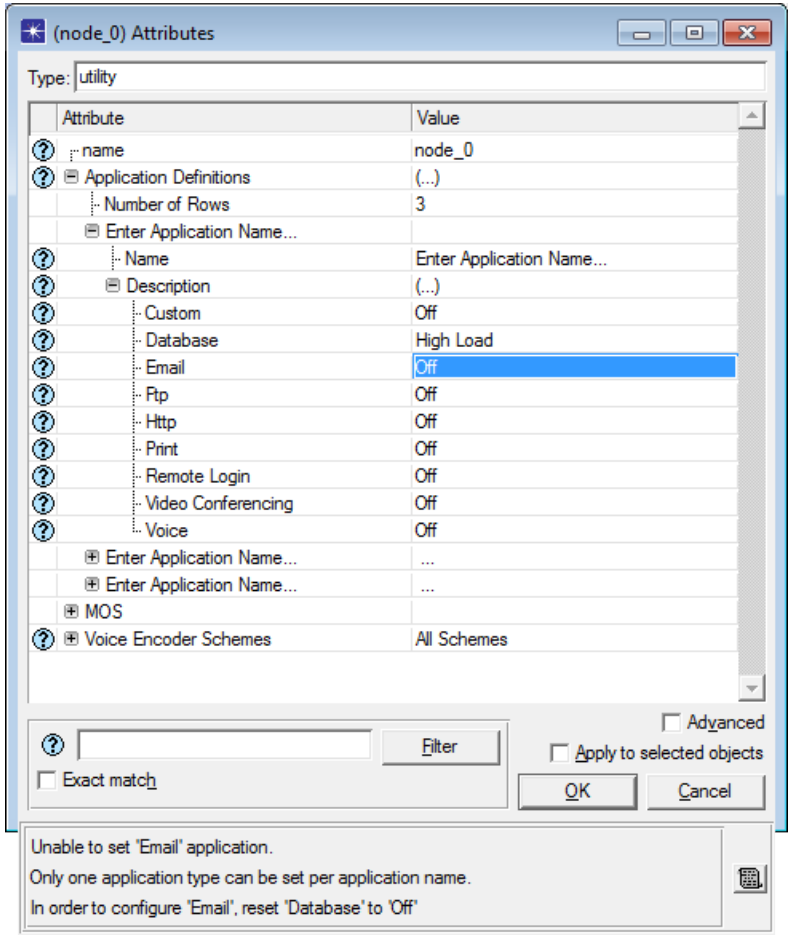




Application Config (cont'd)

- Only one application type can be set per application name!

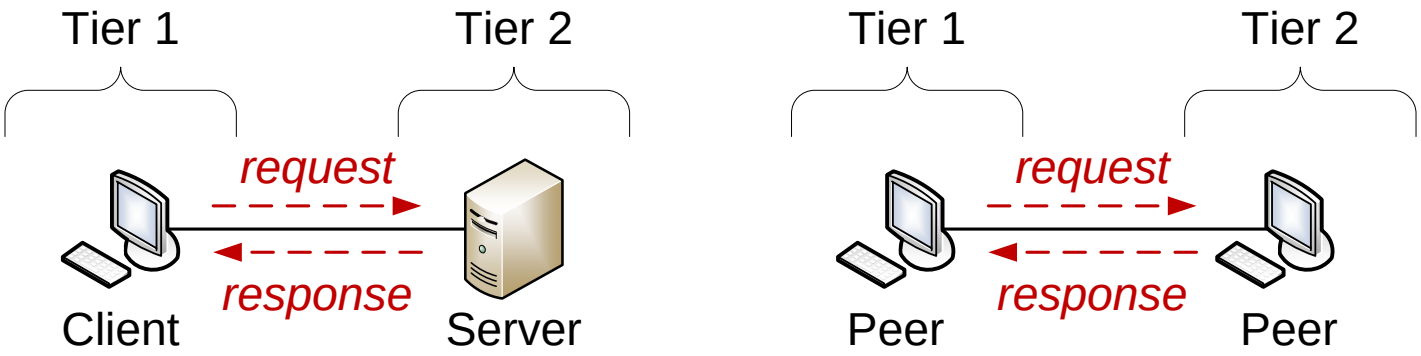




Outline

Configuring Standard Applications

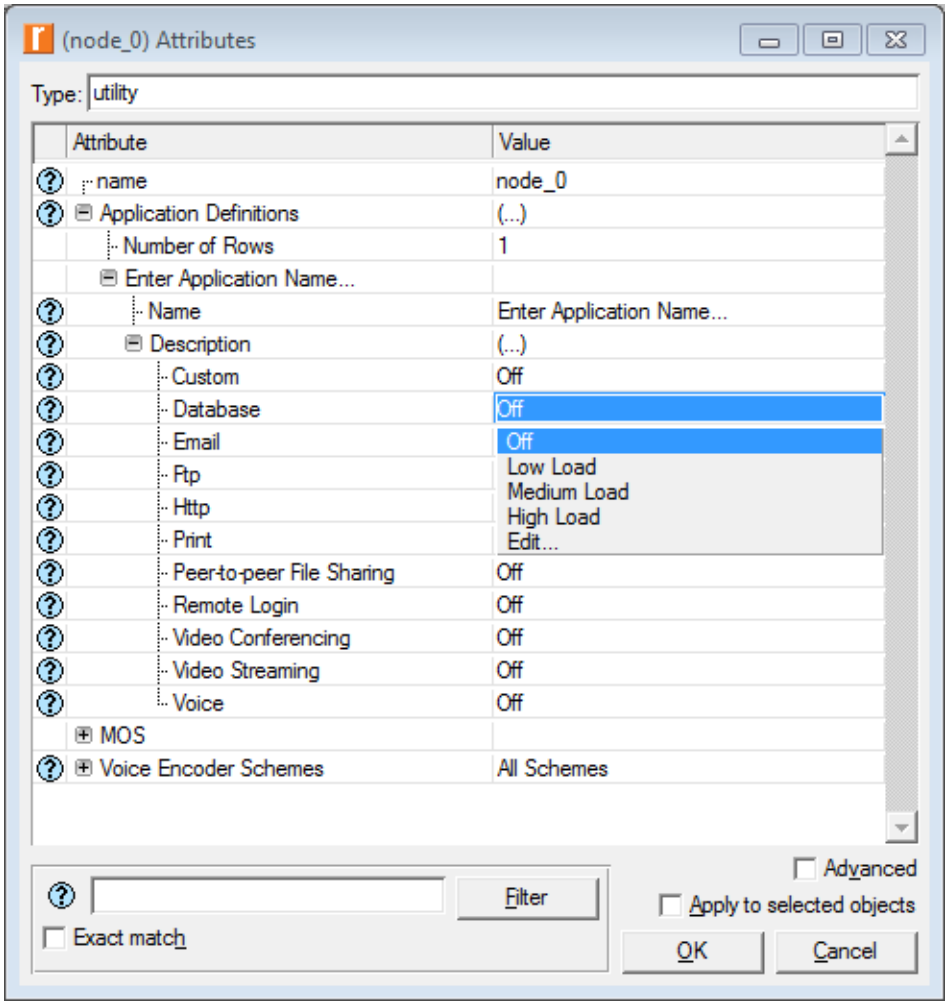
- **Standard applications** are implemented in a **2-tier architecture**
  - A client issues a request and a server or another client (peer) receives the request and returns a response
  - This request-response exchange typically happens within one communication session between client and server or between peers



- **Custom applications** allow modeling applications with more than 2 tiers, where a client request can be forwarded through multiple servers before a response message is sent back

Configuring Standard Applications (cont'd)

- **Database** – models a protocol that executes 2 types of database operations: query and entry



Configuring Standard Applications (cont'd)

- **Database query**
  1. A 512-byte query message that carries the database request
  2. A response message that carries the data
- **Database entry**
  1. An entry message that carries the data
  2. A 512-byte response message that carries the database acknowledgement of the operation
- All database queries or entries are transmitted over a single transport connection
- The default transport protocol = **TCP**

Configuring Standard Applications (cont'd)

- **Low Load** vs. **Medium Load** vs. **High Load**
  - Each of the preset values is configured to execute 100% of query transactions
  - The only difference is the size of the transaction response and the frequency of transaction arrival

(Database) Table

Attribute	Value
Transaction Mix (Queries/Total Transactions)	100%
Transaction Interarrival Time (seconds)	exponential (30)
Transaction Size (bytes)	constant (16)
Symbolic Server Name	Database Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Details Promote OK Cancel

(Database) Table

Attribute	Value
Transaction Mix (Queries/Total Transactions)	100%
Transaction Interarrival Time (seconds)	exponential (12)
Transaction Size (bytes)	constant (512)
Symbolic Server Name	Database Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Details Promote OK Cancel

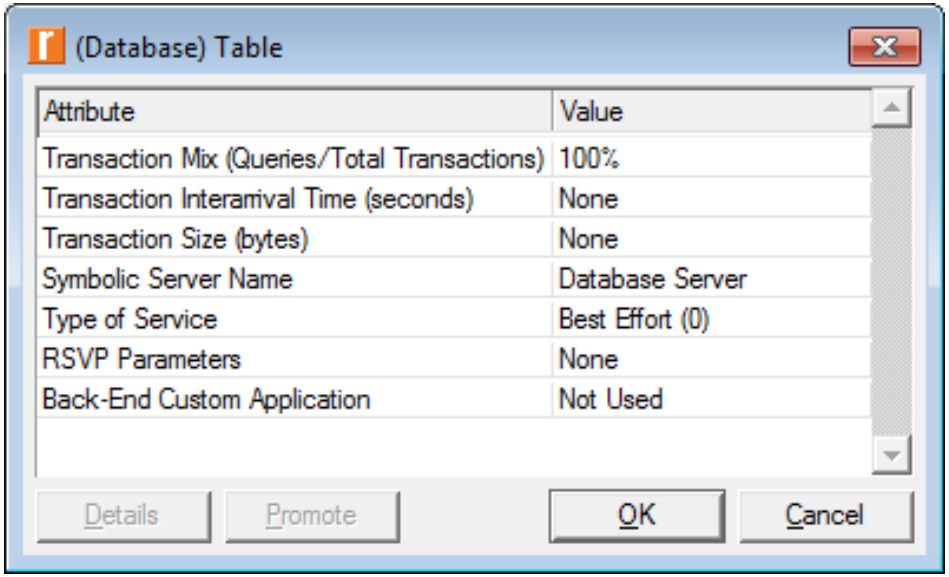
(Database) Table

Attribute	Value
Transaction Mix (Queries/Total Transactions)	100%
Transaction Interarrival Time (seconds)	exponential (12)
Transaction Size (bytes)	constant (32768)
Symbolic Server Name	Database Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Details Promote OK Cancel

Configuring Standard Applications (cont'd)

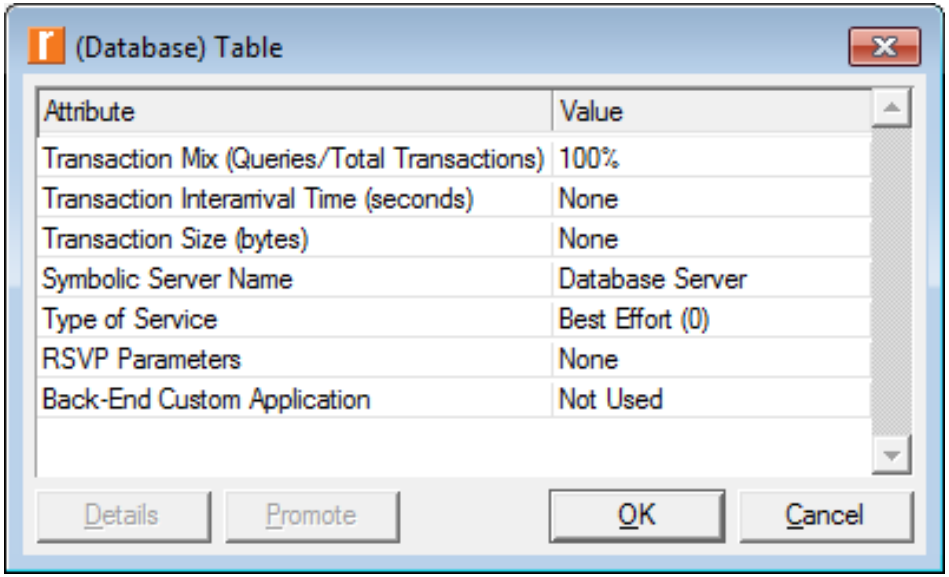
- **Transaction Mix (Queries/Total Transactions)** – percentage of queries among all database transactions
  - 0% – all database operations are entries
- **Transaction Interarrival Time (seconds)** – time between transactions
  - The start time of the next transaction is computed by adding the value of this attribute to the time when the previous transaction started
  - The next transaction can be initiated before the previous one completes



Configuring Standard Applications (cont'd)

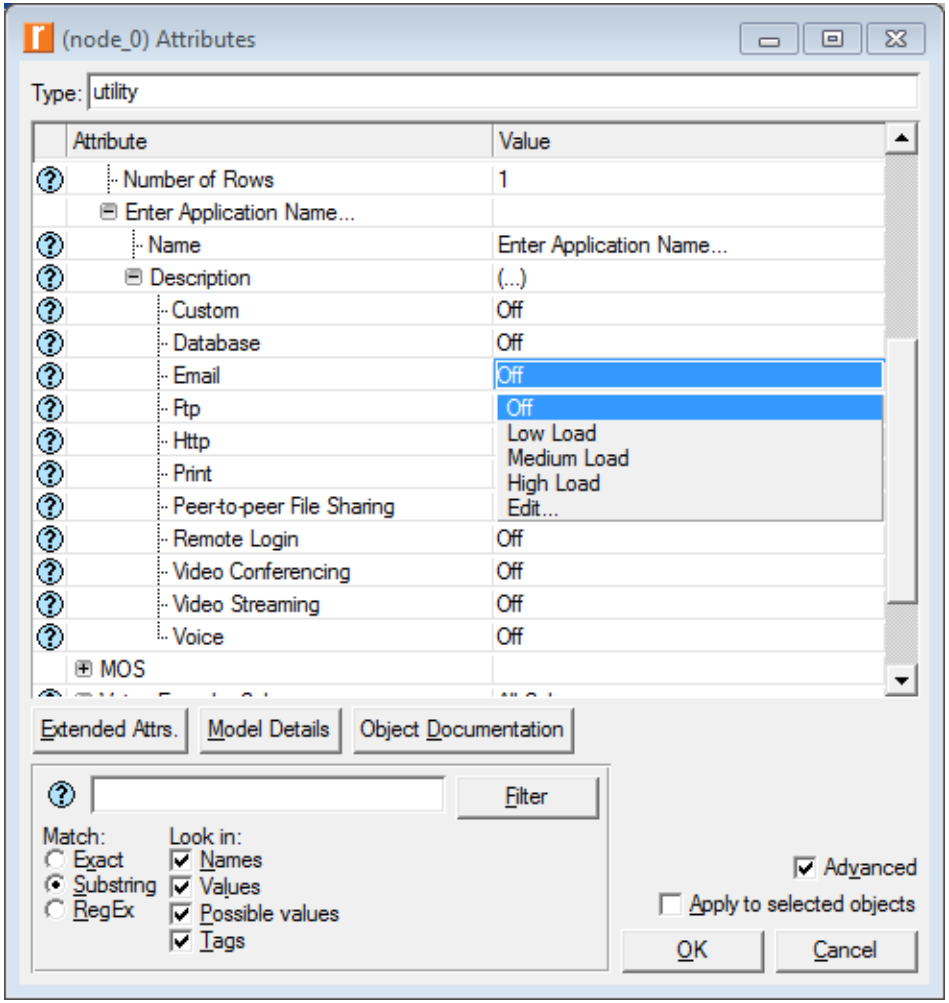
- **Transaction Size (bytes)** – size of an entry message or a response message to a database query

