Effective Mission Management through Service-aware Streaming Infrastructure

Marc Roelands,
Wolfgang Van Raemdonck
Bell Labs, Alcatel-Lucent
Antwerp, Belgium
Email: {marc.roelands,
wolfgang.van_raemdonck}
@alcatel-lucent.com

Stéphane Ménoret,
Laurent Navarro
Communication & Security, Services
Thales Group
Paris, France
Email: {stephane.menoret,
laurent.navarro}@thalesgroup.com

Ana Bildea,
Sébastien Creiche
Arago Systems
Nice, France
Email: {ana.bildea,
sebastien.creiche}
@aragosystems.com

***Abstract*— In this paper, we introduce an innovative, service platform-based approach to dynamic, world model-derived stream prioritization and selection, going beyond today's practice in urban security solutions. By validating the approach in a realistic urban simulation and resource-constrained wireless sensor network context, we demonstrate a significant improvement in situation awareness and effective resource management for security mission operations.**

Keywords-urban security; surveillance mission management; dynamic stream control; service platform; real world simulation; wireless sensor network.

I. Introduction

With the trend towards *Internet of Things* and the emergence of ubiquitous wireless connectivity in general, the capability to stream information, live and on demand, is pervading our society ever more, also extending the potential for urban security solutions. States and governments, ministries of defense and homeland security agencies, but by delegation also critical infrastructure operators or large private security agencies, aim to protect citizens and public infrastructure, and aim to monitor and act upon emergencies with various damage-mitigating actions, on a local up to internationally collaborative scale. With those aims, they invest in dedicated infrastructure roll-outs, potentially leveraging public / privately owned civilian infrastructure as well.

From a value proposition canvas perspective [1], the typical *pains* that organizations in charge of security face are: the unpredictability of each new security situation, the practical heterogeneity of information acquisition systems to be integrated, the unfeasibility of manual browsing through the available abundance of information streams (especially during crisis situations), and personnel budget scarcity (leading to cognitive overload of teams budgeted too small – as one operational person traditionally is assumed capable of handling around ten simultaneous video feeds, surveillance in large cities with thousands of cameras implies quite an extensive staffing).

Consequently, organizations in charge of security are seeking solutions that provide: faster incident-to-safety return response time during missions, flexibility to define before each mission the tool-based support needed, proactive situation awareness support, ranking information according to its relevance to the (dynamic) situation, and autonomous system behavior with respect to system overload handling in (often also mission-critical) high-load operational phases.

We see such requirements confirmed in examples such as the surveillance system in Mexico City [2]. Since 2010, more than 8000 video cameras, gunshot detectors, and license plate recognition systems, including even unmanned aerial vehicles (UAV), have been deployed in this case, requiring an extensive infrastructure with several C2 (Control and Command) and higher level C4I (Command, Control, Communications, Computers, and Intelligence) centers, and a force of 3000 specialized police agents operating the centers.

Security missions, such as high profile events and VIP protection (VIP: Very Important Person, e.g., a nation's president), are not yet fully managed by today's surveillance solutions, as they typically involve temporary, ad-hoc rolled-out security mission field infrastructure and services. From a security assessment point view [3], the main critical asset to protect then becomes the VIP, being subject to various lethal threads, including potential terrorist attacks in visited public areas. VIP protection, the live screening, monitoring and tracking of the VIP, as well as of suspected-malicious people, is complicated by a number of typical mission constraints. The area to be covered by a mission may be wide, freely accessible to the public and overcrowded, e.g., an exhibition centre with both indoor and outdoor sections to which a VIP visit was publicly announced. It may at the same time however be desirable to keep the actual protective measures hidden from the general public. Moreover, each mission follows a unique scenario, implying that history or routine from other missions often cannot be reused straightforwardly. A pre-assessment of the situation is performed before each mission, and a corresponding (multi path option) evacuation plan is typically prepared.

Next to the operational asset protection preparations and related risks, the rolled-out technical infrastructure itself also may have its vulnerabilities. Mission management often critically depends on a wireless network that can be affected by jamming, communication interception and replay. Therefore, mesh network technology, with link redundancy and multiple simultaneously used radio technologies are considered. In security missions where video streaming is

 ISBN: 978-1-61208-421-3

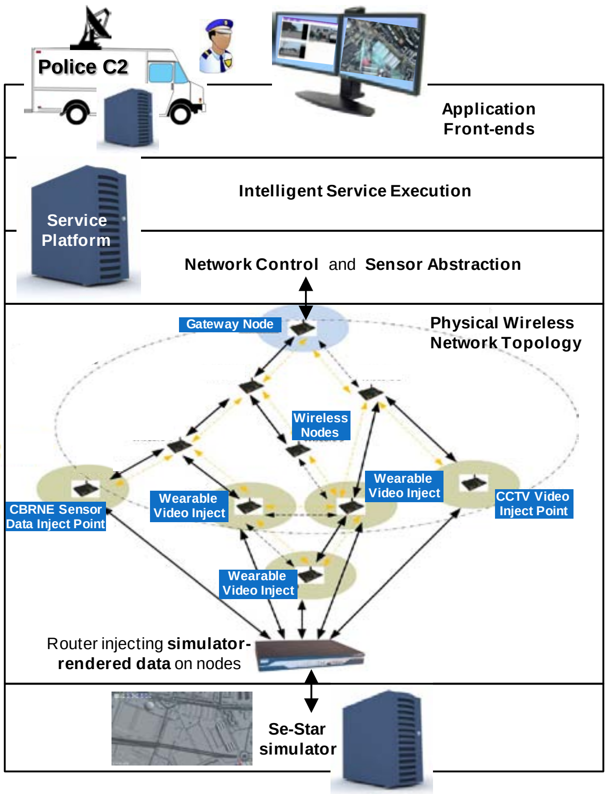


Figure 1. Overview of the experimental setup.

needed, the solutions should especially be able to cope with the limited bandwidth and battery autonomy of the ad-hoc rolled-out wireless network nodes.

