CrossTalk: Cross-Layer Decision Support Based on Global Knowledge

Rolf Winter and Jochen H. Schiller, Freie Universität Berlin
Navid Nikaein and Christian Bonnet, Institut Eurecom

ABSTRACT

The dynamic nature of ad hoc networks makes system design a challenging task. Mobile ad hoc networks suffer from severe performance problems due to the shared, interference-prone, and unreliable medium. Routes can be unstable due to mobility and energy can be a limiting factor for typical devices such as PDAs, mobile phones, and sensor nodes. In such environments cross-layer architectures are a promising new approach, as they can adapt protocol behavior to changing networking conditions. This article introduces CrossTalk, a cross-layer architecture that aims at achieving global objectives with local behavior. It further compares CrossTalk with other cross-layer architectures proposed. Finally, it analyzes the quality of the information provided by the architecture and presents a reference application to demonstrate the effectiveness of the general approach.

INTRODUCTION

Layered protocol stacks as used in almost all modern communication systems have a number of key characteristics that make them the primary architectural choice when it comes to designing a new protocol stack. Compared to the alternative of a monolithic stack, the advantages of a layered approach are the reduced design complexity due to well-defined functional entities, the improved maintainability due to the modular nature, and the high degree of flexibility, since layers function independently of each other. This, in principle, allows for arbitrary combinations of protocol classes. Recently, cross-layer approaches have increasingly attracted the attention of researchers and communication system designers as a third design option. This wave of attention stems from the large and active community dealing with challenging networking environments such as mobile ad hoc networks, next-generation cellular networks, or sensor networks.

A HIGH-LEVEL VIEW OF CROSS-LAYERING

The concept behind cross-layering is rather intuitive. Instead of treating a layer as a completely independent functional entity, information can be shared among layers. This information can be used to adapt protocol functionality in the presence of changing networking conditions, for decision processes such as route selection and as input for algorithms. It is even possible to create new kinds of adaptive applications such as multimedia applications, which are sensitive to changing networking conditions.

The ability to share information across layers is the central aspect of cross-layer design. So instead of a mere replacement, cross-layering can be seen as an enhancement of the layered approach. The ultimate goal is to preserve the aforementioned key characteristics of a layered architecture and in addition to allow for performance improvements and a new form of adaptability.

OPEN ISSUES WITH CROSS-LAYERING

Although the idea is straightforward and intuitive, several questions arise when dealing with cross-layer approaches [1]. From an architectural point of view, the information has to be shared somehow in a structured way. That includes access to the information itself and the interpretation and representation of the data, as well as the issue of altering it. Often, in the literature, some central information entity spanning all layers is used to illustrate a cross-layer protocol stack, leaving out detailed information about the internals. Furthermore, when there is interaction between layers, designing protocols truly independently might not be possible anymore.

In addition to the data-centric problems, other, more functional issues arise such as security and protocol stability. Suppose two layers try to optimize their protocol behavior in two different directions. For example, one protocol tries to be extremely energy efficient whereas the other one tries to increase the throughput. This might imply conflicting optimization goals and could result in an adaptation loop. In addition, such optimizations can implicitly have an effect on other protocols. Changing the transmission power on the physical layer might break links to neighboring nodes, which has a direct impact on the routing protocol. Considering these unintended consequences, the general question comes up whether cross-layering is a "good" architectural choice.

0163-6804/06/$20.00 © 2006 IEEE

The cross-layer extensions in WIDENS provide state information and parameter mapping between adjacent layers to increase protocol reconfigurability and adaptability. The cross-layer information is utilized in the case that in-layer optimizations cannot prevent performance degradation.

