CONVINCE: A Cross-Layer Modeling, Exploration and Validation Framework for Next-Generation Connected Vehicles

Bowen Zheng[†] Chung-Wei Lin[‡] Huafeng Yu[§] Hengyi Liang[†] Qi Zhu[†]
[†]University of California, Riverside, Riverside, CA
[‡]Toyota InfoTechnology Center, Mountain View, CA
[§]Boeing Research & Technology, Huntsville, AL
(Invited)

ABSTRACT

Next-generation autonomous and semi-autonomous vehicles will not only precept the environment with their own sensors, but also communicate with other vehicles and surrounding infrastructures for vehicle safety and transportation efficiency. The design, analysis and validation of various vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) applications involve multiple layers, from V2V/V2I communication networks down to software and hardware of individual vehicles, and concern with stringent requirements on multiple metrics such as timing, security, reliability and fault tolerance. To cope with these challenges, we have been developing CONVINCE, a cross-layer modeling, exploration and validation framework for connected vehicles. The framework includes mathematical models, synthesis and validation algorithms, and a heterogeneous simulator for inter-vehicle communications and intra-vehicle software and hardware in a holistic environment. It explores various design options with respect to constraints and objectives on system safety, security, reliability, cost, etc. A V2V application is used in the case study to demonstrate the effectiveness of the proposed framework.

1. INTRODUCTION

Autonomous driving has made significant progress during the past decade. Many companies and academic institutions started testing their autonomous or semi-autonomous vehicles on real roads. Typical autonomous driving vehicles utilize a variety of sensors (e.g., LIDAR, radar, GPS, cameras and ultrasound sensors) to percept the environment, conduct real-time processing on collected data, make driving decisions through planning modules [10, 25, 36, 49], and send instructions to actuators from control modules for motions such as steering, accelerating and braking. However, accidents have been reported for both test autonomous vehicles and commercialized vehicles with semi-autonomous driving applications [1, 2]. The design and operation of a safe, reliable and secure autonomous-driving system still face tremendous challenges, in particular under stringent resource constraints for commercial vehicles.

First, modern automotive systems have become more complex than ever, in terms of both functionality and architecture. From the functional perspective, there is a wide range of emerging applications including autonomous functions and Advanced Driver Assistance Systems (ADAS), such as adaptive cruise control and lane keeping assist. To fulfill these applications, various software programs are implemented to play important roles in sensing, signal processing, control, decision making, etc. From year 2000 to 2010, embedded software increased from 2% to 13% of a vehicle's total value, and the number of lines of code increased from one million to more than ten million [11, 34, 41]. From the architectural perspective, the number of Electronic Control Units (ECUs) in a standard car has gone from 20 to over 50 in the past decade [11]. The traditional *federated architecture*, where each function is deployed to one ECU and provided as a black-box by Tier-1 supplier, is shifting to the *integrated architecture*, in which one function can be distributed over multiple ECUs and multiple functions can be supported by one ECU [14]. This leads to significantly more sharing and contention among software functions over multicore and distributed platforms. In addition, new computational components such as Field Programmable Gate Array (FPGA) [19, 43] and Graphical Processing Unit (GPU) [21, 31] as well as next-generation communication protocols such as those based on Ethernet [23, 24, 39, 40], have emerged to form a heterogeneous automotive platform.

Furthermore, there are a variety of objectives and metrics that need to be addressed during the design and operation of automotive systems, such as safety, performance, fault tolerance, reliability, extensibility and security. Many of these metrics are heavily influenced by system timing behavior [35, 42, 46], and often lead to conflicting requirements [12, 13, 18, 48, 47]. For instance, shorter sampling periods and end-to-end latencies of control loops usually lead to better sensing and control performance [13], but may be detrimental to schedulability, extensibility and even security (as there is less timing slack for adding strong security techniques [47]). It is important yet challenging to address these metrics in an integrated framework.

The problems become even more challenging when connected environments and applications are considered. Vehicle-

ICCAD '16, November 07-10, 2016, Austin, TX, USA
 ISBN 978-1-4503-4466-1/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2966986.2980078

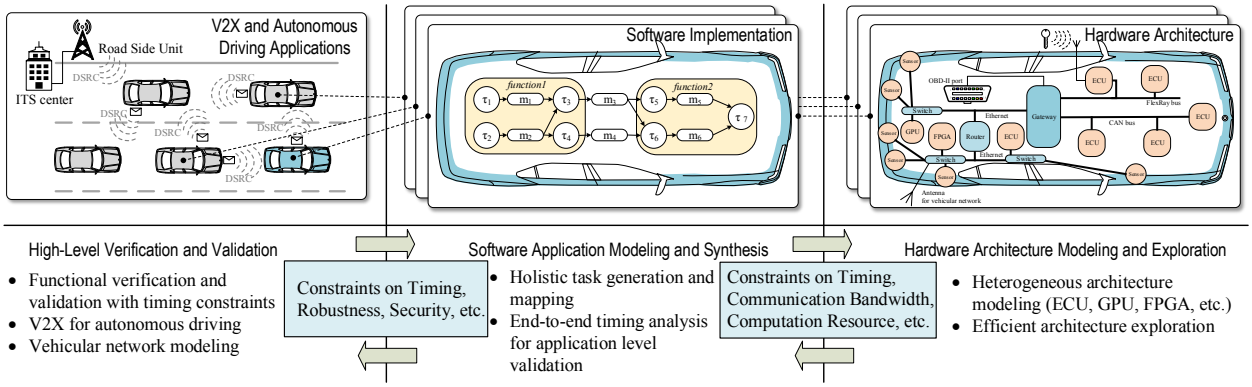


Figure 1: Cross-layer design for connected vehicles.

