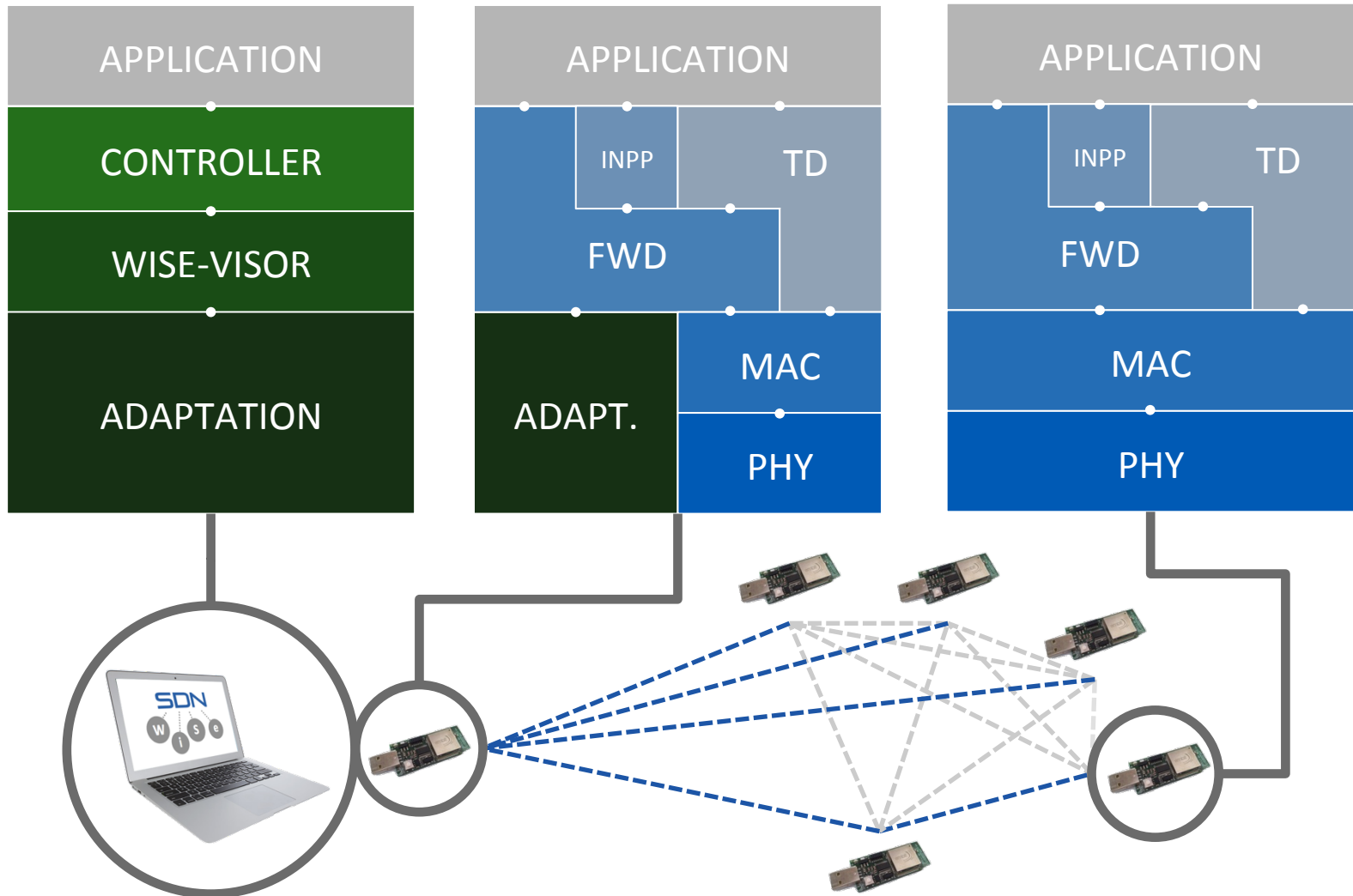


L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo. SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for Wireless SEnsor networks. Proc. of INFOCOM2015. April 2015.

Architettura



Control Plane

CONTROLLER

Tra i suoi compiti abbiamo:
gestione della topologia della rete,
routing e configurazione dei nodi

WISE-VISOR

Astrae le risorse di rete in modo da
avere più reti virtuali coesistenti
sulla stessa rete fisica

ADAPTATION

Adatta il formato dei pacchetti, da e
per la rete in modo da poter
utilizzare dispositivi reali e/o
simulati

Data Plane

TOPOLOGY
DISCOVERY (TD)

Configurazione dei parametri,
beaconing, reporting dei vicini

IN-NETWORK PACKET
PROCESSING (INPP)

Data aggregation
(Riprogrammabilità)

FORWARDING (FWD)

Gestisce i pacchetti in ingresso
come specificato all'interno della
WISE Table

Caratteristiche

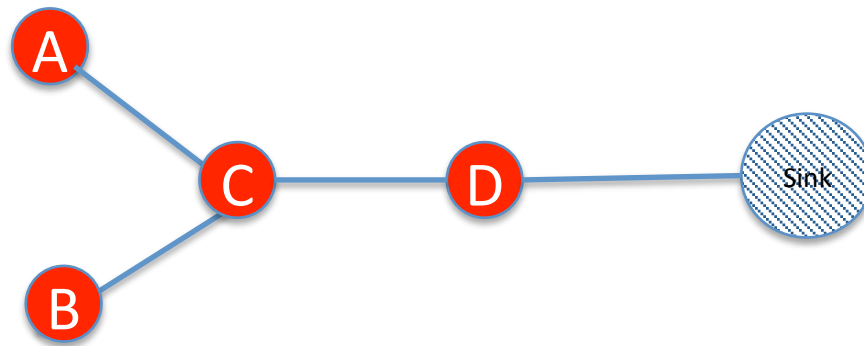
- Stato
- Definizione delle regole
- Multitenancy

Stato

- I router SDN sono stateless
- In una WSN decentralizzare alcune decisioni può ridurre il numero di messaggi inviati
- Stato -> Un struttura dati che può essere scritta o letta in risposta a un messaggio in ingresso

