Xiaohui Liu

Wayne State University
Department of Computer Science
5057 Woodward Avenue, Suite 3209
Detroit, Michigan 48202, USA

Phone: +1 313-318-8421
xiaohui@wayne.edu
http://www.cs.wayne.edu/xliu

Research Interests

Wireless and Sensor Networks: MAC, routing, and cross-layer protocols

Wireless Networked Control Systems: predictable, reliable, and real-time wireless networking for sensing and control

Education

2008–2014 (Expected) **Wayne State University**, *Ph.D. in Computer Science.*

2004–2008 **Wuhan University, China**, *B.S. in Computer Science.*

Awards

2012 **Outstanding Graduate Research Assistant (GRA) Award**, Wayne State University

2009 **Microsoft Imagine Cup US Software Design Top 15 Finalist** (out of about 2,000 teams)

2005 **National Scholarship**, China

2008, 2011, 2012 Graduate Research Travel Award, Department of Computer Science, Wayne State University

2006 Second Prize of Mathematics Contest, Wuhan University, China

2005 Second Prize in 10th "Ziqiang Cup" Extracurricular Contest of Science and Academics, Wuhan University, China

2005, 2006 Second Class Academic Scholarship, Wuhan University, China

2007 Third Class Academic Scholarship, Wuhan University, China

Publications

Journals

Hongwei Zhang, Xin Che, Xiaohui Liu, Xi Ju.
Adaptive Instantiation of the Protocol Interference Model in Wireless Networked Sensing and Control.
In ***ACM Transactions on Sensor Networks (ToSN)****, Volume 10 Issue 2, Article No. 28, January 2014.*

Xiaohui Liu, Hongwei Zhang, Qiao Xiang, Xin Che, Xi Ju.
Taming Uncertainties in Real-Time Routing for Wireless Networked Sensing and Control.
In ***IEEE Transactions on Smart Grid (TSG)****, special issue on "Smart Grid Communication Systems: Reliability, Dependability, and Performance", Vol. 4, No. 1, March 2013.*

Qiao Xiang, Jinhong Xu, Xiaohui Liu, Hongwei Zhang, Loren J. Rittle.
When In-Network Processing Meets Time: Complexity and Effects of Joint Optimization in Wireless Sensor Networks.
In ***IEEE Transactions on Mobile Computing (TMC)****, 10(10), pp. 1488-1502, October 2011.*

Conferences & Workshops

Xiaohui Liu, Hongwei Zhang, Qiao Xiang, Xin Che, Xi Ju.
Taming Uncertainties in Real-Time Routing for Wireless Networked Sensing and Control.
In ***13th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)****, 2012.*

Xiaohui Liu, Hongwei Zhang, Qiao Xiang.
Towards Predictable Real-Time Routing for Wireless Networked Sensing and Control.
In ***Cyber-Physical-Systems (CPS) Week Workshop on Real-Time Wireless for Industrial Applications (RealWin)****, 2011.*

Xin Che, Xiaohui Liu, Xi Ju, Hongwei Zhang.
Adaptive Instantiation of the Protocol Interference Model in Mission-Critical Wireless Networks.
In ***7th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)****, 2010.*

Qiao Xiang, Jinhong Xu, Xiaohui Liu, Hongwei Zhang, Loren J. Rittle.
When In-Network Processing Meets Time: Complexity and Effects of Joint Optimization in Wireless Sensor Networks.
In ***30th IEEE Real-Time Systems Symposium (RTSS)****, 2009.*

Posters & Demos

Hongwei Zhang, Xiaohui Liu, Chuan Li, Yu Chen, Xin Che, Feng Lin, Le Yi Wang, George Yin.
Poster Abstract: PRK-Based Scheduling for Predictable Link Reliability in Wireless Networked Sensing and Control.
In ***4th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)****, 2013*

Technical Reports

H. Zhang, C. Li, X. Liu, Y. Chen, X. Che, F. Lin, L. Y. Wang, and G. Yin.
PRK-based scheduling for predictable link reliability in wireless networked sensing and control.
In ***Technical Report DNC-TR-14-01, Wayne State University****, 2014.*

H. Zhang, X. Che, X. Liu, and X. Ju.
Adaptive instantiation of the protocol interference model in wireless networked sensing and control.
In ***Technical Report WSU-CS-DNC-TR-12- 01, Wayne State University****, 2012.*

X. Liu, H. Zhang, Q. Xiang, X. Che, and X. Ju.
Taming uncertainties in real-time routing for wireless networked sensing and control.
In ***Technical Report DNC-TR-11-04, Wayne State University****, 2011*

