Kodo: Implementation and News on the Network Coding library

Morten V. Pedersen - Aalborg University / Steinwurf ApS
mvp@steinwurf.com

Background

Academia

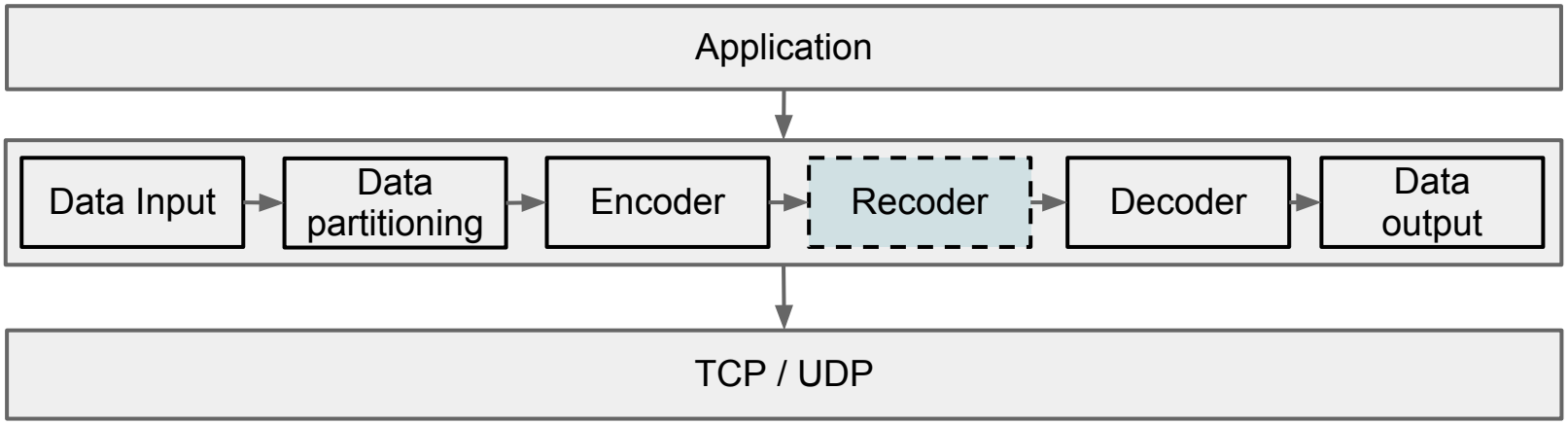
- Network coding key enabler for efficient user cooperation (p2p).
- Kodo developed during a 3 year research project CONE (COoperation and NEtwork Coding). *Concluded 2012*.

Industry

- On campus start-up Steinwurf ApS founded in 2011.
- Taking over the rights for Kodo and development.
- Library source code fully available. Licenses:
  a. Free for Research / Educational
  b. Paid Commercial

Where does Kodo fit?

- Many different requirements
  - Deterministic vs. random, inter- vs. intra-flow, physical to application / transport layer.
- Current versions of Kodo implement
  - Software & Digital Random Linear Network Coding (RLNC)
  - Suitable for transport / application layer protocol implementations
  - Focus on the coding



Kodo (the library)

- C++11 *(staying compatible with major compilers).*
- Designed to allow for easy experimentation and a high degree of code reuse.
- Very flexible design technique used called "mixin-layers" or "parameterized inheritance" using C++ templates.
- Low-level = ample ways of shooting yourself in the foot. With API specs. we try to mitigate this.
- High Performance - code generated by compiler comparable to single monolithic implementation.
- Helper libraries.
  - Resource management
  - Finite Fields

Since Orlando (IETF 86) - external

- **High-level C Binding**
  - https://github.com/steinwurf/kodo-c-bindings
  - Pre-built binaries available

- **NS-3 examples**
  https://github.com/steinwurf/kodo-ns3-examples

- **Basic discrete time simulator**
  https://github.com/steinwurf/kodo-basic-simulations

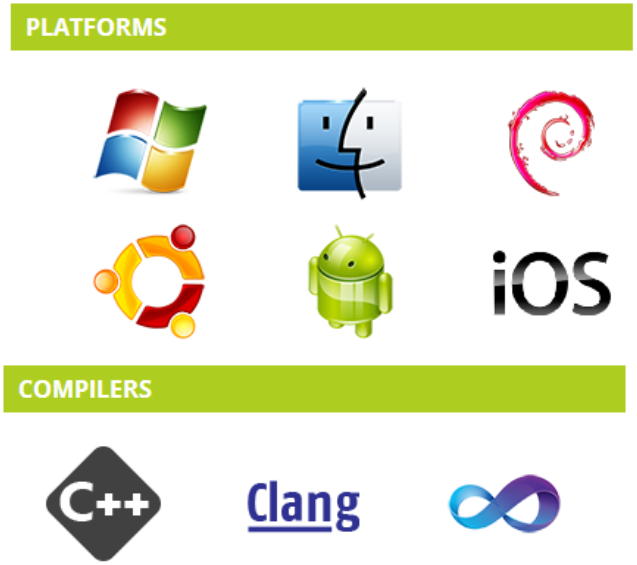
Since Orlando (IETF 86) - internal

v8.0.0 to v11.2.0:

- Bug-fixes + minor improvements
- On-the-fly, sliding window, online encoding/decoding
  - Unique to network coding
  - Progressively include packets into the encoding
  - Progressively extract packets from the decoding
  - Important to efficiently support streaming and interactive applications.
- Sparse coding (development ongoing)
  - Efficient way of increasing performance of encoding decoding
- Additional benchmarks + examples

Kodo and the IRTF NWCRG

- Provides a solid building block for
  - Protocol development.
  - Experimentation with different code variants.
- It is well tested.
  - Visit our buildbot: http://buildbot.steinwurf.dk:12344
- It has traction:
  - New features.
  - Supported platforms.
  - Several companies and university research groups already using it.



Getting started

- Code
  - http://github.com/steinwurf/kodo
  - See example of encode/decode in the examples folder
- Documentation (we are working on it)
  - http://readthedocs.org/docs/kodo/en/latest/
- Status buildbot: http://buildbot.steinwurf.dk:12344/

The End

- Questions?
- Contributions + bug fixes please
  - Simple procedure with sign-off
- Feedback / comments /questions are all very welcome!

Morten V. Pedersen
morten@steinwurf.com