APPLICATION DOMAINS

Cross-layer approaches seem to be a promising new paradigm in that they open up a whole new set of possibilities in terms of performance and adaptability. On the other hand, they add a degree of complexity and could create adaptation loops. This might, if not thoroughly deliberated, jeopardize the stability and security of a communication system, which could reduce the longevity of such an architecture [1]. In centrally administrated, large, infrastructure-based, reliable, and commercially operated networks the advantages of a strict layered approach are predominant. Performance in those kinds of networks is usually increased by means of hardware whereas potential increases through architectural alterations are not considered attractive enough. In dynamic networked environments such as ad hoc networks, there are a lot of intrinsic performance bottlenecks and an obvious need to adapt to rapidly changing conditions. The wireless communication medium is shared and interference-prone resulting in bit errors, packet collisions, high and varying delays, and an overall lowered throughput. Aside from the difficult communication aspects, devices in such networks are likely to be battery-driven and relatively weak in terms of computational power. The mobility of nodes is also a significant system dynamic. It can affect the stability of routes, which in turn could cause broadcast storms to re-establish routes consuming large amounts of scarce resources such as bandwidth and energy. In such dynamic environments, cross-layer approaches are promising since performance and scalability can significantly be improved.

PROPOSED ARCHITECTURES

Different cross-layer frameworks and architectures have been proposed in the past. The Wireless Deployable Network System (WIDENS) project [2], supported by the European Community's research fund under the Sixth Framework Programme, for example includes cross-layering as a fundamental principle into the system design. The WIDENS architecture is an ad hoc communication system for future public safety, emergency, and disaster applications. The cross-layer extensions in WIDENS provide state information and parameter mapping between adjacent layers to increase protocol reconfigurability and adaptability. The cross-layer information is utilized in the case that in-layer optimizations cannot prevent performance degradation. Adaptation loops are avoided by only allowing interactions between adjacent layers. Information from nonadjacent layers can only be accessed through the mapping functions of an adjacent layer. This way, unnecessary and unintended cross-layer operations are avoided.

MobileMan [3] is another project supported by the European Community's research fund. It aims at investigating the potentialities of ad hoc networks in general by defining, designing, and evaluating a metropolitan ad hoc network. Their cross-layer architecture's main component is an information management entity called *Network Status* that is accessible by each layer. It is responsible for storing and organizing protocol data. The insertion and the access to that data are controlled by standardized procedures to achieve layer separation. This way none-cross-layer optimized protocols can still function within their framework. Since those protocols do not add information to the Network Status and do not use the information available, the optimization potential is unutilized but the protocol stack remains fully functional.

Some systems regard cross-layer interactions in a much broader sense. The GRACE project [4] for example considers four different layers which are the network layer, the application layer, the hardware layer, and finally the operating system layer all connected through a resource manager. GRACE differentiates between two kinds of adaptations, global and local. Global adaptations are triggered by the resource manager, which chooses the optimal configuration of each layer. They are global in the sense that all device components (software and hardware) have to be reconfigured. Local adaptations, on the other hand, only take place within a layer as defined by GRACE. The overall system is driven by application needs. The resource manager tries to choose the configuration combination which meets the application needs (utility metric) and at the same time utilizes the least resources (cost metric). The expected steady-state mode is the local adaptation mode which responds to minor changes in the system as long as the application requirements are not violated and the pre-allocated resource utilization is not exceeded.

Other systems are less generic in their design and are tailored towards specific goals using cross-layer interactions. One example would be the framework for data accessibility in ad hoc networks proposed in [5]. The cross-layer component is used for data exchange between a middleware and routing layer. The shared data comprises system profiles containing information such as location, mobility information, and transmission range just to name a few. Using that information data accessibility can be significantly improved.

Other approaches exist that only utilize data from different layers to optimize protocol behavior towards certain metrics like energy efficiency [6] without going through the trouble of describing an underlying architecture.

CROSSTALK: A NEW ANGLE ON CROSS-LAYERING

The work on cross-layer interactions carried out so far is very diverse, which reflects the dimension of the application domain. The one thing the different approaches have in common is the utilization of the information of one layer to improve the performance of a different layer's protocol. Specific problems that have been addressed are power and topology control, energy efficiency, quality of service (QoS) and more.