In the next section we propose an innovative solution providing a considerable improvement over the existing solutions in use today for security mission management. Section III elaborates on the intelligence we established on the service platform, and Section IV makes a first qualitative and quantitative evaluation of the solution.

II. MISSION MANAGEMENT SOLUTION

Considering the mentioned security mission management challenges, we designed a solution allowing:

- *flexible addition of any required sensors*, be it ad-hoc for a mission or accessed via permanently installed, possibly privately owned infrastructure;
- *flexible addition / customization of mission services* (at a programmable service priority) that can *easily be launched during missions*;
- *situation awareness platform support* as relevant to the services requested during missions, so as to *lower cognitive overload* for the security team; and
- *autonomous prioritization of the (video) stream load* on the wireless network (links, nodes and devices), as dynamically dependent on the needs of the requested services (and so also leveraging the same platform situation awareness).

We distinguish following solution infrastructure layers:

- the physical (typically geographically spread) **devices** sensing the real world - our experiments include the simulation of, and content rendering for wearable and CCTV (Closed-Circuit Television) cameras, GPS (Global Positioning System) localization, and specialized toxic particles-detecting (CBRNE, Chemical, Biological, Radiological, Nuclear and Explosives) sensors,
- the **network** connecting the sensor range to a central service platform - our experiments use a prototype dual radio sensor board network,
- a **network control** and **sensor abstraction** platform level - our experiments use SDN-like (SDN: Software-Defined Networking) network control [4] and OGC-compliant (OGC: Open Geospatial Consortium) sensor abstraction [5],
- an **intelligent service execution** platform level - our experiments use a single-machine instance of the generic service platform running at a mobile C2 centre,
- a set of **application front-ends**, providing a user interface to the C2 team for video surveillance and CBRNE sensor monitoring services.

As a solution blueprint, applicable more broadly than in the security mission cases of our experiments, the design assumes a *horizontal, generically reusable* intelligent service platform. Key to the intelligence of this platform is that it leverages (service-independent) **world models** for enhanced situation awareness and prediction. While our experiments show that elementary approximations of such models can already be effective for the optimizations the platform provides, it is absolutely crucial to the *validation* of the solution to be able to embed it in a context of realistic and live real-world data streams. Particularly, as security missions cannot be easily 'rehearsed' in the real world, we validated the solution by means of the **sector-professional human behavior simulator SE-Star**, which we instrumented to provide the control interfaces and to render the (video and sensor) streaming content as would be available from the actual physical devices in actual field operation scenarios. (Such simulation is also used commercially today for system validation, field force training and large-scale exercises.) During actual missions, cameras and sensors (worn by policemen, ad-hoc fixed to walls, or as groups of fixed CCTV infrastructure) would each be connected to a sensor network node, with a *gateway* node eventually connecting the ad-hoc mesh network to the service platform in the C2 centre. Figure 1 summarizes the experimental setup, allowing to realistically validate the solution's effectiveness for real-world scenarios, without the actual real geo-physical deployment and action.

We consider a **guidance and evacuation scenario during a VIP visit** at a large exhibition center in Paris, considering also a potential toxic bomb threat, as the validation case. We use a realistic 3D mesh model of the exhibition center in SE-Star. With the SE-Star human behavior modeling (e.g., VIP-following or panic-motivated) and environment features (e.g., toxic gas cloud dispersion and impact), even a terrorist attack can be realistically simulated in the scenario.

 ISBN: 978-1-61208-421-3

From the application front-ends, that in practice would be installed in a truck near the mission scene, the C2 staff coordinates the protection team in the exhibition center, and should thus achieve effective live situation awareness concerning relevant events in the mission scene. For that purpose, an actual Wireless Sensor Network (WSN) is included in the experimental setup, as would also be rolled out in practice. In the scenario, the ad-hoc network is considered to be connected to the exhibition center's CCTV network. As such, about thirty cameras and a similar amount of chemical sensors, a GPS sensor worn by the VIP, and several cameras worn by policemen are connected to the physical network from Se-Star, at the corresponding network node boards. Their data streaming can be controlled from the service platform, as such dynamically injecting the data in the physical network.

Actual services made available to C2 staff in the service platform for activation during the mission (in line with what would be prepared during an actual mission planning phase) are:

- a *scene overview* service, providing the C2 staff with a general overview of the area they plan the VIP to be visiting,
- a *person monitoring and guidance* service, allowing the C2 staff to monitor the VIP on a planned visit trajectory, and command and guide evacuation when needed,
- a *crowd monitoring* service, allowing the C2 staff to detect and track crowds dynamically occurring in the scene (potentially hindering VIP evacuation), and
- a *toxic gas cloud monitoring* service, allowing the C2 staff to detect and observe the live impact of a chemical/bomb incident (complemented with a front-end for detailed CBRNE sensor readings) (again impacting VIP evacuation).

The next subsections zoom in on our WSN prototype and SE-Star.

A. *Wirless Sensor Network prototype*

We composed our WSN of extendible board prototypes as shown in Figure 2, each of which has a high (10 Mbps) and low (250 kbps) data rate unit. They support required higher level protocols such as CoAP (Constrained Application Protocol) and OGC SOS (Sensor Observation Service) [5]. (OGC SOS is used as a southbound interface to the service platform, as a standards-based way to interact with wireless sensors.)

