Chapter 6

Conclusion and Future Work

6.1 Summary of contributions

We have investigated formal frameworks for specification, construction, and performance evaluation of protocol behavior in mobile and embedded systems. Protocol behavior is an important aspect of embedded systems since such systems are typically networked for the purposes of remote monitoring and control. Formal approaches are important for analyzing protocol behavior in such systems owing to their applicability in performing safety-critical functions such as medical drug-delivery. Today there is an increasing interest in networked embedded systems that are evolving into the next generation cyber-physical systems [82]. Our work directly leads to a better understanding of such systems, for example, those operating in safety-critical automotive environments.

The $\pi Z$ language is a contribution to the study of process calculi for mobile systems such as the $\pi$-calculus. It shows how we can introduce the concept of state and durable communication into this framework. Such an extension is important because it enables a succinct formulation of a number of problems such as message queues, filters, and stateful communication interfaces. The hidden node problem discussed in the thesis illustrates the application of mobility, statefulness, and data abstraction facilities offered by $\pi Z$ in modeling protocol behavior in a ‘real-world’ mobile system. One of the contributions of the thesis lies in showing that the important properties of $\pi$-calculus and $Z$, namely, bisimilarity and data refinement, are captured in $\pi Z$. The labelled semantics for $\pi Z$ is compositional in nature. As
results from [11] show, a compositionally defined labelled semantics facilitates proofs using straightforward induction, which could be mechanized in a theorem proving environment.

As internet-enabled devices are increasingly replacing legacy systems for performing critical functions such as real-time sensing and control, we investigated construction of a formally verifiable internet protocol stack. The formal verification approach, namely the split verification approach, discussed could be used to construct and verify internet protocol stacks having strong robustness guarantees, that can be used for performing critical functions. Split verification is a strategy for program verification wherein tactical operation-level properties are expressed as data refinement conditions and verified using theorem-proving and global control flow properties are expressed using temporal logic and verified using model-checking. Since operation-level properties are represented as abstract specifications, independent of any particular protocol stack implementation, we could build a library of property specifications which are re-usable (for verification) across multiple stack implementations. Though split verification approach was developed in the context of internet protocol stacks, it could be used for verifying other types of programs as well that could be constructed using our C++ subset.

In order to understand and analyze the performance of embedded protocol stacks under contention scenarios, we investigated performance analysis of real-time protocol stacks under a specific form of contention-based attack, namely indirect-contention-in-hosts (ICiH) attacks. Understanding and mitigating ICiH attacks are important as Internet users are increasingly relying on VoIP and other forms of real-time media for communication, information, and entertainment. Apart from targeting end devices such as VoIP phones, the ICiH attacker could also target intermediate systems that are in the real-time media pathway such as media gateways. Since ICiH exploits the degree of interference between RTMS and other concurrent services, such attacks could be hard to identify and mitigate. Operation-trace analysis and the mitigation strategies described in this thesis provide a platform for further research in this area.
6.2 Directions for Future Work

Some of the future directions for πZ are summarize below.

- We intend to investigate symbolic transitions for πZ. This will allow us to explore late and open bisimilarities [67] for πZ processes.

- It appears that we may weaken the condition for data refinement that the two datatypes must have identical signatures (Definition 10, Section 3.4) to a weaker condition of having compatible signatures. Owing to the possibility of mutual references in πZ datatype signatures, formally specifying signature compatibility seems non-trivial. We intend to investigate this in the context of πZ type systems.

- Developing a distributed version of πZ by adding the notion of locations and local resources as in [42] is also an interesting future work.

- Model checking finite control π-calculus has been shown to be decidable [23]. Investigating decidable fragments of πZ for model checking purposes as well as developing practical automated tools such as the π-calculus’ mobility model checker [98], is also an important area of future work.

For the split verification, the choice of C++ lowers barrier to adoption as well as results in efficient and practical implementations. We are encouraged by the success of SystemC [38] in this respect. A future work would be the construction and verification of a complete embedded TCP/IP stack with formally verified security properties.

Though ICiH attacks motivated the development of operation-trace analysis, the notion of operation traces may be used in a more general setting as well. We are investigating the use of operation traces as a means to construct experimental models for protocol stacks. By expressing relevant properties as temporal logic formulae we could automatically check whether traces in these experimental models satisfy those properties.

Today’s embedded systems are pre-cursors to the next generation cyber-physical systems [82] in which computing devices are closely integrated to the physical environment. We intend to investigate formalisms for modeling and analyzing cyberphysical systems [82] from within
the frameworks of both process-calculi as well as temporal logics. This necessitates studying timed and hybrid extensions to the formalisms that we have investigated in this thesis.