

Running Real-World Software on Simulated Wireless Sensor Nodes

Georg Wittenburg, Freie Universität Berlin

ACM Workshop on Real-World Wireless Sensor Networks (REALWSN'06), Uppsala, Sweden



- Background / Motivation / Idea
- Conceptual Perspective / Comparison
- Implementation Details
- A Look into Scalability Issues
- Results / Conclusion

ScatterWeb WSN Platform:

Background (1)

- Developed by AG CST at FU Berlin.
- Project started in 2002.
- Components commercially available.
- Embedded Sensor Board:
 - TI MSP430 microcontroller
 - TR1001 radio transciever
 - 868 MHz, up to 19.2 kbit/s
 - 2 KB RAM, 8 KB EEPROM
- Embedded Chip Radio
- eGate (USB/Web)









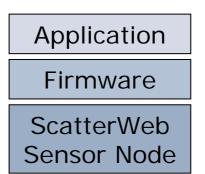






Background (2)

- ScatterWeb Applications:
 - Ad-hoc routing, DSDV
 - Directed Diffusion
 - SQL-like queries
 - RSA, MD2/4/5
 - Localization
 - Virtual Machine Abstraction
 - Rule-Oriented Middleware
- Firmware provides hardware abstraction.
- Applications link against firmware API (111 C functions).
- More at http://scatterweb.mi.fu-berlin.de.

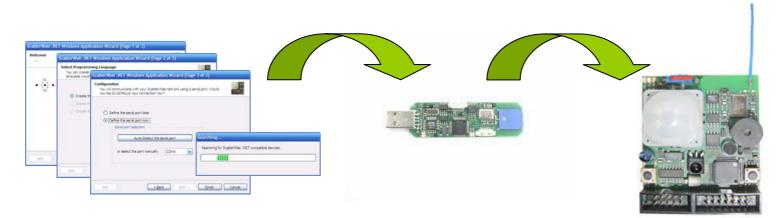




Motivation

Freie Universität

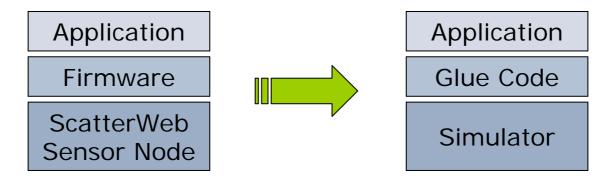
- Development Cycle:
 - Implement, compile, flash to sensor node, test, debug.



- Generally works, but has disadvantages:
 - Real ScatterWeb hardware must be available.
 - Flashing and debugging may be tedious.
 - Testing algorithms on large networks is not feasible.



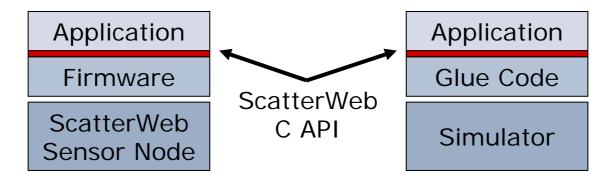
• Use a simulator to develop and test applications.



- Advantages:
 - Faster development cycle.
 - Developer can concentrate on algorithmic issues first and deal with hardware-specific issues later.
 - Algorithms can actually be tested in realistic scenarios under reproducible conditions.



• Use a simulator to develop and test applications.



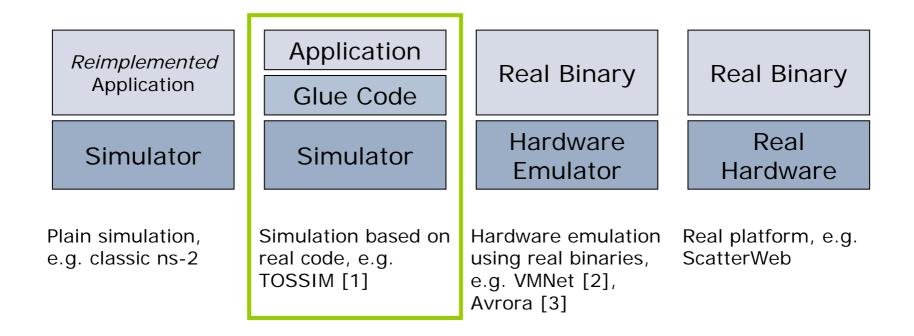
- Preconditions:
 - ScatterWeb firmware has a stable C API.
 - A method exists to port this C API to a network simulator.
- But is there a way to port a C API to a network simulator that is both transparent and generic?



