Deep Science Publishing
https://doi.org/10.70593/978-81-989050-5-5_5

DeepScience
Open Access Books

Chapter 5: Integrated optimization techniques across device hardware and network communication layers

5.1. Introduction

Our society is now heavily reliant on remote communication since a substantial portion of our life functions has become dependent on the Internet. The increasing density of mobile devices no longer supports only phone calls, which has stagnated over the past two decades. Mobile broadband has been growing rapidly over the past decade, and mobile devices now support high-definition video streaming, mobile online gaming, and immersive client-cloud virtual reality processing, attracting huge amounts of traffic. With the fast-growing use of wireless mobile devices comes the challenge of meeting the high demand for quality user experiences through improved wireless communications (Ali & Tran, 2025; Nakamura & Evans, 2025; Ortega & Farid, 2025).

Various integrated optimization approaches have been proposed and implemented to continuously update multiple optimization objectives ranging from network-wide cost reductions, closed-loop mitigation of mobile device energy consumption, through improved network communication performance, to enhanced or guaranteed quality of experience for location-based mobile web services. These efforts target derivatives of the energy consumption or latency of device operation, network-wide operational costs, quality of service, or quality of experience. Many of these derivative functions can be effectively optimized in an off-line, continuous, but no longer feasible, centralized manner since the rapidly increasing number of mobile devices or endpoints would incur appreciable communication overhead. In addition, mobile device latencies, including energy consumption and device processing, cannot be effectively reduced if optimization solutions are updated less frequently due to communication latencies (Shen & Wallace, 2025; Wu & Garcia, 2025; Nakamura & Evans, 2025).

We review the continuously interactive, integrated communication optimization techniques deployed across the networking stack layers incorporating mobile device hardware, wireless resource allocation, and mobile application or network service

operation. We consider the energy constrained and latency benchmarked location-aware mobile web services, which are proponents of the above-listed targets due to the growing demand for augmented reality applications in supporting our daily activities. To deliver quality user experiences, localization of mobile web service users should be prioritized and accounted for in mobile device resource allocation at the wireless communication layer.

5.1.1. Background and Significance

A careful and thoughtful analysis of the design space of MCD requires going beyond identifying the opportunities and challenges it presents. Here, we first provide a background of MCD, including its relevance compared to other optimizations and its potential pitfalls, and we look back at how conductance and mobility optimization have evolved. Thereafter, we give an overview of its significance to technology stakeholders, providing a schematization of the major players involved, key fundamental concepts, and the related metrics that they are interested in. This sets the stage for a more concrete vision of how MCD can be optimized for varying objectives related to varying circuits using varying technologies within a heterogeneous system.

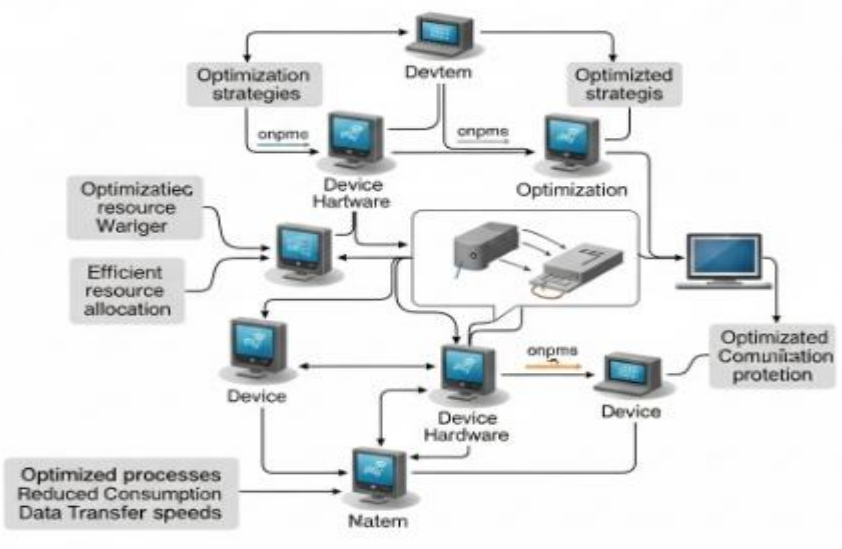

Fig 5.1 : Integrated Optimization Techniques Across Device Hardware and Network Communication Layers.