Q. Xiang, J. Xu, X. Liu, H. Zhang, and L. J. Rittle.
When in-network processing meets time: Complexity and effects of joint optimization in wireless sensor networks.
In ***Technical Report DNC-TR-09-01, Wayne State University****, 2009.*

Under Review

Hongwei Zhang, Xiaohui Liu, Chuan Li, Yu Chen, Xin Che, Feng Lin, Le Yi Wang, George Yin.
PRK-Based Scheduling for Predictable Link Reliability in Wireless Networked Sensing and Control.
In ***The 14th International Conference on Information Processing in Sensor Networks (IPSN)****, 2015.*

Xiaohui Liu, Hongwei Zhang.
A Maximal Concurrency and Low Latency Distributed Scheduling Protocol for Wireless Sensor Networks.
In ***IEEE Wireless Communications and Networking Conference (WCNC)****, 2015.*

Xiaohui Liu, Hongwei Zhang.
Real-Time Convergecast Scheduling in Lossy Multi-hop Wireless Sensor Networks.
In ***12th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)****, 2015*

Patent

Hongwei Zhang, Xiaohui Liu, Chuan Li, “PRK-Based Scheduling for Predictable Link Reliability", U.S. Provisional Application #61/788,445, International Application #PCT/US2014/27055

Wireless Networked Embedded Systems Developed (selected)

PRK-based Scheduling Protocol (PRKS)

PRKS guarantees the reliability of a wireless link is no less than requirement in the presence of interference while maximizing throughput. This is done by identifying the optimal parameter K in the PRK interference model, which in turn is formulated as a distributed control problem. **Numerous underpinnings of TinyOS, such as execution model and resource sharing mechanism, intend to shield programmers from many complexities of programming a concurrent system, which comes at the expense of nondeterministic execution. To cater to PRKS's needs of predictable timing, I tailor these fundamental underlying services.** More specifically, there are five primary challenges in the implementation: (1) the controller is distributed, and nodes can have inconsistent view of plant states due to diffusion delay; (2) the protocol is based on TDMA and requires global time synchronization, but the de facto time synchronization protocol FTSP for TelosB is buggy and yields intolerable synchronization errors; (3) most protocol codes run in asynchronous mode, causing concurrency issues; (4) TinyOS's task execution model makes execution delay very nondeterministic; (5) many computations are intensive and experience highly varying delays. **I single-handedly tackled these challenges as well as other challenges posed by extreme memory constraints of TelosB motes using TinyOS. In addition to PRKS, I also implemented 4 other protocols and 2 PRKS variants for comparison from the ground up:**

- *RTS-CTS*: a contention-based MAC protocol that uses CSMA/CA and RTS-CTS to ameliorate the impact of co-channel interference and hidden terminals;
- *RIDB*: a TDMA scheduling protocol that uses a TDMA protocol similar to the one used in PRKS and that uses the physical interference model to derive interference relations between nodes but ignores cumulative interference in networks;
- *CMAC*: a contention-based MAC protocol where a node transmits at a time instant only if the SINR of this transmission and the SINRs of other concurrent transmissions overheard by the node are above a certain threshold;
- *SCREAM*: a TDMA scheduling protocol using the SCREAM primitive to schedule concurrent transmissions according to the physical interference model;
- *PRKS-R*: same as PRKS but formulates the PRK model instantiation problem as a deadbeat PID regulation control problem instead of as a minimum-variance regulation control problem;
- *PRKS-L*: same as PRKS but directly use the linear model instead of its refined model.

Among these protocols, RTS-CTS and the default CSMA represent the protocol-model-based techniques in existing industry standards such as IEEE 802.15.4 and 802.11p; RIDB, CMAC, and SCREAM represent the techniques used in existing physical-model-based scheduling. Through measurement study in sensor network testbed NetEye and Indriya, we observe that PRKS enables predictably high link reliability (e.g., 95%) in different network and environmental conditions without a priori knowledge of these conditions, and, through local distributed coordination, PRKS achieves a channel spatial reuse very close to what is enabled by the state-of-the-art centralized scheduler while ensuring the required link reliability, and improves substantially over all other protocols in terms of reliability, delay, and throughput. The source code is publicly available at `https://github.com/xhliu/prks`.