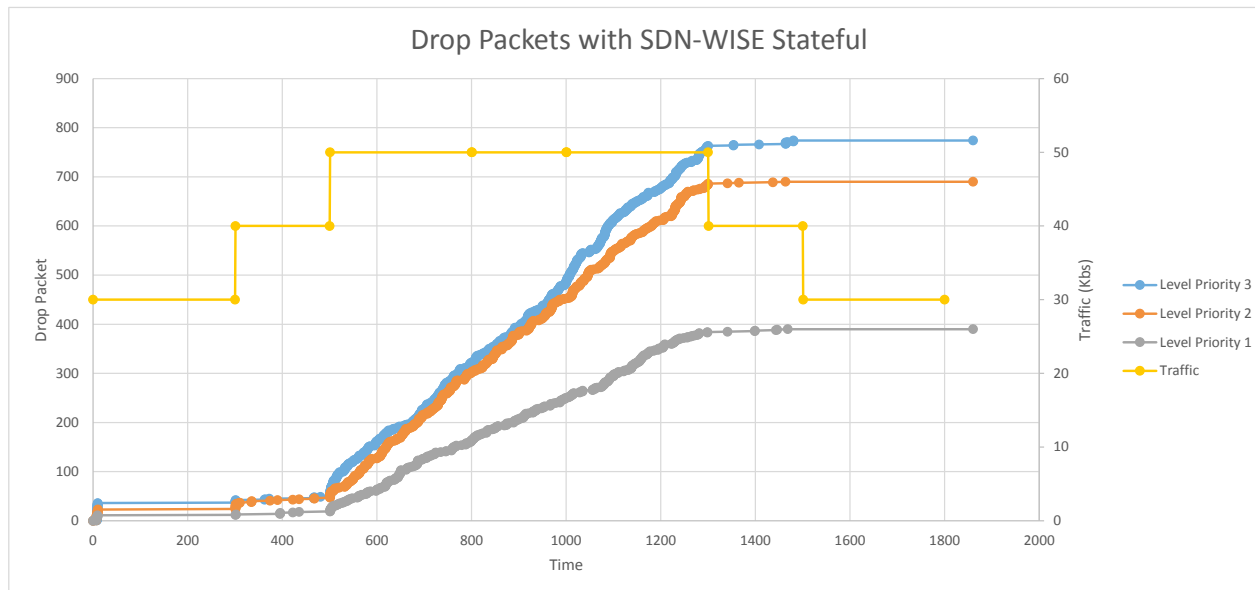
Casi d'uso

- Forwarding condizionale: C forwarda i messaggi provenienti da A, se e solo se i messaggi provenienti da B, soddisfano una certa regola



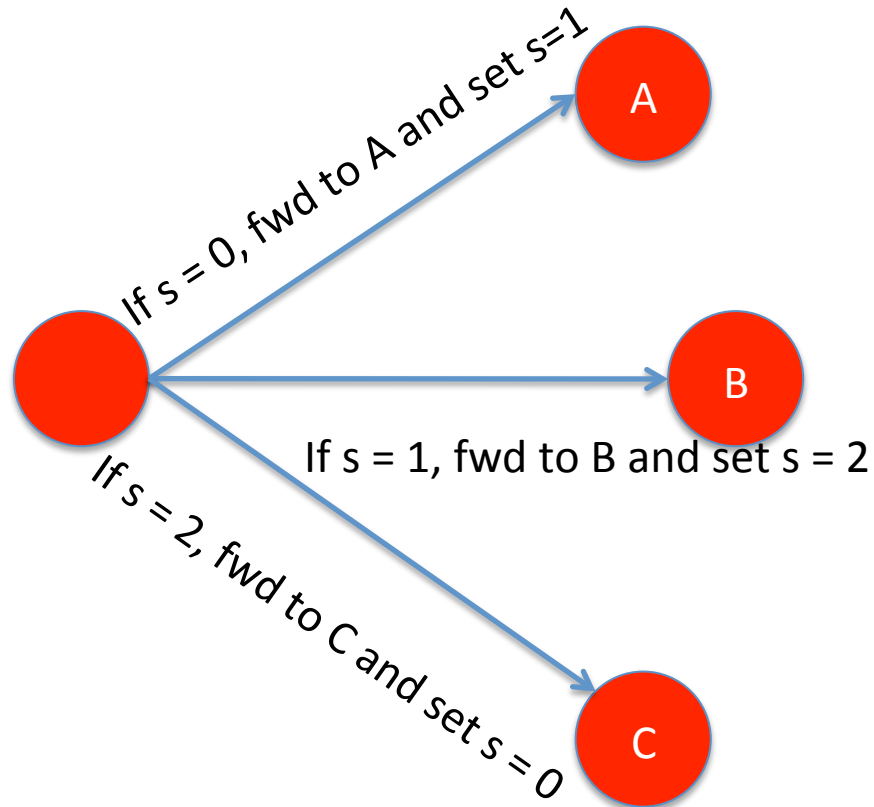
Casi d'uso

- QoS: Il livello di congestione di un nodo è memorizzato all'interno dello stato. In base a tale livello decido se forwardare o droppare i messaggi appartenenti a un certo flusso



Casi d'uso

- Multipath routing



Regole

- Matching + Action
- Permettono di verificare fino a 3 condizioni su 1-2 byte del messaggio in ingresso/dello stato
- Le azioni che possono essere eseguite sono
 - Forward Unicast
 - Forward Broadcast
 - Drop
 - Modifica
 - Invia all'applicazione X
 - Sleep per X secondi
 - ...

Regole

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Type	Multimatch	Location	Offset	Value	TTL	Count
=	2	PACKET	SRC_ADDR	B	>=	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	0	MODIFY	TRUE	STATE_ARR	0	1	255	0
=	2	PACKET	SRC_ADDR	B	<	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	1	MODIFY	TRUE	STATE_ARR	0	0	255	0
=	2	PACKET	SRC_ADDR	B	-	-	-	-	-	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	0	-	-	-	-	-	DROP	FALSE	PACKET	0	100%	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	1	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0

Multitenancy vs Slicing

- Nelle reti SDN wired la divisione di una rete in più slices è effettuata tramite l'assegnazione di porzioni disgiunte dello spazio dei messaggi a diversi controller
- E se invece volessimo condividere gli stessi nodi tra più utenti?
- Problemi di isolamento -> **WISE Visor**

Header SDN-WISE

LEN	ID	SRC		DST		TYP	TTL	NX HOP	
x	1	0	2	0	0	0	20	0	1

- LEN: Lunghezza pacchetto (1 byte)
- ID: identificativo della rete (1 byte)
- SRC: indirizzo sorgente (2 byte)
- DST: indirizzo destinazione (2 byte)
- TYP: tipologia pacchetto + Request Flag (1 byte)
- TTL: Time To Live (1 byte)
- NX HOP: informazione per il forwarding (2 byte)