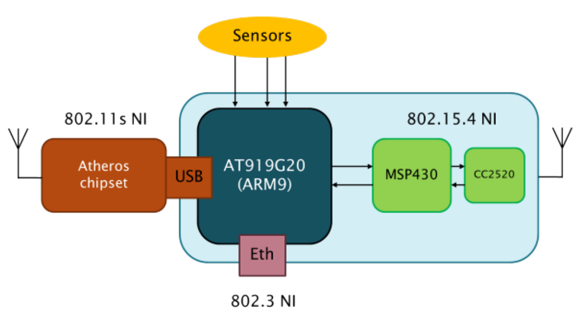


Figure 2. Network node block diagram

The **high data rate hardware unit** runs Linux on a 32-bit ARM processor with 64 MB SRAM (and extendible Flash memory), and has a IEEE 802.11b/g/n USB dongle for the video stream transport (and wired Ethernet for simulated stream injection). We use a IEEE 802.11s driver and Hybrid Wireless Mesh Protocol (HWMP) for routing the video streams in the typical Multi-Point-to-Point situation of the class of use cases at hand. Videos are streamed using RTP/RTSP (Real time Transport Protocol / Real Time Streaming Protocol) [6][7]. A CGI (Common Gateway Interface) web server is used for board configuration (particularly, flow configuration from the service platform).

The **low data rate / low power hardware unit** runs the small footprint Contiki operating system supporting IEEE 802.15.4 and 6LoWPAN [6], on a TI MSP430 16-bit microcontroller with 16 KB internal RAM and 256 KB Flash, with the TI CC2520 chipset for IEEE 802.15.4 low power networking in the 2.4 GHz ISM band. The Contiki IPv6 stack offers low power standard RPL (Routing Protocol for Low-Power and Lossy Networks) routing [7] and CoAP [8], making it ideal for the CBRNE sensor data video control streams.

We designed for **interoperation of the two hardware units**, allowing in principle to turn to an energy-saving low power mode for signaling, configuration and low bandwidth sensor data streaming only, at times when no video needs to be routed through a particular network node. Each video stream has been measured to add 150 mW of power consumption to a network node (added to a base consumption of 750 mW, which could be lowered in idle state by means of a duty cycle mechanism).

Figure 3 shows the network topology as enforced on the boards in the lab (where network nodes were laid out on tables), corresponding to the actually depicted *geographically spread* mesh organization that would be imposed in reality. (The manual enforcement is needed in the lab setup, because, without this, the nodes would connect as a full IEEE 802.11s mesh, while in reality the geographical spreading of the nodes would prevent links to exist between geo-distant nodes.) In the figure, dashed lines represent existing radio links in the topology whereas solid lines represent a chosen set of default routes within the topology.

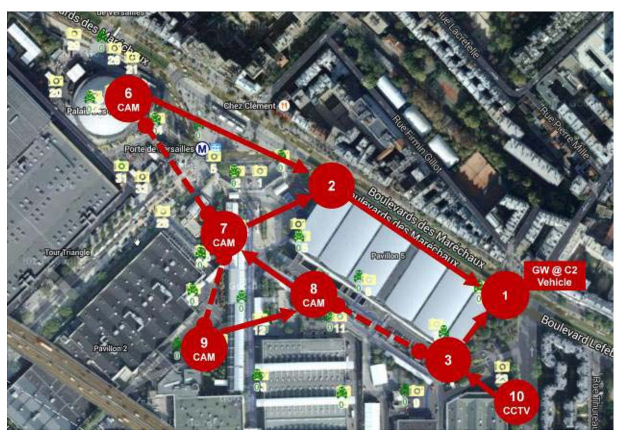


Figure 3. Enforced wireless network toplogy

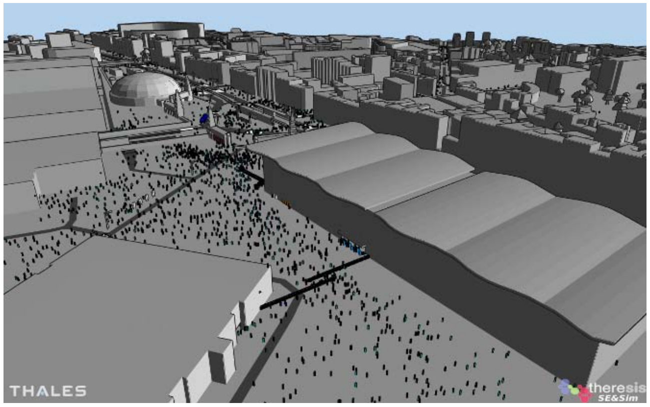


Figure 4. SE-Star urban security city simulator

Given a practical 5 Mbps driver limitation observed for the high data rate hardware unit and the known mesh multi-hop limitation (per-hop bandwidth halving in shared medium), a practical maximum of 5 video streams was observed for the actual setup (at an average 200 kbps per ‘high quality’ stream instance). This demonstrates that – even in later improved production-ready hardware conditions – it is of high importance to **smartly select streams dynamically**, and to use the alternative low data rate network for critical control signals.

B. SE-Star human behavior simulator

SE-Star, as illustrated by Figure 4, is a multi-agent simulator focusing on reproducing human behavior in large-scale environments [9]. It aims to provide an adaptive and modular tool for planning, decision-making and training purposes, on any kind of real or fictive scenario. To do so, it relies on a bio-inspired motivational engine, animating thousands of individual agents within real-scale critical infrastructures, interacting with each other and with objects reproducing real-life equipment. SE-Star is fully customizable, allowing the user to define the environment mesh, the agents' and objects' characteristics and behaviors, and the scenario script, in an easy way. It thus provides a reliable and realistic simulation basis for our experiments.

