

WHAT'S INSIDE

- 1 Overview
- 2 The Problem
- 3 Testing with Real Networks
- 4 Network Emulation
- 5 iTrinegy

INTRODUCTION

Gaming has become more and more reliant on a constant, high-quality internet connection to provide players with better gameplay experiences. Whether it's maintaining a high-calibre online multiplayer experience, ensuring a mobile game can handle flitting between different service providers and network qualities, uploading a campaign save file to the cloud, or releasing patches to resolve live issues, nearly all facets of modern gaming feature a level of dependence on maintaining a steady internet connection.

As a consequence, a game's ability to gracefully deal with a wide range of different network conditions is absolutely key to the player's experience, and the game's success.

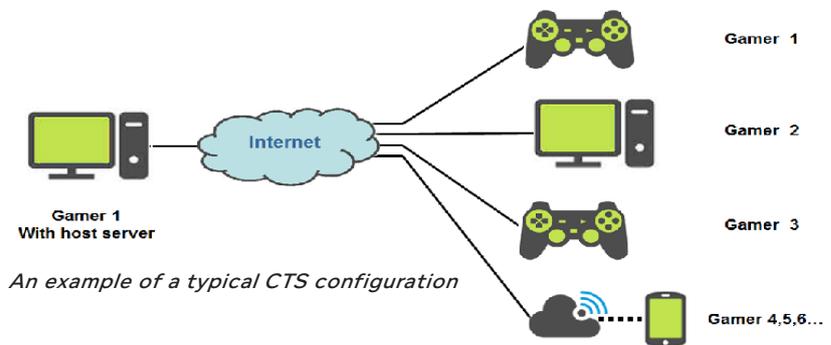
A poor gaming experience due to Internet issues can have dramatic consequences. Any game developer knows just how quickly players will act to publicise bad experiences on all platforms, and understand these reviews can generate a poor reputation which may not reflect the overall game, but can significantly put off new players. As a consequence, it's in a studio's best interest to test games against different network types whilst the game is still in development.

However, this can prove to be particularly difficult in practice.

OVERVIEW

Online games typically operate in one of two ways:

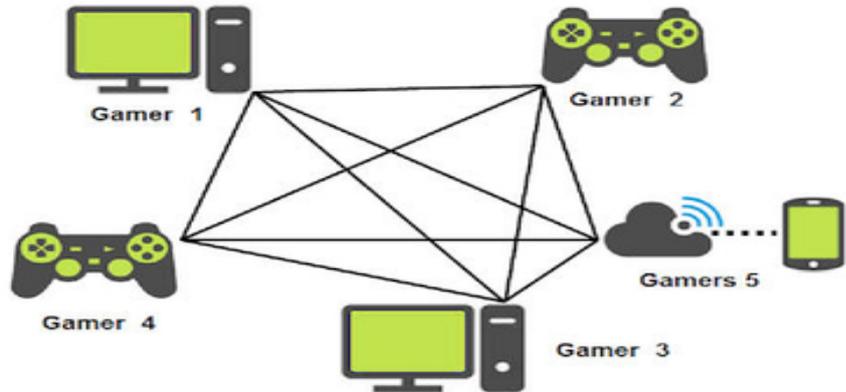
**Client to Server (CTS):** Where the server is hosted by either the game provider, or, for some games, the game is hosted by a player, to whom the other players connect to in order to play.



CTS configurations involve simple, easier connections to establish and maintain, with typically only one connection link to consider- that being the link between player and server. Most multiplayer games operate this way on console, pc, and mobile game formats. These configurations are reliable and established within the industry- however, they rely heavily on all players having a strong, stable connection for smooth gameplay. One player having a sub-par connection can lead to disruptive gameplay, unfair advantages, or a host of game lobby errors.



**Peer to Peer (PTP):** Where the game is a multiplayer scenario and the game consoles are synchronized; a player issues a command, and this is executed in exactly the same way, at the same time, on the machines of other players.



*An example of a PTP network configuration*

PTP configurations are much more complex, representing scenarios where some or all of the components in the network are constantly interacting with one-another; these are called Meshed networks. These configurations, whilst offering greater sophistication and allowing for more specialised and widespread topologies, can quickly become very complicated, in which the number of 'links' between clients can grow at an alarming rate.