The growing demand for services that require terminal processing and switching logic resources has increasingly motivated research on optimal hardware design behind these processes. Designing such resources starts with logically decomposing the needed

services to identify the required functions and then selecting a set of devices to implement this function set with adequate performance and cost characteristics. While the logic functions could be implemented on one or more chips, designed with unique constraints, a typically present and dependable cost consideration often results in an implementation that minimizes circuit delays and areas. The flexibility offered by multi-devices and networks solutions comes at the cost of performance overheads during interdevice communications, and using a shared communication structure adds delays, switches configuration time, and sensitivity to activity distributions, reducing the overall gain.

5.2. Background and Motivation

Adopt for this vision, tx and rx integrated optimization techniques that span HW and SW entities in mobile gateways, and across layers in the communication stack in order to improve the effective throughput, latency, and energy cost of flows traversing the mobile gateways in their common role as the interface between mobile devices and the Internet. The optimization space contains parameters associated with the wireless device TX hardware (e.g., TX gain, rate for wireless TX, and active vs. idle mode), the wireless device RX hardware (e.g., RX gain, exact RX configuration – for both mobility and filtering), and the gateway (e.g., layer 3 routing policy, flow level detection of video, voice, especially transoy or trans video flows), as well as the layer 2 protocol running stack.

With today's cutting-edge wireless technology – MIMO antennas, adaptive transmission coding, high-bandwidth wireless chips, and energy-efficient modulation schemes – users can now download high-definition content to their portable and handheld devices. Coupled with the development of a huge online catalog of movies and TV shows, consumers can increasingly enjoy video on their portable devices. In this setting, mobile gateways devices that support real-time video streaming, imposing stringent delay constraints, can consume a considerable amount of resources, especially at the gateway; they may need to buffer multiple packets at the gateway to hide the delay over the wireless link.

5.2.1. Research Design

A systems-oriented approach to the integration of optimization techniques across different hardware and communication layers is essential to enable the design of future communication. Identifying and addressing limitations and constraints in any of the involved design dimensions may lead to suboptimal performance solutions and preclude the realization of a highly integrated system design. In our approach, functional blocks

in the physical layer are as critical as the link layer characteristics. Central to our work is the incorporation of physically informed models of constituent elements, either as detailed module operations for estimating performance and/or optimization objectives, or as profile functions for providing lookups of key performance characteristics.

We free the optimization design flow for layered and hierarchical approaches that target individual layers, by involving the effects of different design dimensions in the optimization, through module-level performance models. Major building blocks in a communication system include error-control coding, modulation and physical-layer design, decoding at the receiver, scheduling resource allocation at the transmitter. These functional blocks in both the transmitter and receiver jointly determine communication performance, and need to be integrated in an optimization across the functional blocks that is guided by an application-level objective, such as quality or application-specific utility. Further, the design of all modules need to be integrated across different communication components that connect over a communication channel, such that the constraint conditions over the operation of these components is satisfied.

5.3. Overview of Device Hardware

As computational technology rapidly progresses and device capabilities increase, technical solutions in design and manufacturing are required. Original equipment manufacturers (OEM) who develop device hardware typically have software layers that abstract the hardware details. These software layers include kernel drivers, and are well defined for a device category with minimal variations across devices with similar hardware architectures. Although the software variation is minimal, onboard resource settings such as frequency and voltage can be configured, but within bounds determined across OEM and geo-regions for warranty considerations. With these software-defined abstractions, device hardware architecture tends to be hidden from the cloud or application developers, but these abstractions do not really offer the low-level control that can be utilized to achieve device optimizations. Almost all software that directly optimize device hardware are proprietary and created by OEMs to achieve specific use cases, including energy efficiency, building-level efficiencies, user experience, and others.

