Software-Defined Cyber-Physical Multinetworks

Zhijing Qin*, Ngoc Do*, Grit Denker†, Nalini Venkatasubramanian*
* University of California, Irvine, USA † SRI International, USA

***Abstract*—Advances in sensing technologies, mobile and pervasive computing, and cyberphysical systems are making new modalities of information and new channels of communication available. We discuss challenges in leveraging heterogeneous networking technologies to create resilient and flexible infrastructure for information collection and information dissemination in cyber-physical environments. We will also discuss how middleware technologies can assist in supporting cross-layer interactions amongst (a) the devices that collect and receive information, (b) multiple networks that communicate the content, (c) the platforms that process and store the information and (d) applications that use the information for diverse purposes. Drawing on examples from cyber-physical systems, we show how multinetworks can help drive semantic middleware to incorporate diverse sensors and inputs in a structured manner to generate situational awareness.. In particular, we discuss the role of multinetworks in emergency situations (whether natural hazards or manmade disasters) where infrastructure is typically unavailable or partially damaged. Multinetworks provide more reliability and better performance of information sharing and alert dissemination to responders and citizen at large than traditional single communication networks. The reason is that multinetworks allow combining connectivities that open up new possibilities for resilient, adaptive and scalable societal scale systems of the future.**

I. Introduction

Recent advances in sensing, networking, and embedded computing technologies have made it possible to create Cyber-Physical Systems (CPS) that are capable of sensing and affecting their environment for improved utility. A common definition of a cyber-physical system is one that integrates computing and communication capabilities with the monitoring and/or control of entities in the physical world dependably, safely, securely, efficiently and in real-time. CPS technologies can be used to create situation-aware [1], and often safety- or mission-critical ecosystems and services [2] with increased capabilities and performance. CPS systems are beginning to have an impact on multiple domains that impact our day-to-day life including - intelligent transportation systems (air and ground), smart power grids, structural monitoring and control of civil infrastructures such as bridges and dams; medical/healthcare systems (for assisted living, patient monitoring in hospitals, automated laboratories); smart spaces (buildings with surveillance and microclimate control); smart agriculture; flexible manufacturing systems (with self assembling structures); systems and processes used in defense, homeland security and emergency response (ad hoc ground/airborne combat teams, intelligent firefighting etc.).

A popular view of Cyber-Physical Systems (CPS) is that of physical devices that are controlled by cyber components. This underlying model assumes a loop where sensing components at the device-end gather data that is analyzed by software entities that determine what to change and trigger the necessary controls to actuate the desired change. Our goal in this paper is to bring to the forefront the role of networks (in fact, multiple heterogeneous networks) in a CPS system. We argue that a CPS network is a collection of multiple, heterogeneous physical networks that must work in unison to realize the overall goal of the CPS application. We refer to these networks as CPS multinetworks - for example healthcare monitoring systems use a combination multiple networks (BlueTooth, adhoc, WiFi, wired, ultrasound) to capture and communicate patient state; intelligent transportation systems use mixed networks (WiFi/DSRT, VANETS, cellular, onboard vehicular nets) to improve travel times; many of the above-mentioned networks are combined with SCADA systems to provide energy-efficient buildings. The multiple physical layers over which CPS communication occurs across devices may include wired Ethernet (optical fibres, coax), wireless spectrum (RF, IR, microwave), with varying physical features and different protocols/encoding techniques (MIMO, MAC layer). Physical properties may be unique to a single network (RF/IF properties, MIMO-related, fading of RF signals in 802.11) ; physical properties/phenomena may also be a result of composite networks (e.g. channel interference across WiFi and Bluetooth). Supporting resilient communication in a multinetworks is complex – an interesting observation is that multinetworks are both the source of the problem and its solution. The presence of interfering networks hampers communication; alternately, the different physical media provide alternate communication opportunities when a network is overloaded or has failed. Explicitly mapping networks into the CPS infrastructure is interesting since networks have multiple layers that span both the cyber and physical aspects. Individual access networks have physical attributes, some of which can be controlled. In the following we refer to CPS that consists of broad variety of devices and multinetworks linking them as Cyber-Physical Multinetwork Systems(CPMSs). In this paper, we illustrate the role of upcoming software defined networking (SDN) philosophy in designing CPMS. Using specific use cases of data collection and data dissemination in a smartspace CPS, we argue how a layered, software inspired approach is well suited for CPMS applications.

