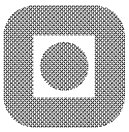
NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY, MATHEMATICS AND ELECTRICAL ENGINEERING



PROJECT ASSIGNMENT

Student's name: Håkon Rørvik Aune

Course: TTM4510

Title : **Electronic newspaper over WiFi**

Text: The project assignment shall evaluate the challenges associated with introducing newspapers distributed on electronic paper with WiFi. It shall include the following:

- a theoretical study of the different electronic paper technologies.
- a study of the different devices that can be used for receiving electronic newspapers.
- a study of the different technologies that can be used to adapt a newspaper to the electronic paper format.
- a study of previous and current attempts at e-newspapers
- a theoretical study on different solutions for distributing e-newspapers over WiFi.

Deadline: 2007

Handed in: 2007

Carried out at: Department of Telematics

Supervisor: Lene Maria Myhre, Trådløse Trondheim

Trondheim, ……2007

Yuming Yiang
Professor

Electronic Newspaper over WiFi

Håkon Rørvik Aune

December 2007

Abstract

Electronic paper (e-paper) and wireless networks is a combination that could bring forth new exiting services. E-newspapers is one of these, and can be seen as a newspaper delivered to an e-paper device over a network. This paper will evaluate different aspects related to such delivery of e-newspapers over WiFi networks.

Layout and structure is one of the areas where different solutions is apparent. Usual (e)books is navigated by pageflipping while web sites uses hyperlinks. Viable approaches for e-newspapers would be either of the two, or maybe a combination.

Distribution is another area with many alternative approaches, which also needs to cover the shortcomings of mobility and wireless networks. Distribution can be done by push or pull means, where push methods is most likely the most convenient method for subscribers. Using a Publish/Subscribe kind of middleware for an e-newspaper system will uncouple the publishers and subscribers and can provide an efficient tools for subscribers to subscribe to a plethora of content from several sources.

Wireless delivery of content makes the subscribers and their devices prone to the issues of mobility. Means to solve this can be store and forward of content by the Publish/Subscribe system that lets the subscriber resume broken transfers. Push-Pull can also be a possibility; the middleware will push a notification that the device uses to pull content when it is in a state favourable for commencing a download.

Preface

This is the 2007 Fall Project on the Masters Program at the Department for Telematics, NTNU. It was written by Håkon Rørvik Aune with supervising by Lene Maria Myhre, Wireless Trondheim and faculty supervising by Professor Yuming Jiang.

I would like to thank my supervisor Lene Maria and professor Yuming for the help provided during the project. I would also like to thank my office mates Johan and Erik for an entertaining semester working on this project assignment.

It would not have been the same to write this report if not LaTeXnor Ubuntu existed.

Nothing is impossible.
-*Anonymous*

Contents

List of Figures

List of Tables

1 Introduction

How can devices with electronic paper display and WiFi capability help a newspaper company realise electronic newspapers distributed over a wireless network such as the one provided by Wireless Trondheim (Trådløse Trondheim)? This paper seeks to answer that question. The local newspaper Adresseavisen will be used to provide examples and scenarios when suitable.

1.1 Background

Electronic paper(e-paper[1]) is a new technology emerging that combines the readability of ordinary print on paper with the possibility of carrying the equivalent of multiple encyclopedias in a unit the size of a PDA. Wireless networking gives a user the possibility to access Internet and other resources from where ever he is, given that a wireless network is available. In Trondheim such a wireless network is established by Wireless Trondheim and are covering most of the downtown area of Trondheim. A combination of such an e-paper device and WiFi capability can give great possibilities. In this paper these possibilities shall be explored with focus on how a newspaper can be adapted for e-paper devices as well as distributed with wireless network, and other challenges related to this.

1.2 Problem to be addressed

The project assignment shall evaluate the challenges associated with introducing newspapers distributed on electronic paper with WiFi;

[1] If nothing else is stated "e-paper" will be used for the electronic paper display technology, and "e-paper device" for the reader device that is equipped with electronic paper display technology. Further, the notion e-newspaper is a newspaper that are loaded on an e-paper device unless said otherwise.

- different electronic paper technologies.
- devices that can be used for receiving electronic newspapers.
- different technologies that can be used to adapt a newspaper to the electronic paper format.
- previous and current attempts at e-newspapers.
- different solutions for distributing e-newspapers over WiFi.

1.3 Limitations of problem

This paper does not delve much into the aspects of authentication nor authorisation. Business models and how subscriptions should be paid and such is not covered in any depth either.

1.4 Structure

First the theory related to electronic paper (2.2) and devices (2.3) using it is presented in order to give an understanding of their properties and why it is chosen for this task. Then theory related to content networking (2.4) is presented, to give a picture on how a newspaper can be distributed to such e-paper devices. This includes content adaption (2.5), content delivery networks (2.4.1) and pre-caching on wireless devices (2.4.2). Then (3) the theory is tied in with the problem at hand, before a possible solution is described. In the last part a conclusion (4) is provided.