Tipologia Pacchetti

- SDN_WISE_DATA = 0
- SDN_WISE_BEACON = 1
- SDN_WISE_REPORT = 2
- SDN_WISE_REQUEST = 3 (deprecato)
- SDN_WISE_RESPONSE = 4
- SDN_WISE_OPEN_PATH = 5
- SDN_WISE_CONFIG = 6

Data

HEADER	PAYLOAD
	X

- PAYLOAD: dati da/per il nodo

Beacon

HEADER	No. HOP	BATT
	0	100

- Ciascun nodo invia ogni SDN_WISE_DFLT_CNT_BEACON_MAX secondi il beacon
- No. HOP: distanza in numero di hop dal sink (1 byte)
- BATT: livello in percentuale della batteria (1 byte)

Report

HEADER		N	ADD		RSSI	ADD		RSSI
		2	0	2	180	0	3	175

- Ciascun nodo invia ogni SDN_WISE_DFLT_CNT_REPORT_MAX secondi il report dei vicini
- N: numero di vicini del nodo (1 byte)
- ADD: indirizzo del vicino (2 byte)
- RSSI: indicatore qualità del canale (1 byte)

Response



- Il payload del messaggio è costituito dalla regola che viene installata sul nodo di destinazione (17 byte)

OpenPath

HEADER	ADDRESS	...
	value	...

- Scorciatoia per creare un percorso all'interno della rete
- Il payload è costituito da una lista di indirizzi (2*numero di nodi nel path byte)
- Ciascuno nodo che riceve questo messaggio, cerca se stesso all'interno della lista e impara le regole necessarie per inviare al primo e all'ultimo nodo nella lista

Config

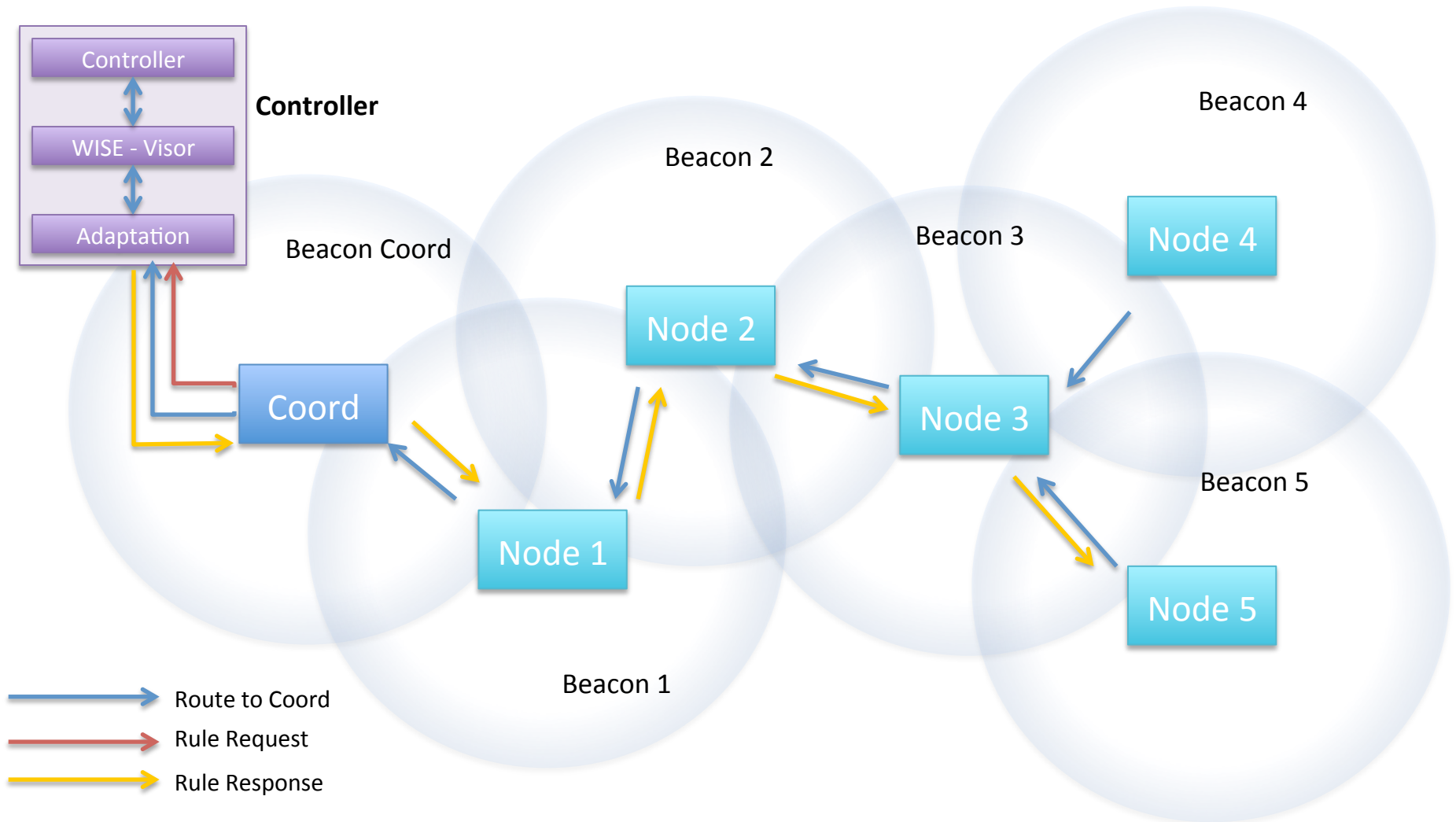
HEADER	PROPERTY	
	r/w + id	value

- Permette di configurare vari aspetti del nodo
- Leggere il contenuto della WISE Table
- Impostare periodi, indirizzi, TX power, ecc
- Il primo byte è un id per identificare la proprietà
- Il secondo e il terzo sono i valori di tale proprietà

Routing in SDN-WISE

- Beaconing
- Reporting
- Il primo nodo che non sa come gestire un messaggio, modifica il Request Flag e lo invia al Controller Plane
- Il Controller Plane invia un messaggio Response o un OpenPath
- E rimanda il messaggio originario