III. LEVERAGING REAL WORLD KNOWLEDGE FOR DYNAMIC OPTIMIZATIONS

With the experimental setup where the SE-Star virtual cameras and sensors can be controlled to inject requested live data and rendered video into the physical entry points of the network, configured in a realistically enforced network topology, the service platform can control and manage streams in a **fully realistic context**, and can be evaluated on serving the actual case-specific applications effectively. The intelligent service execution level of the service platform selects and prioritizes video stream loading of the network using **real world knowledge modeling** as a key enabling mechanism to add situation awareness and prediction in the service execution context. The mechanism allows the service platform to **dynamically select** the most critically needed and most relevant video streams for each service context, while simultaneously determining an overall platform **prioritization** for those streaming needs for the near future, as a dynamic and proactive means of network route and resource reservation.

Applicable beyond our current experiments, some platform design aspects need to be noted.

First of all, we make an explicit distinction between any *service goals* that may be requested by a user (i.e., during a mission in this case) and the real world facts expressed in the *world model*, which are pre-articulated by a *domain expert.* Service composition can be done by means of *service templates*, making referencing of world model elements from the templates straightforward. This serves as an **inherently scalable context-awareness approach**. Indeed, as if it were, context engine ‘model fractions’ are *woven into* the service composition during the process, inherently scaling up context processing according to service instance needs. (Sub-Section II.A. will discuss how this is achieved.) This thus avoids the well-known bottleneck seen in traditional presence servers in communication services [9] or the similarly implied need for elasticity solutions in publish-subscribe systems [10].

Using a world model in this way, in general, enables **proactive service-aware resource management** of the network and the deployed service processing, resulting e.g. in dynamic bandwidth reservation or distributed code placement, complementing reactive service-agnostic resource management autonomously by the system itself. In the case focused on in this paper, we consider de constrained WSN links as the resources to be proactively managed.

A. Service composition with knowledge weaving

The service platform thus has a knowledge base storing world model elements, describing the **behavioral constraints of particular real-world phenomena,** and a set of service templates. Figure 5 shows an example of such a phenomenon behavioral constraint as used for the current prototype use case. In this example, we express a first order prediction of *the movement of a person* for which the current position can be observed and a set of course waypoints are known (as the case with a prepared VIP visit plan). Particularly, the code in Figure 5 shows a function representing this knowledge, returning a person’s 2D geo-position predicted for a time *lead_time* ahead, based on the current observed position *c_pos* (at current time *c_t*), the last observed position *l_pos* (at time *l_ts*), and a set of planned positions (*path*) from course waypoint known from the VIP visit plan. The (somewhat arbitrarily chosen) heuristic captured by the function is that the extrapolation based on the speed estimate is vector-wise corrected by averaging its direction with the direction perpendicular to the line segments of the nearest course planned waypoints (i.e. towards the planned path).

Models of similarly elementary nature are devised for other relevant phenomena in the application domain and validated in the experimental setup, most notably, occurrence and prediction of *crowds*, based on observed people density and individual speeds, and predicted *dispersion of toxic gas*


```
ppos_predict(lead_time, path, (c_ts,c_pos), (l_ts, l_pos))
    dt = l_ts - c_ts
    speed = dist/dt
    vip_dir= track.compute_ori(l_pos, c_pos)
    to_path= perpendicular_lines(path, c_pos)
    est_dir=track.normalize(vec2_sum(vip_dir,
                                     track.normalize(to_path)))
return track.latlon(vec2_sum(c_pos,
                    vec2_mul(est_dir, speed*lead_time)))
```

Figure 5. Moving person as example of a real-world phenomenon; above: planned trajectory on map; below: phenomenon world model fragment

clouds, based on above-threshold CBRNE sensor measurements and a first order circular expansion model.

Much more complex world models could be used, where available or when they can be generated. When also sufficient live observation data would be available, this could even lead to more accurate predictions. However, our experiments have shown that the used of just **coarse estimates can already be turned into considerable operational advantage**.

As an example service template, all *monitoring* type of services in our experiments use the service template shown in Figure 6. The service templates are implemented as highly parameterized directed graph descriptions of connected execution primitives. In the example, with parameters in bold text, the graph composes the essential elements of phenomenon visual tracking, pan-tilt-zoom control of cameras, selection and activation of camera streams and prediction of such for network control, and ultimately video mixing for displaying the video streams in the designated user interface area. Apart from the set of camera sources that

```
showme [icore:phenomenon]
       [icore:camera/set]
       [icore:sink]  {
[icore:phenomenon/observation/pos] -> geos_filter ->
    select_ctrl -> ptz -> axis_ctrl[icore:camera/ptz_config];
[icore:phenomenon/observation/pos] ->
    ptz[icore:camera/ptz_config];
[icore:camera/set] -> geos_filter[icore:camera/coverage] ->
    c2service -> streams -> flow_ctrl;
c2service -> mixer -> [icore:sink/video];
streams -> [icore:camera/video] -> mixer;
[icore:phenomenon/observation/prediction] -> flow_ctrl;
}  [icore:showme_service]
```

Figure 6. Example service template logic

the service should consider (in this case, all cameras available in the exhibition centre as registered for the mission), destination sink for the (raster-mixed) video streams (in this case, always a designated area in the C2 application front-end user interface, i.e., network-wise behind the gateway node of the WSN), the real-world phenomenon to be observed and status-predicted is a template parameter. When instantiating the template, it is fed with the appropriate phenomenon description, e.g. the observation and status-prediction of a person, in this case the VIP. The knowledge base holds the implications of that choice, e.g., the type of further sensor input (Global Positioning System readings of the device carried by the person), the way how to observe and track the person via a camera, and how to predict the behavior of the person, e.g., how to predict position, i.e., the example behavioral knowledge expressed in Figure 5. In this way, the **full description of the service instance graph can be derived for a *service request***, as a weaving of the applicable service template logic and the applicable phenomenon behavioral knowledge.