General computing equipment such as PCs, laptops, and high-frequency servers enable the execution of a wide variety of computing workloads. These systems are distinct from purpose-built, task-specific hardware such as GPUs, TPUs, ASICs, array co-processors, FPGAs, etc. However, with the exclusivity of processing units, these components tend to be designed for massively parallel workloads, typically found in areas such as video analytics, gaming, 3D modeling/graphics rendering, AI/ML, and other areas also referred to as hyperscale use cases.

5.3.1. Types of Device Hardware

When sensor and computing devices are coupled in a wireless sensor network (WSN), interesting and useful functions are enabled. Applications benefiting from coupled devices are numerous. For example, in future smart spaces, wired and wireless devices will be coupled to manage power and resources efficiently. Most people are interested in the potentially innumerable applications. However, the coupling technologies are less emphasized.

Device hardware can greatly affect the energy consumption level and design of the communication layer. It can also offer many new features to communication layer design. Time synchronization is an example of a function not easy to implement via message passing yet easy to support via signal processing algorithms exploiting the pre-existing smart environment sensor apparatus already boosting wireless monitoring capability at a lower cost. Communication layer support of device hardware to parallelize resource consuming heavy lifting tasks like multi-dimensional signal processing is another example. Parallelization can be done conveniently via the wireless device mesh network. Also, communication design can help adapt software associated with device hardware for application fitting and switching. It can help support use of flexible and adaptable interfaces to device hardware for collaborative applications across a smart space.

Not all computational core and hardware interfaces are created equal. There are a plethora of devices in existence and devices yet to come. Common classes of devices include cameras, microphones, infrared, radio frequency, laser detection and ranging sensors, accelerometer arrays, broadband ultrasonic transceivers, in and out wearable processors of computer and other specialties, and nanosensors. However for the statement to be true as laboratory samples evolve and transmute to mass produced chips functional equivalent in real applications, the thrust is devices working together to solve a single problem. Similar but yet not alike devices contribute diversely to the solution process.

5.3.2. Performance Metrics

In the previous sections, we have reviewed the wireless network communication and device hardware-related components in layers and tried to explain the decoupling and abstraction of communication and hardware layers. The layers of the hardware devices are underexplored and poorly defined, with no uniformity across different design paradigms with respect to the cost, performance, and overall feature set provided by these hardware layers. To better understand the hardware layer protocol design, we need to have an understanding of the hardware performance metric functions.

For the wireless sensor network, various types of performance metrics can be identified. For ease of understanding, we organize them into the following broad categories: capacity and communication performance metrics, deployment model-based performance metrics, coverage and hole identification-related metrics, topology and routing related metrics, directed diffusion concepts related metrics, and communication-transduction related metrics. Several design metrics, such as energy efficiency, reliability, coverage, delay and latency, capacity, delay jitter, security, detection, and tracking prediction can be identified for the wireless datacentric smart integrated system paradigm. Here we present only a few performance metric functions for understanding the working of the wireless device hardware protocol-related functions. The wireless datacentric smart integrated system common layer may use the available metric set to analyze and optimize the various protocols built on top of this unified hardware device layer.

5.4. Network Communication Layers

Computer Networks connect a multitude of devices together. Each device is a point conveying information in the form of bits, comprising addresses and codes for operation. The concept of interconnecting devices started with the inception of the telegraph. Data network application domains have exploded with wireless and optical technologies, mobility and location awareness, multimedia exposure via Voice over IP, music and movies, sustainability needs through eco and energy conscious designs, larger and mixed data flows requiring strong Quality of Service guarantees. Technologies used for the physical implementation of these devices may vary: copper wire or fiber optics, electromagnetic waves or light pulses, wireless signal, power lines etc. The main goal of all these implementations is to transport blocks of information as bits with a desired maximum delay, possibly with a required minimum bandwidth, reduced bit-error rate, and at affordable cost.

Network Communication Layers provide the basic enabling service to user-level applications running on the various end systems, reliably transferring messages containing blocks of information. The end-user application is often unaware of the layered protocol which defines the rules for the dialogues and transfers between the service users and the provider. The network's architecture is such that, from the end-user point of view, the various abstraction levels do not show: the provider creates a service which hides, according to the quality of service required, both the technologies used and the layers which are implemented. Application domains in fact are sensitive only to the transfer delay and the throughput of the offered service.