Routing in SDN-WISE



SDN-WISE Statefulness: Solution

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Type	Multimatch	Location	Offset	Value	TTL	Count
=	2	PACKET	SRC_ADDR	B	>=	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	0	MODIFY	TRUE	STATE_ARR	0	1	255	0
=	2	PACKET	SRC_ADDR	B	<	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	1	MODIFY	TRUE	STATE_ARR	0	0	255	0
=	2	PACKET	SRC_ADDR	B	-	-	-	-	-	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	0	-	-	-	-	-	DROP	FALSE	PACKET	0	100%	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	1	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0

Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value
=	2	PACKET	SRC_ADDR	B	>=	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	0
=	2	PACKET	SRC_ADDR	B	<	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	1

SDN-WISE Statefulness: Solution

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Type	Multimatch	Location	Offset	Value	TTL	Count
=	2	PACKET	SRC_ADDR	B	>=	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	0	MODIFY	TRUE	STATE_ARR	0	1	255	0
=	2	PACKET	SRC_ADDR	B	<	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	1	MODIFY	TRUE	STATE_ARR	0	0	255	0
=	2	PACKET	SRC_ADDR	B	-	-	-	-	-	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	0	-	-	-	-	-	DROP	FALSE	PACKET	0	100%	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	1	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0

Type	Multimatch	Location	Offset	Value
MODIFY	TRUE	STATE_ARR	0	1
MODIFY	TRUE	STATE_ARR	0	0

SDN-WISE Statefulness: Solution

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Type	Multimatch	Location	Offset	Value	TTL	Count
=	2	PACKET	SRC_ADDR	B	>=	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	0	MODIFY	TRUE	STATE_ARR	0	1	255	0
=	2	PACKET	SRC_ADDR	B	<	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	1	MODIFY	TRUE	STATE_ARR	0	0	255	0
=	2	PACKET	SRC_ADDR	B	-	-	-	-	-	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	0	-	-	-	-	-	DROP	FALSE	PACKET	0	100%	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	1	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0

Operator	Size	Location	Offset	Value
=	2	PACKET	SRC_ADDR	B

Type	Multimatch	Location	Offset	Value
FORWARD	FALSE	PACKET	0	D

SDN-WISE Statefulness: Solution

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Type	Multimatch	Location	Offset	Value	TTL	Count
=	2	PACKET	SRC_ADDR	B	>=	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	0	MODIFY	TRUE	STATE_ARR	0	1	255	0
=	2	PACKET	SRC_ADDR	B	<	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	1	MODIFY	TRUE	STATE_ARR	0	0	255	0
=	2	PACKET	SRC_ADDR	B	-	-	-	-	-	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	0	-	-	-	-	-	DROP	FALSE	PACKET	0	100%	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	1	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0

Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	1

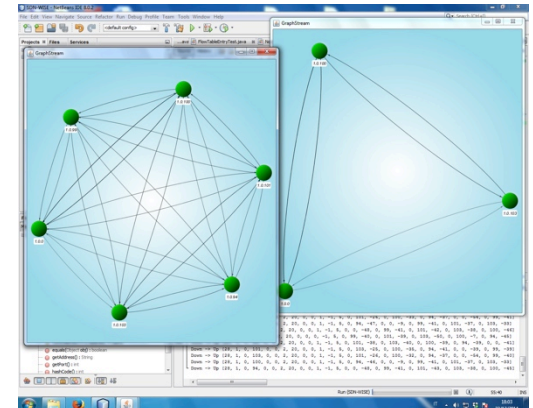
SDN-WISE Statefulness: Solution

Matching Rule					Matching Rule					Matching Rule					Action					Statistics	
Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Operator	Size	Location	Offset	Value	Type	Multimatch	Location	Offset	Value	TTL	Count
=	2	PACKET	SRC_ADDR	B	>=	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	0	MODIFY	TRUE	STATE_ARR	0	1	255	0
=	2	PACKET	SRC_ADDR	B	<	2	PACKET	DATA	Xthr	=	1	STATE_ARR	0	1	MODIFY	TRUE	STATE_ARR	0	0	255	0
=	2	PACKET	SRC_ADDR	B	-	-	-	-	-	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	0	-	-	-	-	-	DROP	FALSE	PACKET	0	100%	255	0
=	2	PACKET	SRC_ADDR	A	=	1	STATE_ARR	0	1	-	-	-	-	-	FORWARD	FALSE	PACKET	0	D	255	0

Type	Multimatch	Location	Offset	Value
DROP	FALSE	PACKET	0	100%
FORWARD	FALSE	PACKET	0	D