B. *Execution of service instances*

The system thus *instantiates* a service template upon receiving a user-issued *service request*, extracting parameter values from the request, and referencing any relevant real world phenomena behavior from the knowledge base. Upon full composition of the requested service instance (as was discussed in Section III.A.), the executable description of the service instance is **deployed and started as a data stream processing graph**.

From the template example in Figure 6, we see that all service instances produced from that template have their graph wired in the overall execution graph of Figure 7 according to the links denoted with "->" in the template. Based on their nature, some of the composing processing nodes are instantiated per service instance (e.g., mixer), while others are common to all service instances or considered infrastructure components (e.g., flow_ctrl and streams, as the *Flow Ctrl* and *Stream Mngr* blocks displayed in Figure 7, respectively). The template parameters (bold in Figure 6) are used to inject specific bindings, among which also the specific phenomenon observed, as corresponding to the issued service request. E.g., when a *person monitoring and guidance* service is requested by the C2 staff, a service instance is deployed and executed filling the template with the observation and prediction logic for *movement of a person* is used for the phenomenon parameter. This implies via further template and world model elaboration, e.g., that the function ppos_predict from Figure 6 is embedded in a processing node called path_estimation in Figure 7.

Figure 7 thus shows four such concurrent service instances resulting from service requests in the example scenario experiments. The shaded ellipsoid graph parts show where world model dependencies are inserted, thus leveraging properties of one or more relevant phenomena either for selecting or controlling cameras, or for predicting near-future observations of the phenomenon, for requesting a corresponding network provisioning. The *Stream Mngr*

 ISBN: 978-1-61208-421-3

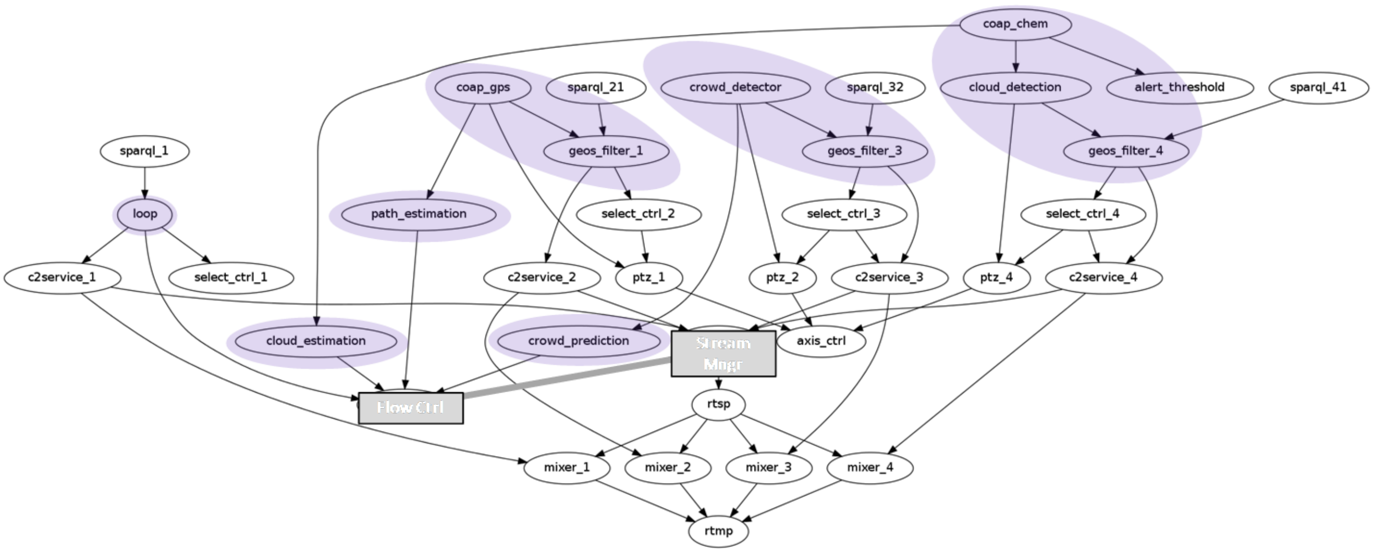


Figure 7. Example data processing graph resulting from 4 example service requests

block collects the camera selection requests and handles the actual stream control for the cameras over the network.

C. *Service-aware stream priority control*

In analogy to the SDN network control concept [4], the service platform considers the notion of *'flow'* as a programmability abstraction for the network. As part of the service platform, a **flow controller** (*Flow Ctrl* block in Figure 7), associated one-on-one to a particular network, dynamically declares which **flows** – defined by a source and destination address and required **QoS** (Quality of Service) characteristics – need to get a particular **(reserved) network priority**. When an actual **stream** is requested to be transported over the network by the **stream manager** (*Stream Mngr* block in Figure 7), after a given maximum *network reconfiguration time* $t_{maxconf}$, the stream gets assigned to a particular flow, and so is handled according to the flow's declared priority and QoS. In the experimental setup, flows correspond to particular camera to network gateway combinations and an associated bandwidth requirement, as a simplified QoS characteristic, either "High Quality" (HQ), corresponding to an appropriate stream bandwidth for a "main" video screen area of a service, or "Low Quality" (LQ), for all other video streams.

