Motivation

• Many of today’s mobile devices are equipped with multiple network interfaces that can be used to connect to the Internet.
  • Ethernet, Wi-Fi, 3G and Bluetooth.
  • Current OSs typically choose only one of the available network interfaces and assign all the traffic to it, even if more than one is connected to the Internet.

Architecture

• Our DNIS middleware is divided into two subtasks:
  • Estimation the applications’ needs and characteristics, as well as the properties of the network interfaces.
  • Scheduling different connections to different network interfaces based on the estimated parameters.

Conventional Algorithms

○ Random selection
  • It selects one of the available interfaces for the incoming connection at random.

○ Round Robin
  • It selects one of the available interfaces for the incoming connection in a rotating basis.

○ Selection using GetBestInterface()
  • It selects one of the available interfaces that has the best route to the specified destination IP using the Windows API GetBestInterface() method.

DNIS Algorithms

○ Profile-based
  • It constructs profiles for both the incoming connection and the interfaces, then it selects the interface that has the best fit profile to the connection’s one.

○ Max. Throughput
  • It selects one of the available interfaces that maximize the overall throughput regardless of its stability.

Implementation

• DNIS uses the same concepts used by firewalls and network proxies to control the network flow.

• Each provider has a different LSP chain (i.e., LSP chain for TCP, UDP, etc.)

Results

Note: \( \lambda \) represents the average number of process X per minute.