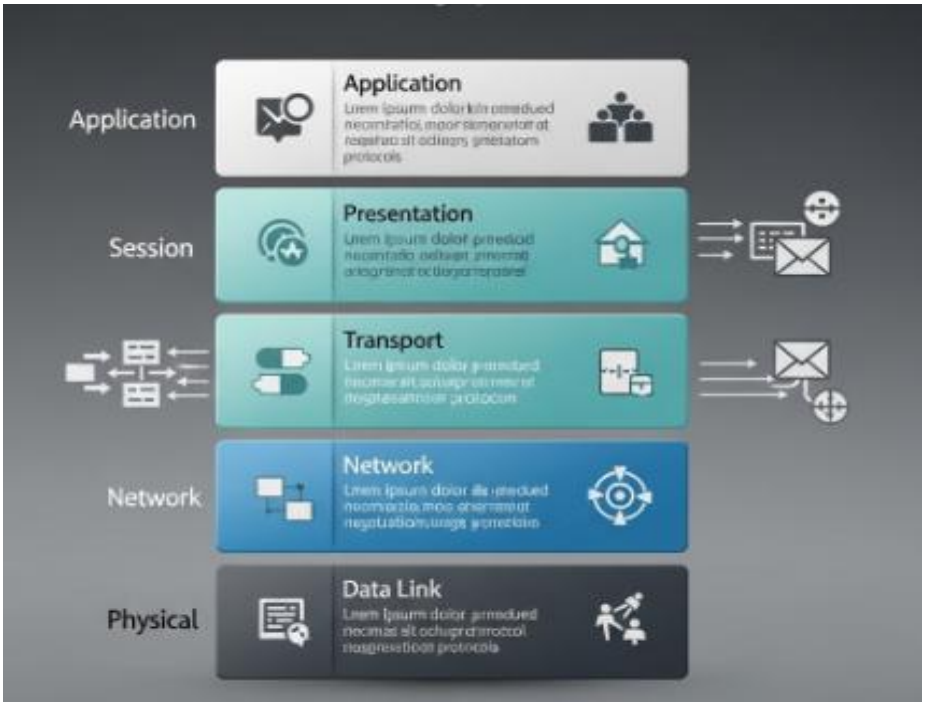

Fig 5.2 : Network Communication Layers.

5.4.1. Layered Architecture

A layered architecture is used to implement most computer network systems. This architecture has several advantages. First, it allows for easier identification of the specific requirements and services provided at each layer of the protocol stack. Second, it usually lends itself to a modular object-oriented software implementation with the specific code for each layer residing in a separate module. Third, the specification at each layer is modular, and thus a problem may be addressed at only one specific layer while the other layers remain unchanged. Fourth, new protocols and standards may be easily introduced by modifying and periodically innovating at one specific layer while maintaining backward compatibility with the rest of the existing protocol stack. Finally, the implementation of a protocol at each layer is relatively simpler, which often leads to reduced design and development time.

Layers typically perform the services that are provided themselves entirely by modifying, and possibly encapsulating, the services from the lower layers, and passing the modified data to the next higher layer. Each layer also performs operations that make it capable of performing the service for the higher layer. The protocols used at a layer must maintain domain knowledge of the next higher layer in order to perform these operations. Only limited domain knowledge is retained at one specific layer, and it is

this modular specificity along with encapsulation that allows for the protocol stack modularization. This shows two protocol stacks, one at an application client, and the other at an application server. The application server and client communicate application data through their transport layer protocols, but the implementation of these transport packets is modular as each layer is communicating only through its layer interface with the next layer. Each layer also contains only limited domain knowledge about the other layers in the stack.