The flow controller determines overall network flow priorities and QoS level dynamically by linearly combining the relative priorities predicted for each candidate requested stream for each service instance, based on the priority of *each service instance* (as determined by the C2 staff) and the dynamic *likelihood of relevance of each candidate stream* in that service instance. The dynamic likelihood of relevance for the candidate streams is determined by the respective services' phenomenon predictors (arrows from prediction graph edges to *Flow Ctrl* block in Figure 7), which thus implies predictions need to happen $t_{maxconf}$ ahead of time, to allow for network reconfiguration.

Furthermore, the flow controller takes into account the *limited capacity of the network* (overall or for particular flows) ensuring no overloading in any upcoming time interval, thus timely and dynamically provisioning the network for only the highest-priority-ranked flows. The stream manager consequently only allows actual streams in provisioned flow paths, and **blocks non-flow-reserved stream requests** to prevent network overloading.

In the current setup, the actual **routing** for requested flows is dynamically chosen according to one out of a set of overall routing plans that have been pre-determined to be suitable for the network topology at hand. As flow requests arrive to it, the flow controller thus heuristically requests the network to keep the current, or switch to another overall routing plan. The typical mesh bandwidth division effect and the shared medium indeed essentially limit the transmission capacity of the particular network setup, rendering finer-grain routing optimization for overall bandwidth improvement less useful for the particular use case. Beyond bandwidth utilization, the heuristic routing plan switching is beneficial for **load spreading across the network**, avoiding individual wireless network node battery drain, which would degrade the network critically.

IV. EVALUATION

We conducted a range of experiments, assuming the VIP guidance and evacuation scenario, and typical C2 user interactions and user service requests. In the next subsections, we revisit the requirement assumptions, the resource use effectiveness, and the actual mission management effectiveness, as concluded from the real-world validation context for the intelligent service platform.

A. *Qualitative validation of requirements*

Using the realized prototype as a premise, we interviewed commercial domain experts. They confirmed a correct requirements focus, and recognized the operational

 ISBN: 978-1-61208-421-3

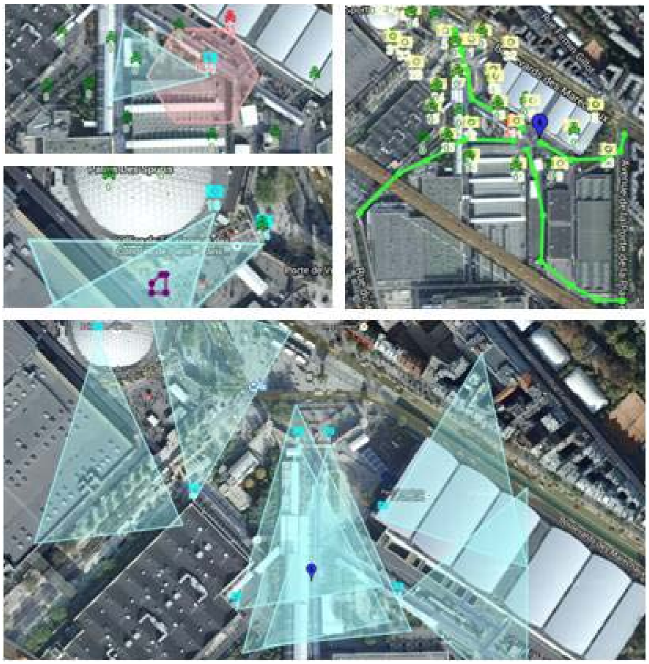


Figure 8. Snapshot impression of live geo-overlay C2 user feedback: (top left) crowd and toxic gas cloud anticipations, (top right) candidate VIP evacuation paths, (bottom) active-stream camera orientation

improvements targeted by our system as highly valuable indeed, however adding as a remark that, even when kept rudimentary, the **world model, being an evident dependency** for the system should be provided by an ‘impact-conscious’ domain expert, for ensuring reliable system behavior. As another remark, C2 staff should anyhow also be able to **verify and potentially override the decision support** stream selection.

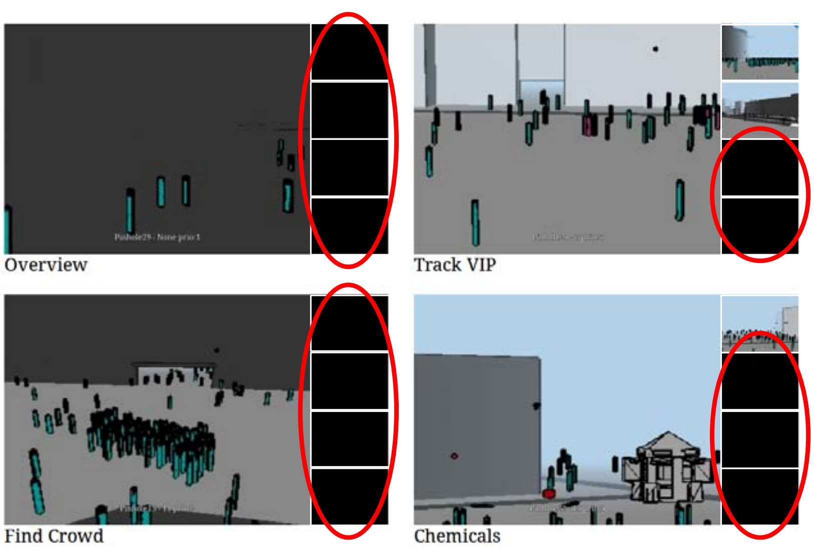
To partially address the decision-overridability concern, as shown in Figure 8, we added **live visual user feedback** in the application front-ends for C2 staff to be able to verify the system’s world model-based analysis and selection, by means of geo-map overlays showing phenomena predictions, camera streaming status and orientation, and CBRNE sensor values. The figure also shows the clickable, situation-dependent evacuation route proposals foreseen in the guidance service (example trajectories in top-right snapshot of the figure).