- Background / Motivation / Idea
- Conceptual Perspective / Comparison
- Implementation Details
- A Look into Scalability Issues
- Results / Conclusion

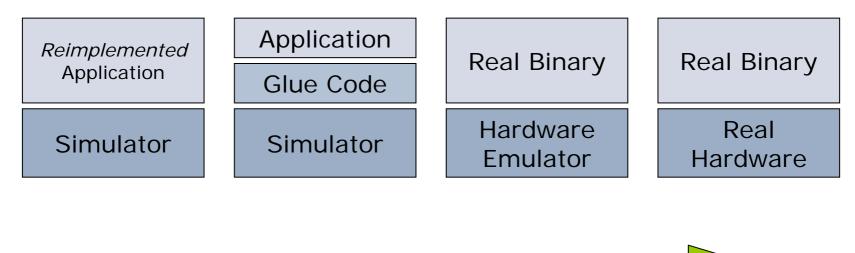


• Where would this leave us conceptually?





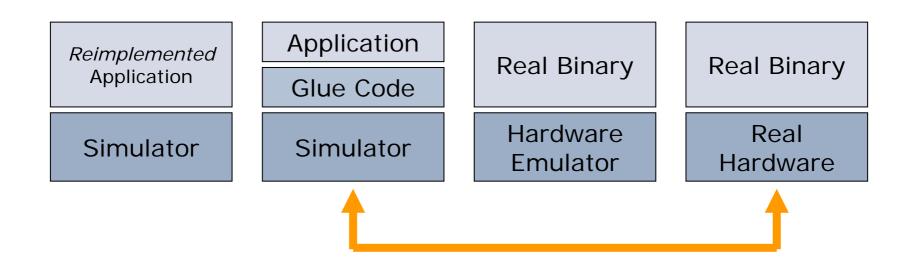
• Advantages / Disadvantages





Georg Wittenburg, FU Berlin

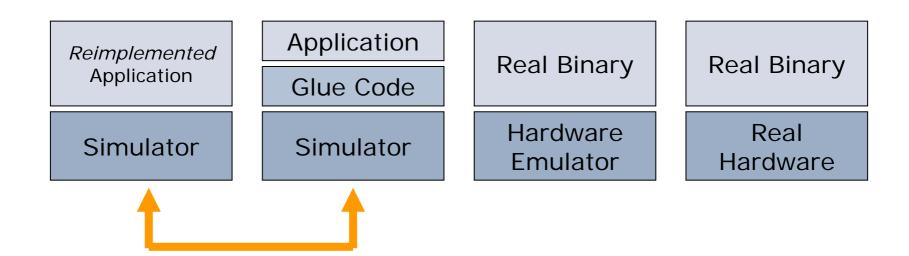




- Pro:
 - Faster development cycle.
 - Algorithms first.
 - Testing under reproducible conditions.

- Contra:
 - Less accurate.

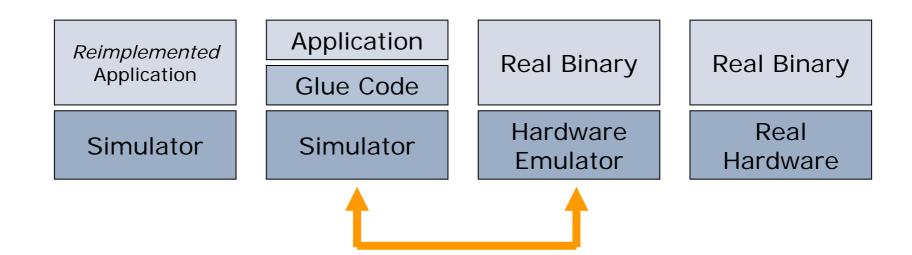




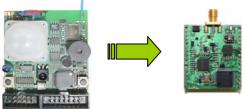
- Pro:
 - No reimplementation, no programming inaccuracies, more realistic simulation.
- Contra:

• ?