CrossTalk [7] abstracts from specific problems since it is a framework for cross-layer optimizations (Fig. 1). It differs from proposed cross-layer architectures in that not only locally available information is used to influence protocol behavior, but it establishes a global view of the network according to one or multiple metrics. Such a metric could be energy level, communication load or neighbor degree (amount of one-hop neighbors), just to name a few. Having such a global view at each node of the network, a node can utilize that information by comparing its own local status against the global status of the network. This comparison allows a node to evaluate its own *relative* status that can be used for decision processes such as the choice of algorithms, routing decisions, the placement of objects and services, load balancing, position estimation, and so forth. For example, if a node runs at 60 percent of its capacity, this value on its own has only little meaning. But compared against a global capacity utilization of only 10 percent for example, the 60 percent imply that the node is clearly overloaded compared to the rest of the network. That might lead to a higher collision probability at that node and quicker depletion of its batteries, causing routes to break. The knowledge of its relative state allows the node to act accordingly and lower its communication burden. More generally speaking, the idea behind CrossTalk is to base local actions on global knowledge.

Local actions as opposed to global actions are lightweight in terms of resource utilization. The clear disadvantage is, of course, the potential lack of accuracy. Global actions on the other hand are very expensive, but can achieve network-wide optimal results. Consider a reactive routing protocol and a route request as the global action. A request is flooded through the network to find the destination. Every node in the network might participate in finding the destination node. Ultimately, only a small amount of nodes will, after establishing the route, participate in the forwarding of data packets. But by involving a huge amount of nodes the route could be found and it is quite likely that an optimal route was found. Local actions do not involve other nodes, making them lightweight. But they lack information beyond the node's scope.

CrossTalk ultimately combines the advantages of both approaches to achieve global objectives at a low overall cost.

ARCHITECTURAL DETAILS

The CrossTalk architecture consists of two data-management entities. One is responsible for the organization of locally available information. This information can be provided by each layer of the protocol stack or other system components. Such data could be the current battery status, load, neighbor count, signal-to-noise ratio (SNR), transmit power, location information, or velocity. Each protocol can also access that data to utilize it for local optimizations. The sum of this information represents the state of the node or *local view* on the network.

The other data-management entity establishes a network-wide or *global view* of the same type of information collected in the local view. To produce the global view, CrossTalk provides a data dissemination procedure. Whenever a packet is sent, CrossTalk can enrich the packet with data from the local view by piggybacking the information. Every CrossTalk node receiving a packet extracts that information and adds it to its global view. This way, numerous samples are collected at every node in the network. By not generating data dissemination or control packets, the overhead is minimized. Additionally, only the source of a packet is adding information. Forwarding nodes only extract the origin's local information and do not add any additional overhead. This way, the packet sizes are only marginally increased, keeping the overall overhead low and making fragmentation due to increased packet sizes unlikely. When adding a sample to the global view, additional information could augment the collected samples. Besides giving it a timestamp, for example, distance information can be added if available. The topological distance in hops could be extracted from the packet header.