to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications (generalized as V2X communications) have been proposed to enhance driving safety and transportation efficiency as individual vehicles may suffer from blind spots and precision limitations. Standards such as Dedicated Short Range Communication (DSRC) are under the development at the United States [20, 27]. In the standard, Basic Safety Messages (BSMs) which contain vehicle state information (such as speed, acceleration and location) are exchanged among vehicles and surrounding infrastructures through wireless channels. Vehicles can estimate the position and movement of other vehicles based on the received messages and take early actions to avoid potential collisions or improve driving efficiency. However, designing a connected vehicle or a connected application faces many challenges, not only because of the increasing complexity but also because of the openness of the connected environment:

- *Timing*. The timing behavior of V2X communication is usually less predicative than that of in-vehicle networks since it is affected by the surrounding physical environment. As a results, it is more difficult to model and analyze the performance of a V2X network and guarantee the satisfaction of timing constraints which are extremely critical for automotive systems.
- *Robustness*. The connections between vehicles or infrastructures may not be stable. It is by the nature of wireless communication and moving vehicles. A connected application on a single vehicle or a set of vehicles should be robust enough to tolerate faults and deal with changing environments.
- *Security*. In-vehicle network security has been discussed in previous works [22, 28, 30, 38]. The open environment of V2X communications further broadens the potential attack surface. A connected vehicle application has to address security concerns while meeting other design constraints, in particular timing and resource constraints.

With these challenges, the modeling, exploration and validation of connected automotive systems should be considered *across system layers* including applications, software implementations and architecture platform. The concept of such cross-layer design is illustrated in Figure 1. The top layer is the application layer where V2X and autonomous driving applications are considered. In the application layer, functional verification and validation are done and constraints on timing, robustness and security are decomposed to individual vehicles. For example, if the application is Cooperative Adaptive Cruise Control (CACC), where every vehicle in the group communicates with other vehicles to adaptively maintain a safe distance from its preceding vehicle (referred to as *gap* in the rest of the paper), the performance mainly depends on timing, the error rate of the messages and the security level of the system [17]. Through verification and validation, constraints are decomposed to individual vehicles, for example, the constraints on end-to-end latency, the constraints on error correction ability and the constraints on security level. Inside each individual vehicle, these constraints guide the task generation and task to platform mapping at the software implementation layer. In this vision of cross-layer design, if the constraints obtained from the high-level application layer can not be fulfilled, the software implementation layer can provide feedbacks for the application layer to relax some constraints. The software implementation layer and the hardware architecture layer inside one individual vehicle also communicate with each other through the constraints on timing, communication bandwidth, computation resource, etc. Similarly, if the constraints can not be fulfilled after hardware exploration, the higher levels can trade-off among design metrics and relax some constraints.

To achieve the vision in Figure 1, we propose CONVINCE, a cross-layer modeling, exploration and validation framework for connected vehicles. The framework includes mathematical models, synthesis and validation algorithms, and a heterogeneous simulator for addressing inter-vehicle communications and intra-vehicle software and hardware in a holistic environment.

The rest of the paper is organized as follows. Section 2 introduces CONVINCE and its design methodology. Section 3 discusses a case study in CONVINCE for analyzing performance and security of an V2V application, and Section 4 presents its simulation results. Section 5 concludes the paper.

2. CONVINCE FRAMEWORK

The overview of CONVINCE is shown in Figure 2, including modeling, mathematical analysis, exploration, verification and validation components. Modeling is conducted across multiple layers, including the high-level V2X appli-