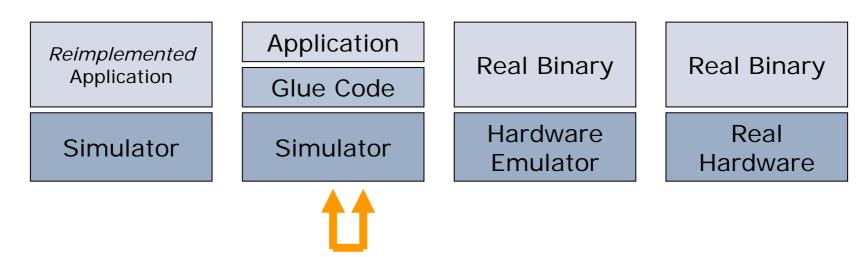
- Pro:
 - Portable across platforms.



Possibly less accurate.

Contra:





- TOSSIM integrates simulator implemented from scratch.
- Pro:
 - Our approach reuses existing simulation technology.

• Contra:

• ?



- Background / Motivation / Idea
- Conceptual Perspective / Comparison
- Implementation Details
- A Look into Scalability Issues
- Results / Conclusion

Implementation Issues

- Why use the ns-2 network simulator [4]?
 - Written in C++, supports linking C code.
 - Open source, compatible license.
 - Large user base.
 - But need to consider scalability issues.
- Key problems when linking application C code into a C++ simulator:
 - Our C code is static, but we need to have dynamic sensor node objects in the simulation.
 - From the point of view of the application, these object must appear as normal sensor nodes.
- The goal is to run ScatterWeb applications within the simulation with as little changes to their code as possible.