Multi-Timescale Adaptation (MTA) Routing Protocol

MTA identifies minimal energy paths that can meet probabilistic deadlines of real-time traffic, given the notorious dynamics and uncertainties of path delays in wireless networks. **TinyOS is not designed with real-time support in mind, so it's not surprising to find many of its features detrimental to real-time routing, such as task and split-phase APIs, to name a few. Many core APIs are altered in a systematic way to enable real-time networking without devising an entirely new real-time OS.** More precisely, the following obstacles appear in realizing it on resource-scarce TelosB motes running TinyOS: (1) the standard TinyOS task queue is served in FIFO order so urgent tasks can be postponed by less urgent ones, causing deadline misses that can otherwise be avoided; (2) the default CC2420 radio stack is non-blocking and synchronous, making critical codes like packet transmission and reception susceptible to excessive delay; (3) tracking relative deadlines of traffic necessitates time synchronization across a link while there are bugs in the default packet-level time synchronization for TelosB, especially under heavy load; (4) the number of paths in a directed-acyclic-graph (DAG) maintained by MTA increases exponentially with the network diameter, making it impossible to store all of them in memory to choose the optimal path from. **Despite the aforementioned obstacles, I managed to implement the whole protocol independently. Besides MTA itself, I also implemented 4 other protocols and 7 MTA variants from scratch:**

- *MCMP*: a multi-path QoS routing protocol where end-to-end QoS requirements on reliability and timeliness are uniformly divided into per-hop reliability and timeliness requirements, upon which a node chooses the minimum number of next-hops to satisfy the per-hop requirements in data delivery;
- *MMSPEED*: the geographic routing protocol MMSPEED that routes and schedules packet transmissions based on nodes' distances to destinations, packet delivery deadlines, and mean link delays; MMSPEED also tries to improve packet delivery reliability by transmitting packets along multiple paths;
- *MMSPEED-CD*: same as MMSPEED but, instead of using the mean link delay, uses a conservative estimate of link delay that equals the sum of the mean delay and three times the standard deviation of the delay;
- *SDRCS*: similar to MMSPEED but, instead of using geographic distance, uses data-forwarding hop-count as the measure of distance, where the hop-count is computed based on received-signal-strength (RSS) between nodes; data forwarding is through receiver contention similar to that in opportunistic routing;
- *M-DS*: same as MTA but directly estimate path delay quantiles using non-parametric method P^2 and path delay samples that are collected in a distributed manner;
- *M-DB*: same as MTA but estimates the mean and variance of path delay directly through path delay samples;
- *M-ST*: same as MTA but estimates the mean and variance of path delay as the sum of the mean and variance of the sojourn time at each node of the path, without decomposing the sojourn time into the individual packet-times;
- *M-MD*: Same as MTA but maintains the data forwarding DAG based on mean link/path delay instead of link/path ETX;
- *M-mDQ*: same as MTA but forwards packets to the next-hop candidate with the minimum path delay quantile instead of the one with minimum path ETX;
- *mDQ*: same as M-mDQ but does not use the data forwarding DAG of MTA for stability control;
- *M-FCFS*: same as MTA but uses FCFS instead of EDF for intra-node transmission scheduling.

Two testbeds of 127+ motes have verified MTA's significant advantages over the state of the art for a variety of settings. The source code is publicly available at `https://github.com/xhliu/mta`.

Technical Skills

Programming languages	C, nesC(network embedded systems C), LaTex, Matlab, C++, Java, ASP, HTML, Javascript, PHP
Tools	TinyOS, gdb, MySQL
Standards	802.15.4, 802.11
Operating systems	Linux, Mac OS X, Windows

Experience

Fall 2009 – Present	**Research Assistant**, *Wayne State University*, Dr. Hongwei Zhang.
January 2010 – January 2012	**President**, *ACM Student Chapter at Wayne State University*.
Fall 2008 – Fall 2009	**Teaching Assistant**, *Wayne State University*.
December 2007 – March 2008	**Software Engineer**, *Wicresoft Company*.
May 2007 – December 2007	**Chief Development Officer**, *Trinity Studio*.

Professional Activities

TPC Member

ICCCN'14

Referee

ToN, ToSN, TPDS, TC, IPL, IJDSN, TPDS, JPDC

Sensys, MASS, SECON, WCNC, AHSN, ICCN, SAS, Globecom, SAC, ICC, MSWiM, NAS, CPNS, ADHOC, WMET, HiPC, QShine, TVT, MUE, WAVE

Reference

Hongwei Zhang.
Associate Professor
Department of Computer Science, Wayne State University
Suite 3010, 5057 Woodward Avenue, Detroit, Michigan 48202
+1 313 577 0731
hongwei@wayne.edu
`http://www.cs.wayne.edu/~hzhang`

Nathan Fisher.
Associate Professor
Department of Computer Science, Wayne State University
Suite 14200, 5057 Woodward Avenue, Detroit, Michigan 48202
+1 313 577 5421
fishern@wayne.edu
`http://www.cs.wayne.edu/~fishern`

Le Yi Wang.
Professor
Department of Electrical and Computer Engineering, Wayne State University
Room 3135, 5050 Anthony Wayne Drive, Detroit, Michigan 48202
+1 313 577 4715
lywang@wayne.edu
`http://www.ece.eng.wayne.edu/~lywang`