5.4.2. Protocols and Standards

Each layer of the communication subsystem is composed of protocols. A protocol is a set of rules or requirements that describes how two devices at a given layer interact. In an architecture with N layers each device has $N - 1$ protocols to handle all interactions with remote devices. Protocols specify aspects such as addressing, initialization, or the structure and meaning of exchanged messages. Protocols must be unambiguous to allow devices with different logical implementations to interact. The number of requirements can be very large for a protocol. As a consequence, higher layer protocols can be built upon protocols specific to lower layers. The set of rules for a certain layer of the protocol can define a protocol family or a platform. Standard protocol families at each layer allow devices from diverse manufacturers to communicate. Standards can be formalized by a standardization organization. Market feedback or unregulated initiatives can also lead to informal standards. Obviously, the existence of standards is a condition necessary for devices from diverse manufacturers to interact. However, the adoption of a particular standard is not always guaranteed. Early cellphones used different protocols from diverse manufacturers until an informal standard was established.

Table 5.4 contains several examples of protocols and open or informal standards defined over layers of the communications subsystem. Both OSI and Internet model layers are represented. In the latter case, the upper layers correspond to the Application, Presentation, and Session layers while the lower layers correspond to the Network Interface, Internet, and Transport layers. Note that for the Internet model, multiple standards coexist as protocol families or platforms for the same layer. For instance, multiple implementations of the FTP, Telnet, or SMTP protocols coexist. The more such implementations coexist, the richer the experience in validating the protocol, reducing bugs and optimizing it.

5.5. Optimization Techniques

We will now review some optimization techniques using integrated communications and computational capabilities of devices that might be deployed near physical sensing or actuation resources and the supporting computer and communication networks. In our survey of optimization techniques from various fields, ranging from algorithms to heuristic and neural approaches, we observe that many overlap or are variants of the same fundamental concepts. However, these variants are often employed in different manners across different disciplines and application domains.

In general, optimization methods can be classified as either algorithmic or heuristic. Algorithmic approaches provide guaranteed results within acceptable computation times for low-dimensional spaces. However, optimization problems that are encountered in large-horizon, dynamic sensor and control systems are generally nonlinear and high-dimensional and thus require the use of heuristics, ranging from educated guess-and-check to evolution-based methods and deeper techniques derived from the principles of artificial intelligence, statistical learning, or genetic processes. Machine learning tools exploiting large datasets can and should also be used to debug or enhance the performance of optimization methods as basic components of a given underlying backend. We briefly review the state of these three disparate areas in what follows.

Traditional optimization techniques include the simplex or barrier method and sequential quadratic programming for convex linear or quadratically-constrained convex problems. The well-known coding theorem guarantees that the optimal representation relies on a dictionary formed from the detected correlated sequence, or jointly optimal prediction. Generalizations to arbitrary and possibly multimodal distortion functions are well-established. Other specific examples include a method for meeting probabilistic performance requirements by relating the solution of the Fourier-based bandwidth optimization to a probabilistic formulation of Boolean and analytic function behaviors.

5.5.1. Algorithmic Approaches

An integrated optimization technique toward the maximization of the node utilization is proposed in sequential and joint algorithmic forms. Different optimization problems for a single device node under different operating states are formulated: data-reporting, sensing, energy-harvesting, and recharging. Due to these energy-related constraints, a directed multi-sink Hybrid Wireless Sensor Network structure is used. The first approach achieves sequential and near-optimal solutions, while the second achieves multi-dimensional delivery-time and energy-minimized cost optimization. Complete utilization of any individual recharging stations is not considered, however, the study indicates the significance of data-reporting delay on the overall energy consumed by the

hybrid system. In the second approach, a new algorithmic method for joint delivery-cost and sensing-cost optimization is proposed. The heuristic optimizes the path and cluster sizing controlling parameter values by integrating an estimation method with a particle-swarm movement-based search superimposed in bubbles. The results show an elaborate picture regarding the importance of controlling parameters on the overall performance.

The various algorithmic optimization techniques have their advantages. However, it can be concluded that the application of heuristic and algorithmic approaches toward the cost minimization of Hybrid Wireless Sensor Networks can yield efficient wireless communication systems compared to the existing communication, augmentation and management techniques.

5.5.2. Heuristic Methods

In this section, we present a few heuristic methods for optimizing across different layers based on specific applications. These heuristic approaches are attractive for how quickly they return optimized configurations, e.g. reducing video size in some cases to fractions of their original sizes, while producing excellent output quality.