- **Symbolic Server Name** – common symbolic name for all the nodes that operate as database servers for a given application definition
  - Typically, the value of this attribute remains unchanged
  - This attribute is part of every standard application definition



Configuring Standard Applications (cont'd)

- **Email** – models message exchange between an e-mail client and an e-mail server

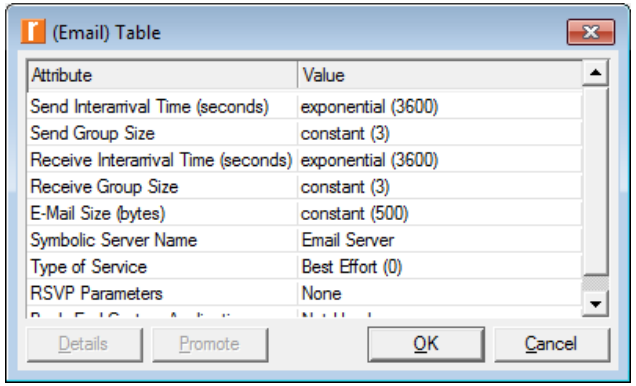


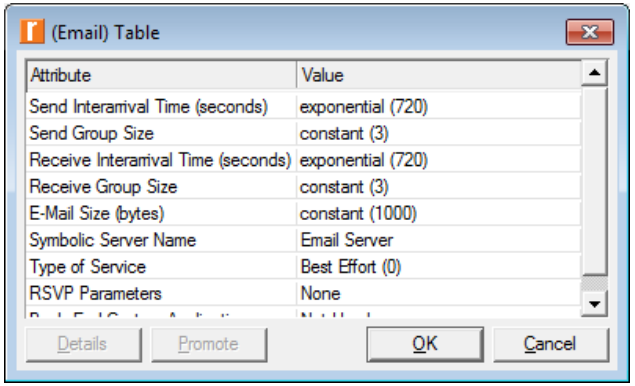
Configuring Standard Applications (cont'd)

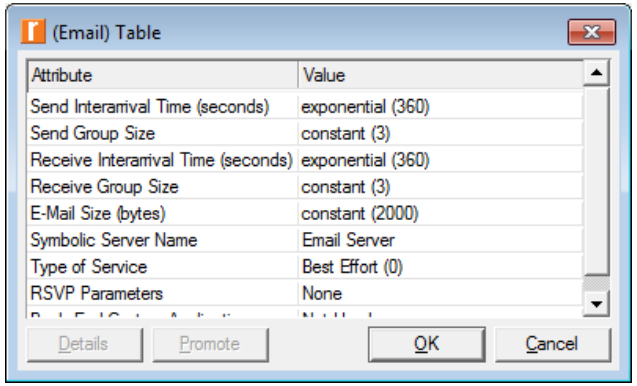
- **Sending a group of e-mail messages**
  1. A group of e-mail messages from the client
  2. A 16-byte response message from the server
  3. A 8-byte close confirmation message from the server
- **Receiving a group of e-mail messages**
  1. A 16-byte request message from the client
  2. A group of e-mail messages from the server
  3. A 8-byte close confirmation message from the server
- Each group of e-mails to be uploaded or downloaded creates a new transport connection with the server
- The default transport protocol = **TCP**

Configuring Standard Applications (cont'd)

- **Low Load** vs. **Medium Load** vs. **High Load**
  - Each of the preset values is configured to send and receive 3 e-mail messages per group
  - The main difference is the size of e-mail messages and the frequency of transactions

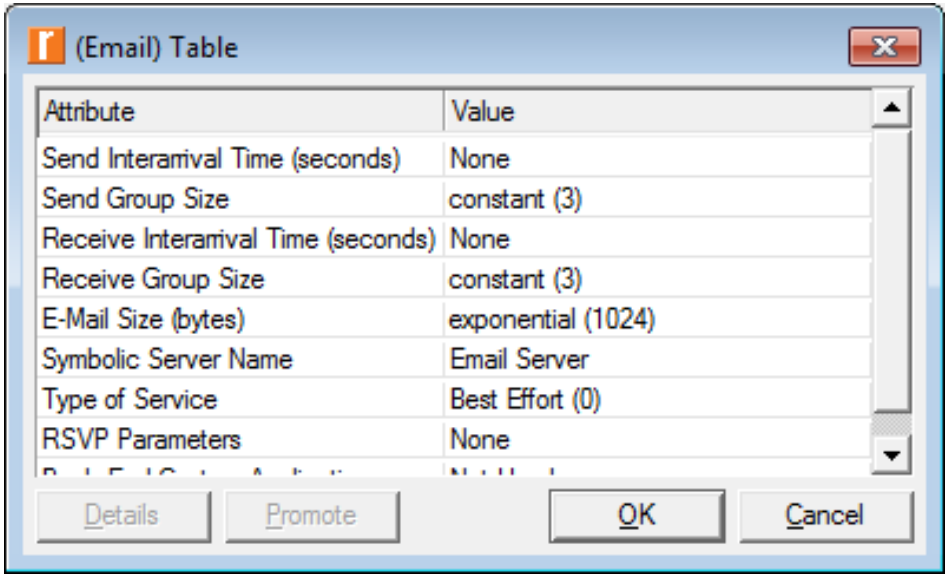






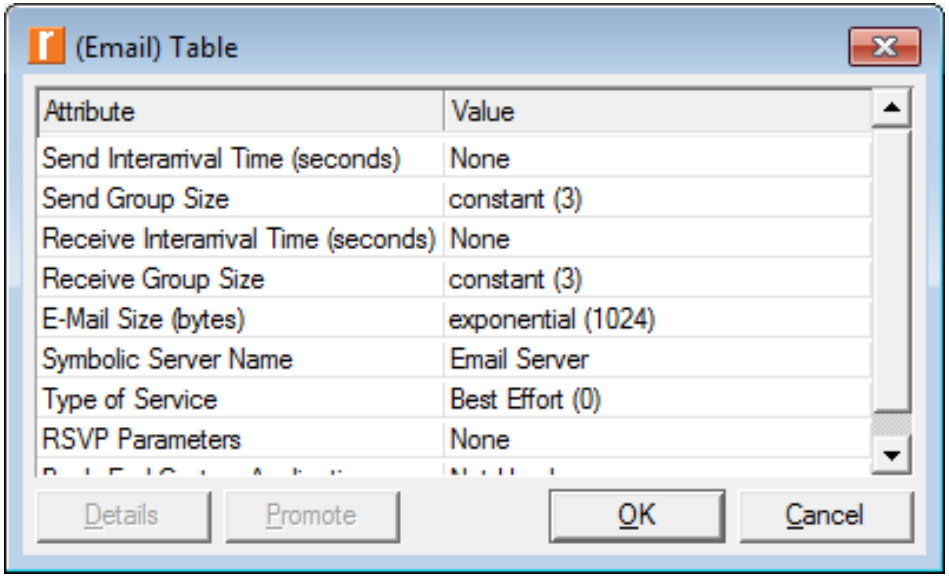
Configuring Standard Applications (cont'd)

- **Send Interarrival Time (seconds)** – time between groups of e-mails sent from the client
  - The start time of the next transaction is computed by adding the value of this attribute to the time when the previous transaction started
  - The next transaction can be initiated before the previous one completes
- **Receive Interarrival Time (seconds)** – time between groups of e-mails received from the server
  - Send and receive interarrival times are independent (e.g., a client can be a frequent sender of messages but an infrequent recipient)



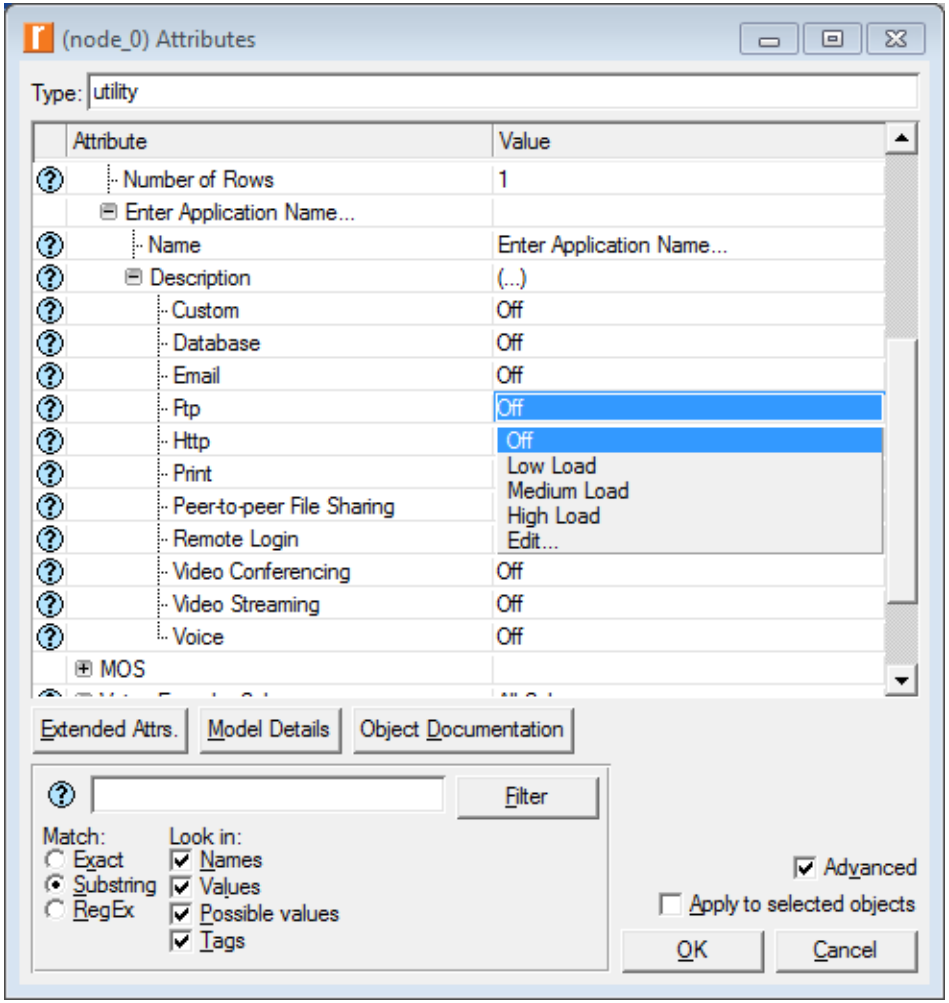
Configuring Standard Applications (cont'd)

- **Send Group Size** – number of e-mail messages per group of e-mails to be uploaded
- **Receive Group Size** – number of e-mail messages per group of e-mails to be downloaded
- **E-Mail Size (bytes)** – size of a single e-mail message



Configuring Standard Applications (cont'd)

- **Ftp** – models 2 primary FTP operations for data transfer between a client and a server: GET and PUT

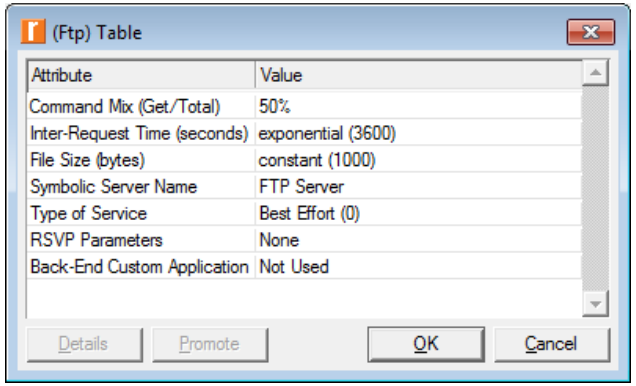


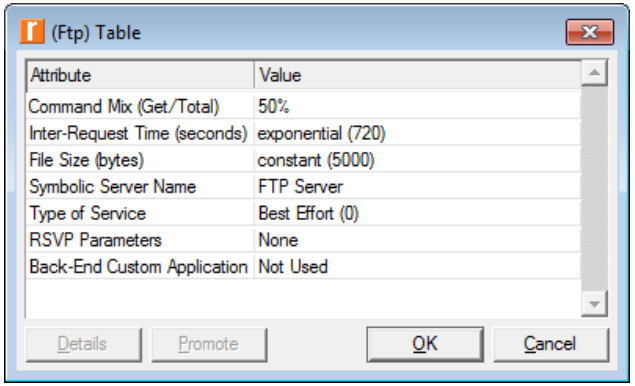
Configuring Standard Applications (cont'd)

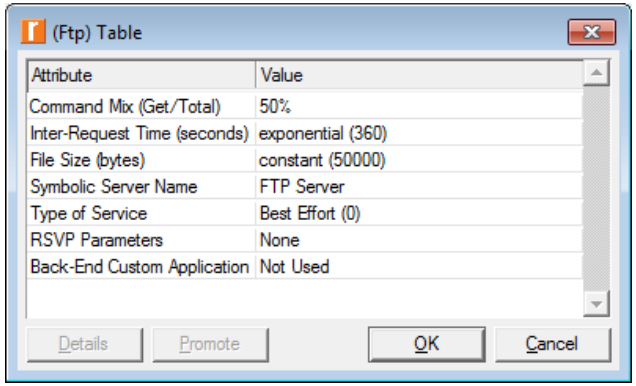
- **GET**
  1. A 512-byte file request from the client
  2. A file downloaded from the server
  3. A 8-byte close confirmation message from the server
- **PUT**
  1. A file uploaded by the client
  2. A 512-byte confirmation response from the server
  3. A 8-byte close confirmation message from the server
- Unlike in real networks, this FTP model sends control and data messages over the same transport connection
  - Each file transfer creates a new transport connection with the server
- The default transport protocol = **TCP**

Configuring Standard Applications (cont'd)

- **Low Load** vs. **Medium Load** vs. **High Load**
  - Each of the preset values is configured to have all FTP operations evenly distributed between GET and PUT operations
  - The main difference is the size of transferred files and the frequency of transactions

