



# neat

# Neat

**NEAT will make it easier for European software developers to write and deploy applications that efficiently communicate across the Internet, and will enable innovations across the Internet protocol stack.**

## AT A GLANCE

### Project title:

NEAT – A New, Evolutive API and Transport-Layer Architecture for the Internet

### Project coordinator:

Simula Research Laboratory (NO)

### Partners from:

Celerway Communication AS (NO); EMC Information Systems International (IE); Mozilla Denmark APS (DK); Karlstads Universitet (SE); Fachhochschule Münster (DE); University of Aberdeen (UK); Universitetet i Oslo (NO); Cisco Systems (FR)

### Duration:

March 2015 – February 2018

### Total cost:

3.9 M€

### EC Contribution:

3.9 M€

### Programme:

H2020-ICT-2014-1

### Further information:

<https://www.neat-project.org>

## Context and motivation

The Internet's transport architecture is, in some aspects, still dependent on developments dating from the 1980's. Modern online applications need a more flexible transport architecture to rest upon, given the substantial changes that the usage of the Internet has undergone in the past decades. The new reality includes mobile users and no longer shows a clear separation between data communication and telecommunication services.

## Challenge

Smartphones and tablets can run an extensive variety of Internet applications, connected via networks such as WiFi and LTE; terminals such as laptops and desktop PCs add wired connections to the above mix of wireless network technologies. Besides, each application has its own requirements for how it wishes its data to be carried over the Internet. The communication needs of a sensor application differ from those of web browsing or remote conferencing applications. This diversity of needs and technologies currently requires application developers to consider which network(s) to support, and how to tune transport and network parameters for the chosen network(s) – not an easy task.

The problem above is compounded by the fact that the current transport architecture of the Internet – as embodied by common APIs – exposes communication protocols to applications instead of services, and ties applications to a priori choices.

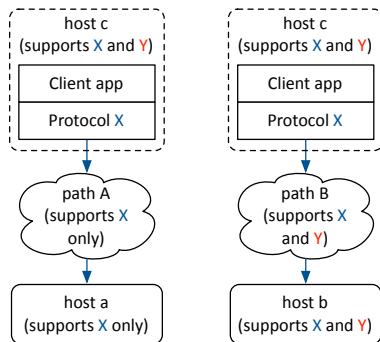


Fig. 1: Due to an a priori choice, the client application is bound to using protocol X when connecting to a new server, even if both the new server, the client and the network path support a better option Y. Implementing mechanisms for a flexible choice of transport and/or network protocols is not a trivial task for application programmers.

Moreover, ossification of the network infrastructure – as embodied by the ubiquitous middleboxes deployed on many networks – makes it very difficult for applications to pick and use modern transports that could provide a better service and improved performance.

## Solution

NEAT will provide a new approach enabling developers to specify an application’s requirements in terms of rate, delay, reliability, cost, etc. This will allow NEAT to choose, or help choose, the best communication service. By separating services offered to applications from the underlying protocols and OS features, NEAT will enable automatic and transparent choice of the best transport options available. As network technologies and protocols continue to evolve, NEAT-enabled applications will immediately be able to take advantage of new functions to reduce web page download times, make teleconferences more responsive or reduce the cost of downloading a software update, getting whatever the application needs.

The NEAT *transport system* is being designed to take care of:

- Offering an enhanced API between applications and the transport layer, so that it exposes transport services to applications.
- Implementing transport services in a best-effort manner, based on the

protocols and functions that are available along a path.

- Providing transparent support for: (a) dealing with middleboxes; (b) discovering and leveraging end-to-end features; (c) interacting with the network for a better application experience.
- Enabling user-space protocol stacks.

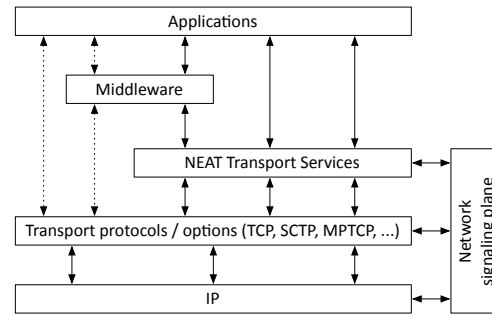


Fig. 2: The NEAT transport architecture.

Such features will allow the NEAT system to: (a) decouple applications from a priori choices of underlying protocols and technologies; (b) support incremental evolution and deployment of new transports; (c) relieve applications from the burden of implementing common mechanisms, and avoid “reinventing the wheel”.

## Expected impact

NEAT will release an open-source reference implementation of the transport system that realises its core functionalities. Also, parts of the NEAT system are targeted for inclusion in future releases of the open-source Firefox web browser.

Building an in-house transport system can yield the best performance for applications that have special network requirements, and this is the path taken by a few industry giants. However, this path cannot be afforded by most software developers and is fraught with many technical hurdles. By providing a free open-source transport system with a large set of functions for efficient communication, NEAT will offer a more level playing field for innovative small companies, from which novel Internet applications often come.

Moreover, to facilitate the broad adoption of NEAT outcomes, the project consortium is promoting its results for standardisation at the Internet Engineering Task Force (IETF), in relevant working groups such as the Transport Services Working Group (TAPS) and the Transport Area Working Group (TSVWG).