Testbeds

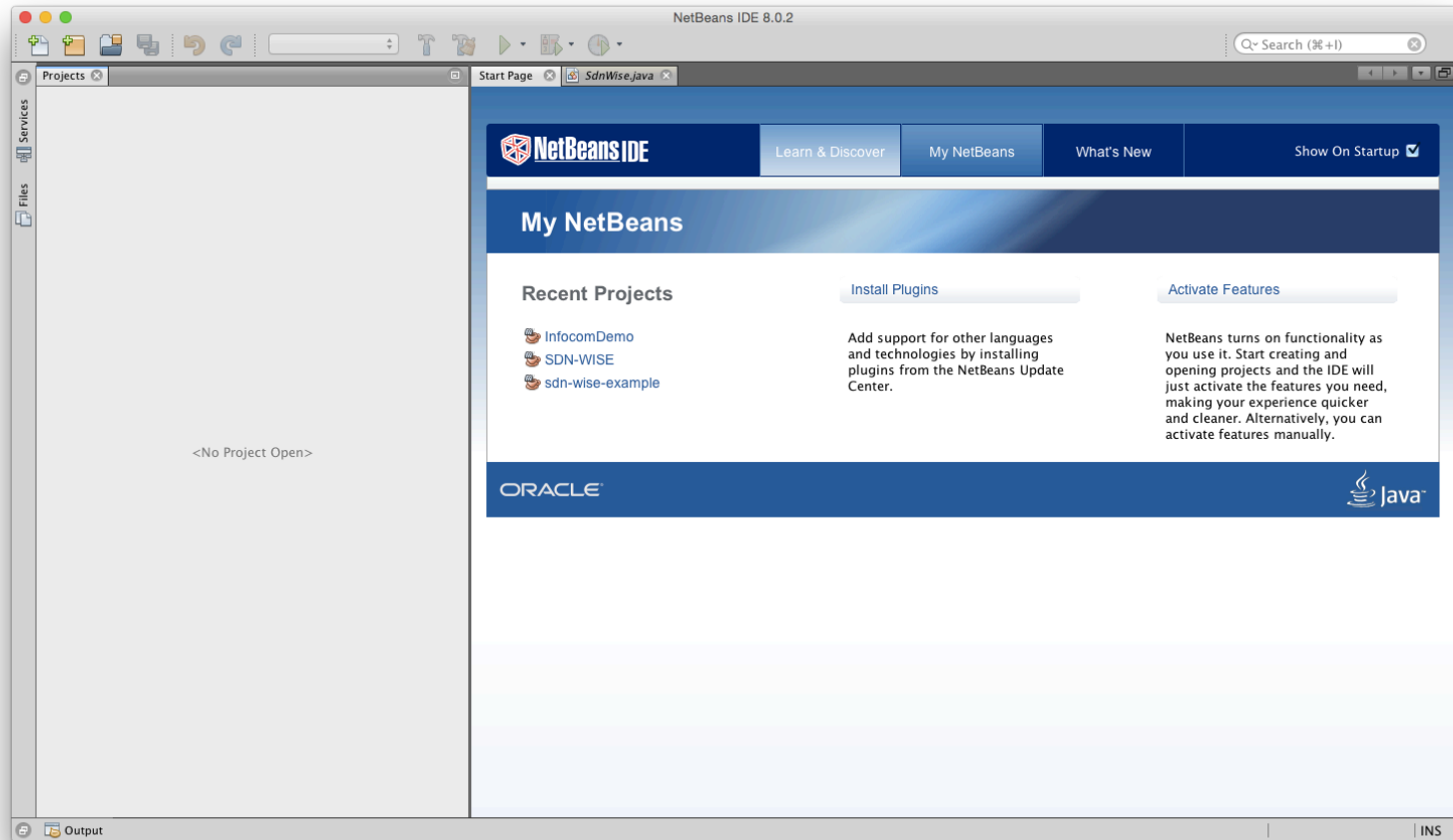
- Simulated Testbed
 - OMNeT++
 - Java
- Real Testbed
 - 6 Embit EMB-Z2530PA
 - IEEE 802.15.4
 - TI CC2530 single chip device - 8051 8-bit controller
 - 8kB of RAM and 256kB of Flash memory
- Controller Plane + Application
 - Intel Core 2 CPU @ 2.40 GHz
 - 4GB of RAM
 - Windows 7 - 32 bit + Java 7



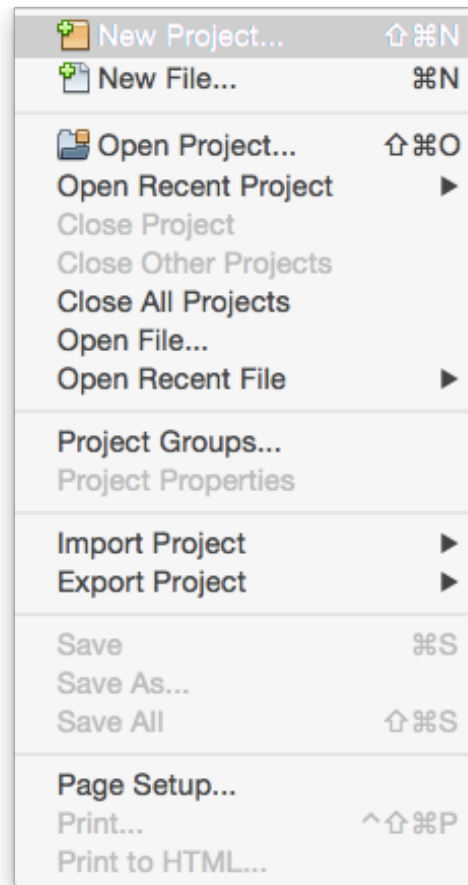
Un po' di codice

- Java + C -> Netbeans + Maven + IAR
- Iniziamo con qualcosa di semplice...