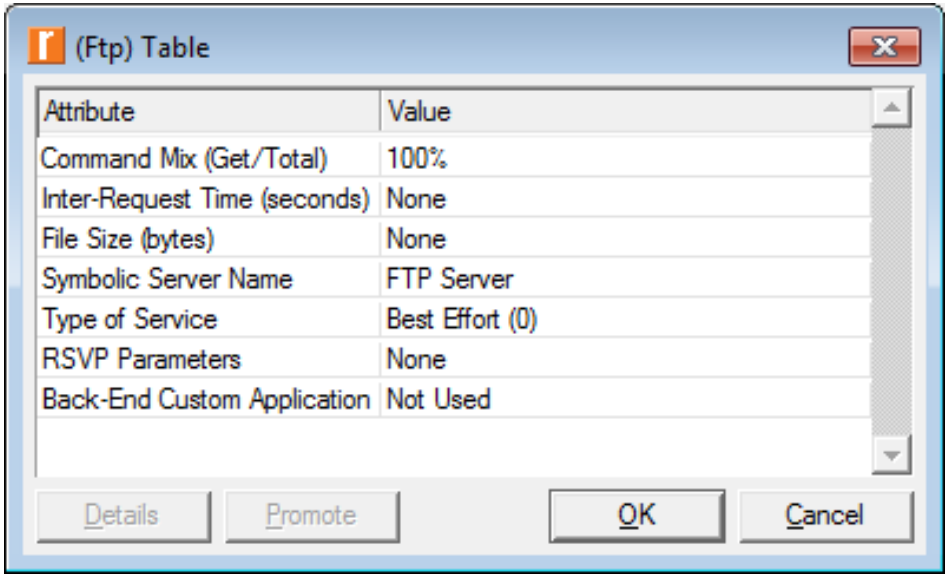






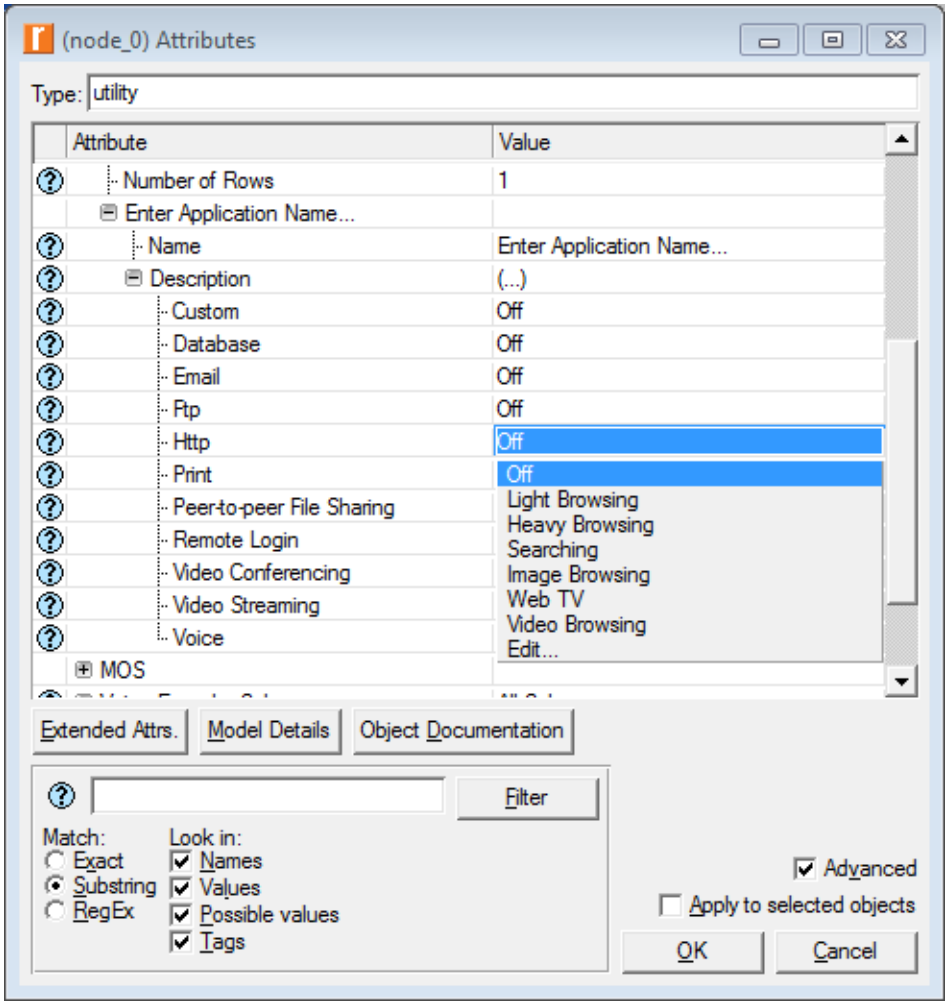
Configuring Standard Applications (cont'd)

- **Command Mix (Get/Total)** – percentage of GET operations among all FTP operations
  - 0% – all FTP operations are PUT ones
- **Inter-Request Time (seconds)** – time between consecutive FTP operations
  - The start time of the next FTP operation is computed by adding the value of this attribute to the time when the previous operation started
  - The next operation can be initiated before the previous one completes



Configuring Standard Applications (cont'd)

- **Http** – models Web browsing activity where a client periodically contacts servers to retrieve web pages



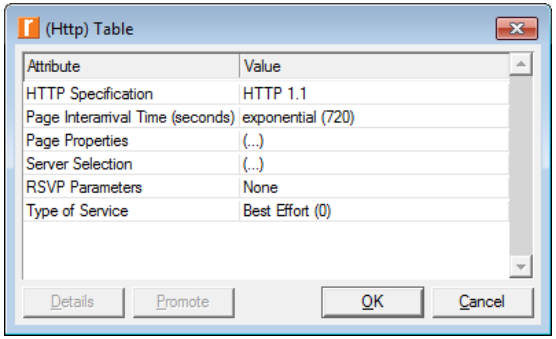
Configuring Standard Applications (cont'd)

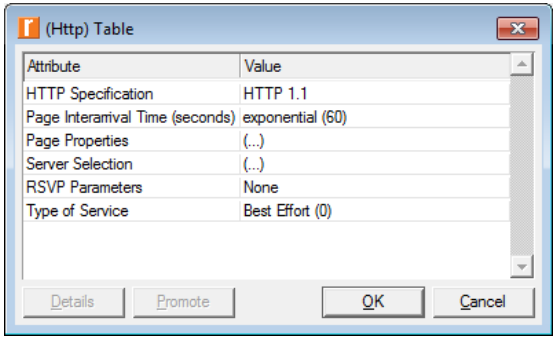
- **Web browsing**
  - The user downloads a page from a Web server
  - The page contains text and graphic information, and sometimes video
  - These page elements are collectively referred to as 'inline objects'
  - Each HTTP page request may result in opening multiple TCP connections for transferring the contents of the inline objects embedded in the page
  - The number of concurrent TCP connections is determined by the application configuration

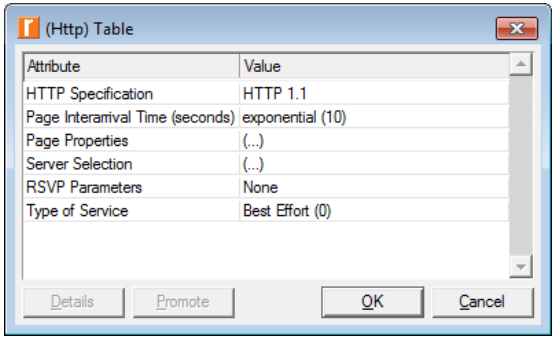
- The default transport protocol = **TCP**

Configuring Standard Applications (cont'd)

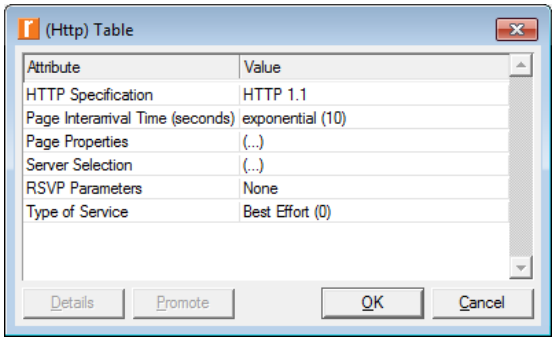
- **Light Browsing** vs. **Heavy Browsing** vs. **Searching**

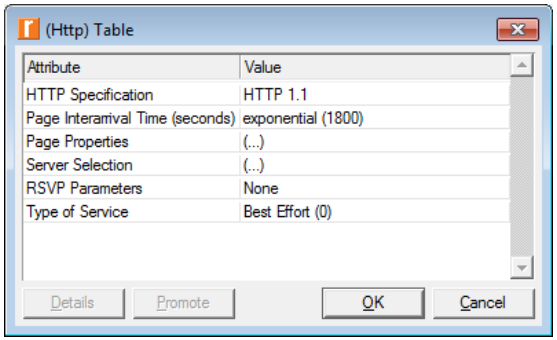


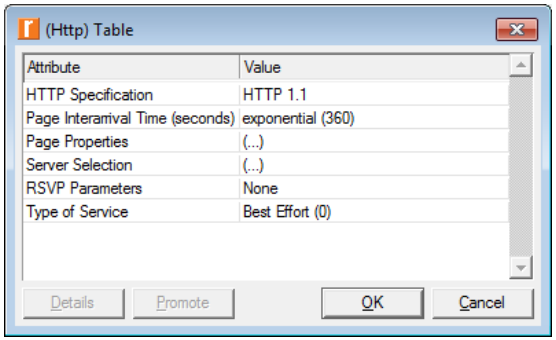




- **Image Browsing** vs. **Web TV** vs. **Video Browsing**

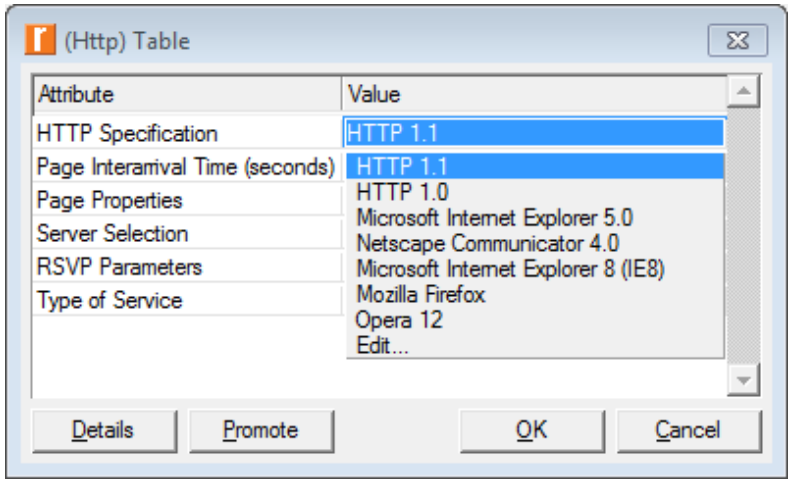


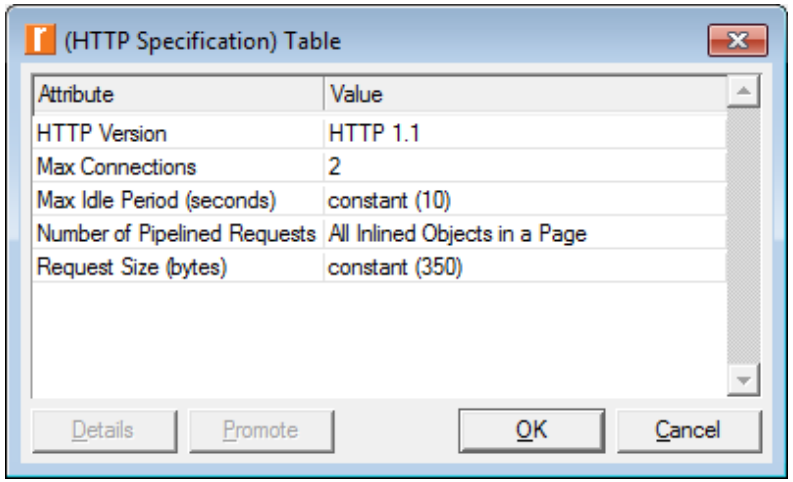




Configuring Standard Applications (cont'd)

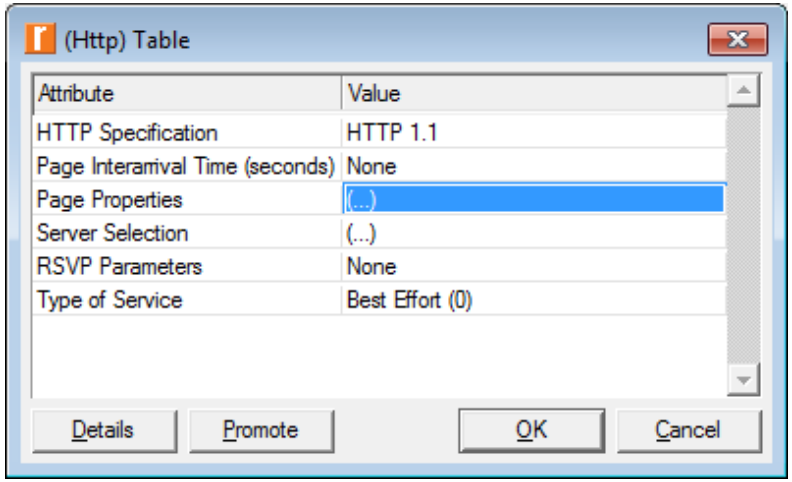
- **HTTP Specification** – how the inline objects are requested and transferred
  - HTTP 1.0 – uses a nonpersistent connection where each inline object is requested and transmitted over a separate TCP connection
  - HTTP 1.1 – persistent connections where the inline objects are transmitted over the same TCP connection
  - Pipelining – HTTP 1.1 combines multiple requests for the inline objects into a single message

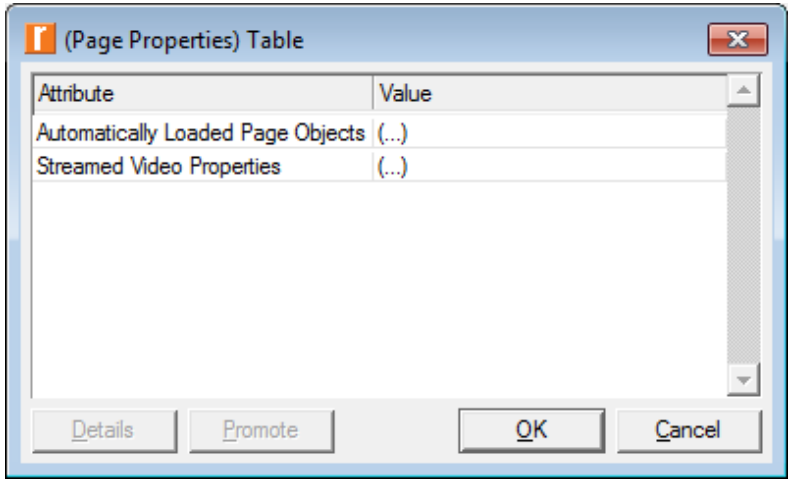




Configuring Standard Applications (cont'd)

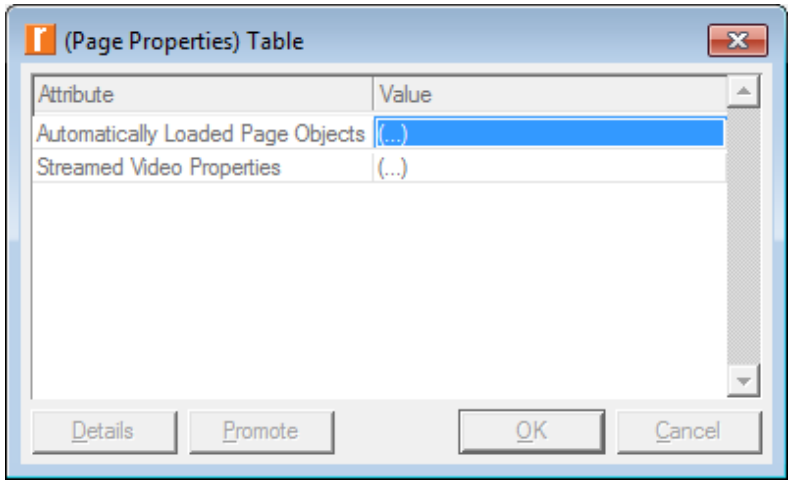
- **Page Properties** – a web page may contain many objects
  - Text, images and short videos (like ads) are loaded automatically
  - Streamed videos are loaded as soon as a user clicks 'Play'
  - One page can contain at most 1 streamed video object