II. Cyber-Physical Multinetwork Systems Meet Software-Defined Networks

As mentioned before, the heterogenity of networks and hardware resources in CPMS and the diversity of services and applications running on those devices and networks is both an opportunity and challenge. The opportunity is that for any set of given application requirements, there are various potential solutions that make use of different network and device resources to achieve the needs. The challenge is in coordinating and optimizing the use of the various resources (device, networks, applications) with the goal of satisfying as many tasks as possible. Additionally, given the dynamically changing nature of the underlying environment (e.g.mobile users, congested networks, application generated data), the determination of how the various devices can adapt the use of the multiple networks for delivery of information is further complicated. To address these concerns and challenges, we envision a controller based framework that is inspired by the Software-Defined Network (SDN) paradigm.

Software-Defined Networking (SDNs)[3], as the name suggests, is an emerging software-centric approach to designing network architectures that are dynamic, manageable, cost-effective, and adaptive; it is well suited to the high-bandwidth, dynamic nature of today's content and applications. We conjecture that the SDN paradigm is a good candidate to help solve the resource management needs of CPMSs for multiple reasons.

- SDN allows for a clear separation of concerns between services in the control plane (that makes decisions about how traffic is managed) and the data plane (actual mechanisms for forwarding traffic to desired destinations). The decoupling encourages abstractions of low-level network functionality into higher level services and consequently simplifies the task of network administrators.
- SDN mechanisms aim to provide a balance between the degree of centralized control/coordination through the presence of an explicit SDN controller and decentralized operation through flow-based routing and rescheduling within the network components; this balance is realized via interactions between controllers and controlled devices.

However, the current realization of SDN technologies are still far from addressing the heterogeneous and dynamic needs of CPS multinetworks. The popular use of SDN technologies today is in Data Center Networks (DCN) [4][5], where the focus is on the collection of specific network statistics (e.g. bandwidth consumption) from nodes networked via fast interconnects within the data center. In contrast, a typical CPMS setting gathers state information from devices distributed over a more loosely coupled (and possibly wide area) network. Second, performance metrics of interest in CPMS go beyond bandwidth consumption; with more heterogeneous and time-sensitive traffic as in CPMS, reducing the collection overhead while keeping the effectiveness of the collected data needs is equally important. Unlike the case of DCNs, whose network requirements primarily revolve around link utilization and throughput, CPMSs settings present additional timing related needs - such as delay, jitter, packet loss, throughput etc.. Third, unlike the situation in a data center network, link and node capabilities in a CPMS are very heterogeneous. This implies that the single objective optimization techniques in DCN flow scheduling, such as bin packing [4]and simulated annealing [5] are not directly applicable in CPMS. Finally, the nature of interactions in current realizations of SDN (e.g. Openflow [6]) is limited to *south-bound*, i.e.lower layer interactions, i.e. between controller and devices such as switches. The so-called *north-bound* interactions between applications/service and controller have received much less attention and are not standardized [7]. Although there are proposals [8], [9] that advocate the use of a network configuration language to express policies such as "ban a device if its usage over the last five days exceeds 10 GB", these policies still focus on lower layer parameters of the network stack.