Both of these configurations will rely on strong connections between clients, often with great distances between each other once the application is live, to maintain a high quality of service. In addition to these two differentials, there are many ways in which unreliable networks can provoke issues that will severely hinder game performance under live, real-world networks; latency bursts, 'lag', packet-reordering and packet loss are just a few of the impairments which, if severe enough, can severely damage the gameplay experience.

## TESTING WITH REAL NETWORKS

Both CTS and PTP configurations are essential for multiplayer services, in-game marketplaces, campaign saves and more, and both can be severely impacted by network issues such as lag, packet loss, and rubber banding if left unchecked in development. As a consequence, optimizing games against bad networks is essential to stop bugs ruining a game's launch. However, trying to configure even the most basic CTS networks on command-line formats can prove to be very difficult, whilst simply testing over a live network gives you very little insight into what network impairment thresholds will stop your game from working.

Listed below are a few of the major issues that unassisted testers frequently face.

### **Configuring the right network type for testing can be difficult**

It's difficult and time consuming to introduce multiple network types into a test environment manually, and even harder to have this network contain tailored transmission issues on demand. This is especially true of mobile gaming, where the networks are always changing. Seeing how a game handles flitting between WiFi and 4G, or changing from 4G to 3G connections, are important considerations to make, yet can be extremely difficult to configure manually.

### **Real Networks aren't consistent**

Under the stresses of a modern village, town, or city, the network quality can fluctuate massively, meaning that testing only on a perfect, lab-quality network would not give you an appropriate understanding of how your game will perform in the hands of players using real-world networks.

### Manual tests can't provide repeatable results

As the live mobile networks in cities and towns are constantly changing, taking a game in development out into the streets to test against poor networks will only provide unclear or unreliable data. Due to the erratic nature of internet networks, it can be very unclear as to what network conditions you are experiencing, and what conditions are causing game-breaking errors. Identifying an error whilst testing under these conditions leaves you with the knowledge that something is wrong, but with absolutely no indicators as to what really caused it- you're not any closer to fixing a potentially harmful bug that you now know exists within your game.

## NETWORK EMULATION

iTrinegy Network Emulators allow you to create different network types, and network impairments, in a controllable, repeatable environment. Simply connect your test consoles to the emulator, connect the emulator to a strong internet connection, enter the parameters you'd like to test against, and begin the emulation. After these have been achieved, your game will begin to experience customised levels of:

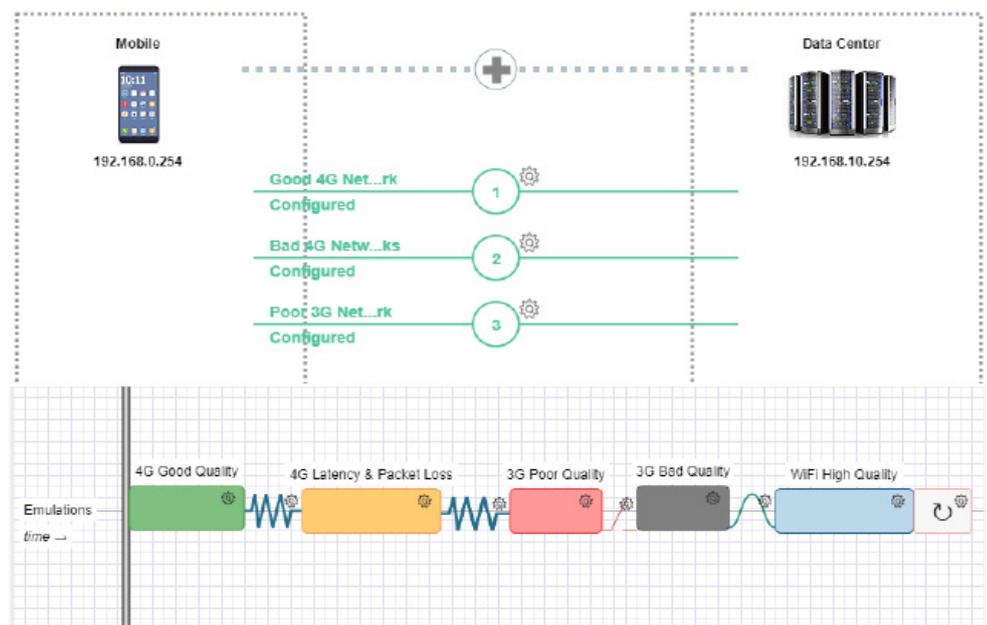
- Latency
- Packet loss
- Bandwidth congestion
- Packet Reordering

and more, in a controllable, repeatable environment.