The simplest heuristic method is to apply heuristics sequentially at different layers. For example, if a sequence of DCT is to be DCT coded while another DCT coefficient table is to be Huffman coded, the problem can be solved by using a simple two-pass processing technique. It just processes the result of the first stage, creating a function of the values at different encoding levels, while ignoring the second coding process. This two-pass approach is refined further if the first encoding process selects sample values which are not from the original image or video and which do not preserve perceptual fidelity.

Another extension is based on a more hierarchical design process. For example, we can first assign all visual content to a few high-fidelity channels. Then the assignment is expanded and customized to achieve a global target bit rate and visual system fidelity according to user preferences. Furthermore, the DCT encoding layer can adaptively adjust its parameters based on assigned characteristics. Or a group of channels can communicate using a few central DCT encoders, which can be designed at the same time if it allows efficient joint DCT coding with lower perceptual fidelity. These two-pass extend more generally for layers at different subband layers based on perceptual criteria. In particular, in the first analysis pass, the multiresolution/time-frequency decompositions can ignore perceptual errors in a perceptual invisibility sense based on the subsequent compression stage.

5.5.3. Machine Learning Applications

Machine Learning (ML) offers flexible solutions to many optimization problems. Given sufficient amounts of training data, ML models can achieve accurate inference and demystify some hard-to-craft heuristics. ML, unlike classical optimization techniques, can use solution quality-related information implicitly captured by past training data to make decisions and dramatically speed up the solution search. In many real-world optimization applications, the complexity of the optimization problems, the rapid changes of the problem spaces, and the dynamic uncertainties against the optimally-calculated solutions under a fully-defined model can easily render traditional optimization approaches impractical. Moreover, many optimization problems are known to be NP-hard, and it is unrealistic to expect a classical approach to adopt the least computation time and universal training samples to always generate optimal solutions. Learning from the standard computing model, as we move into the varied facets of hardware architecture, platform and communication networks, we reach a selective conclusion that within each of these varied facets, we have a new challenge imposed on resource allocation or utility maximization or cost/execution time minimization. However, even within similar forms of varied optimization problems and through our understanding of the ML domain, we still cannot find a unique boundary towards a universal optimization approach.

The designing of a universal optimizer still eludes us, and learning from the other faces of data usage in order to push the importance of efficiently handling the data during internal computation is always desirable, especially with new ML models of enormous sizes and workloads at hand. To this end, we suggest using selective, partly-trained, multilayer collaborative ML models, and apply them towards jointly optimizing ML model monetization. The individual facets in the models assume the contribution of individually optimization towards address the full model obligation during external interactions, with the mutual contribution from the other parts foots the affordability of their contribution overhead which in return lowers the communication and collaborative overhead, and limits the power and latency overhead introduced by the inference process. More interestingly, this multilayer model should seamlessly allow users to communicate directly with the model appropriately at the level of information specification according to the dynamism of the input queries, and thus optimizes an entire end-to-end model lifecycle that must include the entire model construction, preserving, collaborative inference or post-processing, and monetization.

5.6. Integration of Techniques

While integrated optimization can be applied to any one of a number of layers (e.g. hardware, operating system, application, network), and in many domains it is common

to target a vertical stack of layers in a single domain, it is more rare to coordinate and optimize across multiple vertical stacks that represent different devices working to achieve a common goal, as in the work we review in this section. This type of interaction, which we define here for the purpose of this survey as “cross-layer optimization," has been well-studied in the embedded domain and can use the fact that packets will traverse a suite of devices connected by various hardware implementations and using various protocols, but generally is not well-studied for general purpose computing at all tours. The second major approach for integration leverages the interaction between the computing devices and the communication devices that is often ignored by device-centric optimization: that multiple devices are co-located at one site and therefore the communication interface is shared. Generally, this type of cross-layer optimization is found using specialized or mission-specific hardware; we detail the existing work in these areas below, as well as the challenges faced in moving toward more general purpose approaches.

Cross-Layer Optimization.