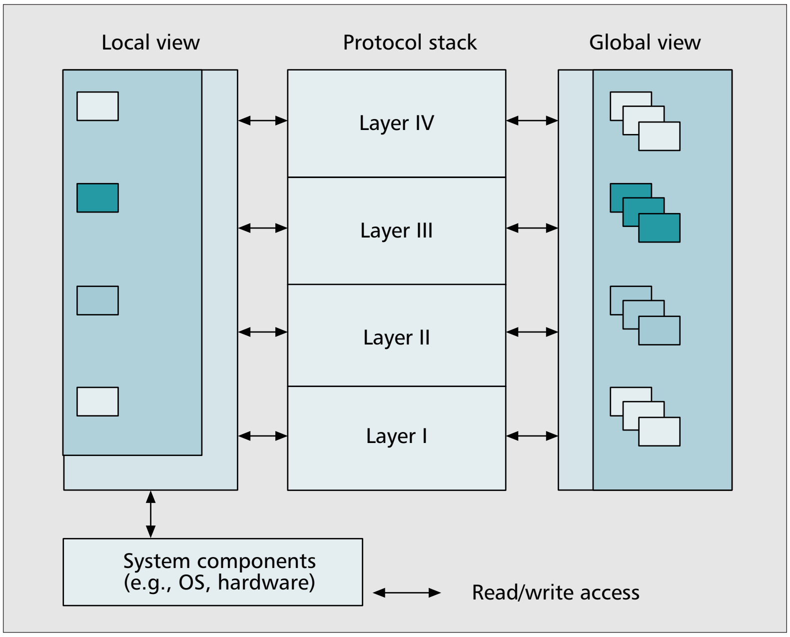

Figure 1. *The CrossTalk architecture.*

The collected samples themselves have to be aggregated somehow in order to represent more significant data. CrossTalk provides a set of algorithms to compute the global view. The simple mean value and weighted moving averages are the two types of averaging operations available. For the weighted moving averages, different variants following different intuitions exist. In addition, each of the two types can choose to include or exclude samples collected from neighbor nodes. The reason behind excluding a node in transmission range is that such nodes might, due to the nature of ad hoc networks, have similar local information influencing the global view. Direct physical neighbors will also more often add information to the global view, since every packet can be overheard. Consider the neighbor degree of a node: it is quite likely that neighboring nodes have a similar degree (amount of neighbors) due to the physical proximity of nodes. Including the local value samples from neighbors makes sense for metrics like battery status, though, since that does not necessarily have a spatial dependence.

The variants mentioned above differ in weighting factor and weighting function. The two weighting factors are distance and time, since time reflects the up-to-dateness of the global view and distance the spatial dependencies found in ad hoc networks. Both can be assumed to be available. The insertion time into the global view is used as the temporal weighting metric and the topological distance in hops is used as the spatial weighting metric.

The temporal weighted moving average has two variants for the weighting function. A linear weighting function is used giving the most weight to the most recently entered sample and proportionally less to samples entered into the global view after that. The other weighting function gives exponentially more weight to recent samples. The intuition behind the linear function is that it can more effectively smooth the effect of small temporal fluctuations, whereas the exponential function can more rapidly respond to sudden metric changes.

The spatial weighted moving average also has two different weighting functions. A linear weighting function is used too, giving more weight to samples coming from distant nodes. Here, the intuition is the same as in the case where neighbor nodes were excluded. Closer nodes might dominate the global view since packets from those nodes can more likely be overheard. In addition, depending on the metric, spatial dependencies might exist. The second weighting function is a triangular function, giving the least weight to samples from the closest nodes and also to the most distant nodes and the highest weight to samples from nodes with a medium distance. For the ascending part of the function, the intuition matches the one of the linear function. The descending part takes border effects into account. The edges of an ad hoc network have certain properties that might significantly influence the local view of those nodes. For example, the degree of nodes at the edges is most probably lower. Therefore, nodes far away are regarded as border nodes and are weighted less using the triangular function.

No matter which function is used, the global view can never reflect the exact global state of the network, and that is not what CrossTalk is aiming at. Instead, with CrossTalk a reasonably up-to-date view of the network can be generated.

No matter which function is used, the global view can never reflect the exact global state of the network, and that is not what CrossTalk is aiming at. Instead, with CrossTalk a reasonably up-to-date view of the network can be generated. Using the reasonably correct global-state information, the relative node state can be evaluated by comparing the global and local views. This comparison is the central aspect of CrossTalk.

The quality of CrossTalk's global view has to be assured when used for cross-layer optimizations. For example, a global view based on only a few samples will most probably not reflect the accurate global state. Therefore, thresholds can be set which control the global view computation. For example, if the number of samples is lower than a threshold, the global view will not be calculated. Whenever the global view is not calculated, the protocol using it will either have to rely on local optimizations or will be forced to resume the "normal layered" mode.

CROSSTALK IN COMPARISON

CrossTalk shares some similarities with the aforementioned architectures. The local view can be compared with the MobileMan architecture. A key characteristic is that none-cross-layer protocols can still work within the framework of CrossTalk. Cross-layer protocols themselves have to be able to run in a pure layered mode as well, since it cannot be assumed that the required information for optimization processes is always available. In other words, protocols should not be dependant on the cross-layer information. Consider a protocol optimized by utilizing GPS location information. As soon as the node enters a house, GPS might not be available anymore. To keep the network operational, CrossTalk protocols need to have a fallback mechanism to "regular" layered operation.

Compared to WIDENS, CrossTalk does not provide the additional structured access of information between layers. There are no mapping functions and no control of adjacent layers, giving CrossTalk a higher degree of freedom at the cost of having less control over optimization processes.