It's just that simple.

In addition to the huge range of customizable network impairments available, iTrinegy NE-ONE emulators allow you to programme and test up to 20 different point-to-point network experiences at any one time, and even more with the INE! This means that you can blitz through different parameters, operate multiple tests on troublesome conditions, or 'brute force' test your game against the harshest networks available to see how your game will handle the worst network stresses.

Both ranges of emulators come with an extensive array of further tools and features created with feedback from our existing gaming clients- too many to go through properly here! However, an breakdown of some of our popular features of the NE-ONE emulator range can be found [at our '5 reasons why developers use iTrinegy' press release!](#)



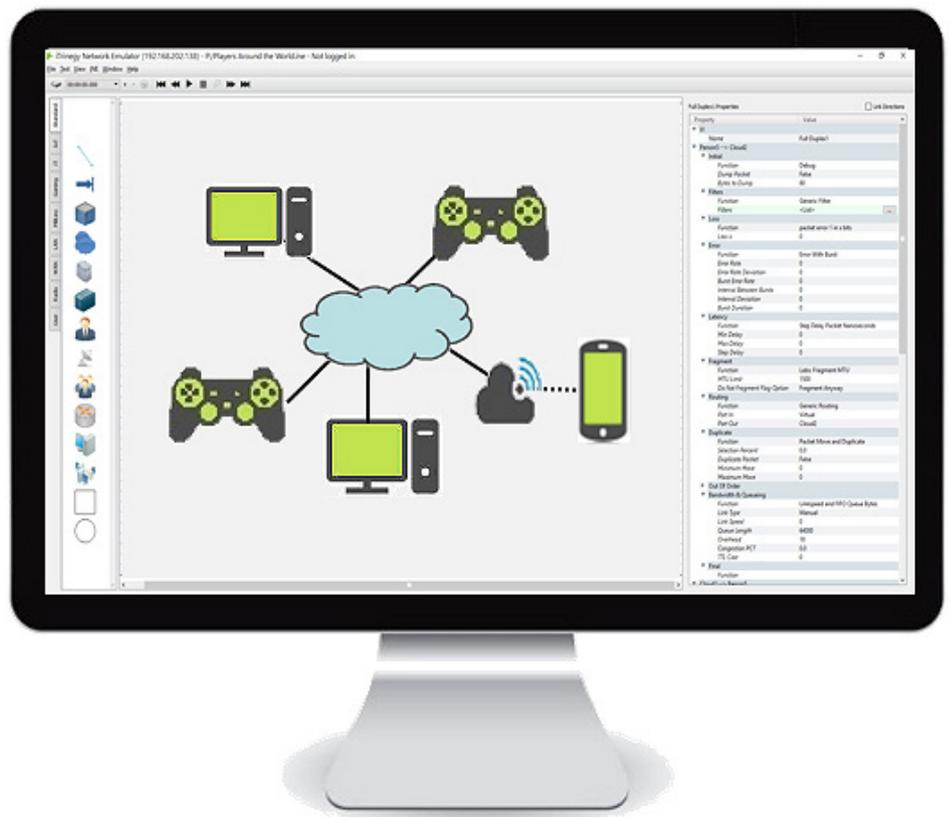
*The NE-ONE's multi-link capabilities and 'Scenario Builder' have proven to be highly popular with game developers*

## ITRINEGY

Over the last 12 years, iTrinegy has strived to provide reliable test equipment for a world in which network quality holds a very high-importance in application performance. We've listened to our customers, undertaken a lot of research, and have worked to provide a solution that allows users without a background in network engineering to perform comprehensive tests against complicated networks, within minutes. The end products of our hard work are the NE-ONE and INE network emulator ranges, which support the most demanding requirements of developers and testers across multiple fields. By combining a robust, reliable design with an elegant, user-friendly GUI, we provide a high-quality solution in service at hundreds of locations across the globe, in multiple industries.

From Game development, error identification, marketplace functionality and cross-platform testing, iTrinegy emulators have been used within the game industry for a while now, with our customer base ranging from new, up-coming indie studios, to studios running tests on Triple-A releases in some of the biggest game series on the market.

iTrinegy demonstrations can be requested [at our demo request page](#), and require a minimum of one week to be successfully scheduled. Fill in the form, and an iTrinegy representative will be in touch soon after!



*A simplified representation of the iTrinegy INE GUI, capable of emulating realistic Meshed Network Scenarios and Point-to-Point configurations in controllable conditions*