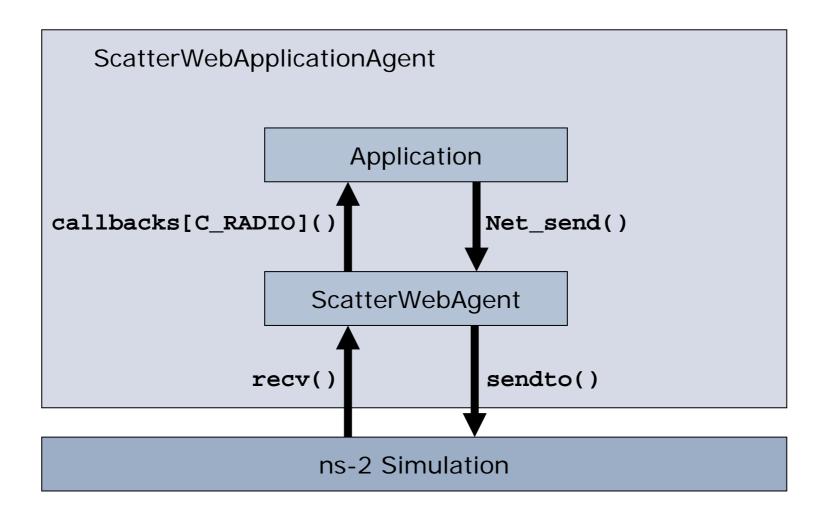
RFALWSN'06



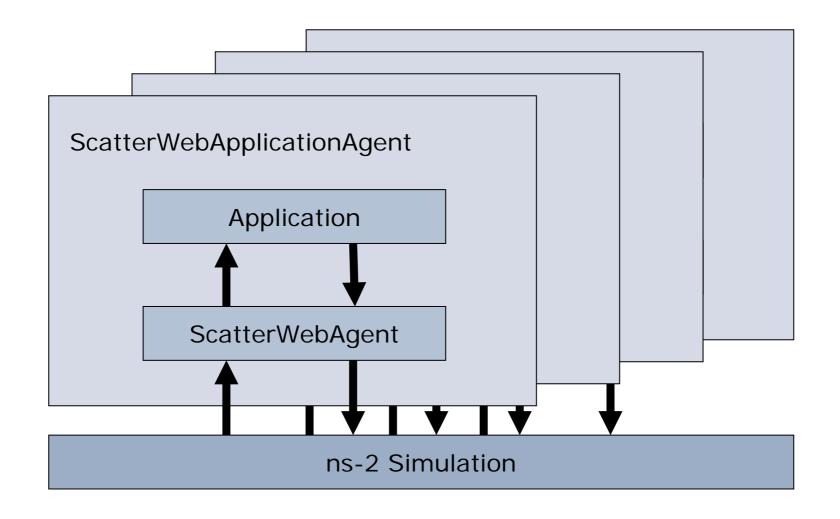
Simulator







Implementation Overview (2)



Freie Universität

Berlin



• How does this work?

```
[...]
#define Net_send ScatterWebAgent::instance->Net_send
[...]
```

Excerpt from ScatterWebFirmwareWrapper.h

```
[...]
#include "ScatterWebFirmwareWrapper.h"
[...]
#include "ScatterWeb.Event.c"
#include "ScatterWeb.Process.c"
```

Excerpt from ScatterWebApplication.cc

Implementation Details (2)



• What changes are necessary in the application code?

```
[...]
#ifndef SCATTERWEB_ON_NS2
int global_variable;
#endif
[...]
```

Excerpt from ScatterWeb.Event.c

```
class ScatterWebApplicationAgent {
    public:
        [...]
        int global_variable;
};
#define global_variable ((ScatterWebApplicationAgent*)
        ScatterWebAgent::instance)->global_variable
```

```
Excerpt from ScatterWebApplication.cc
```



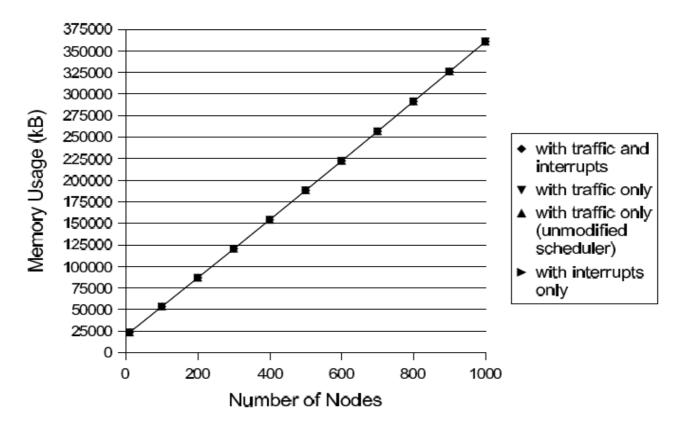
- Background / Motivation / Idea
- Conceptual Perspective / Comparison
- Implementation Details
- A Look into Scalability Issues
- Results / Conclusion



- ns-2 is not exactly famous for its scalability.
 - Addressing this is out of scope.
 - However, make sure not to add *additional* overhead.
- Consider both memory consumption and simulation time.
- Specifically, look at:
 - Different network sizes.
 - Different network densities / average node degrees.
 - Different interrupt loads on simulated sensor nodes.
 - Different traffic loads as incurred by different network densities



• Number of Nodes vs. Memory Usage:

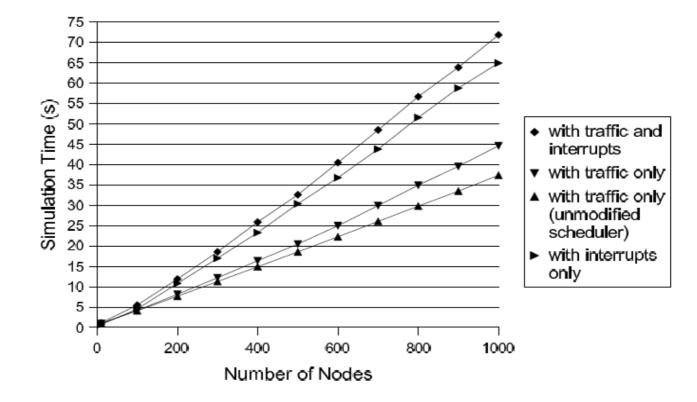


 Memory usage increases proportionally at about 345 kB per simulated ScatterWeb sensor node.