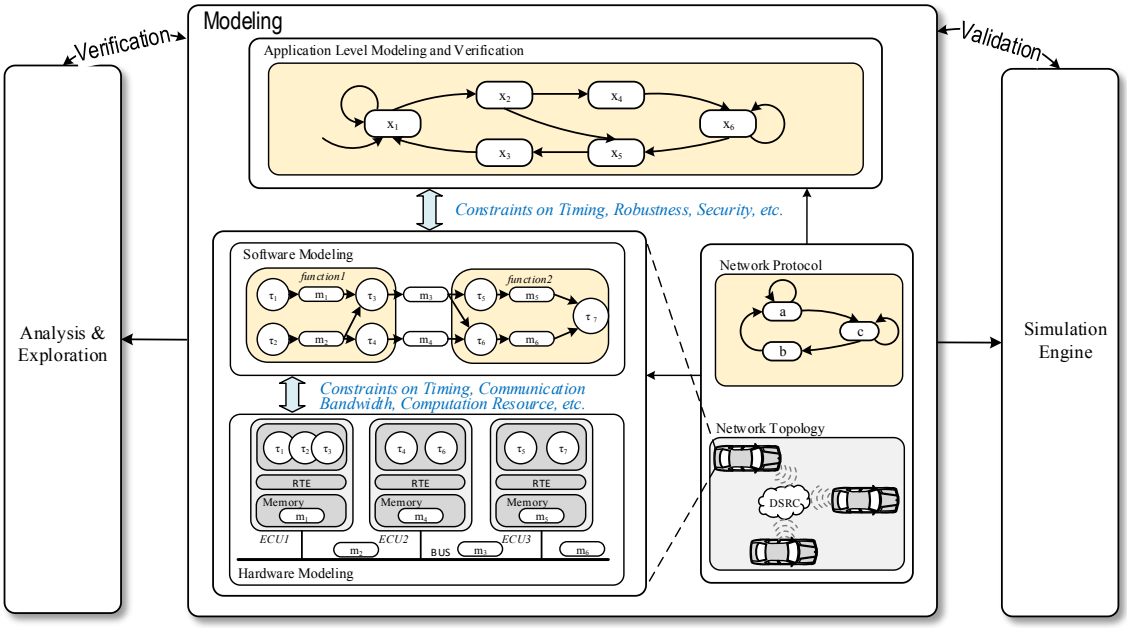


Figure 2: CONVINCE: cross-layer modeling, exploration and validation framework for connected vehicles.

cation modeling, the software modeling inside one vehicle, and the hardware modeling. These models can be abstracted and used by the *analysis and exploration engine* to optimize the design and verify whether design constraints are met. The models can also be leveraged by the *simulation engine* to validate system designs, identify potential issues and provide design insights.

In the rest of the section, timing property will be used as an example to demonstrate the modeling, analysis and exploration of the CONVINCE framework.

Computation Model: The software layer is captured by a set of tasks $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$. The tasks can be mapped to multiple computation units as shown in Figure 2. The timing property of each task τ_i is captured by a worst-case execution time C_{τ_i} (for a specific platform) and an activation period T_{τ_i}. Every task is required to finish its execution before its deadline (e.g., sometimes set as its period). Fixed-priority preemptive scheduling is modeled in the framework as every task is assigned a priority offline and lower priority tasks can be preempted by higher priority tasks. The worst-case response time r_{τ_i} (the longest time it may take to complete task τ_i) can be formulated as the following equation [50]:

$$r_{\tau_i} = C_{\tau_i} + \sum_{\tau_k \in hp(\tau_i)} \left\lceil \frac{r_{\tau_i}}{T_{\tau_k}} \right\rceil C_{\tau_k}. \quad (1)$$

The first term of the equation denotes the worst-case execution time C_{τ_i} and the second term represents the preemption time from higher priority tasks in set $hp(\tau_i)$ on the same computation unit.

Communication Model: In V2X and autonomous driving applications, messages are exchanged at different levels. At application level, messages are transmitted through wireless channels from one vehicle to others. Inside the vehicle, the messages can be transmitted on bus or exchanged through memory among modules and tasks. In autonomous driving system, the in-vehicle bus system can also be heterogeneous like CAN and Ethernet [23, 24, 39]. The analysis of the communication latency is essential as automotive systems are timing-critical systems and failures to fulfill the timing requirements may lead to catastrophic outcome.

Intra-Vehicle Communication: In our model, the message access delay for memories is modeled as a small constant, and the mathematical models to capture CAN bus and Ethernet are discussed below.

1) CAN Bus: CAN bus is prevalent in current automotive systems. The protocol is priority based and non-preemptive. The worst-case response time r_{m_i} for message m_i is as follows [50]:

$$r_{m_i} = C_{m_i} + B_{max} + \sum_{m_j \in hp(m_i)} \left\lceil \frac{r_{m_i} - C_{m_i}}{T_{m_j}} \right\rceil C_{m_j}. \quad (2)$$

The timing property of each message m_i is captured by worst-case transmission time C_{m_i} and period T_{m_i}. As CAN protocol is non-preemptive, the message may have to wait for the longest transmission time of any lower priority messages, denoted as B_{max}. The third term denotes the waiting time due to higher priority messages in set $hp(m_i)$.

2) Ethernet: Ethernet is discussed to be the potential candidate for autonomous driving, including Time-Sensitive Networking (Ethernet AVB) and Time-Triggered Ethernet (TTEthernet) [44]. Time-Sensitive Networking extends traditional full-switched network by adding eight priorities (three bits) for priority scheduling, and Credit-Based Shaping (CBS)

algorithm is used to select transmission schemes for different classes. The Time-Sensitive Networking classifies traffic into Class-A, Class-B, and best-effort class. Class-A has the highest priority and typically with 2 ms latency and Class-B has the second highest priority and typically with 50 ms latency [44]. The best-effort class assigns its traffic with the rest lower priorities. The packets with the same priority are queued in a FIFO in the corresponding class. We adopt the Compositional Performance Analysis (CPA) as shown in [15] to quantitatively analyze the worst-case timing behavior of Time-Sensitive Networking.

Time-Triggered Ethernet is also extended from the switched Ethernet by assigning the transmission of messages to time slots following the time division multiple access (TDMA) fashion. As TDMA scheme assigns time slots offline, it makes the message delay deterministic and predictable. However, synchronization protocol is needed to deal with clock jitter. Besides time-triggered communication, Time-Triggered Ethernet also provides rate-constrained messages and best effort messages that are event triggered. The rate-constrained messages are those with less strict timing requirement and best effort messages are for traditional Ethernet applications with less or no timing constraints. We adopt the TDMA analysis in [33] to quantitatively analyze the timing behavior of TDMA-based network.

