



SIMUTools 2010

Torremolinos (Málaga), Spain



Towards a modularized INET

Alfonso Ariza (Univ. De Malaga) & Juan-Carlos Maureira (INRIA)

Current Status of the INET/INETMANET

- Presently the INET has a big size :
 - 870 source files archives in the basic inet-framework.
 - 1600 source files archives in InetMANET branch.
 - Several models mixed into a single monolithic big model.
- Lack of formalism to produce formal releases and maintain/add new functionalities.



Current Status of the INET/NETMANET

The size code continues growing...



... and the problems are beginning to appear



Current Status of the INET/NETMANET

But... .

until when?



Current Status of the INET\NETMANET

Sooner or later the problem will turn up

The code size cannot keep on growing indefinitely
without presenting serious problems



For example, now it is impossible to compile the whole InetMANET framework in Windows OS , the linker cannot link all the files .

**It is urgent to find a solution that allows
the code to keep on growing**



Discussion

What alternatives do we have? and What can we do?

- Divide and Conquer
 - Modularize it into small projects
 - Multiples dynamic libraries
 - Dynamic Dependencies

But, in which way?

- Wireless and Wired?
- OSI Layers?
- Network and Applications?



Modularized INET/netMANET

Advantages

- Smaller executables
 - It creates an executable file with the minimum code necessary for execute the model
 - Lower footprint.
- Shared models!!! (a model repository).
 - More easy extensibility (plug and test new models).
 - Specific module regression-tests provided by the authors .
 - It allows the modules to be updated by multiple maintainers.
- Less elements in the IDE palette when designing.

Disadvantages

- How to split the whole thing?
 - A dependency tree is required:
 - Each project should specify its dependencies.
 - Developers need to take responsibility on their models.
-



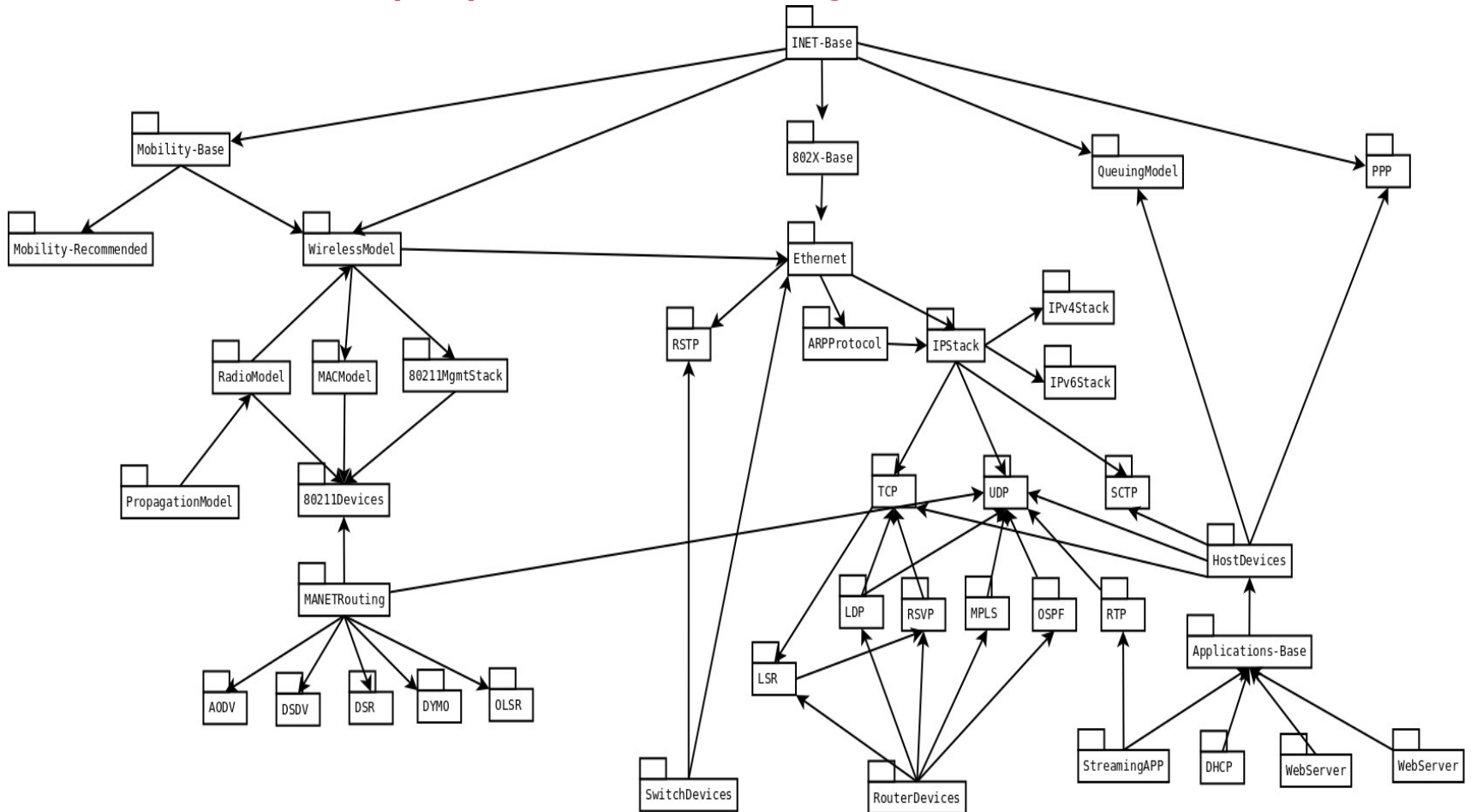
Modularized INET/netMANET

- Each project (modularized model) needs to have a clear structure:
 - IEEE802.11
 - Src
 - Ned
 - Examples
 - Test
- Each project needs to specify the dependencies.
- IDE could download the dependencies from a repository!!



Modularized INET/netMANET

This is a proposal for dividing the INET/netMANET



Modularized INET/netMANET

Every node of the dependency graph can be a software packet that would be downloaded independently and that could be compiled in order to generate a shared library.

Problems

- Dependencies between modules.
- How to solve this?

Reuse the apt-get utility!!



Summary

- INET/InetMANET is getting bigger day after day.
 - Hard to maintain.
 - Complexity.
 - Etc,
- Proposed solution: modularize it
 - Split the whole model into small packages (projects)
 - Implement a dependency tree (apt-get like)
 - Create a repository (apt-get like)
 - Assign maintainers to each package.



Summary

- Advantages
 - Easy to share final simulations.
 - You share your project (only your project)
 - The IDE will download the last version of the required models, compile them, run regression test provided by each model, compile your model and link everything together by using dynamic shared libraries (.so or .dll).
 - Easy to extend. Just plug your new model to the dependency tree and your are done.
 - Model releases will depend on their maintainers
 - Anyone can open a new repository (like linux ones)
 - We could keep numbering advantages, but we have only 15 minutes to talk.



Summary

- Disadvantage
 - We need to keep coding!!
 - We need maintainers
 - Do we already have them? Of course!
 - We need to dive into apt-get to reuse its code.