Integrative approaches to cross-layer optimization have their roots in the embedded systems domain, where the networking hardware is a critical component of vertical application and device optimization. Such techniques generally work on the principle that the packets will be received that pass through an entire stack of devices with very specific capabilities and limitations for the timing, size, protocol, blocking behavior, and number of packets transmitted. The general-purpose cross-layer optimization has found some footing but is usually limited to protocol stack optimizations at the application and network transport layer, or utilized individual hardware capabilities across devices combined with active networking infrastructure to assist with packet scheduling and other timing at the network device. These widely used approaches limit the ease and accuracy of communications at the multiprocessor network, and thus enable optimizations at the individual device level to take priority over any collaborative work.

5.6.1. Cross-Layer Optimization

New, revolutionary products and services, enabled by the emergence of the Internet, are becoming profitable. Bandwidth costs and service provider operating expenses are falling. Incremental improvements do not fully address the economic challenges posed by the increasing costs of Internet connectivity. Processors, memories, and networking technologies continue to experience dramatic increases in price-performance ratios, bringing new capabilities to every device. There is a need for techniques and mechanisms that allow us to fully exploit this new hardware and networking infrastructure. This exploitation can fundamentally change the ways computers do what

they do best — processing — while improving the service experience without requiring more expensive transport.

To achieve these general goals, new uses of available resources — of processors, storages, etc. — must be developed. For example, the fall in broadcast bandwidth costs is beginning to make push technologies a viable alternative to present pull-based systems. Push technologies are attractive because they reduce the rental cost of transport resources needed for services that have some sort of periodicity: news, sales, tide predictions, etc. The rental cost is driven down to the subscription cost, as you no longer have to pay for each page you access. The push idea has also become a useful back-office technology, where services fraudulently use these protocols to deliver unsolicited web pages to on-line users, cheapening the cost of bulk advertising strategies.

Cross-layer optimization techniques aim to reduce the specialization of given layer operations on a given device's resources — memory, processing, etc. — and of network properties — node specializations, delay, bandwidth, etc. For example, the push solution can exploit the limitations of your device's configuration — primarily its processing resource, which is often very small in mobile devices — and result in delays larger than the average page transfer time, thus minimizing the probability that the access will collide.

5.6.2. Cooperative Strategies

Incorporating a dependency or user benefit model and utilizing feedback for cooperative device and network level optimization, adaptive applications are a natural candidate for integrated optimization strategies. An example of this optimality is provided in a video application which recognizes device capabilities and current network conditions and adapts the video type, resolution, rate, and content dynamically. Such an application operates by recognizing the degree of user tolerance and prompting actions in the user. But more ambitious applications not only adapt user actions, but also act autonomously and interact with the network and the device to improve user experience. Existing applications with only device and network user adaptive features could be developed into cooperative applications autonomously reacting to changes in device, network, and user.

An application can operate in a cooperative approach by detecting a device lacking in capabilities, like during video playback, and prompting a device control to lighten the device workload or reduce video size based on how fast the application can upload or by sending a message for delay. Also during times of limited network resources any application would like to prompt the flow control for temporary reduction in video quality or resolution during this playback for priority over reduced bandwidth.

Traditional network layer flow control executes an end to end user data flow optimization based on a flow model or parameters like data flow size and inherent delay, besides connection delay. Such an end to end utility criterion model, without the advanced in-device capability and action prompts or adaptation, could delay download or video quality reduction based on how fast its connection delay to use.

5.7. Conclusion

This chapter has highlighted an increasingly coherent view on closed forms of optimization functions that enables a systematic end-to-end integration across temperature effects, device performance trade-offs, the communications security/privacy, and higher-layer data security/content. Here layers correspond primarily to a decomposition of device layers and a physical-device-level soft-sensor. This clear and modular view allows for a closer look at how device and communication layer device-measured information can help optimize layers at all levels, and at much higher utility than present-day approaches allow. We believe that these relationships will allow us to move orders of magnitude closer to a continuous-data-world optimization at all levels.