Inter-Vehicle Communication: DSRC is the standard for vehicular communication in the United States. The protocol stack of Wireless Access in Vehicular Environments (WAVE) is developed for DSRC. The WAVE protocol stack supports two kinds of applications at the application layer: safety applications and Internet applications. For Internet applications, the transportation layer and network layer cover the traditional TCP/IP stack. For safety applications, the WAVE Short Message Protocol (WSMP) replaces TCP/IP stack for transportation layer and network layer. All applications share the same data link layer protocol and physical layer protocol.

As safety messages are time-critical, the IEEE 802.11p protocol that covers the data link layer and physical layer has been studied in [45]. The IEEE 802.11p protocol allocates seven 10 MHz wide channels for multi-channel operation, among which one control channel (CCH) is for safety communication only, and six service channels (SCH) for regular communication. To deal with the media access contention, the Enhanced Distributed Channel Access (EDCA) is utilized to classify the messages into four priority categories and set corresponding contention window and arbitration inter-frame spaces for the back-off procedure CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance). In [45], the authors establish two Markov chains for two different priority groups to analyze the delay distribution in the broadcast mode. We adopt the probability density function of message latency in [45] to analyze the latency for V2X communication.

Besides latency, packet loss is another major concern when designing safety-critical systems [17]. Although CSMA/CA has been adopted in the IEEE 802.11p protocol, it can only reduce the collisions instead of eliminating them. Furthermore, its performance can saturate if large amount of requests are generated or jamming is performed. According to the standard, if two broadcasting messages collide, both messages are lost and no retransmission will be scheduled. If vehicle-to-vehicle messages collide, the messages will be retransmitted within the limit of the retransmission times. If the maximum retransmission times have reached, the message will be discarded. Packet loss may also happen when a wireless communication channel is affected by fading and shadowing. Packet loss may significantly affect the performance of V2X safety applications as vehicles need timely information to predict danger and take actions.

Security Model: The emerging of autonomous driving and vehicular communication provides the attacker with a variety of attacking surfaces, including the On Board Diagnostics-II (OBD-II) port [29], the various sensors and the wireless communication interfaces such as DSRC, Bluetooth and keyless entry system [9]. The authors in [29] successfully compromised a real vehicle by hacking into its engine control system, brake control system, and other electronic components. The security-aware design for CAN-based and TDMA-based intra-vehicle network has been studied in [32, 33]. Besides intra-vehicle security, inter-vehicle communication brings in more concerns for safety applications. As summarized in [7, 16, 26, 37], the V2X security issues can be classified into the following categories.

- *Message falsification*: An attacker sends falsified messages to victim vehicles to make them behave as the attacker intends.
- *Impersonation*: An attacker uses fake identity to cheat other vehicles for malicious attacks, e.g., masquerade attack, replay attack and sybil attack.
- *Message tampering*: An attacker gathers, intercepts and/or tampers a message when routing it. Typical attacks include man in the middle attack, wormhole attack and blackhole attack.
- *Denial of service attack*: An attacker maliciously floods or jams communication channels to stop vehicles from sending or receiving messages in normal communication.
- *Privacy issues*: An attacker leverages location information, electronic IDs or other types of information for malicious purpose.

3. CASE STUDY

In this section and the next, we use Cooperative Adaptive Cruise Control (CACC) as a case study to demonstrate the effectiveness of CONVINCE in analyzing the impact of security attacks in vehicular communication and ultimately the application performance.

CACC Application: CACC is the technology that utilizes V2V wireless communication to enhance the traditional single-vehicle adaptive cruise control (ACC) by communicating with other vehicles to cooperatively maintain a safe gap. Platooning, where a leading vehicle leads a group of closely-following vehicles to move like a train, can be formed with CACC enabled vehicles. As platooning can maintain a shorter gap between vehicles and reduce speed variations, it may enhance traffic efficiency and reduce emission. In [8], the authors have designed and implemented a CACC platooning management protocol. In this case study, we will leverage this protocol to study the security issue across multiple layers.

In the protocol designed in [8], every CACC-enabled vehicle receives the acceleration of its preceding vehicle through V2V messages, and obtains the location and speed of the preceding vehicle from sensors such as radar. With these

information, each vehicle can maintain a safe gap to its preceding vehicle. As in [8], the equation to calculate the safe gap g_{safe} is

$$g_{safe} = 0.1v_f + \frac{v_f^2}{2D_f^{max}} - \frac{v_p^2}{2D_p^{max}} + 1.0, \quad (3)$$

where v_f denotes the speed of the following vehicle and D_f^{max} denotes the maximum deceleration of the following vehicle, and similarly, v_p denotes the speed of the preceding vehicle and D_p^{max} denotes the maximum deceleration of the preceding vehicle. The minimum gap required is 1.0m.

After receiving the location of the preceding vehicle d_p, the current gap between two vehicles can be calculated as $g = d_p - d_f - l_p$, where d_f is the location of the following vehicle, and l_p is the length of the preceding vehicle. Depending on g, the following vehicle may enter different modes and decide its acceleration.

1) Collision Avoidance Mode: If $g < g_{safe}$, the following vehicle will enter the collision avoidance mode. In this mode, the vehicle will decelerate with its maximum deceleration D_f^{max} until the gap becomes safe again. Therefore, in this mode, the new acceleration for the following vehicle is $a_{control} = D_f^{max}$.