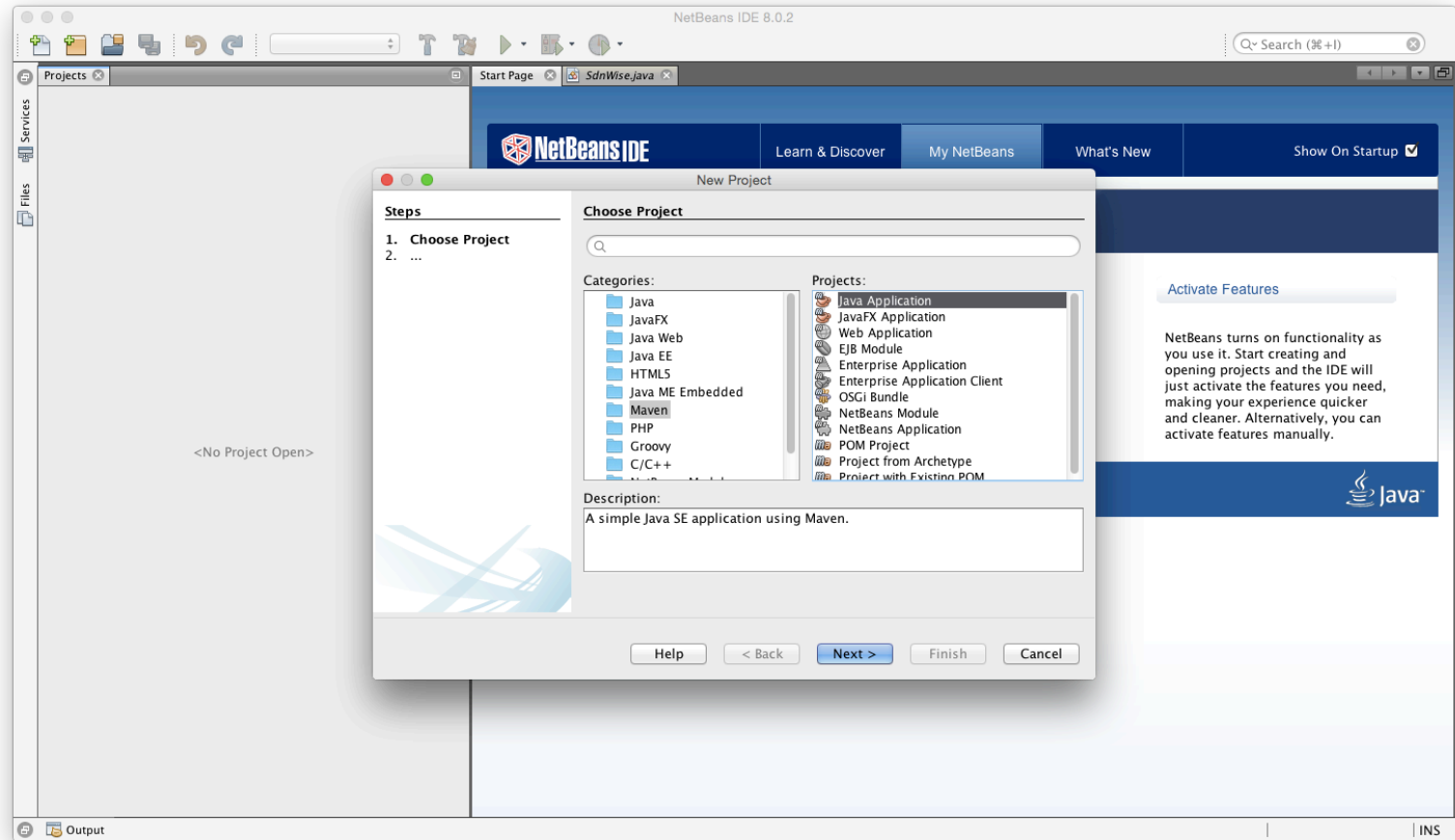
Netbeans + Maven



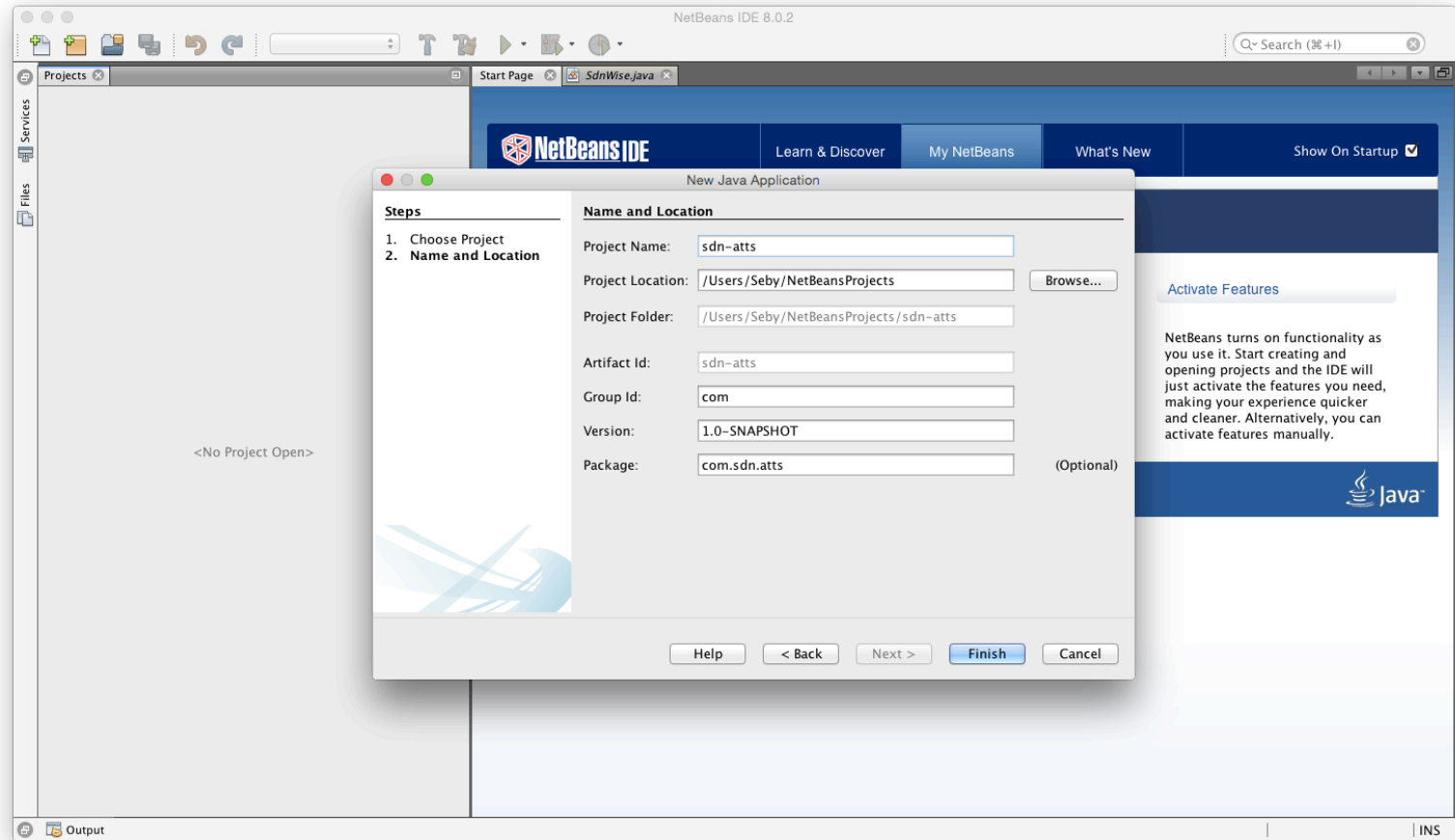
Netbeans + Maven



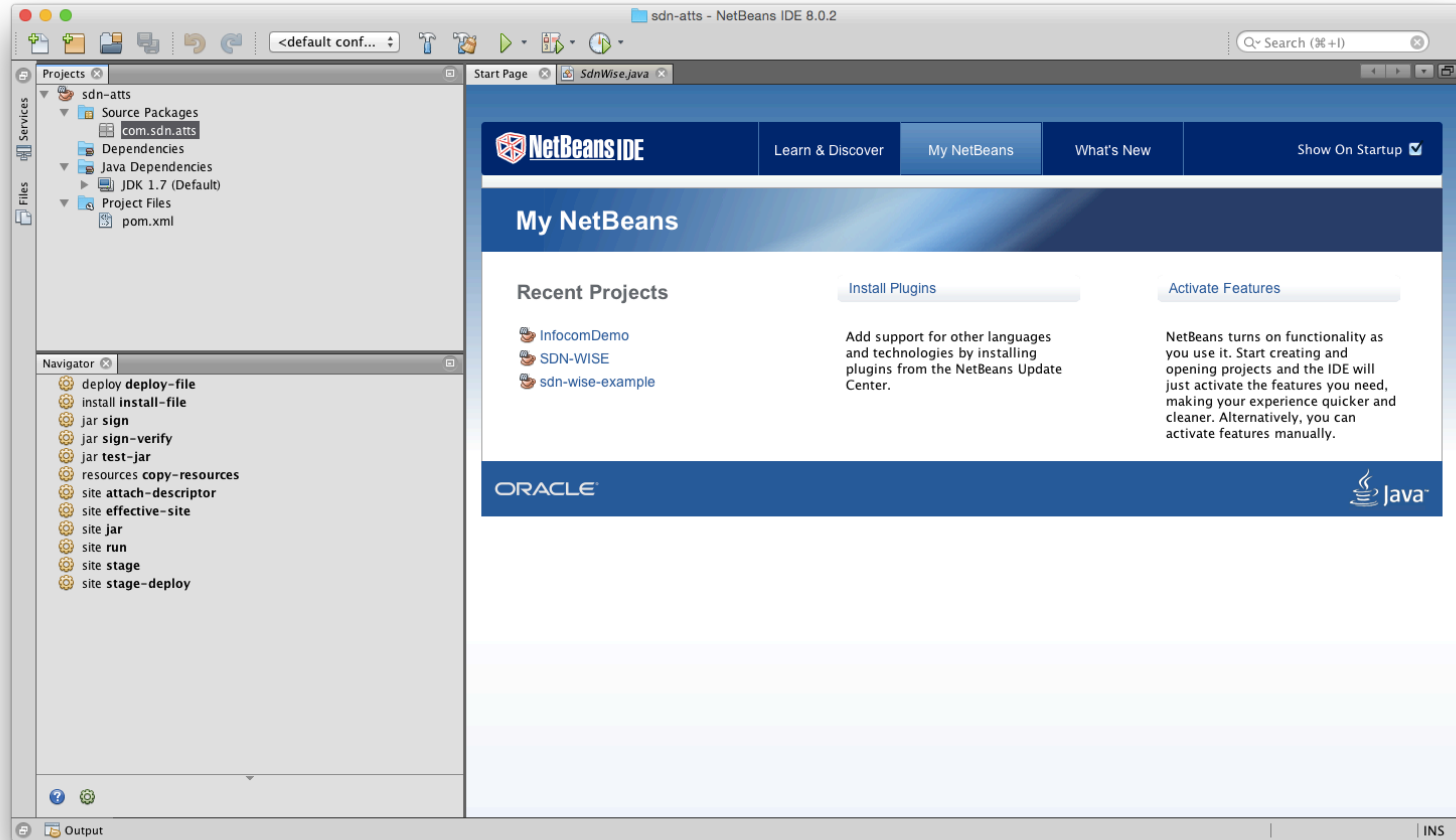
Netbeans + Maven



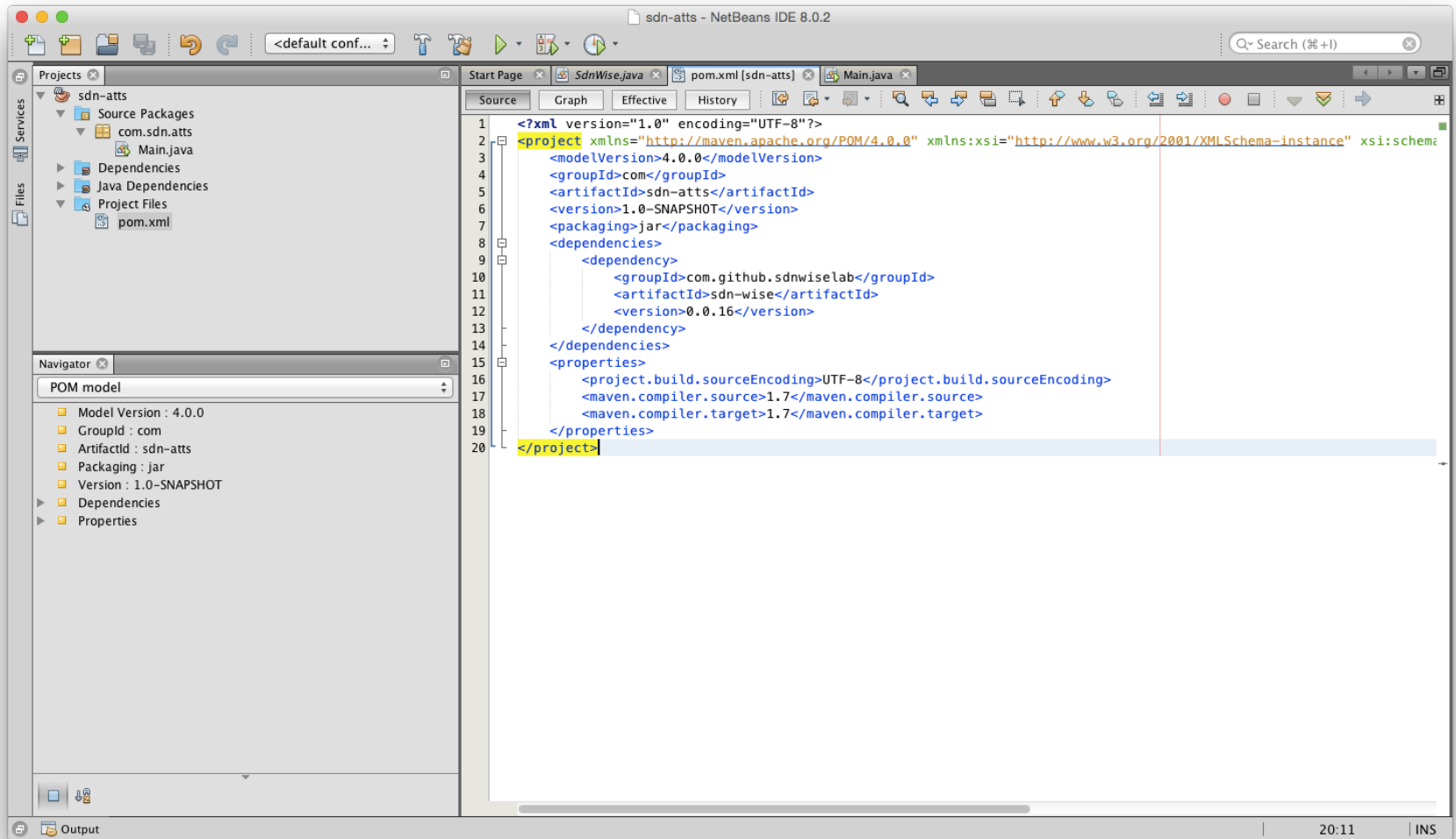
Netbeans + Maven



Netbeans + Maven



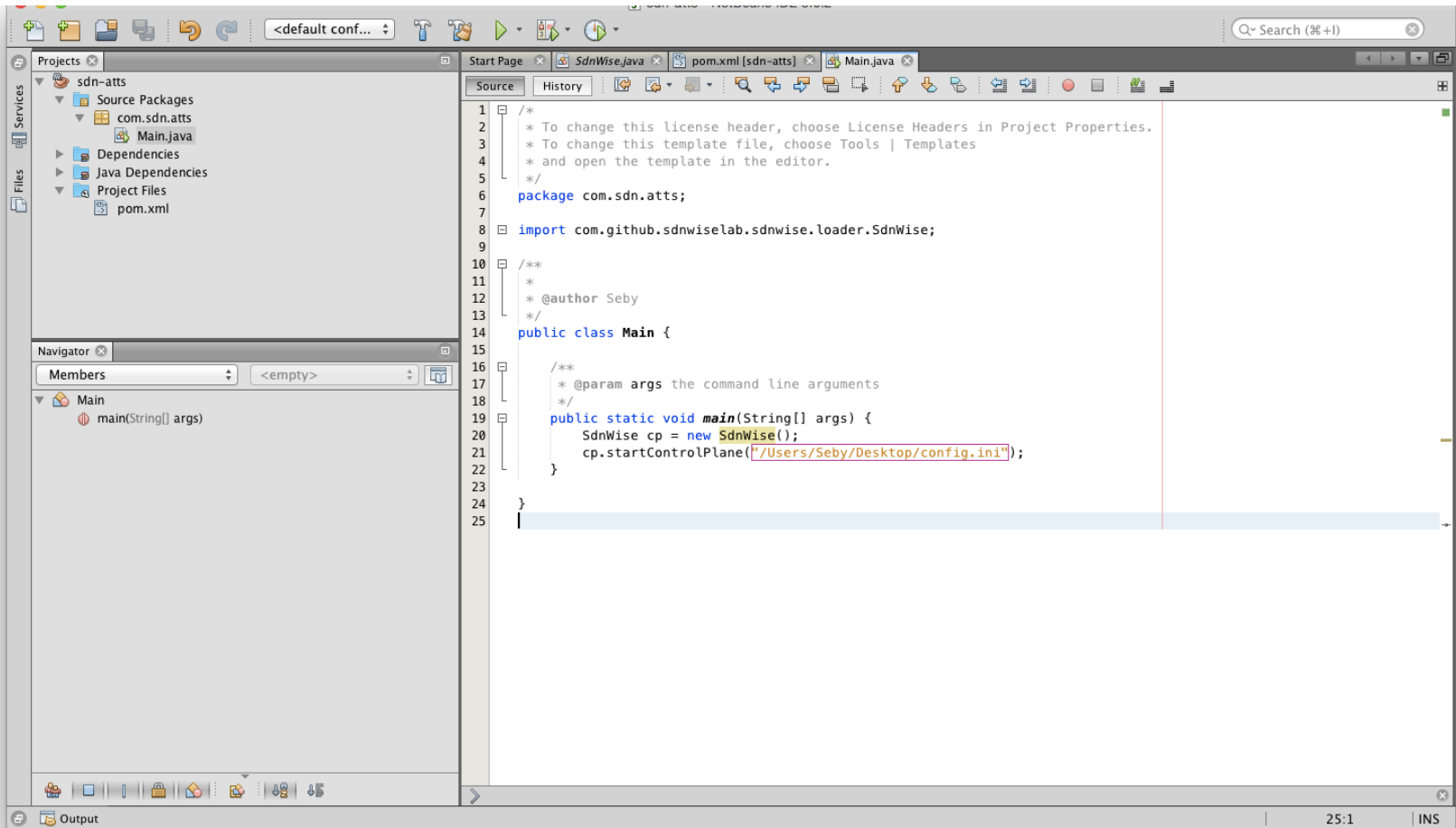
POM



Config file

- Download un config file di esempio da:
- <http://sdn-wise.dieei.unict.it/config/config.ini>

Scriviamo un main



RXTX - <http://fizzed.com/oss/rxtx-for-java>

The screenshot shows a web browser window with the address bar displaying `fizzed.com/oss/rxtx-for-java`. The page content includes:

- A paragraph about the build process: "Built using Microsoft Visual C++ 2008 - not MinGW. The x86 and x64 versions are native and do not rely on any other non-standard windows libraries. Just drop in the compiled .dlls that are specific to the version of Java you run. If you installed the 64-bit version of the JDK, then install the x64 build."