More recently, SDN techniques are being applied to wireless networks. OpenRadio[10] suggests the idea of decoupling the control plane from the data plane to support ease of migration for users from one type of network to another easily, in PHY and MAC layers. CellSDN[11] enables policies for cellular applications that are dictate by subscriber needs, instead of physical locations - providing finer control of network flow than previously possible. The OpenWireless[16] prototype supports seamless handoff between wifi and wimax networks when video data is streamed, using openflow controllers. The wireless SDN solution provide the necessary building blocks for managing CPS multinetworks, but they are not sufficient. The *southbound* approach retains its focus on connecting to a specific lower-level access networks; its application to CPS multinetworks must support mechanisms that abstract out the network heterogeneity. Furthermore, the framework must support *northbound*,higher layer interactions, i.e. to the heterogeneous applications and their requirements.

In this paper, we proposed a novel CPMS controller architecture, that overcomes the limitations of traditional SDN controllers and takes into account, the characteristics of CPMS. We next provide more details on a proposed CPMS controller framework and illustrate how this architecture works in different communication scenarios.

A. CPMSs Controller Architecture

Figure 1 illustrates the conceptual architecture of a potential CPMS controller.

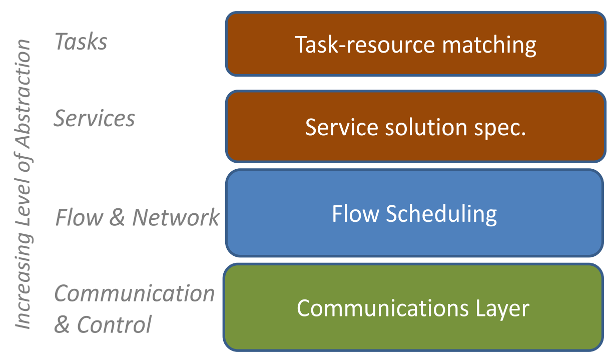

Fig. 1. CPS Controller Architecture

The concept of an abstraction level is fundamental to our vision of CPMSs since it allows us to make use of the heterogeneous multinetwork resources in a flexible manner. Tasks are the highest level of abstractions in a CPMSs that defines what is required; this leaves open the choice of what applications/services, devices and communication networks are used to meet the required task at hand. A simple example might be to determine how many people there are in a room. Services are concrete software/hardware entities that help in the realization of a task. A task may be realized by a single service (capture video from room camera) or a workflow of services that together realize the task at hand (capture video and determine if there is a person in it). A task/service mapping specified the what devices and what applications should be used to complete the task. The lower level Flow and Network layers decide which networks should be used for the application flows and how the application flows should be routed across the network. These decisions will be sent out to the corresponding devices via communication and control layer. Such a layered view has benefits since it hides the details of lower layers (network/devices) so that tasks can be accomplished in a more flexible way. Furthermore, the separate abstraction levels allow dedicated algorithms to be designed a certain layer for improved performance.

Consider a specific instance of a smartspace CPMS (as described earlier). Example tasks here might be *"Locate Tom in Bren Hall"* or *"Alert all cell phone users about a fire in Bren Hall"*. Once such a task is submitted to the controller from a requesting node, the controller components process it through a series of steps below.

- The controller maps the task request onto the existing resources in the multinetwork. For example, the first task (*"Locate Tom in Bren Hall"*)may be satisfied using various devices such as cell phones or hallway cameras that provide services such as GPS locations or providing camera pictures. This leads us to the second task that requires us to determine which cell phone users are currently in Bren Hall.
- The Task-Resource matching component will return several potential solutions that can be filtered by automated policies at the controller or via a human in the loop (i.e. network operator) - this will decide which solution the controller will adopt and further optimize.
- Once a solution is selected, the Service Solution component of the controller transforms the characteristics of the devices and services involved in that solution into specific requirements for devices, networks and application constraints (e.g., minimum throughput). For example, if the solution-"using camera and Ethernet to locate Tom's position", the controller can decide what is the data rate and delay requirement of this video surveillance service, given the video frame resolution, codec and receiver's buffer.
- The Flow Scheduling component takes these requirements and schedules flows that satisfy them. Scheduling and coordination of the resources of a CPMSs is complex due to the heterogeneity of the platform; many devices in the CPMS have low power and computation capability (e.g. sensors, smartphones,etc). Moreover, the heterogeneous application flows imply that the general best effort and traditional packet level routing protocols using unified routing metrics is no longer suitable in CPMSs. The packet-level decision making must now transition ot higher level Flow-based routing and scheduling mechanisms that take into account the constraints of the multiple communication infrastructures. For this to work effectively and in time, we propose the use of a logically centralized management and coordination component.
- Finally the controller triggers the necessary communications in the CPS.