2) Gap Control Mode: If $g \geq g_{safe}$, the following vehicle will enter the gap control mode. In this mode, the following vehicle follows the preceding vehicle to maintain a time gap T_{gap}. The desired acceleration a_{des} of the following vehicle can be calculated as following [8]:

$$a_{des} = 0.66a_p + 0.99(v_p - v_l) + 4.08(g - v_f T_{gap} - 2.0), \quad (4)$$

where a_p denotes the acceleration of the preceding vehicle and g is the current gap $g = d_p - d_f - l_p$. The actual acceleration to control the vehicle can be calculated as below [8]:

$$a_{control} = \frac{a_{des} - a_f}{\tau}\Delta t + a_f, \quad (5)$$

where τ is the controller delay and set as 0.4s. As in [8], $a_{control}$ is bounded by [-3, 3], and Δt is set as the sending rate 0.1s.

ACC Application: As stated in previous sections, packet loss and delay can happen in vehicular network. Upon packet loss or delay, the following vehicle cannot obtain the latest acceleration information of the preceding vehicle (a_p), and can only depend on the speed and location information of the preceding vehicle (obtained from its own sensors) to maintain a safe gap, i.e., entering the ACC mode. In the extreme cases, the preceding vehicle may fully brake with the maximum deceleration D_p^{max}. Therefore, we conservatively assume $a_p = -D_p^{max}$ when calculating the desired acceleration as in Equation (6). As ACC mode lacks acceleration information, the desired gap between vehicles should be larger. According to [8], the time gap T_{gap} is set to 0.55s for CACC gap control model and set to 1.2s for ACC gap control model. As a result, the safe gap of ACC becomes larger.

$$a_{des} = -0.66D_p^{max} + 0.99(v_p - v_l) + 4.08(g - v_f T_{gap} - 2.0) \quad (6)$$

Attacker Model: In the case study, we assume the attacker floods the wireless channels to impair the vehicular communication. The attacker can be a participant of the vehicular network or a malicious attacker from the road side. We study the impact of packet loss (due to flooding on V2X communication channels) on CACC application. The V2X messages are exchanged at a certain rate (set as 10 Hz in the experiments following [8, 27]), and packet loss may lead to outdated information for the following vehicle. If a message is not received within the time window (set as 0.1s in the experiments), the following vehicle has to rely on its own sensors for deciding the safe gap, and CACC is in fact downgraded to ACC as discussed above. We also assume that CACC will be restarted once messages can be successfully received during the time window.

In Section 4, we will demonstrate in our experiments, how different strengths of flooding attack may lead to different degrees of packet loss and ultimately deteriorate the system performance (evaluated based on the gap between vehicles).

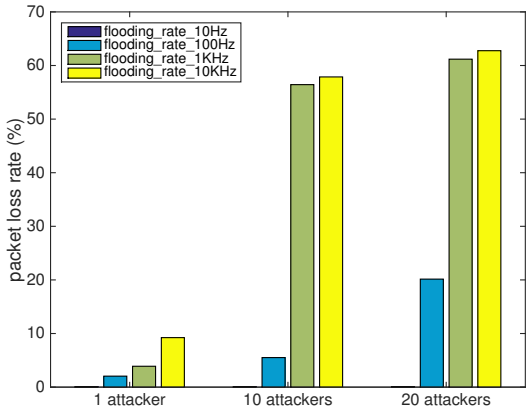
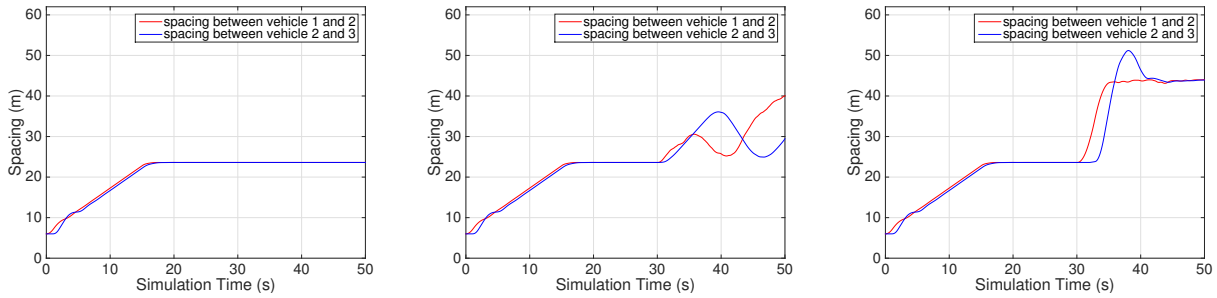


Figure 3: Packet loss rate under different strengths of flooding attack. During the simulation, 50 vehicles are uniformly distributed on a road of length 300m and normal packet sending rate is 10 Hz.