GRACE has a broader scope than CrossTalk. Although CrossTalk provides access to the local view for arbitrary system components, it only focuses on optimizations of the network protocol stack. In addition, the cost and utility model of GRACE is not required as well as preconfigurations of the protocol stack. This makes CrossTalk more flexible (as compared only to the network layer of GRACE) and more light weight (also because the controlling entity is missing in addition to the cost and utility model). On the other hand, it is less application-centric and does not treat the whole device as the optimization domain.

Most importantly, in comparison, none of the approaches offer a network-wide view of the network.

THE QUALITY OF THE GLOBAL VIEW

When used for optimizations, the global view has to be reasonably correct. This section gives details about the quality of the global view by analyzing simulation results. It also presents a reference application to give an example of how the global state information can be applied in a useful way. The simulations were carried out using ns-2 with a huge variety of parameter settings. Since the global view is potentially dependant on several system parameters, all identified dependencies were thoroughly analyzed over a broad range of settings. The network structure can have an influence on the quality of the global view. The network size (50 to 400 nodes), for example, can have a significant impact due to increasing distances between potential communication partners. The same applies to different aspect ratios of the network area (1:1 to 1:9). Different network densities (50 to 150 nodes/km2) were evaluated as well since in a sparse network, nodes overhear fewer messages. In addition, mobility was analyzed using the random way-point mobility model. Apart from the network structure, CrossTalk cannot be analyzed without taking the communication itself into consideration. There are no control or data dissemination packets. Therefore, the quality of the global view directly depends on the communication pattern and the network load (0.5 to 2 packets/s). The pattern distinguishes between two types of communication. One is local and describes communication within a certain number of hops. Global communication takes place with any node outside that radius. The patterns analyzed had a local communication share of 25 percent to 100 percent and local communication was within three hops. Finally, the global view is dependant on the time a node participates in an ad hoc network. The startup phase of a node (where it has to collect samples) is when the global view is the most fragile. Churn was analyzed to capture the effect of frequent initialization phases. Nodes in this scenario failed within 190 s after a lifetime of 60 s.

In all tested scenarios, the global view maintained a very high quality. The quality itself can be expressed using three different metrics. The first metric is the comparison of the average of all local view values from every node in the network with the average of all global views in the network. Ideally, the two values should match exactly at any given time. This, of course, holds only true if the standard deviation is zero. Therefore, the standard deviation of all global views in the network is the second metric, showing the degree of uniformity of the global view across nodes. The final metric is termed ‘correctness’. First of all, the average of all local views within the network is calculated giving the exact value for a perfect global view. Then, this value is compared to each node’s local view, which indicates whether a node’s relative state is above or below the network-wide average. Then, the node’s global view is taken and compared against the local view. If this comparison yields the same result (above or below the network-wide average), the node can estimate its relative state correctly. The fraction of nodes able to evaluate their relative state correctly is the correctness metric.

The disseminated information used to construct the global view was nodal load, as calculated in [7]. We placed the data dissemination functionality in the network layer. The reason for this is that the necessary header information is available here and software-wise this might be the “earliest” access, since lower-layer functionality might be deeply embedded in the firmware of the network interface. All tested scenarios yield similar results. Figure 2 shows the standard deviation of the global view in networks of different sizes with an average load metric around 35 packets per timeframe [7]. The graph does not show the variants that exclude the neighbors since they are similar to the corresponding algorithms that include the neighbor samples. The figure clearly shows the uniformity of the global view. All views have extremely low standard deviations. Surprisingly, the simple mean value outperforms the more sophisticated weighted moving averages (WMA). Nonetheless, with growing network sizes the distance-based global view algorithms seem to be more suitable.

Typical dimensions for the correctness of the global view can be seen in Fig. 3. The correctness slightly decreases with the density of the network but always remains well above 95 percent. This degree of correctness can sufficiently be used for decision processes, as described in the following section.