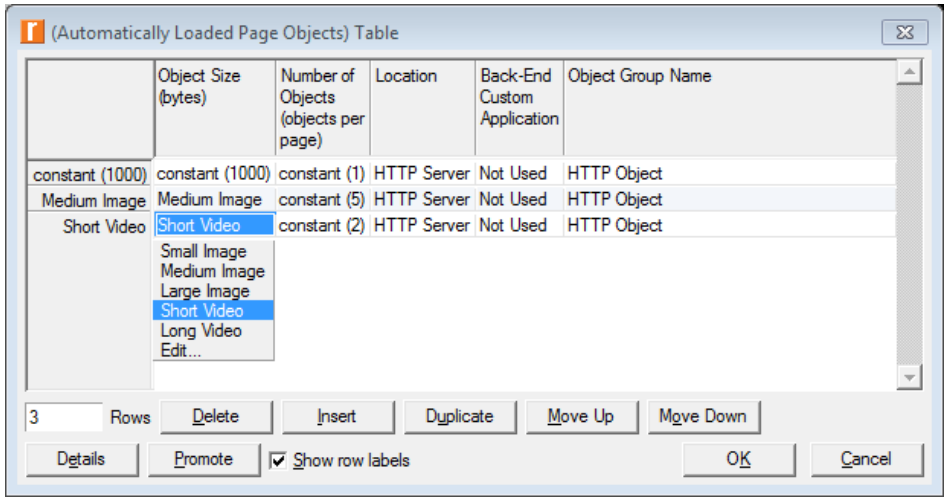




Configuring Standard Applications (cont'd)

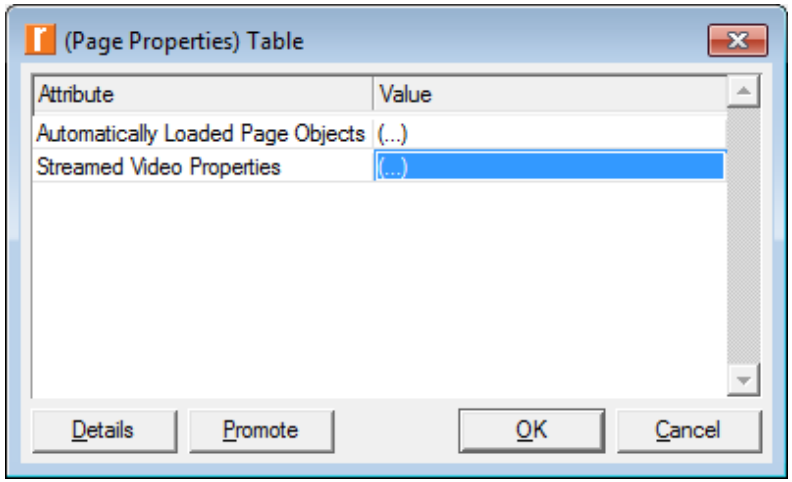
- **Automatically Loaded Page Objects**
  - Each object is represented by a row specification for this attribute
  - The first row represents the 'page' itself
  - The subsequent rows represent the objects within this page
  - The number of objects in the first row is always set to 1 internally

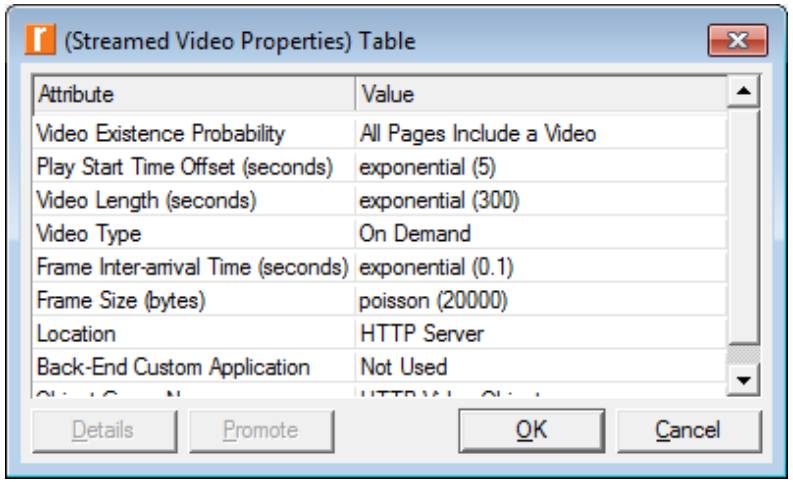




Configuring Standard Applications (cont'd)

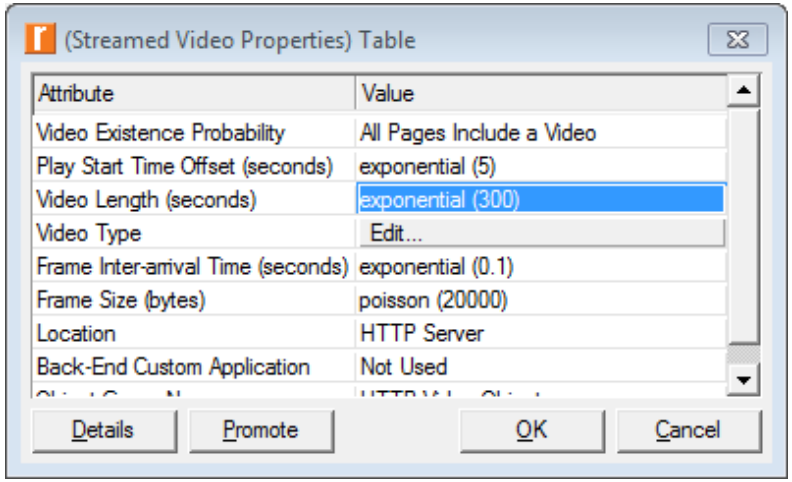
- **Streamed Video Properties**
- **Video Existence Probability** – probability that a video object is contained in the page
  - 0.0 – No Video
  - 1.0 – All Pages Include a Video
- **Play Start Time Offset (seconds)** – difference between the time that the page was loaded and the time that the user clicked 'Play'

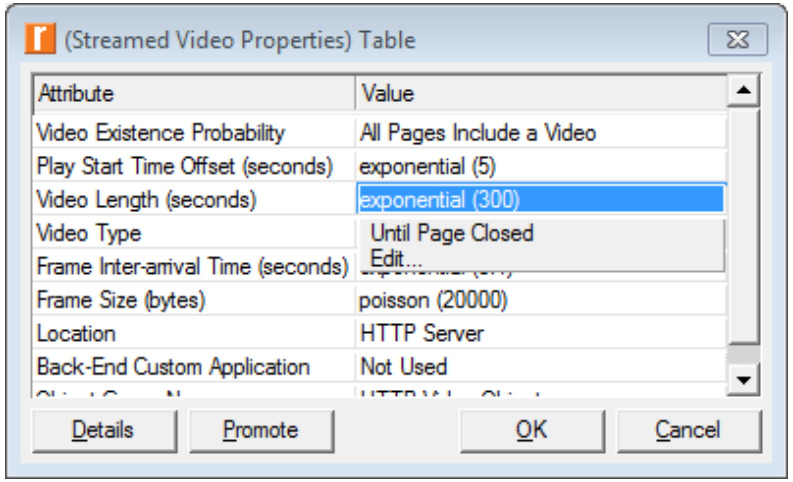




Configuring Standard Applications (cont'd)

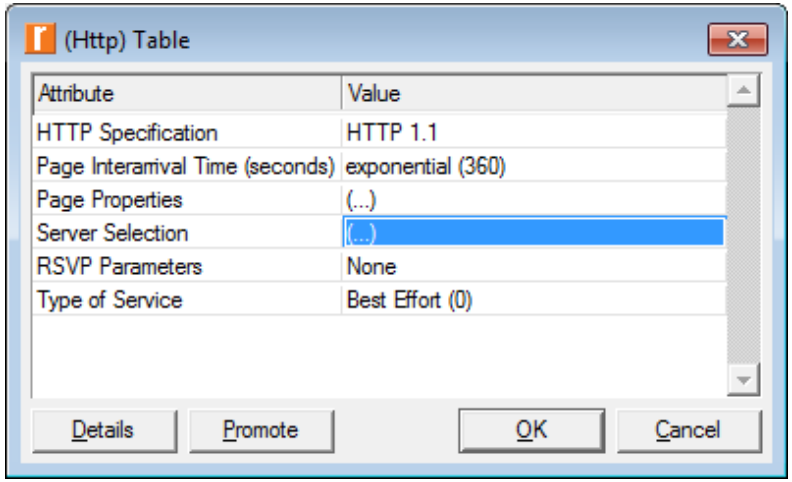
- **Video Length (seconds)** vs. **Video Type**
  - On Demand – the server sends back a file of size, in bytes, equal to = Frame Size * Video Length / Frame Inter-arrival Time
  - Live – a streaming video will be assumed over when its page is closed regardless of the length specified with the 'Video Length' attribute
  - The special value 'Until Page Closed' is accepted only for the live videos

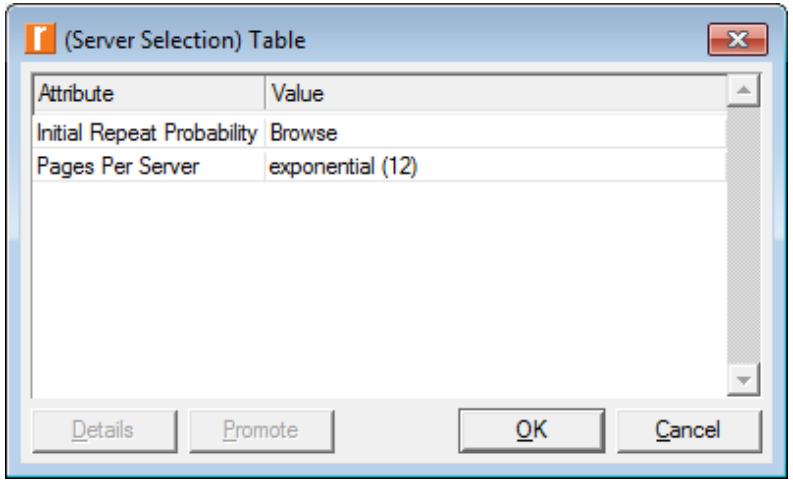




Configuring Standard Applications (cont'd)

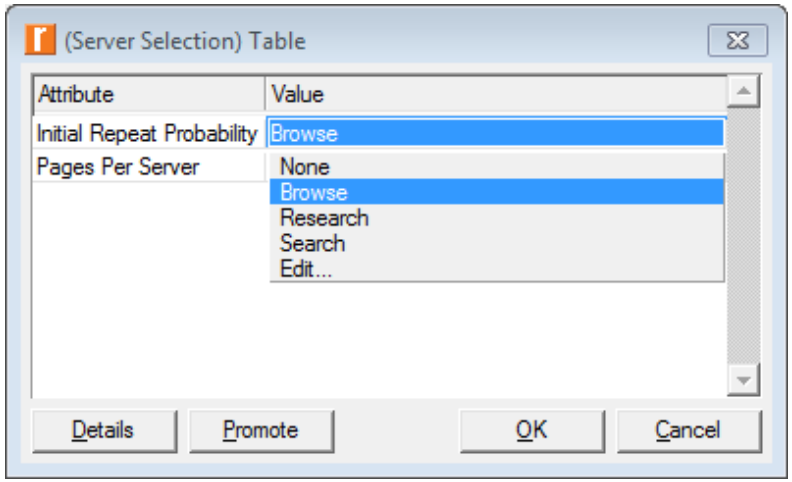
- **Server Selection**
  - Web pages often contain embedded links that point to other web pages
  - This attribute specifies if the embedded links point to web pages located on the same server or not
  - I.e., whether the web pages referenced through embedded web page links will be retrieved from the same server or from another one

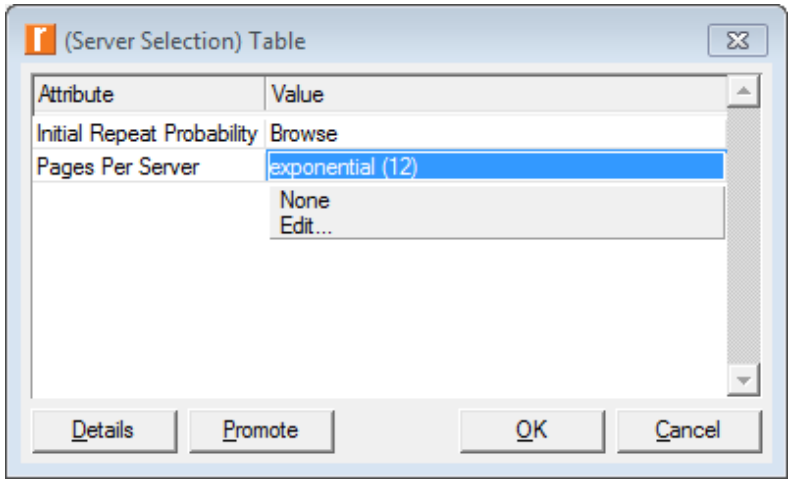




Configuring Standard Applications (cont'd)

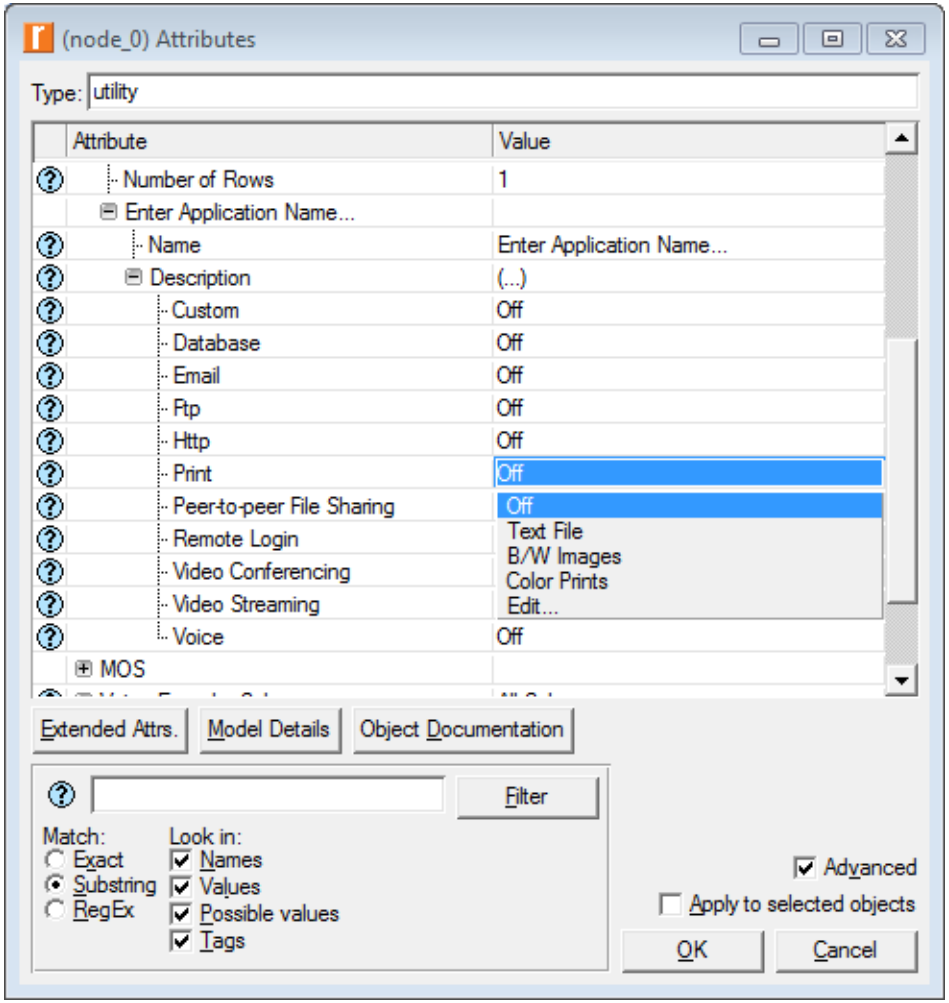
- **Initial Repeat Probability** – probability that a user would request the next page from the same server
  - 0.0 – None
  - 0.6 – Browse
  - 0.9 – Research
  - 0.3 – Search
- **Pages Per Server** – if the same server is chosen, then the client retrieves N consecutive web pages from that server
  - Otherwise, a new server is selected