4. SIMULATION RESULTS

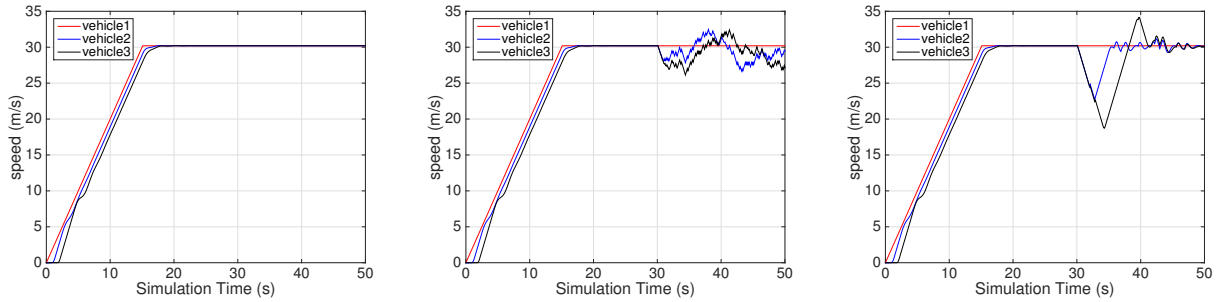
We leverage VENTOS (VEhicular NeTwork Open Simulator) [8] for our simulation, which itself is an integration of several tools with CACC platooning implemented. VENTOS is based on the structure of Veins [6], an simulator that combines the open source traffic simulator SUMO [5] and open source network simulator OMNeT++ [4] with WAVE protocol stack implemented. In addition to OMNeT++, we also leverage NS-3 [3] for packet level simulation of V2X communication networks.

Packet Loss Rate under Flooding Attack: We first study the relationship between the strength of the flooding attack and the packet loss rate. In this study, we assume there are 50 vehicles distributed on a road of length 300m. The transmission power of the DSRC module is 26dBm. The EDCA related parameters are set as follows (CWmin: minimum contention window size; CWmax: maximum contention window size; AIFSN: arbitration inter-frame spaces): CWmin = 15, CWmax = 1023, and AIFSN = 3. The flooding message is of length 500 bytes.



(a) no attack (b) 40% packet lost (30s to 50s) (c) 90% packet lost (30s to 50s)

Figure 4: Spacing-time diagram of 3 vehicles in CACC under different strengths of flooding attacks.



(a) no attack (b) 40% packet lost (30s to 50s) (c) 90% packet lost (30s to 50s)

Figure 5: Speed-time diagram of 3 vehicles in CACC under different strengths of flooding attacks.

We assume some of the vehicles within the 50-vehicle group are malicious attackers. We classify the simulations into three scenarios: 1 attacker, 10 attackers and 20 attackers. In each scenario, every attacker applies the flooding attack with the same strength that varies from 100 Hz (i.e., sending flooding packets at a rate of 100 Hz) to 1 KHz to 10 KHz. The normal vehicles send packets at a rate of 10Hz.

We use NS-3 to simulate these scenarios, and the results of flooding attack regarding packet loss rate are shown in Figure 3. From the figure we can see that for normal traffic, the packet loss rate is around zero. When malicious flooding attack is conducted, the packet loss rate can reach 63% in this case study. We can observe that as the number of attackers increase and/or the attacking strength increases, the packet loss rate also increases (and could be even higher than 63%).

The results show that significant packet loss can be caused by malicious attacks. Next, we quantitatively study how packet loss ultimately affects the CACC performance.

CACC Performance Deterioration under Attack: We assume there are three vehicles joining the CACC application, namely Vehicle 1, Vehicle 2 and Vehicle 3. At time zero, the vehicles are aligned in a line with a gap of 1m between each two consecutive vehicles. Vehicle 1 is set as the leading vehicle. Vehicle 2 and Vehicle 3 will automatically follow Vehicle 1 and maintain safe gaps. The simulation has two phases:

- *Warming up*: From 0s to 15s, Vehicle 1 constantly accelerates with an acceleration of $2m/s^2$, and reaches $30m/s$ at time 15s. Vehicle 2 and Vehicle 3 also accelerate according to the CACC protocol.
- *Keeping speed*: From 15s to 50s, Vehicle 1 stops accelerating and keeps the speed $30m/s$. Vehicle 2 and Vehicle 3 can catch up with Vehicle 1 during this phase.

Then, flooding attack is scheduled at time 30s during the keeping speed phase with different strengths. The performance deterioration due to flooding is demonstrated in Figure 4 and Figure 5.

Figure 4 (a) and Figure 5 (a) demonstrate the normal behavior without any flooding attack. Figure 4 is a spacing-time diagram, where y-axis denotes the spacing between each two consecutive vehicles on the road. The spacing includes the gap between vehicles and the length of one vehicle (set as 5m in our experiments), i.e., it is the distance from the preceding vehicle's front bumper to the following vehicle's front bumper. Figure 5 demonstrates the vehicle speeds as the simulation time increases. In Figure 4 (a), the spacing gradually increases to around 24m, indicating the CACC protocol is functioning well. In Figure 5 (a), we can observe that during the first 15s, Vehicle 2 and Vehicle 3 are catching up with Vehicle 1. During 15s to 50s, Vehicle 2 and Vehicle 3 also reach the cruise speed of Vehicle 1 and maintain the spacing around 24m and a speed at $30m/s$.