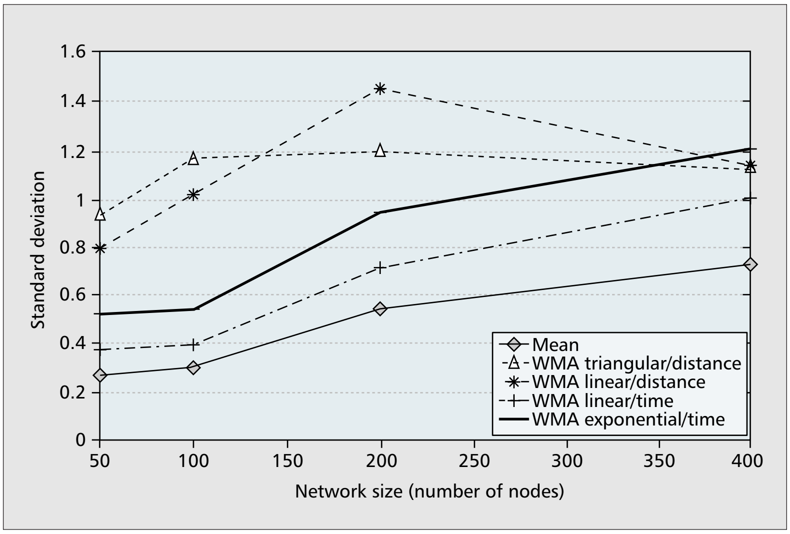

Figure 2. *Global view, standard deviation.*

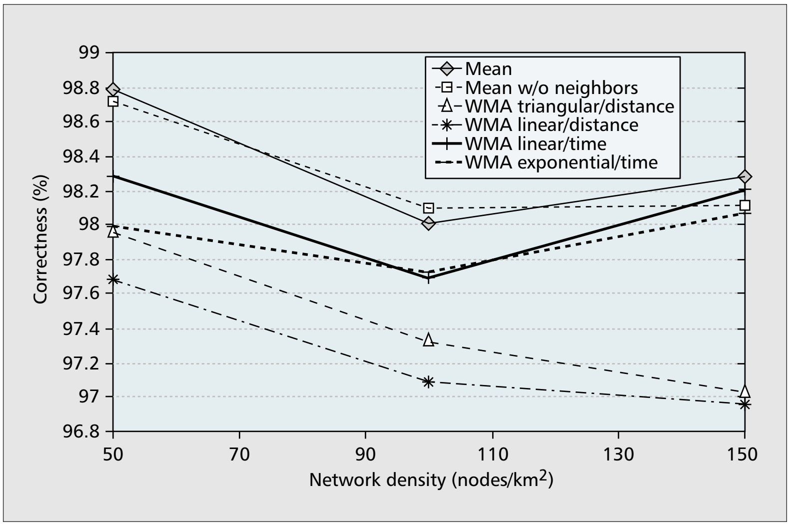

Figure 3. *Global view, correctness.*

CROSSTALK APPLIED

To have a reference application a load-balancing algorithm was developed based on the AODV routing protocol. Naturally, load is used as a metric for our decision processes. AODV worked the “regular” way whenever there was not enough data to construct the global view. It also worked in the original version whenever the global view suggested that the node is not overloaded. The cross-layer optimization algorithm only was invoked whenever a node found itself overloaded. The only operation that was influenced was AODV’s route request procedure. The general idea is that, if a node is overloaded, accepting new routes going through it will result in an even higher routing burden. To prevent that, routes should be established through nodes currently handling less communication traffic.

Assuming a shortest path-routing algorithm, the probability that routes go through the core of the network is high. To achieve load balancing, the routes need to be pushed towards the edges of the network. The load-balancing algorithm designed achieves that.

The algorithm itself consists of two phases (Fig. 4). The first one comprises the point from being overloaded, up to a certain overload degree, the delay bound. The overload degree is the ratio between local and global view (>1 → being overloaded). The second phase covers the region beyond the delay bound up to a second threshold, the drop bound. Up to the delay bound, route request packets are delayed before being forwarded by a delay proportional to the overload degree. Beyond the delay bound, route requests are dropped with a probability proportional to the overload degree. If the request is not dropped, it is delayed by the maximum delay. The first phase gives longer routes the chance to be established. Those routes are likely to lead through less congested network regions. When a node is significantly overloaded, dropping a route request will make sure that a different route will have to be established.