Configuring Standard Applications (cont'd)

- **Print** – models the operation of submitting a printing job to a printer

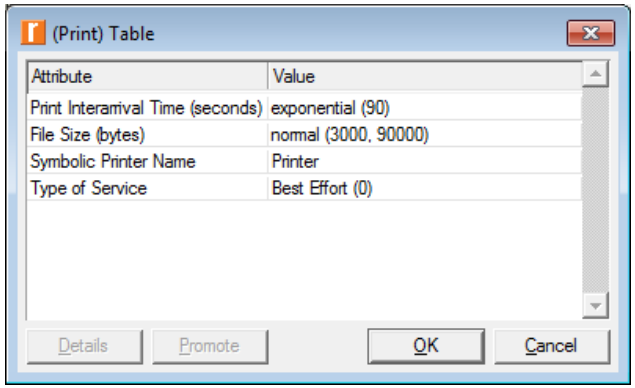


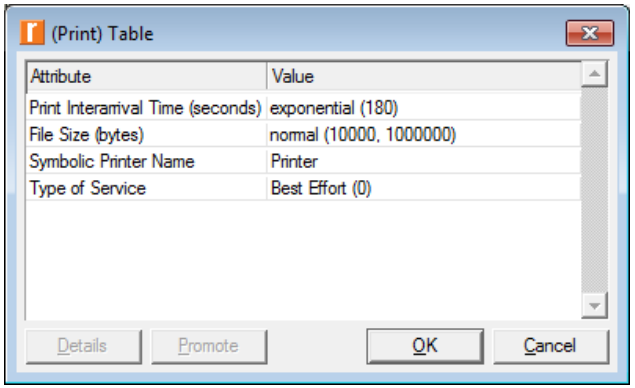
Configuring Standard Applications (cont'd)

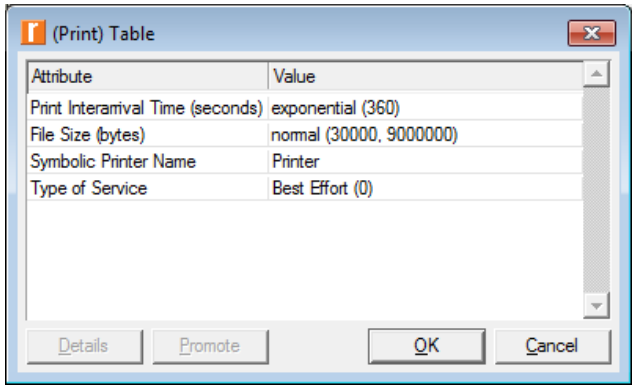
- **Submitting a printing job**
  1. A file from the client
  2. A 8-byte close confirmation message from the printer
- Each printing job creates a new transport connection with the printer
- The default transport protocol = **TCP**

Configuring Standard Applications (cont'd)

- **Text File** vs. **B/W Images** vs. **Color Prints**
  - The main difference is the size of transferred files and the frequency of printing job requests

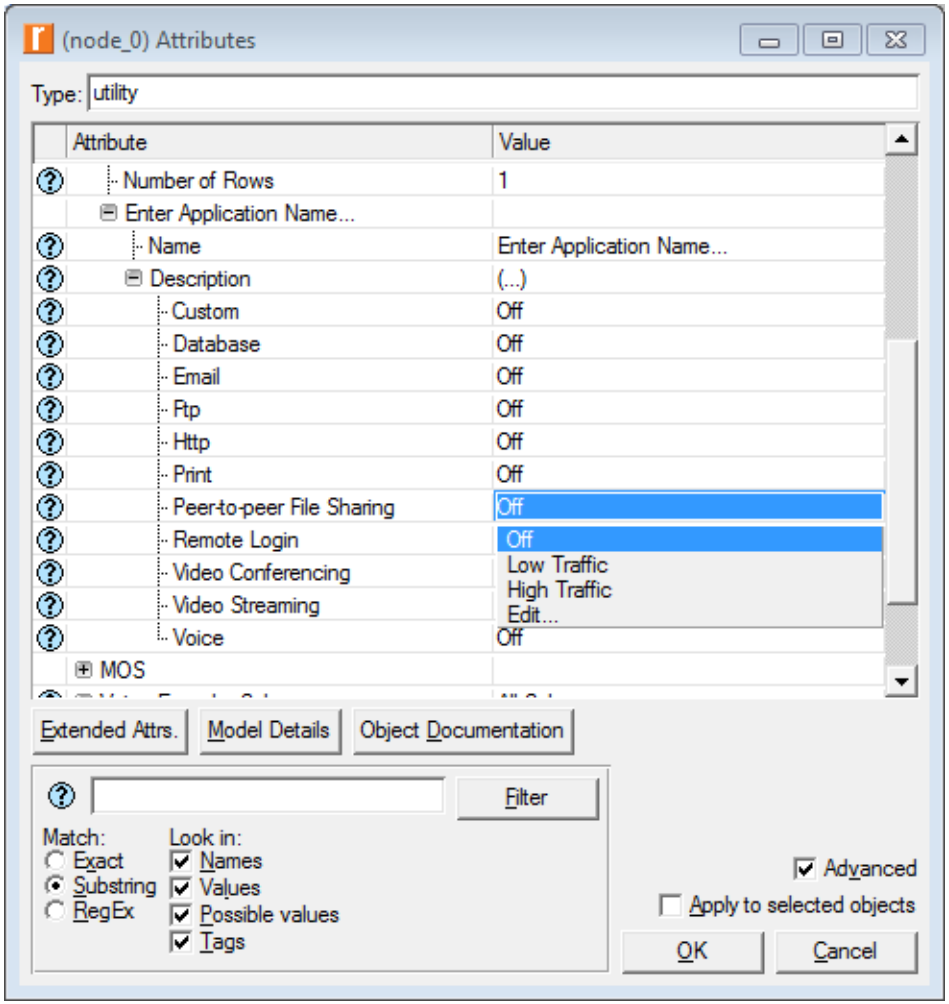






Configuring Standard Applications (cont'd)

- **Peer-to-peer File Sharing** – models a client-to-client protocol without the concept of a server



Configuring Standard Applications (cont'd)

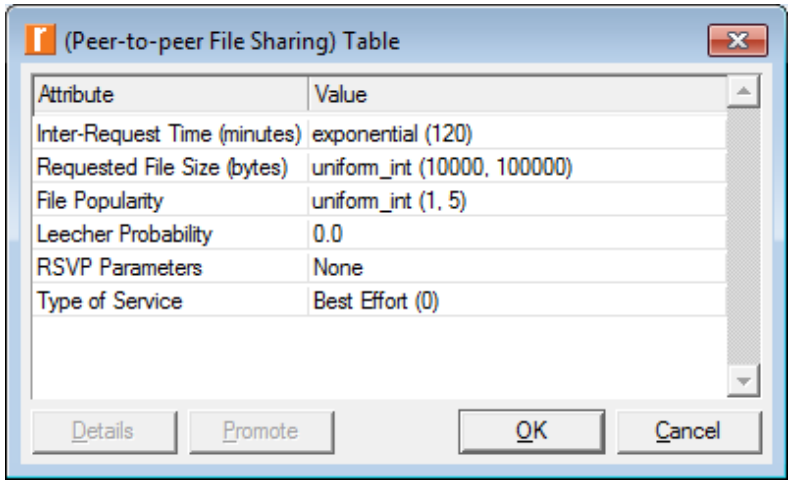
- **P2P file sharing**
  1. A 512-byte request from peer 1 to peer 2
  2. Part 1 of the requested file downloaded from peer 2
  3. A 8-byte close confirmation message from peer 2
  4. A 512-byte request from peer 1 to peer 3
  5. Part 2 of the requested file downloaded from peer 3
  6. A 8-byte close confirmation message from peer 3
  7. etc.

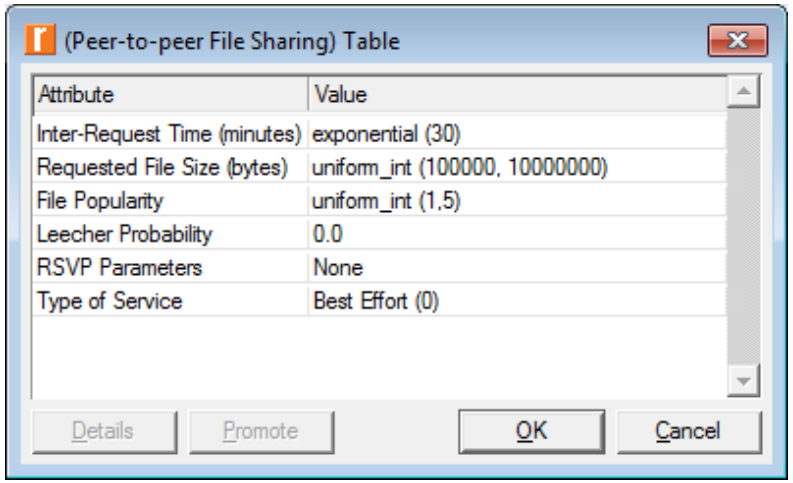
- P2P file sharing differs from HTTP and FTP in that a peer may request a portion of a file from multiple nodes rather than just one
  - Simultaneous requests for portions of a file are supported
  - Trackers are not modeled

- The default transport protocol = **TCP**

Configuring Standard Applications (cont'd)

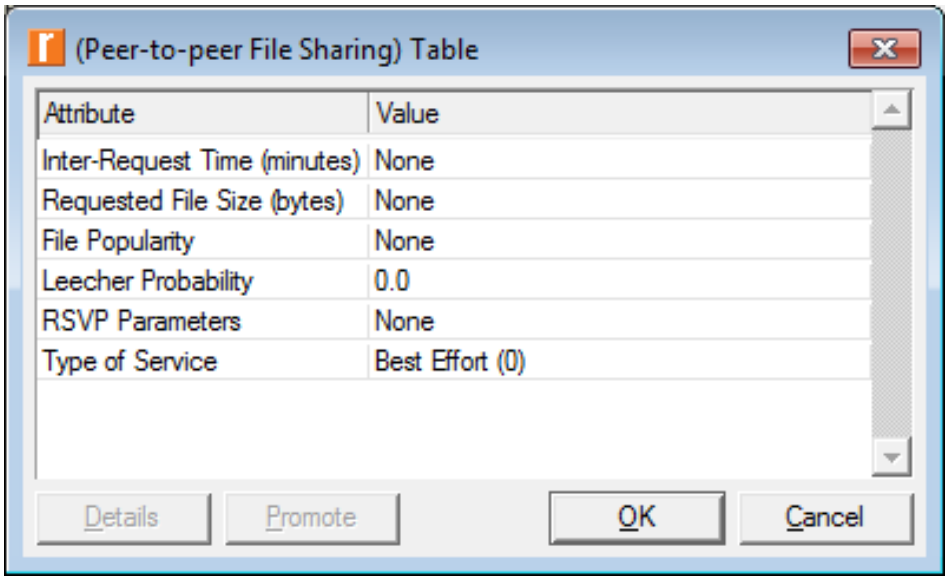
- **Low Traffic** vs. **High Traffic**
  - The main difference is the size of transferred files and the frequency of transactions





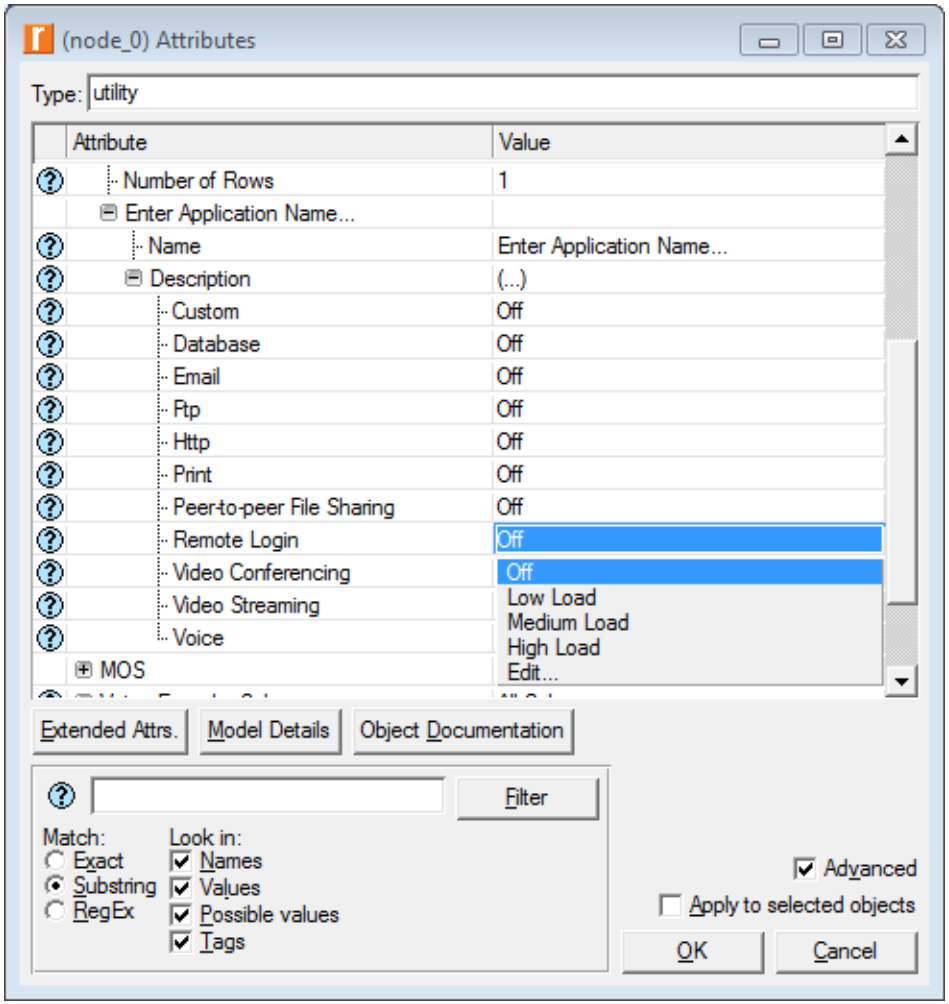
Configuring Standard Applications (cont'd)

- **File Popularity** – number of peers that have the requested file
- **Leecher Probability** – probability that a node that runs the Peer-to-peer File Sharing application will be a leecher
  - **Leecher** – a node that only downloads files from other peers and doesn't contribute any files to the P2P system



Configuring Standard Applications (cont'd)

- **Remote Login** – models a terminal service (aka Terminal Network or Telnet)