Georg Wittenburg, FU Berlin



• Number of Nodes vs. Simulation Time:

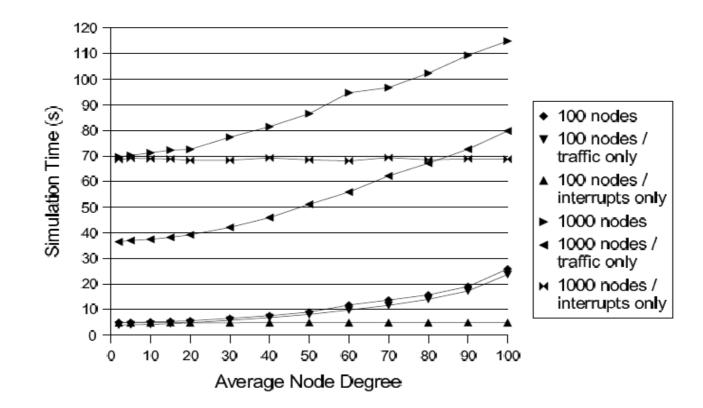


• Time to complete a simulation run increases proportionally with the number of sensor nodes.

Georg Wittenburg, FU Berlin



• Average Node Degree vs. Simulation Time:



 For test cases with network traffic, simulation time increases polynomially with the average node degree.

Georg Wittenburg, FU Berlin



- Background / Motivation / Idea
- Conceptual Perspective / Comparison
- Implementation Details
- A Look into Scalability Issues
- Results / Conclusion

Results (1)

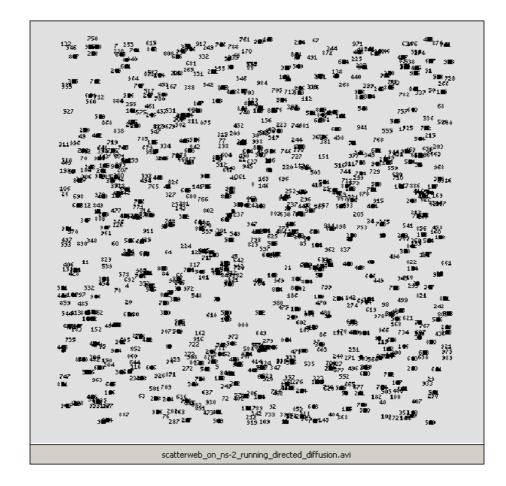


+ ////// + //PACH + ////// + + ////// + ////// + UINTLE + { + STL + UIN + UIN	lleInterestMesage: Handle incoming packets of type INTEREST_ME	SSAGE	→ InterestMessage passed already_known → dstid -*(UINT16*)0x15e4dcae * srcid *(UINT16*)0x15e4dcae payload	(InterestMessage 0 (UINT16 *) 0x15e 65535 (UINT16 *) 0x15e	9Struct *) 0x9c58dd0 4dcac 4dcae
+ //f	<pre>crestMessage - (struct interestnessagestruct) payload; irst of all we increment the hop count - and only here!!! crestMessage->hop counter+;</pre>	::::			
+ 1110	erestnessage->nop_counter++;		an an tha an an an taran t An	taran dara tara tara tara tara tara tar	andaranan darar darar Arabitetetetetetetetetetetetetetetetetetete
	<pre>ssed = consistencyCheck(*InterestMessage);</pre>		5]	Add <u>B</u> reakpoint
+ 11 + {	(!passed)		Location	Address	Add Watchpoint
	if (debug_level > DEBUG_HIGH)		🔶 in handlelnterestMessage	u 0x823b8f0	
+ +	<pre>{ Comm print("[DBG4]received Interest Message did not pass con</pre>	neiete	🔶 in Process_radioHandler(\	oi 0x82423a2	<u>R</u> emove
+++	Comm_print("[DBG4]message dump: \r\n"); Comm_print("[DBG4]dest ID:");	IISISC			<u>D</u> isable
+	Comm_print("%i",*dstid);				View Code
+	Comm_print(" src ID:"); Comm print("%i",*srcid);				
+	comm_princ(*1 , "SECIO);			••	<u>C</u> onditional

• The simulation can be used to implement and debug real ScatterWeb applications.