III. Gathering Situational Awareness in CPMSs

Creating Situational Awareness (SA)[1] is a data-centric process, which requires gathering of data (typically in a database) and analysis of this data for extraction of higher level semantic information. An SA task is typically described using higher level concepts (such as space, entitied, events, interactions) as opposed to what devices and networks are used. Consider for instance, the task *"Locate Tom in Bren Hall"*. Answering this request/task requires data about the space (Bren Hall), people (Tom) and required service (localization). The mapping of this task to services/resources that can accomplish this task requires an understanding of devices and networks in the space of interest. A corresponding Task DB maintains the set of ongoing tasks and potential task-service mappings, drawing on information present in a Service DB that defines all available services. Multiple mapping choices are possible - the location of a person might be obtained from a surveillance camers. from a GPS reading on their cellphone or by WiFi triangulation from a mobile laptop. In the example above, person identification requires translation of tasks (Locate Tom) to sensing services (triangulate Tom's cellphone and/or match Tom's facial image to those in video image sequences from surveillance cameras). Recent efforts have shown that extraction of awareness from pervasive space data is aided by "semantics": [12] - such semantics can be obtained by analyzing data collected in a service DB (e.g. video surveillance service, GPS/Bluetooth location service) and Device DB (e.g. Camera, GPS devices).

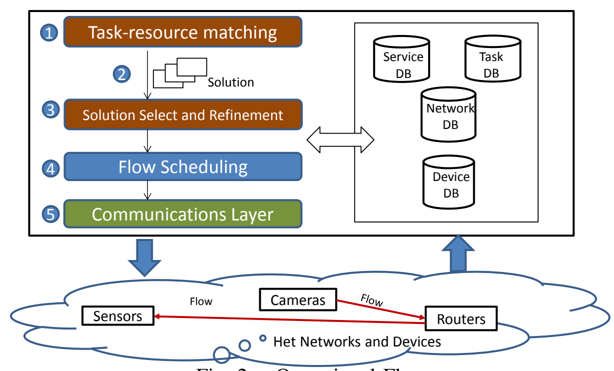

Fig. 2. Operational Flows

The Resource Matching step is often based on availability/suitability of devices - this step aims to compute possible mappings of devices/services to the task and realize those mappings by using multiple network and hardware resources. Once a solution is selected,we must determine more detailed service requirements. Assuming the the solution "Use video surveillance" is selected to accomplish task "Locate Tom in Bren Hall" - detailed parameters such as video resolution(640*800), Frame rate (30fps), Codec(H.264), Client Buffer(100kbytes)are specified. These service requirements are then translated into network and resource requirements: Data Rate of at least 0.7Mbps, Delay less than 1s and Loss Rate less than 5%. The information needed to determine whether the desired datarates and delays are possible is obtained from a Network Information Base (or Network DB) that contains the state of the networks in the space.

In the next step, a Flow Scheduling component of the controller will perform multi-constrained scheduling in order to fulfill flow requirements for multiple flows. Since different services have different requirements, the scheduling algorithm needs to consider multiple constraints. A flow entry is then sent from the controller to the controlled device using open APIs and protocols provided in SDN frameworks, such as OpenFlow[openflow] - the aim is to enforce a global (sytemswide) flow scheduling.