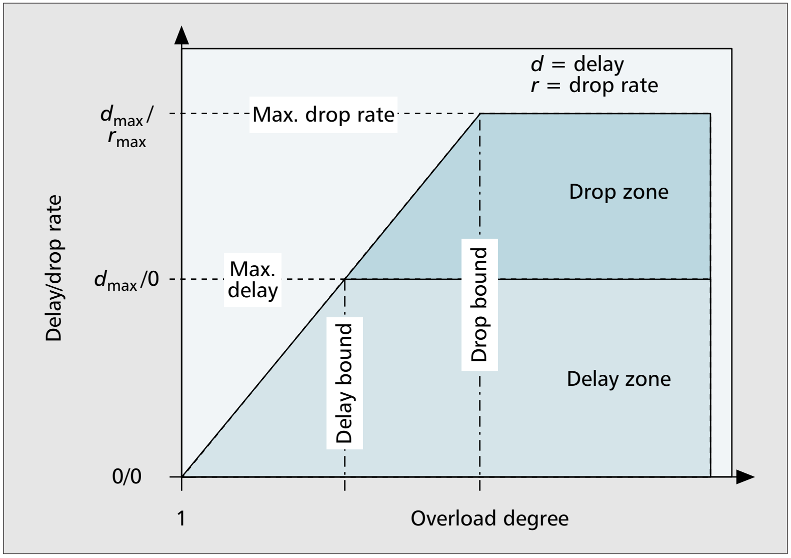

Figure 4. *Load balancing algorithm.*

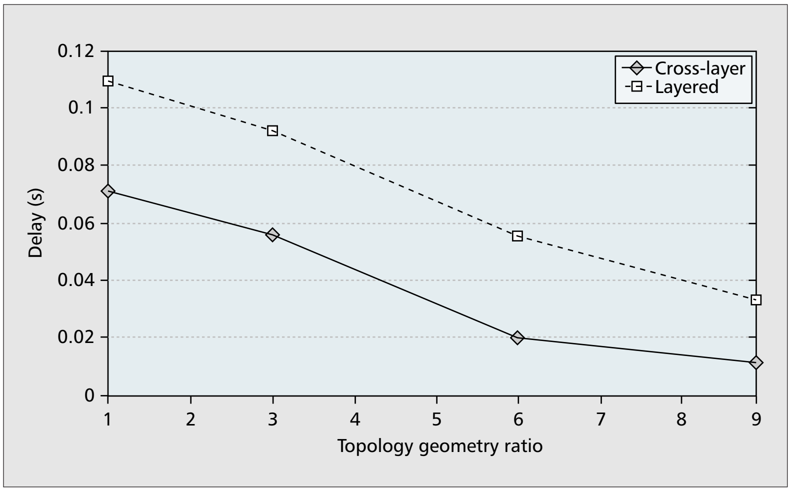

Figure 5. *Average per hop link delay.*

	WIDENS	Mobile Man	GRACE	CrossTalk
Focus	Device/application	Network	Network	Network
Network stack adaptation	Local	Local	Local	Local/network-wide
Added complexity	+	+	+++	++
Flexibility	+	++	+	++

Table 1. *Overview of some proposed architectures.*

Through the application of this load-balancing algorithm several objectives were achieved. The per-hop delay could be reduced by up to 65 percent (Fig. 5). This is only a side effect of a more evenly distributed load in the network. Other side effects were slightly higher packet-delivery ratios and overall less packets sent. The results can be seen more clearly when regarding bottleneck nodes. The load at those nodes was lowered by up to 25 percent. The coefficient of variance of individual loads can be regarded as the network wide effect of the load-balancing algorithm. Using CrossTalk, it could be significantly lowered. The coefficient of variance of the individual nodal loads compared to AODV in its original state was reduced by up to 23 percent.

CONCLUSION

This article has presented CrossTalk, a cross-layer architecture based on data dissemination. The data is used to construct a global view of the network in a distributed fashion. The global view itself is used to evaluate the relative state of a node. Decision processes can be supported based on this relative state information. Those decisions, local in nature, achieve global objectives. An in-depth analysis of CrossTalk's behavior in the face of network dynamics such as mobility and churn can be found in [7].