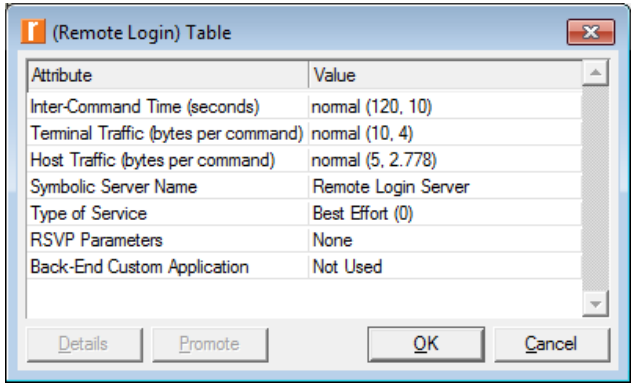
Configuring Standard Applications (cont'd)

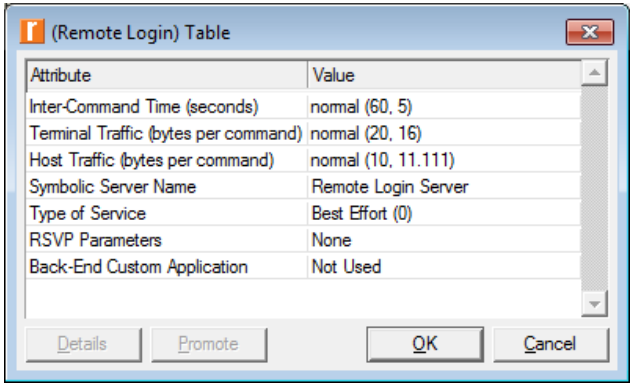
- **Remote login scenario**
  1. A command issued from the local terminal
  2. A copy of the command is sent back
  3. A response generated by the remote host
- All commands and responses are transmitted over a single transport connection
- The default transport protocol = **TCP**

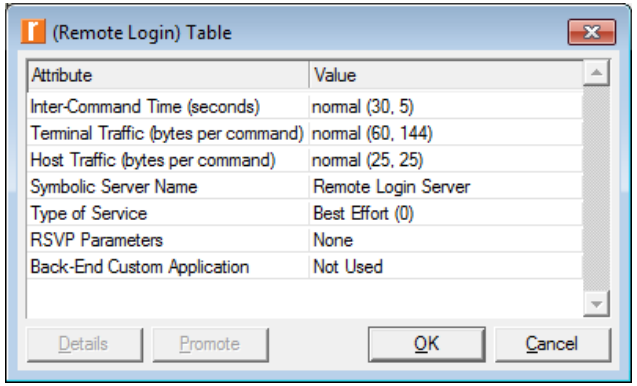
Configuring Standard Applications (cont'd)

- **Low Load** vs. **Medium Load** vs. **High Load**
  - The main difference is the size and the frequency of transactions

- **Terminal Traffic (bytes/command)** – amount of data transferred per command
- **Host Traffic (bytes/command)** – amount of data returned in response to a command

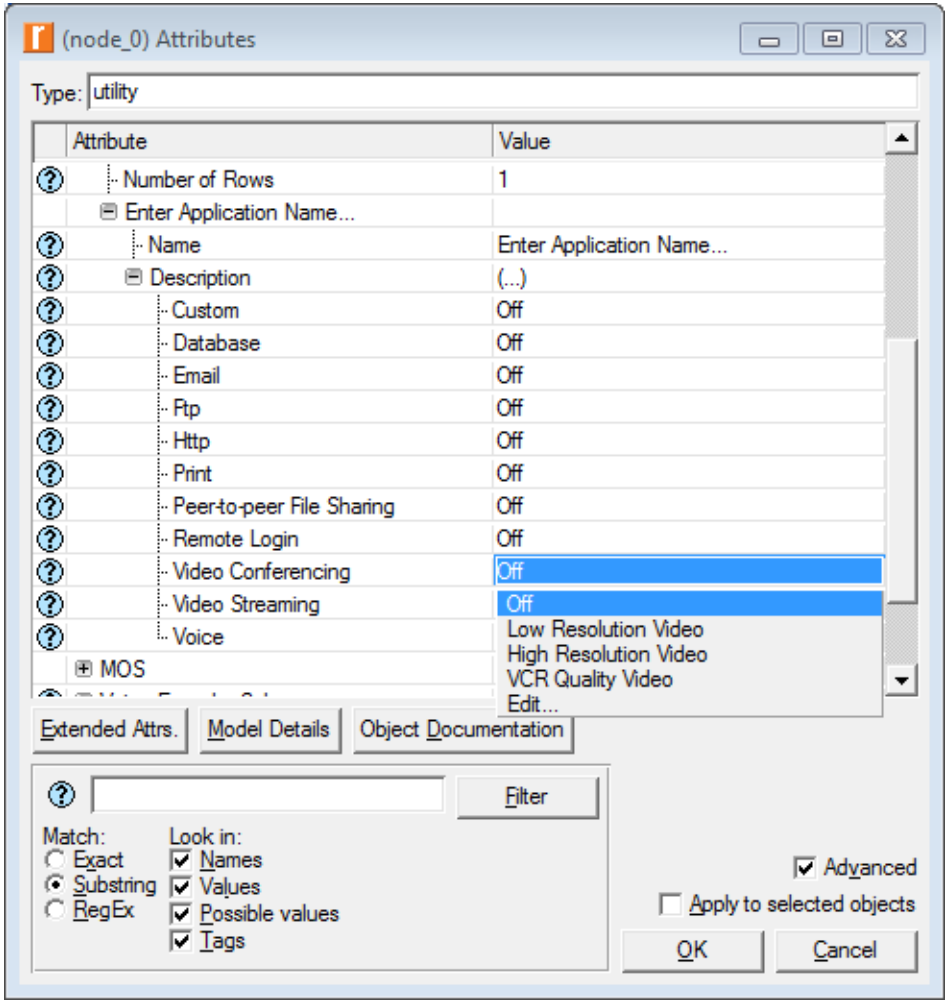






Configuring Standard Applications (cont'd)

- **Video Conferencing** – models transmission of video traffic between 2 nodes in the network



Configuring Standard Applications (cont'd)

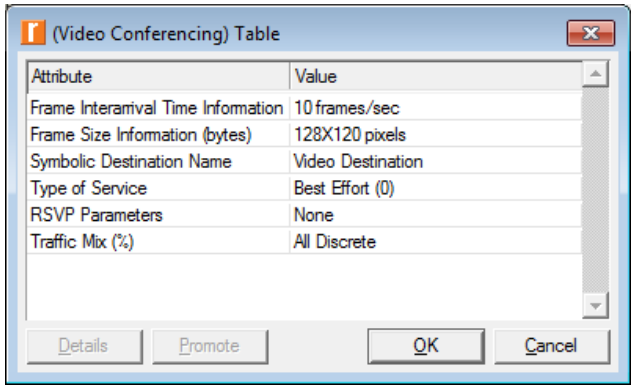
- **Video conference**
  1. A 8-byte request from the local host
  2. A 64-byte response from the remote host
  3. A one-way video stream in each direction

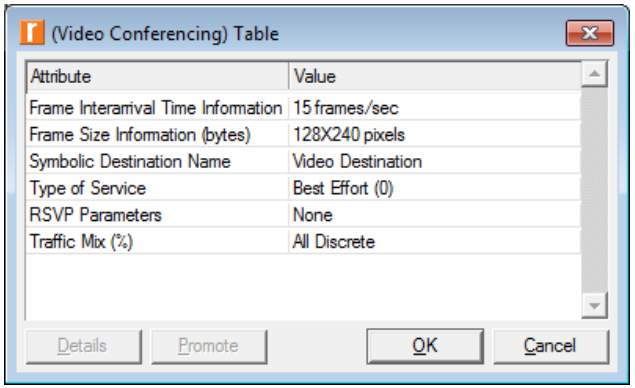
- Typically, a Video Conferencing session is established between 2 clients without the use of a server

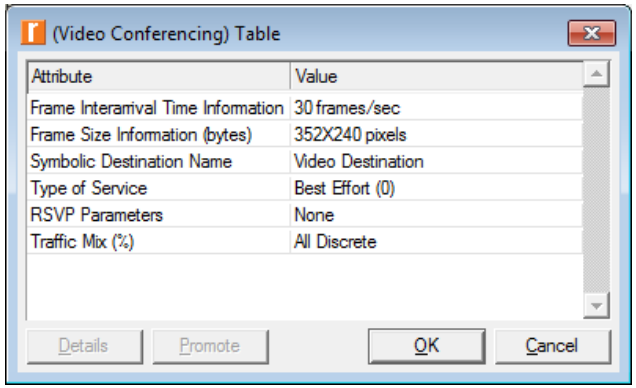
- The default transport protocol = **UDP**
  - If TCP is used as the transport protocol, each host opens an independent TCP connection

Configuring Standard Applications (cont'd)

- **Low Resolution** vs. **High Resolution** vs. **VCR Quality Video**
  - The main difference is the size of the video frames and frequency of their generation

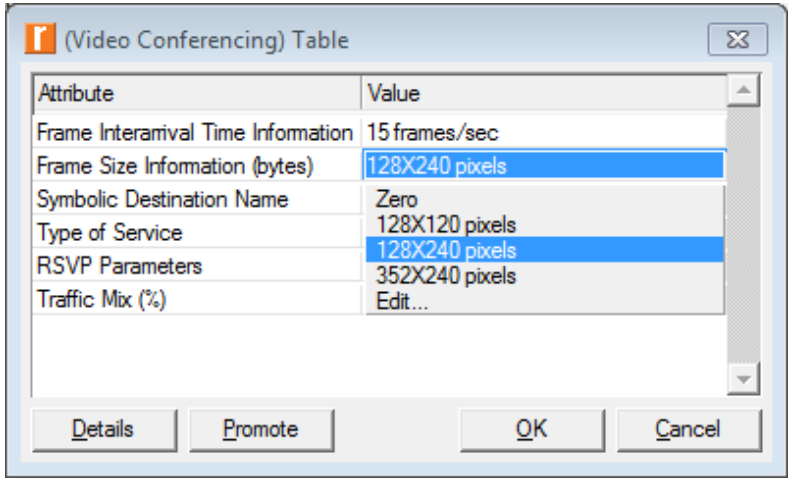


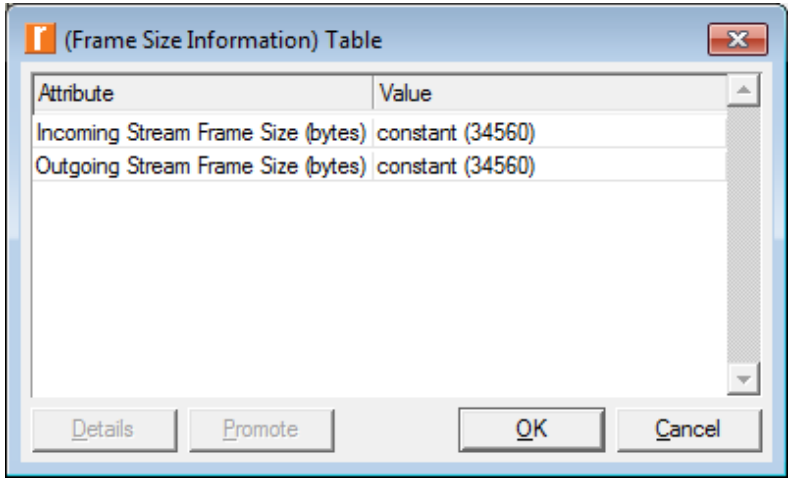




Configuring Standard Applications (cont'd)

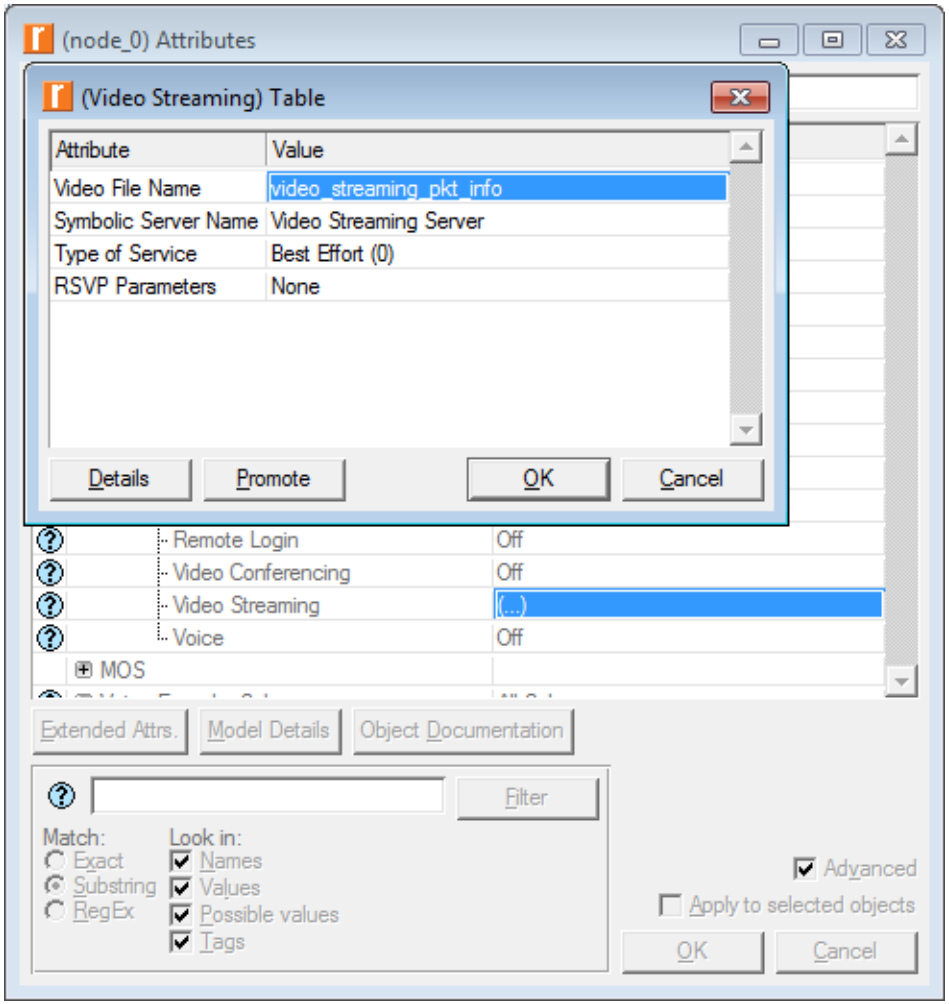
- **Frame Size Information (bytes)** – size of incoming (generated at the remote host) and outgoing (generated at the local host) frames
  - Each pixel requires 9 bits
- **Traffic Mix (%)** – whether the traffic is generated as pure discrete (explicit) or pure background or part discrete/part background
  - 0 – All Discrete
  - 100 – All Background





Configuring Standard Applications (cont'd)

- **Video Streaming** – you can capture the video traffic produced by a given video streaming application into a file, and then use this file to simulate the same traffic in a scenario



Configuring Standard Applications (cont'd)