A real world implementation of the above procedure is not straightforward. First, extracting high level semantics from collected data is not easy. Indoor localization through techniques such as WiFi triangulation, when deployed in practice, have accuracy issues; techniques for person identification through video/images have inherent uncertainties, may be slow and may encroach unnecessarily on the privacy of users. The mapping of tasks to services for timely and accurate SA must consider other factors such as timeliness, reliability. performance of the SA process. Furthermore, the overall environment contains multiple services running on heterogeneous networks - the ability to accommodate those services without violating their various network requirements such as jitter, delay, throughput is challenging.

IV. Information Sharing and Dissemination in CPS Multinetworks:

As is the case with situation awareness and information collection applications, information sharing and dissemination tasks can also be further strengthened through the use of CPS multinetworks. Continuing with the previous example, consider the case of a building fire in Bren Hall. A dissemination task here might be “Deliver a fire warning to people in or close to Bren Hall”. When a fire occurs, the dissemination task is triggered, and the system will proceed to determine a set of recipients who must receive the alert messages. The location of these recipients further dictates the specificity of the message content and how it is delivered. For example, the alert can be sent via a short text message about the ongoing fire to a smart phone user; to assure individuals that this is a real event, an audio announcement from the building manager can be disseminated that warns users to exit quickly from the building, this can be further augmented with images or video of the impacted locations and suitable exit paths.

Reliable alert dissemination is a key criteria when we send critical messages; existing techniques exploit redundancy [13] [14]to ensure resilient communication. Ensuring reliable and timely communication is a challenge with rich media alerts (e.g. those with image and video content); however, recent research aims to exploit multiple networks in the communication path concurrently to ensure both low latencies and reliable delivery [15], [16]. Pub/Sub systems have also been explored for personalization of messages , enabling the delivery of meaningful and actionable content to the relevant recipients [17].

Given a specific task, there are many potential dissemination plans that can be generated. The Resource Matching component in the CPMS controller collates possible solutions. For example, for users registering to receive text message warnings, the component can broadcast the warnings to mobile devices over cellular networks. For users who wish to receive image or video warnings, the component may choose to send the warning messages over WiFi networks. Information required to enable selection of the right services is obtained from the controller-side Databases - this includes information about Device/Resource availability, user preferences (registered apriori), priorities, task criticality etc.

The actual process of information sharing and dissemination also has challenges. It is possible that a solution contains several services that can be composed into a workflow or message graph using sequential/pipeline and concurrent operators. For example, a GPS based service is used to provide information about where the users are located; this, in turn, determines if the users are impacted by the fire. A content delivery service is then employed to send the warning to users with the right content type - this is an example of a sequential service workflow. A video clip and an audio recording of the fire may be disseminated to the end user via the same network infrastructures concurrently - and these are concurrent service workflows.

Once the services are selected, the Flow Scheduling component employs multiple networks simultaneously for information sharing and dissemination - this is essential, especially due to the changing and evolving nature of events. For example, it could be possible that network infrastructure, e.g., WiFi, in Bren Hall is destroyed due to the fire. A straightforward approach is disseminating the warnings to the users using cellular networks. However, large content broadcast over cellular networks is not practical(content is limited to 99 characters for cellular broadcast). There is a growing trend to combine unicast and application layer multicast protocols to support the dissemination process. Unicast-based communication is time consuming; delivery of large content to users one-by-one suffers from high latency and thus cannot provide fast warning services to users. A simple, yet and efficient solution is to employ all possible network interfaces available in the mobile devices for sharing the warnings. One challenge for this solution is the design of a Flow Scheduling component can use the cellular networks and WiFi p2p networks (that do not require infrastructure equipment, e.g., WiFi routers) to disseminate large contents. More specifically, the component unicasts part of the warnings over the cellular networks to each mobile device, which in turn shares the received information with nearby mobile devices over the WiFi p2p networks. This not only speeds up the information sharing process but also reduces/offloads data traffic going through the cellular networks (i.e., avoids congestion and reduces dataplan costs charged by the cellular network providers). The coordinaton and scheduling of such interdependent services and networks remains a future topic of research.