An analysis of the high quality of the global view was shown. Furthermore, a reference application was described and analyzed that exploits a global view on the load. By utilizing the relative load information, the proposed cross-layer protocol can achieve significant load balancing, thus demonstrating the effectiveness of CrossTalk.

Further research on CrossTalk analyzes the actual cost involved to establish a global view and deals with mechanisms to reduce that cost. Furthermore, other cross-layer adaptations are under investigation, especially adaptations to mobility-induced dynamics, which represent one of the unique challenges in wireless networks.

REFERENCES

[1] V. Kawadia and P. R. Kumar, "A Cautionary Perspective on Cross Layer Design," *IEEE Wireless Commun.*, 2005.

[2] R. Knopp *et al.*, "Overview of the Widens Architecture, A Wireless Ad Hoc Network for Public Safety," *Proc. SECON*, Poster session, 2004.

[3] M. Conti *et al.*, "Cross-Layering in Mobile Ad Hoc Network Design," *IEEE Comp.*, Feb. 2004, pp. 48–51.

[4] S. V. Adve *et al.*, "The Illinois GRACE Project: Global Resource Adaptation through Cooperation," *Proc. SHAMAN*, June 2002.

[5] K. Chen, S. H. Samarth, and K. Nahrstedt, "Cross-Layer Design for Data Accessibility in Mobile Ad Hoc Networks," *Wireless Pers. Commun.*, vol. 21, 2002, pp. 49–76.

[6] U. C. Kozat, I. Koutsopoulos, and L. Tassiulas, "A Framework for Cross-Layer Design of Energy-Efficient Communication with QoS Provisioning in Multihop Wireless Networks," *Proc. IEEE INFOCOM*, 2004.

[7] R. Winter *et al.*, "CrossTalk: A Data Dissemination-Based Crosslayer Architecture for Mobile Ad Hoc Networks," *Proc. ASWN*, June 2005.

BIOGRAPHIES

ROLF WINTER (winter@pcpool.mi.fu-berlin.de) received his Master of Engineering degree with distinction from the University of Portsmouth, England. Currently, he is pursuing his Ph.D. within the Berlin-Brandenburg Graduate School on Distributed Information Systems at the Freie Universitaet Berlin in the Computer Systems and Telematics group of Prof. Schiller. His research interests include wireless ad hoc networks, cross-layer design, mobile gaming, and peer-to-peer overlays.

NAVID NIKAEIN received his Ph.D. in communication systems from the Swiss Federal Institute of Technology (EPFL) in 2003. During his doctoral studies, he served as a research and teaching assistant in the Mobile Communications Group of Institut Eurecom, Sophia Antipolis, France. He is currently working as a research engineer on Eurecom's experimental software radio platform. His main research interests are in the area of mobile ad hoc networking and wireless communication. Specific subjects include medium access control, routing, topology control, cross-layer architecture, and software radio architecture.

JOCHEN H. SCHILLER is head of the Computer Systems and Telematics group at the Institute of Computer Science, Freie Universitaet Berlin, Germany. He studied computer science at the University of Karlsruhe, where he also received his Ph.D. in 1996 (summa cum laude). Afterward he joined Uppsala University, Sweden, where he taught and cooperated in several research projects. After being a guest professor at ETS Montreal, Canada, and Kiel University, Germany, he got his habilitation and published *Mobile Communications*, currently in use as a textbook by more than 200 universities. Since April 2001 he has been a full professor at the Freie Universitaet Berlin. In April 2003 he became dean of the Department of Mathematics and Computer Science. His research focus is on wireless, mobile, and embedded devices, communication protocols, operating systems for devices with small footprint, and quality of service aspects in communication systems.

CHRISTIAN BONNET joined Institut Eurecom as an associate professor in 1992. Since 1998 he has been head of the Mobile Communications Department of Eurecom. His teaching activities are in distributed and real-time systems, mobile communication systems, wireless LANs, and protocols for mobility management. His main areas of research are wireless protocols, wireless access to IP networks, and data communications in mobile networks, including mobile ad hoc networks.