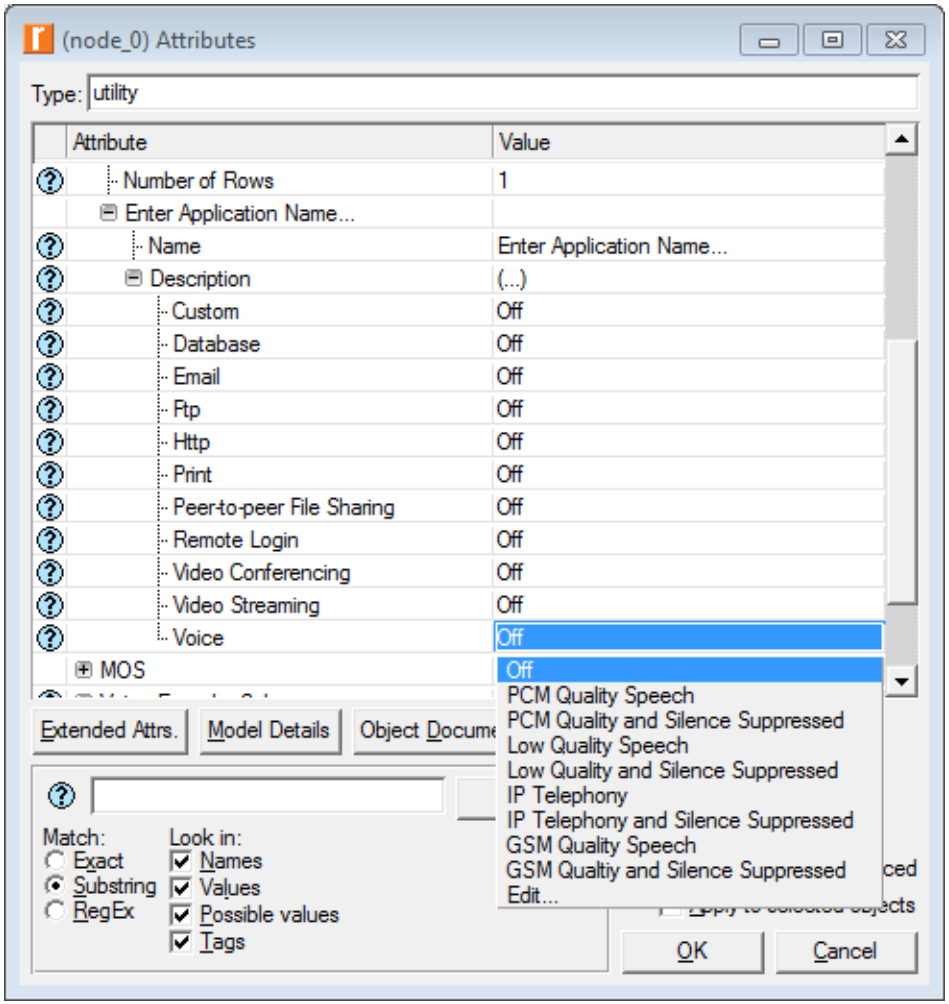
- **Video streaming**
  1. A 8-byte request from the client to the server
  2. A one-way video stream from the server
  3. A 1-byte close message from the server
  4. A 1-byte close confirmation message from the client

- A video capture file is shipped with Riverbed Modeler
  - C:/Riverbed EDU/17.5.A/models/std/example_networks/Application_Configuration_Example.project/video_streaming.gdf
  - The first row specifies the size of the packets
  - The remaining rows specify the packet interarrival times in microseconds

- The default transport protocol = **UDP**

Configuring Standard Applications (cont'd)

- **Voice** – models network communication between 2 clients using a digitized voice signal

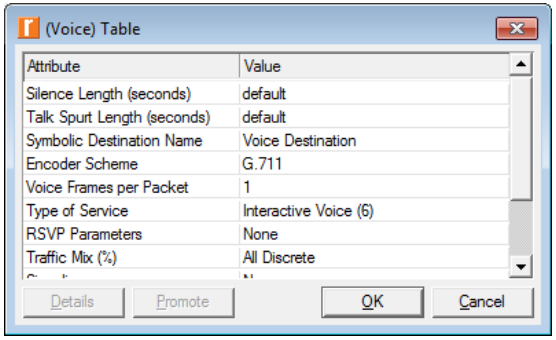


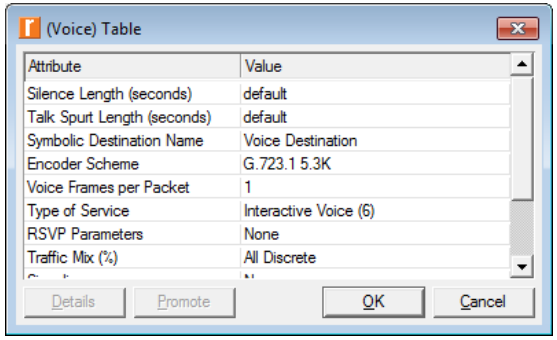
Configuring Standard Applications (cont'd)

- Typically, the total time it takes for the voice signal to travel from one caller to another (aka mouth-to-ear delay) consists of the time it takes to:
  - Encode and compress the voice data on one end
  - Packetize the encoded and compressed voice data
  - Transmit the voice data packet through the network
  - Decompress the received voice data
  - Decode it for playback at the other end
- A voice call is established from one client to another, no server is modeled for voice conversations
- The default transport protocol = **UDP**

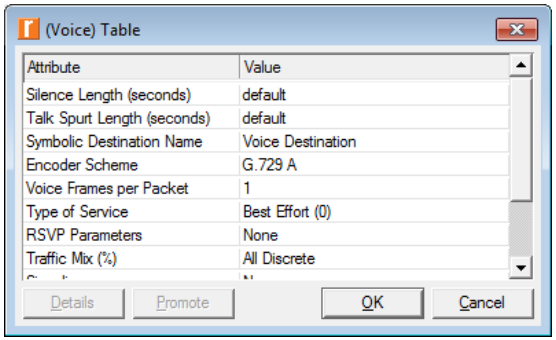
Configuring Standard Applications (cont'd)

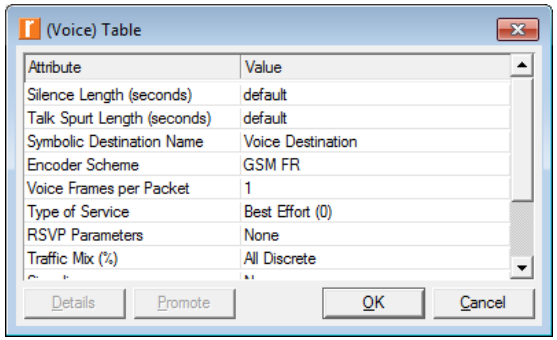
- The main difference is the type of voice encoding scheme and the ToS
- **PCM Quality Speech** vs. **Low-Quality Speech**





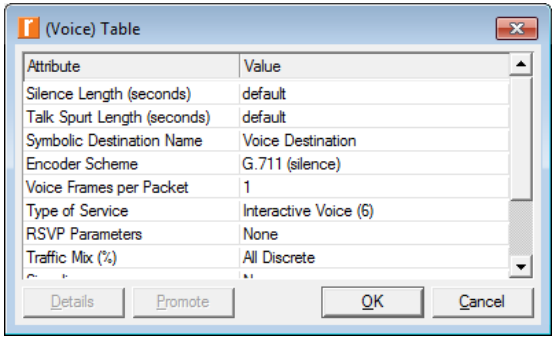
- **IP Telephony** vs. **GSM Quality Speech**

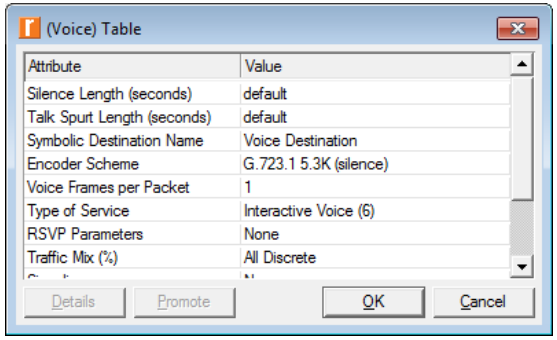




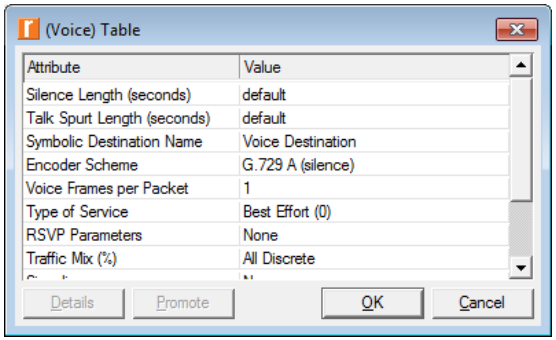
Configuring Standard Applications (cont'd)

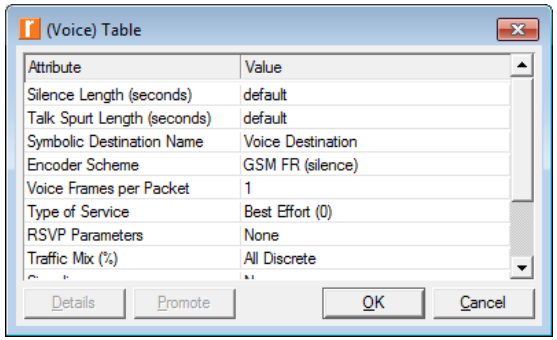
- The voice data arrives in spurts that are followed by a silence period
- **PCM... and Silence Suppressed** vs **Low... and Silence Suppressed**





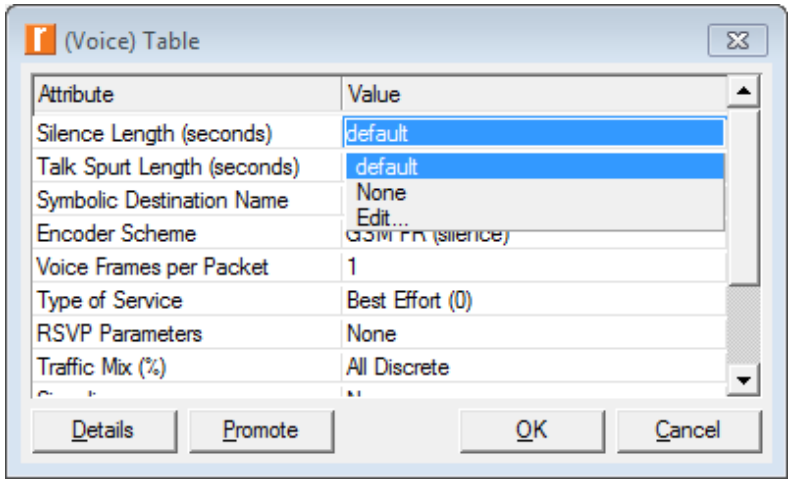
- **IP... and Silence Suppressed** vs **GSM... and Silence Suppressed**

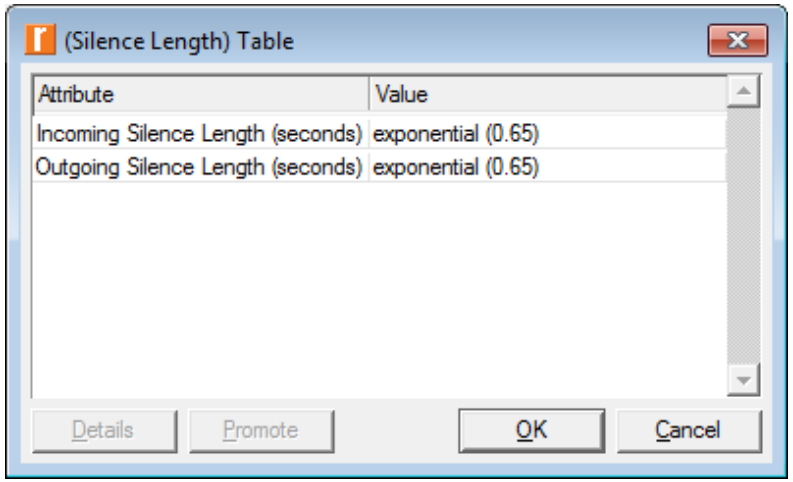




Configuring Standard Applications (cont'd)

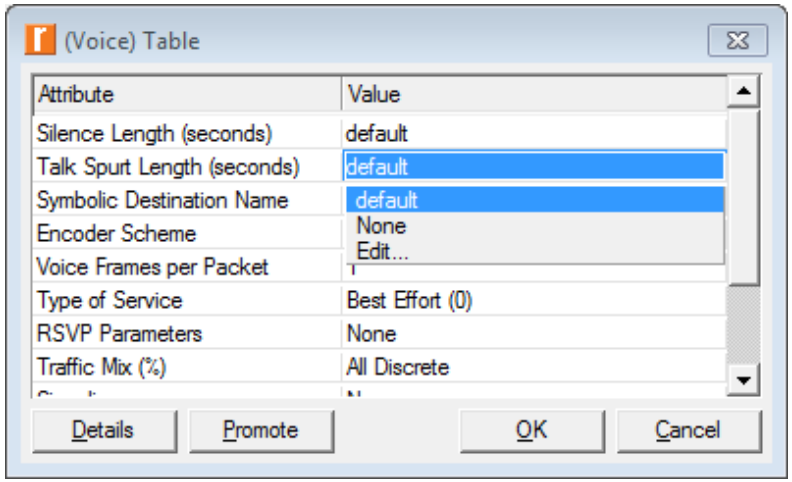
- **Silence Length (seconds)** – time spent by the called party (incoming) and the calling party (outgoing) in silence mode in a speech-silence cycle
  - The incoming and the outgoing periods of silence are independent of each other
  - I.e., it is possible for both clients to talk or to be silent simultaneously, or for one side of the call to talk while the other side is silent

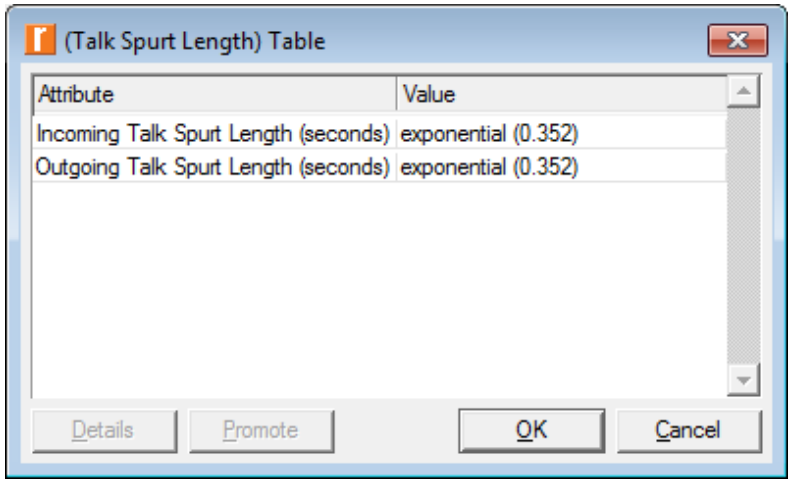




Configuring Standard Applications (cont'd)

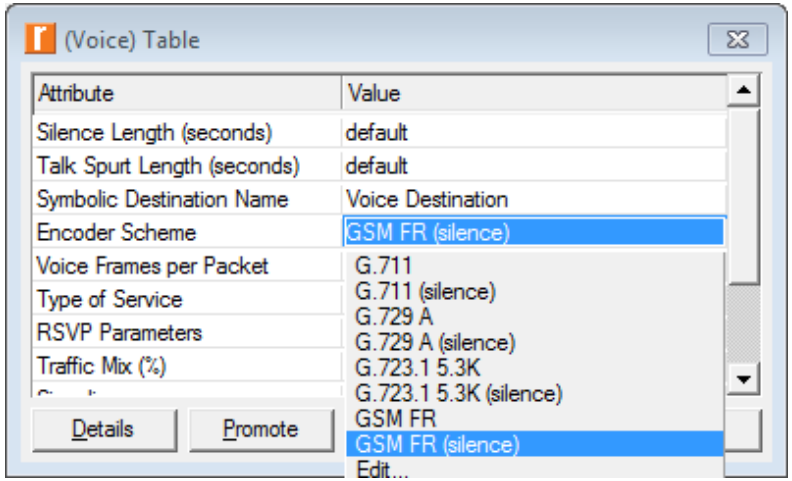
- **Talk Spurt Length (seconds)** – time spent by the called party (incoming) and the calling party (outgoing) in speech mode in a speech-silence cycle
  - The incoming and the outgoing periods of talk spurts are independent of each other
  - I.e., it is possible for both clients to talk or to be silent simultaneously, or for one side of the call to talk while the other side is silent