- A note: "Tested the x86 and x64 versions with Windows 2008, 2003, and Vista SP1."
- A section header: **linux-i386, linux-x86_64**
- Another paragraph about the build process: "Built using CentOS 5.2 and gcc 4.1.2. Just drop in the compiled .dlls that are specific to the version of Java you run. If you installed the 64-bit version of the JDK, then install the x64 build."
- A note: "Tested the x86 and x64 versions with x86 and x64 versions of CentOS 5.0 and 5.2."
- A section header: **Downloads**
- A table with download information:

Version	File	Information
RXTX-2-2-20081207	Windows-x64 Windows-x86 Windows-ia64 Linux-x86_64 Linux-i386	Based on CVS snapshot of RXTX taken on 2008-12-07

Below the table, there is a section titled "Updates? Need assistance?" with the following text:

Follow [@mfizz_inc](#) on Twitter for future updates and latest news.

If you have specific issues, questions, or problems, please [contact us](#) with your inquiry or consulting request.

The footer of the page is dark gray and contains three columns:

- About**: Fizzed is a boutique consulting firm and research lab that incubates the launch and growth of mobile, web, and data
- Latest**: [@fizzed_inc](#): Excited to announce Stork - a better way to deliver your JVM-based application [github.com/fizzed](#)
- Contact**: 1631 15th Ave W Suite 317

A small upward-pointing arrow icon is located in the bottom right corner of the footer.

Conclusioni

- SDN-WISE è una soluzione SDN per WSN
- C'è ancora **TANTO** lavoro da fare
- <http://sdn-wise.dieei.unict.it>
- <https://github.com/sdnwiselab/sdn-wise>

Domande?

s.milardo@hotmail.it