Figure 4 (b) and Figure 5 (b) demonstrate the CACC performance with flooding attack that causes 40% packet loss. Since flooding attack starts from time 30s, the curves are the same as the normal behavior case from 0s to 30s. We can observe that after flooding attack, the vehicle spacing in Figure 4 (b) oscillates around 30m. From Figure 5 (b) we

can observe that after the attacking at time 30s, Vehicle 2 and Vehicle 3 can not follow Vehicle 1 smoothly. Instead, they have to speed up or slow down constantly. This is because some packets are lost, and thus Vehicle 2 and Vehicle 3 have to switch between the CACC safe gap and ACC safe gap. They can correct their acceleration when latest packets arrive, however the driving efficiency of the individual vehicles and the entire system has already been affected.

Figure 4 (c) and Figure 5 (c) demonstrate the CACC performance with flooding attack that causes 90% packet loss. Similar to the 40% packet loss case, the curves are the same as the normal behavior case for the first 30s. When flooding starts at 30s, the vehicle spacing significantly increases to around 44m, as most of the packets are lost. Vehicle 2 and Vehicle 3 can not follow Vehicle 1 smoothly. In this case, the vehicles are in ACC mode most of the time and the driving efficiency has been severely reduced.

5. CONCLUSION

In this paper, we introduce CONVINCE, a cross-layer modeling, exploration and validation framework for connected vehicles. In the framework, computation, communication (including both intra-vehicle and inter-vehicle communication), and system metrics such as timing and security are quantitatively modeled for design space exploration, validation and verification. We use CACC as a case study to demonstrate the usage of CONVINCE for analyzing the timing and security of V2V applications. Such analysis sets the foundation for our on-going work on exploring and validating security designs of connected vehicles (with integration of in-vehicle computation and communication models).

Acknowledgments

This work is supported in part by the Office of Naval Research grants N00014-14-1-0815 and N00014-14-1-0816, and the National Science Foundation grant CCF-1553757.

6. REFERENCES

[1] As U.S. investigates fatal Tesla crash, company defends autopilot system. http://www.nytimes.com/2016/07/13/business/tesla-autopilot-fatal-crash-investigation.html?_r=0. Accessed: 2016-7-21.

[2] Google self-driving car caught on video colliding with bus. https://www.theguardian.com/technology/2016/mar/09/google-self-driving-car-crash-video-accident-bus. Accessed: 2016-7-21.

[3] NS-3. https://www.nsnam.org/.

[4] OMNeT++. https://www.nsnam.org/.

[5] SUMO. http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/.

[6] Veins. http://veins.car2x.org/.

[7] M. S. Al-kahtani. Survey on security attacks in vehicular ad hoc networks (VANETs). In *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*, pages 1–9. IEEE, 2012.

[8] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal. Platoon management with cooperative adaptive cruise control enabled by vanet. *Vehicular Communications*, 2(2):110–123, 2015.

[9] S. Bono, M. Green, A. Stubblefield, A. Juels, A. D. Rubin, and M. Szydlo. Security analysis of a cryptographically-enabled rfid device. In *USENIX Security*, volume 5, pages 1–16, 2005.

[10] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. springer, 2009.

[11] R. N. Charette. This Car Runs on Code. *IEEE Spectrum*, February 2009.

[12] P. Deng, F. Cremona, Q. Zhu, M. Di Natale, and H. Zeng. A Model-Based Synthesis Flow for Automotive CPS. In *Cyber-Physical Systems (ICCPS), 2015 ACM/IEEE International Conference on*, pages 198–207, April 2015.

[13] P. Deng, Q. Zhu, A. Davare, A. Mourikis, X. Liu, and M. Di Natale. An efficient control-driven period optimization algorithm for distributed real-time systems. *IEEE Transactions on Computers*, PP(99):1–1, 2016.

[14] M. Di Natale and A. Sangiovanni-Vincentelli. Moving From Federated to Integrated Architectures in Automotive: The Role of Standards, Methods and Tools. *Proceedings of the IEEE*, 98(4):603 –620, april 2010.

[15] J. Diemer, D. Thiele, and R. Ernst. Formal worst-case timing analysis of ethernet topologies with strict-priority and avb switching. In *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*, pages 1–10. IEEE, 2012.

[16] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero. VANET security surveys. *Computer Communications*, 44(0):1–13, 2014.

[17] Y. P. Fallah and M. K. Khandani. Analysis of the coupling of communication network and safety application in cooperative collision warning systems. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, ICCPS '15, pages 228–237, New York, NY, USA, 2015. ACM.

[18] L. Guo, Q. Zhu, P. Nuzzo, R. Passerone, A. Sangiovanni-Vincentelli, and E. Lee. Metronomy: A Function-Architecture Co-Simulation Framework For Timing Verification Of Cyber-Physical Systems. In *Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2014 International Conference on*, pages 1–10, Oct 2014.

[19] Y. Han and E. Oruklu. Real-time traffic sign recognition based on zynq fpga and arm socs. In *IEEE International Conference on Electro/Information Technology*, pages 373–376, June 2014.

[20] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, and J. Wang. Vehicle-to-vehicle communications: Readiness of V2V technology for application. Technical report, 2014. National Highway Traffic Safety Administration, DOT HS 812 014.

[21] F. Homm, N. Kaempchen, J. Ota, and D. Burschka. Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 1006–1013, June 2010.

[22] T. Hoppe, S. Kiltz, and J. Dittmann. Security threats to automotive CAN networks—practical examples and selected short-term countermeasures. In *International Conference on Computer Safety, Reliability, and Security*, pages 235–248, 2008.