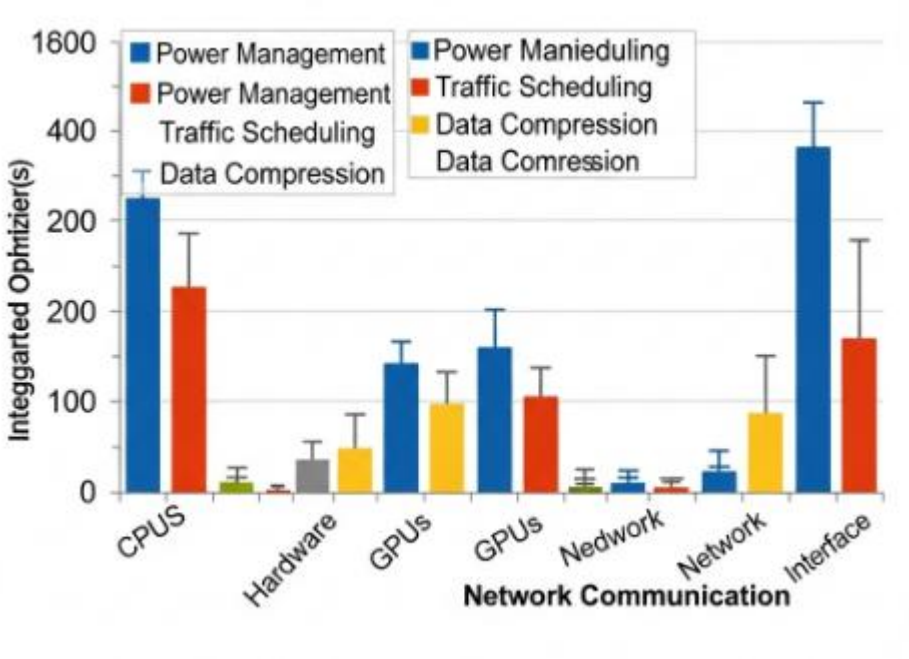

Fig : Integrated Optimization Techniques Across Device Hardware and Network Communication Layers.

Future applications in sensing must not flirt with a fundamental misalignment between objectives and technologies. It should be possible to reduce energy usage in a space with other highly competent micro- and nano-sensors. A major limitation is the high level of misalignment currently constraining understanding of trust data in lower-power, memory-constrained, single-node devices with high on-board feature extraction and computation. Privacy of the underlying information for basic facts of physical events like births and property transfers must also be preserved, and it may return to traditional mechanisms, with use licensing and opaque but reversible transformation functions that clarify the conditions for future access. Finally, attention must turn to how centralized ownership and access to mass data databases can be addressed. Although a true free market solution seems unlikely, unlike data distribution of other types, centralized mass ownership may not be efficient.

5.7.1. Future Trends

Future trends will focus on expanding the energy efficiency frontier using new materials and new architectures. Some of these applications and their effective implementation may necessitate new computing models. For example, a new model may simplify distribution of computations to better deal with reliability issues of emerging chips implemented with new materials, packaging technologies, or 3D stack architectures.

These trends bear new challenges that benefit from integrated optimization techniques across device hardware and communication layers, as distinct from existing techniques that tend to focus on either of them. Stacking multiple layers of memories on top of the logic may enable several orders of magnitude speed benefit. Network chips may also have the same or similar inter-layer communication mechanisms, enabling efficient computing with low-energy footprint. Such chips do not fundamentally change model computation mapping, which too relies on distributed memory. Therefore, utilization of integrated optimization techniques across these two chips is necessary for desired speed benefit while minimizing power or energy utilization.

References

Nakamura, K., & Evans, P. (2025). AI-enhanced RF front-end calibration for integrated 6G chips. In Proc. IEEE Int. Solid-State Circuits Conf. (pp. 78–82).

Ortega, J., & Farid, A. (2025). Secure edge-computing protocol design in intelligent wireless systems. Computer Networks, 234, 109430.

Wu, C., & Garcia, L. (2025). Circuit-level implementation of machine-learning algorithms in analog hardware. Nature Electronics, 8(3), 217–226.

Ali, Z., & Tran, B. (2025). Bandwidth optimization in AI-managed reconfigurable antennas. IEEE Antennas and Wireless Propagation Letters, 24(2), 198–202.
Shen, D., & Wallace, J. (2025). Signal integrity in high-speed AI hardware. IEEE Design & Test, 42(1), 34–45.