B. *Qualitative and quantitative validation of resource use effectiveness*

The platform’s prioritized camera activation results in a user experience as shown on the left side in Figure 9. Shown are the video matrices of four simultaneously active service instances (one for each planned service type planned for the scenario, as an example corresponding to Figure 7). As decided by C2 staff for the given mission, each service has been given a relative priority, e.g. VIP tracking given priority over crowd monitoring. Further, in line with mission management expert preference, the main screens (shown larger to C2 staff, and streamed at HQ stream quality) should *always* display the camera source of highest relevance to the requested service, while camera sources of less relevance to the respective services are displayed on side screens (at LQ stream quality), *as far as* possible within the actual network loading.

Without the prioritized dynamic camera activation by the service platform across all active services, random stream drops would occur at network overload, resulting not only in a bad user experience, but possibly even the drop of the video images most critical to the C2 staff decision taking.

With the dynamic priority flow reservation mechanism described in Section III however, i.e. with the service platform being able to predict based on phenomena behavioral knowledge which camera sources will become most relevant for the active services at any given time, the service platform can **rank flows according to their eventual streaming relevance**. As illustrated in the table on the right in Figure 9, the (dynamically changing and



Priority	Flow ID	QoS	Allow	Strm ID
5005	10	HQ	Yes	19
3005	10	HQ	Yes	21
2005	10	HQ	Yes	14
1005	10	HQ	Yes	28
54	9	LQ	Yes	25
53	10	LQ	Yes	6
34	10	LQ	Yes	10
33	7	LQ	No	27
32	10	LQ	No	5
24	10	LQ	No	12
23	10	LQ	No	31
14	10	LQ	No	7
13	10	LQ	No	18
12	6	LQ	No	29
11	10	LQ	No	20

Figure 9. Video user experience (left), and ranking tabel (right) for highest priority stream selection

 ISBN: 978-1-61208-421-3

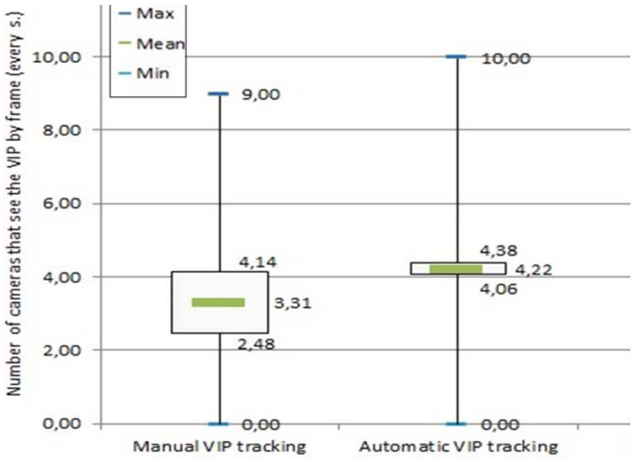


Figure 10. Automated versus human video selection

predicted) single most relevant camera source for each service (by its flow and stream identity) is assigned the highest priority (and, among each other, are ranked according to relative service priority). Camera sources with lower predicted relevance in the table, i.e. those that are estimated to be candidates for side screen display, in LQ, are dynamically priority-ranked according to their predicted relevance for each of the running services. In this way, overall, the least relevant streams are prevented from loading the network in favor of the most relevant ones. As the figure illustrates, this results in black screens (i.e. no content) to occur only for the least critical camera views. As such, **a far more acceptable user experience** is realized, fulfilling the critical mission decision support better than with a non-intelligent solution where blocking occurs randomly.

In fact, a **near-maximal hit-rate** can be seen to be reached with respect to what can be optimally obtained under the maximum overall network capacity for the given setup, as, under the mission preferences and requirements outlined in the beginning of Section IV.B., the (in general NP-hard) Knapsack optimization problem can be approximated by the simple ranking at each flow selection update cycle. (In rare instances where the phenomenon predictions would be radically wrong, there is a risk of not reserving flows properly, but the live experiments under the realistic SE-Star simulation phenomena behavior have shown that the approach appears robust against this.)

As for comparison to other stream prioritization approaches, we found Chen et al. [13] to confirm the urban security sector-relevance of concurrent tracking services, realized via a multi-service-programmable platform with a service priority notion, under the resource constraints of a WSN serving data streams from cameras, CBRNE sensors and other sensor types. Chen's approach uses a similarly elementary model for movement prediction (but considers for that an energy-friendly, distributed, cooperative tracking logic deployed across the sensor nodes) and uses a similar service-aware relevance and priority metric to dynamically determine which streams should get prioritized in the network. The main difference between both approaches lies in the fact that Chen solves resource conflicts by *controlling congestion locally* in the network nodes (favoring the most relevant streams by auction logic among the network nodes), while our approach is rather *proactively avoiding* congestion at a *global* level. Further experiments would be needed to compare the approaches in this respect, although results may be expected to be of comparable merit. While a more fine-grain optimization may be achieved in Chen's approach, at a cost, the specification flexibility of phenomenon models and service templates in our solution, and the fact that no processing is required for it on the constrained, battery-powered network nodes themselves, may however be a practical advantage in C2 or C4I mission management cases.