V. Conclusions

In summary, the Software Defined Networking approach promises to provide increased flexibility for future communication scenarios that support multiple access techniques and heterogeneous flows. We argue that such an SDN based approach is especially interesting in cyberphysical systems; where support for merging advances in networking and device technologies is a key driver for many real world applications. Architectures such as the one illustrated in this paper aim to support separation of higher layers task/services for communications with lower layers that provide the actual physical control of the communication elements - such cross-layer design methodologies are key enablers for future networks.

REFERENCES

[1] S. Mehrotr, N. Venkatasubramanian, M.-O. Stehr, and C. Talcott, "Book chapter: Pervasive sensing, event detection and situational awareness," *Securing Cyber-Physical Infrastructures: Foundations and Challenges*, 2011.

[2] G. Denker, N. Dutt, S. Mehrotra, S. Mark-Oliver, T. Carolyn, and N. Venkatasubramanian, "Resilient dependable cyber-physical systems: a middleware perspective," *Journal of Internet Services and Applications*, vol. 3, no. 1, pp. 41–49, 2012.

[3] white paper, "Software-defined networking: The new norm for networks," *Open Networking Foundation*, 2012.

[4] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, "Devoflow: scaling flow management for high-performance networks," in *Proceedings of the ACM SIGCOMM 2011 conference*, ser. SIGCOMM '11, 2011.

[5] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: dynamic flow scheduling for data center networks," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, ser. NSDI'10, 2010.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*

[7] M. Mendonça, B. N. Astuto, X. N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," 2013, in Submission In Submission. [Online]. Available: http://hal.inria.fr/hal-00825087

[8] A. Voellmy, H. Kim, and N. Feamster, "Procera: a language for high-level reactive network control," in *Proceedings of the first workshop on Hot topics in software defined networks*, ser. HotSDN '12, 2012.

[9] Hinrich, N. Gude, M. Casado, J. Mitchell, and S. Shenker, "Practical declarative network management." in *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking, pp. 110. ACM, New York*, ser. WREN 2009, 2009.

[10] M. Bansal, J. Mehlman, S. Katti, and P. Levis, "Openradio: A programmable wireless dataplane," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 109–114. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342464

[11] L. Li, Z. Mao, and J. Rexford, "Toward software-defined cellular networks," in *Software Defined Networking (EWSDN), 2012 European Workshop on*, 2012.

[12] S. M. R. Vaisenberg and D. Ramanan, "Exploiting semantics for scheduling data collection from sensors on real-time to maximize event detection," in *In Multimedia and Computer Networks MMCN*, 2009.

[13] S. M. Kyungbaek Kim and N. Venkatasubramanian, "Farecast: Fast, reliable application layer multicast for flash dissemination," in *ACM Middleware*, 2010.

[14] K. E. Benson and N. Venkatasubramanian, "Improving sensor data delivery during disaster scenarios with resilient overlay networks," in *proceedings of Third International Workshop on Pervasive Networks for Emergency Management*, 2013.

[15] N. V. Ngoc Do, Cheng-hsin Hsu, "Hybcast: Efficient rich content dissemination over hybrid cellular and ad hoc networks," in *IEEE SRDS*, 2012.

[16] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T.-Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, S. Seetharaman, D. Underhill, T. Yabe, K.-K. Yap, Y. Yiakoumis, H. Zeng, G. Appenzeller, R. Johari, N. McKeown, and G. Parulkar, "Carving research slices out of your production networks with openflow," *SIGCOMM Comput. Commun. Rev.*, vol. 40.

[17] K. K. Ye Zhao and N. Venkatasubramanian, "Dynatops: A dynamic topic-based publish/subscribe architecture," in *ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '13. ACM, 2013.