[23] IEEE. IEEE standard for local and metropolitan area networks - timing and synchronization for

time-sensitive applications in bridged local area networks. *IEEE Std 802.1AS-2011*, pages 1–292, March 2011.

[24] IEEE. IEEE approved draft standard for a transport protocol for time sensitive applications in a bridged local area network. *IEEE P1722/D16, November 2015*, pages 1–247, Jan 2015.

[25] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo. Development of autonomous car - part I: distributed system architecture and development process. *Industrial Electronics, IEEE Transactions on*, 61(12):7131–7140, 2014.

[26] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *Communications Surveys Tutorials, IEEE*, 13(4):584–616, Fourth 2011.

[27] J. B. Kenney. Dedicated short-range communications (DSRC) standards in the United States. *Proceedings of the IEEE*, 99(7):1162–1182, 2011.

[28] P. Kleberger, T. Olovsson, and E. Jonsson. Security aspects of the in-vehicle network in the connected car. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 528–533, June 2011.

[29] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al. Experimental security analysis of a modern automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462. IEEE, 2010.

[30] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *IEEE Symposium on Security and Privacy (SP)*, pages 447–462, May 2010.

[31] C. Lee, S. W. Kim, and C. Yoo. VADI: GPU virtualization for an automotive platform. *IEEE Transactions on Industrial Informatics*, 12(1):277–290, Feb 2016.

[32] C.-W. Lin, Q. Zhu, C. Phung, and A. Sangiovanni-Vincentelli. Security-aware mapping for CAN-based real-time distributed automotive systems. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 115–121, Nov. 2013.

[33] C.-W. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli. Security-aware mapping for TDMA-based real-time distributed systems. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 24–31, Nov. 2014.

[34] J. P. MacDuffie and T. Fujimoto. Why Dinosaurs Will Keep Ruling the Auto Industry. *Harvard Business Review*, 88(6):23–25, 2010.

[35] F. Mueller. Challenges for Cyber-Physical Systems: Security, Timing Analysis and Soft Error Protection. In *High-Confidence Software Platforms for Cyber-Physical Systems (HCSP-CPS) Workshop, Alexandria, Virginia*, page 4, 2006.

[36] U. Ozguner, T. Acarman, and K. Redmill. *Autonomous ground vehicles*. Artech House, 2011.

[37] M. Raya, P. Papadimitratos, and J.-P. Hubaux. Securing vehicular communications. *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, 13(LCA-ARTICLE-2006-015):8–15, 2006.

[38] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar. Security and privacy vulnerabilities of in-car wireless networks: a tire pressure monitoring system case study. In *USENIX Conference on Security*, pages 21–21, 2010.

[39] SAE. Time-Triggered Ethernet. *SAE Standard AS6802*, November 2011.

[40] F. Sagstetter, S. Andalam, P. Waszecki, M. Lukasiewycz, H. Stähle, S. Chakraborty, and A. Knoll. Schedule integration framework for time-triggered automotive architectures. In *Proceedings of the 51st Annual Design Automation Conference*, DAC '14, pages 20:1–20:6, New York, NY, USA, 2014. ACM.

[41] A. Sangiovanni-Vincentelli. Quo Vadis, SLD? Reasoning About the Trends and Challenges of System Level Design. *Proceedings of the IEEE*, 95(3):467–506, March 2007.

[42] A. Sangiovanni-Vincentelli and M. Di Natale. Embedded System Design for Automotive Applications. *Computer*, 40(10):42 –51, 2007.

[43] F. Schwiegelshohn, L. Gierke, and M. HÃijbner. Fpga based traffic sign detection for automotive camera systems. In *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2015 10th International Symposium on*, pages 1–6, June 2015.

[44] T. Steinbach, H.-T. Lim, F. Korf, T. C. Schmidt, D. Herrscher, and A. Wolisz. Tomorrow's in-car interconnect? a competitive evaluation of IEEE 802.1 AVB and Time-Triggered Ethernet (AS6802). In *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pages 1–5. IEEE, 2012.

[45] Y. Yao, L. Rao, X. Liu, and X. Zhou. Delay analysis and study of IEEE 802.11 p based DSRC safety communication in a highway environment. In *INFOCOM, 2013 Proceedings IEEE*, pages 1591–1599. IEEE, 2013.

[46] S. Ying et al. Foundations for Innovation in Cyber-Physical Systems. In *Workshop Report, Energetics Incorporated, Columbia, Maryland, US*, 2013.

[47] B. Zheng, P. Deng, R. Anguluri, Q. Zhu, and F. Pasqualetti. Cross-layer codesign for secure cyber-physical systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(5):699–711, May 2016.

[48] B. Zheng, Y. Gao, Q. Zhu, and S. Gupta. Analysis and optimization of soft error tolerance strategies for real-time systems. In *Proceedings of the 10th International Conference on Hardware/Software Codesign and System Synthesis*, pages 55–64. IEEE Press, 2015.

[49] B. Zheng, H. Liang, Q. Zhu, H. Yu, and C.-W. Lin. Next generation automotive architecture modeling and exploration for autonomous driving. *IEEE Computer Society Annual Symposium on VLSI*, 2016.

[50] Q. Zhu, H. Zeng, W. Zheng, M. Di Natale, and A. Sangiovanni-Vincentelli. Optimization of task allocation and priority assignment in hard real-time distributed systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 11(4):85, 2012.