C. *Objective validation of operational success*

As today's C2 surveillance systems often offer only *manual* video selection and camera control, a **metric of operational success** is how well video streams are selected by the system compared to what a human operator would be able to do when presented with the same live data. Figure 10 makes this comparison for a human operation in the same VIP guidance and evacuation scenario. For obtaining the results in the figure, we conducted a validation experiment repeating the (non-deterministic) SE-Star simulation of the scenario 20-fold. Half of the runs was conducted with a human operator just activating the *person monitoring* service and observing the automated selection via the application front-end on top of the service platform (including its network stream prioritization). The other half of the runs was performed with a human operator manually selecting the virtual camera views directly in SE-Star (without network). To automate the counting of validly selected camera views during each scenario run, SE-Star was enhanced with the ability to count the number of (manually or automatically) selected cameras for which the selection was done successfully, i.e., for which the simulation indeed rendered the VIP visible to the camera view. The resulting statistics in Figure 10 show that the service platform has a higher success score in all runs, at a smaller variance. This gives a confident indication that **our automated system selection is always at least as good, and often better, than manual human selection**, for the considered type of mission scenarios. Apart from prioritizing network loading, the service platform moreover is able to *simultaneously* automate decision support for *multiple* services, tracking several different phenomena.

V. CONCLUSION

In this paper, we reported on the validation of a service platform prototype in an urban security context. We have set up experiments in such a way as to simulate the operational context for the platform as realistically as possible, using a city simulator and a constrained wireless sensor network. We found the dynamic world model-based stream prioritization across ad-hoc requested services to obtain a significant operational enhancement over current mission management tools, with respect to user experience as well as network utilization. Next to inclusion of the proposed features in product candidates, we envision generalizing on the service platform concept for broader cases and full horizontal applicability, and benchmarking it further against alternative approaches for intelligent stream selection and network resource control.

ACKNOWLEDGMENT

This work was performed in the context of the FP7 iCore project as supported by the European Commission.

REFERENCES

[1] A. Osterwalder, Y. Pigneur, G. Bernarda, A. Smith, and T. Papadakos, Value Proposition Design: How to Create Products and Services Customers Want. Wiley, 2014, ISBN: 978-1118968055.

[2] Thales Group. *Mexico City, the world's most ambitious urban security programme.* Available from: https://www.thalesgroup.com/en/worldwide/security/case-study/mexico-city-worlds-most-ambitious-urban-security-programme [retrieved: May 2015]

[3] G. K. Campbell, "Types of metrics and performance indicators appropriate to the security mission," in Measures and Metrics in Corporate Security, G. K. Campbell, Elsevier, pp. 21-56, 2014, ISBN: 978-0-12-800688-7.

[4] Open Networking Foundation. *Software-Defined Networking: The New Norm for Networks. ONF White Paper*, Apr. 2012. Available from: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf [retrieved: May 2015].

[5] Open Geospatial Consortium. *OGC Sensor Observation Service Interface Standard Version 2.0.* Apr. 2012. Available from: http://www.opengis.net/doc/IS/SOS/2.0 [retrieved: May 2015].

[6] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. *Internet Standard RFC 3550: RTP: A Transport Protocol for Real-Time Applications*. July 2003. Available from: https://tools.ietf.org/html/rfc3550 [retrieved: May 2015].

[7] H. Schulzrinne, A. Rao, and R. Lanphier. *Proposed Standard RFC 2326: Real Time Streaming Protocol (RTSP)*. April 1998. Available from: https://tools.ietf.org/html/rfc2326 [retrieved: May 2015].

[8] J. W. Hui and D. E. Culler, "6LoWPAN: Extending IP to Low-Power, Wireless Personal Area Networks," IEEE Internet Computing, vol. 12, no. 4, pp. 37-45, July 2008, doi: 10.1109/MIC.2008.79.

[9] L. Navarro, F. Flacher, and C. Meyer, "SE-Star: a large-scale human behavior simulation for planning, decision-making and training". The 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 2015, pp. 1939-1940, ISBN 978-1-4503-3413-6.

[10] T. Winter, et al.. *Proposed Standard RFC 6550: RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. March 2012. Available from: https://tools.ietf.org/html/rfc6550 [retrieved: May 2015].

[11] Z. Shelby, K. Hartke, and C. Bormann. *Proposed Standard RFC* 7252: *The Constrained Application Protocol (CoAP).* June 2014. Available from: https://tools.ietf.org/html/rfc7252 [retrieved: May 2015].

[12] L. Lin and A. Liotta, "A Critical Evaluation of the IMS Presence Service," The 4th International Conference on Advances in Mobile Computing and Multimedia (MoMM 2006) Austrian Computer Society, Dec. 2006, pp. 19-28, ISBN 3-85403-215-3.

[13] M. Li, F. Ye, M. Kim, H. Chen, and H. Lei, "BlueDove: A Scalable and Elastic Publish/Subscribe Service," IEEE International Parallel & Distributed Processing Symposium (IPDPS 2011), IEEE Press, May 2011, pp. 1254-1265, doi: 10.1109/IPDPS.2011.119.

[14] L. Chen et al., "Dynamic service execution in sensor networks," The Computer Journal, vol. 53, no. 5, pp. 513-527, June 2010, doi: 10.1093/comjnl/bxp051.