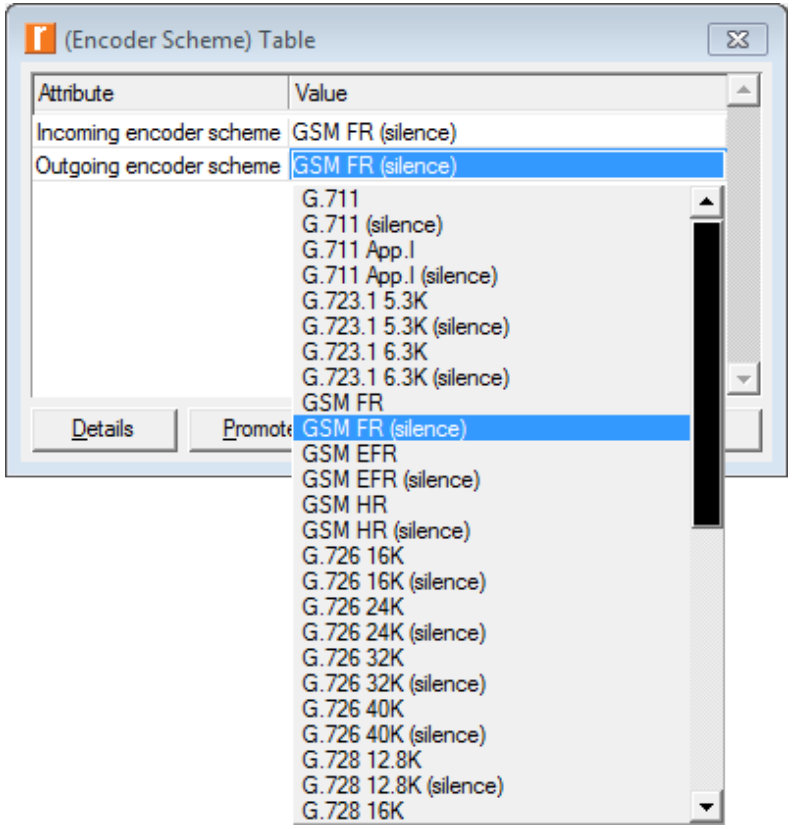




Configuring Standard Applications (cont'd)

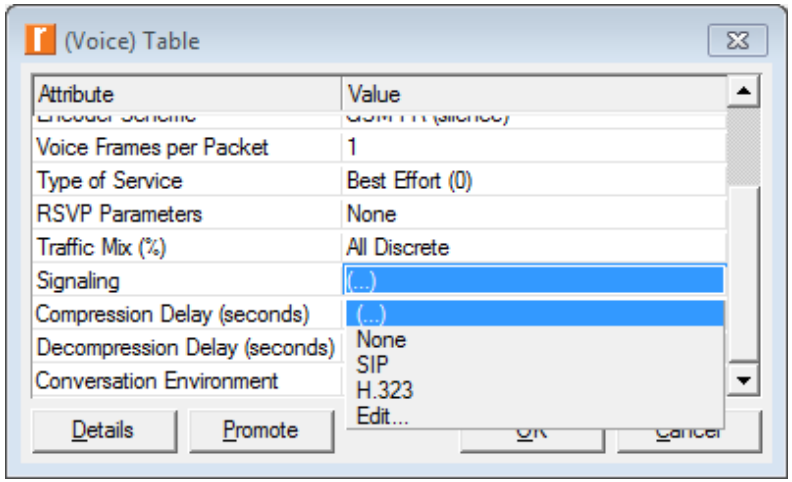
- **Encoder Scheme** – algorithm for encoding voice into a digital signal to be used by the calling and called parties

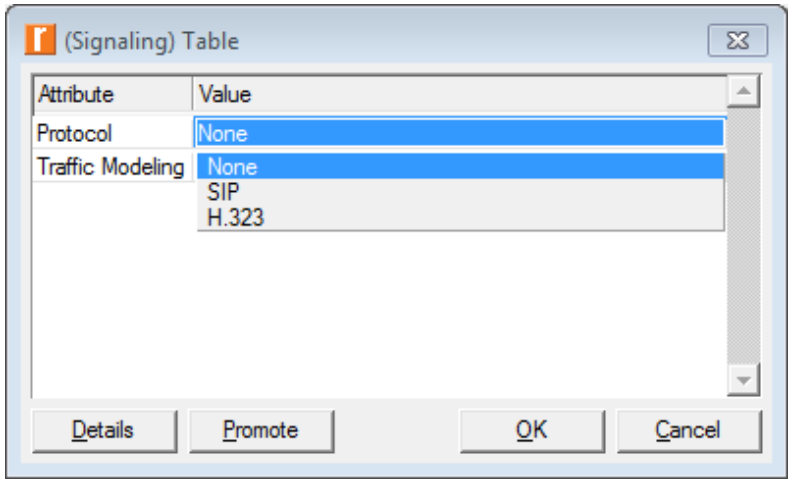




Configuring Standard Applications (cont'd)

- **Signaling** – method for establishing and tearing down a voice call
- **Protocol** – signaling protocol for setting up and releasing the voice connection
  - None – no signaling and the calls will start and end without any setup/release procedures
  - SIP – will be used for signaling
  - H.323 – will be used for signaling

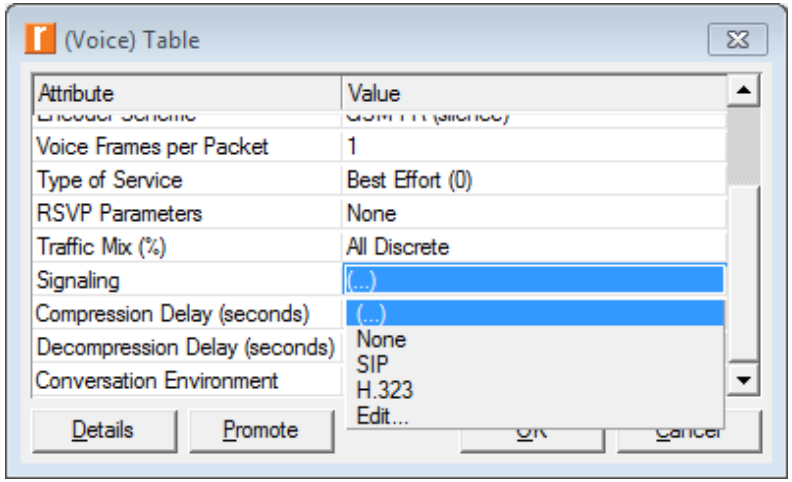


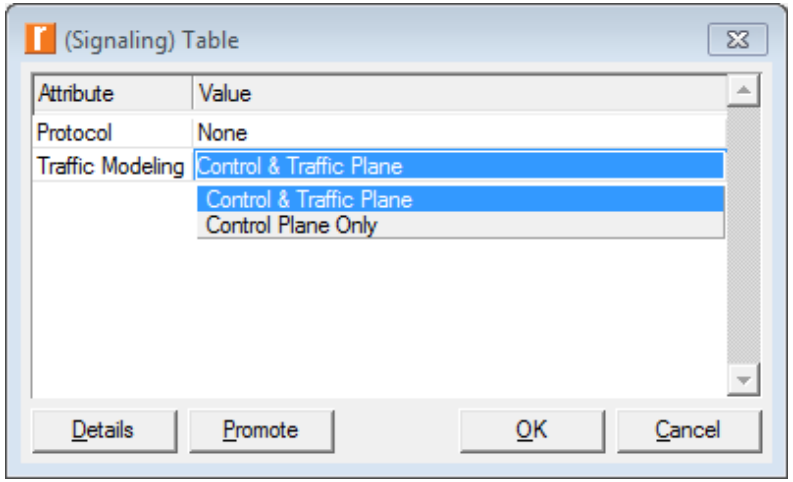


Configuring Standard Applications (cont'd)

- **Traffic Modeling** – what portion of the voice traffic will be modeled
  - Control Plane – flow of packets associated with the signaling protocol
  - Traffic Plane – flow of packets associated with the application data

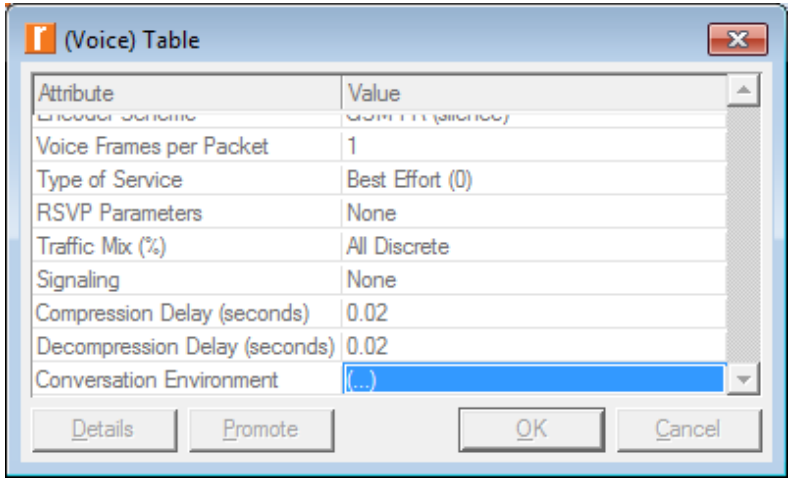
- If 'Control Plane Only' is selected, no application traffic will be simulated from the time the call is setup to the time it is released

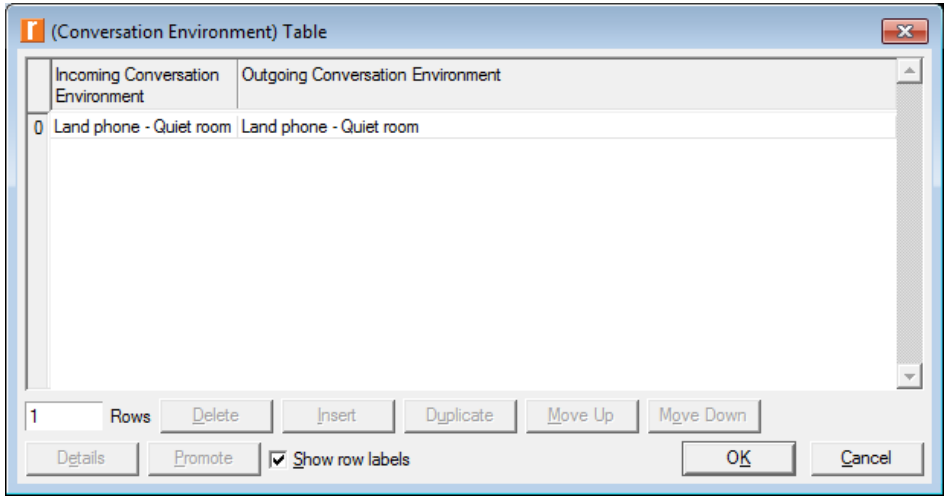




Configuring Standard Applications (cont'd)

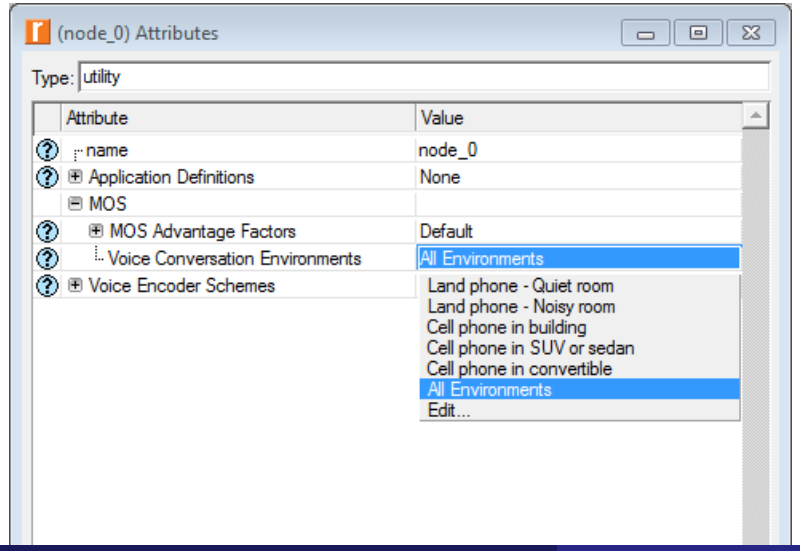
- **Conversation Environment** – incoming and outgoing conversation environments for the purpose of **Mean Opinion Score (MOS)** estimation
  - Land phone – Quiet room
  - Land phone – Noisy room
  - Cell phone in building
  - Cell phone in SUV (Sport Utility Vehicle) or sedan
  - Cell phone in convertible

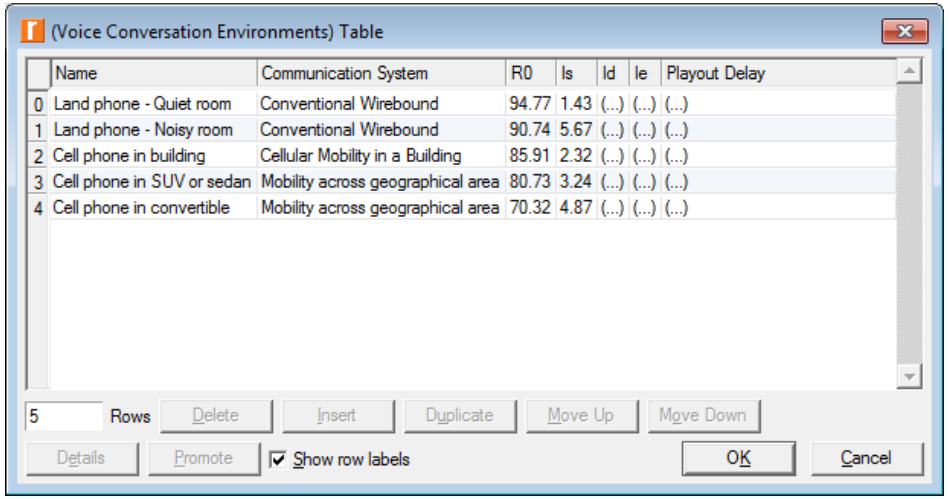




Configuring Standard Applications (cont'd)

- ITU-T G.107 'The E-model: a computational model for use in transmission planning'
- $R = Ro - Is - Id - Ie + A$, $0 \leqslant R \leqslant 100$
- Ro – basic signal-to-noise ratio (SNR)
- Is – sum of all impairments due to voice transmission
- Id – impairment due to the delay
- Ie – impairment due to packet loss
- www.itu.int/ITU-T/studygroups/com12/emodelv1/index.htm





Configuring Standard Applications (cont'd)

- **Voice Encoder Schemes** – voice encoding schemes are defined through attributes such as codec type, frame size, etc.