Results (2)





• Existing algorithms can be visualized for the first time.

Georg Wittenburg, FU Berlin

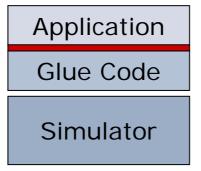


- Porting the API of an existing WSN platform to an existing simulator is possible.
- This approach has several advantages:
 - Faster development cycle with focus on algorithms.
 - Reuse of existing simulation technology.
 - Preserves abstraction / portability of API.
 - Applicable to other projects that use C for hardware abstraction.
- No *additional* run-time overhead is incurred.
- Next step is to quantify the accuracy of our simulation and compare it with the other approaches.



Thank you for your time! Any questions?





	New Execution Encalgoritt Window Settings Help			
inal.	C 6 18 7 8 7 8 7 8 8 8 8 9 10 10 10 10 10 10 10 10 10 10 10 10 10			
		C)		
	PACIET RANDLING FUNCTIONS			
+ 305	MADRENARTSANTING CONTRACTOR CONTRACTOR CONTRACTOR CONTRACTOR CONTRACTOR CONTRACTOR CONTRACTOR CONTRACTOR CONTRA		0	
1.00	mmm	aircady_known	0	
	number of the second state incoming packets of type INTEREST MESSAGE		(UNNT16*) 0:15	eridses
	manen -	- "UNITS") 015+6dcac		
•••	WTE handleleiteresidessay (UNWTE* dated, UTWTE* second, UTWE* payloa		(UNT16*) 0d5	stácae
	struct Interesticovereferent' Interesticover:	"(UNI16") 0.15e9dcae		-
	UNIN stready more it:	payload	(ON IT -) USSES	ISSN BOXBIELY
	UNTR passed - 0;			
	Incornationage - (annat: Intercationage/itract) payload;	_		
	//first of all we increment the hop count - and only here!!!! Internationsate-then countered:			
	meneralage and change of			
	passoi - consistency/hock ("InterestRessage) :			Add Election
	12 ([passed]			
	if (delage level a MERIC FICE)	Location	Address	- Add Weld pro
	(and sever a surve most)	 In handbilderocMiccapi(In Process redictiondarie 		Remove
	Comm print["[DBC4]received interest dermaps did not pass compute	 In Process, region landwry/ 	N 080.747.587	
	Comm print("[0554]mrosage chapp \r(n");	3H		Decision
•	Comm_print([][0664]detal.[10:7]); Comm_print(["Wi", "doubd);			Veerfinde
	Com print " er: 10:"1:			VANLOSA
	Component ("An", Arrend);	1		Continual
	H P	-		

More at:

http://www.inf.fu-berlin.de/inst/ag-tech/scatterweb_net/software/ns2.html



- P. Levis, N. Lee, M. Welsh, and D. Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003), 2003.
- H. Wu, Q. Luo, P. Zheng, B. He, and L. M. Ni. Accurate Emulation of Wireless Sensor Networks. In Proceedings of Network and Parallel Computing, IFIP International Conference, NPC 2004, pages 576–583, Wuhan, China, Oct. 2004.
- B. Titzer, D. Lee, and J. Palsberg. Avrora: Scalable Sensor Network Simulation with Precise Timing. In Proceedings of the Fourth International Conference on Information Processing in Sensor Networks (IPSN'05), pages 477–482, 2005.
- 4. The Network Simulator ns-2. <u>http://www.isi.edu/nsnam/ns/</u>.