2 Theory

First, Wireless Trondheim and Adresseavisen is presented and their role in this project is described. Next, electronic paper and the devices that uses this technology is introduced. Proceeding, content delivery for mobile units is described. The theory presented here are of the kind that should be relevant to the problem at hand, delivering newspapers to e-paper devices over WiFi. It will partly also be touched upon theory that might be of relevance for future extensions of an e-newspaper, such as the next iteration of e-paper devices with extended possibilities (i.e. video and colour) and what such devices might need to perform.

2.1 Wireless Trondheim & Adresseavisen

2.1.1 Wireless Trondheim

Wireless Trondheim is a research and development project with a goal of covering central Trondheim with a wireless network. This network is to be used for research and development on and about wireless services; See Table 2.1 for the most important topics of interest. The three first areas of research in Table 2.1 are covered by Wireless

Table 2.1: Research areas Wireless Trondheim [3]

1	Business models for mobile telecom services
2	Usage Patterns
3	Service production and evaluation
4	Telecom Protocols
5	Radio interfaces

Trondheim's own service research and development lab. This lab supports the traditional telecom services (e.g. GSM, SMS, WAP) over Parlay, as well as providing building blocks for making services that include location information, streaming possibilities and point of interest (POI) map information. On top of this different development environments are available to build usable services, such as Sun's J2EE and J2ME. Evaluation and testing of services is mostly done in the WLAN Wireless Trondheim is currently building.

For the two last areas of research in Table 2.1, they are covered by the networking lab and its *Street'n Roof* Lab. These *Street'n Roof* Lab's (14 street + 4 roof in total) are located on street and roof level in the vicinity of NTNU and Wireless Trondheim's headquarters. They are equipped with among other things high speed network equipment, host computers and the necessary mountings for testing different wireless technologies (e.g. WiMax and UMTS). These labs are further connected with a dual fibre ring, and each project usually operates within a VLAN.[3]

2.1.2 Adresseavisen

Adresseavisen is the biggest regional newspaper in the region of Trøndelag, where Wireless Trondheim is sited. In addition to operating a traditional paper newspaper they have lately expanded to radio and television (Radio-Adressa and TV-Adressa). The newspaper is also published online on the website `http://www.adressa.no`, as well as a WAP edition on `wap.adressa.no`.

2.2 Electronic paper

Electronic paper (e-paper) is a general term for technologies that aim to produce an experience close to that of print on paper. The different e-paper technologies try to achieve the same level of contrast as print on ordinary paper, without the use of a backlight. Other important aspects include low power usage, the ability to maintain the same text/image with low to none power usage, and to have somewhat the same physical attributes of paper such as flexibility and light weight [51].

There are several different approaches that tries to achieve the same qualities as mentioned earlier. In the following, descriptions of the most important and promising e-paper display technologies are provided. First the two electrophoretic[1] technologies E Ink and Gyricon is explained before the use of electrowetting as an e-paper technology is explored. Then the possibilities of using cellulose for the paper feeling is examined. The last technology is the use of bistable nematic liquid crystals as a means of realising electronic paper. This section will be ended with a few notes on colour e-paper and flexible e-paper.

[1] Electrophoresis are the motions of particles in fluid that is being subjected to an electric field.

2.2.1 E Ink technology

The E Ink Corporation employs a technology that makes electronic paper with the help of millions of tiny microcapsules. These capsules, with the approximate diameter of a human hair, contains electrically charged white and black particles. The capsules are placed between an electrode at the back and a transparent electrode at the front of the e-paper. The positively charged white particles will in the event of a negative electric field move to the top, while the black will stay at the bottom out of sight. Vice versa for a positive electric field. If the electrode at the back is in the form of a circuit that can apply/manipulate an electric field at each capsule, it is possible to arrange the black and white particles in such a way that text or images will appear. See Figure 2.1.

Philips have previously partnered with E Ink, and have provided the display drivers to several products using the E Ink e-paper. Such backplates are similar to those used by LCD displays and exist in both passive and active matrix configurations. The matrix covers all pixels and enables the alteration of each pixel. The difference in passive and active technologies is that passive can only change one row at the time while the active can do changes over the entire matrix at a time. [7, 50]

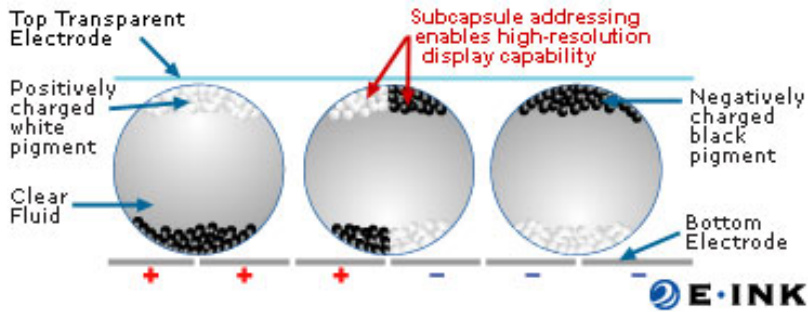


Figure 2.1: Cross section of a display made with E Ink microcapsule technology [7]

2.2.2 Gyricon technology

Gyricon is an e-paper technology researched by Xerox. The technology works in a way not very different from E Ink. A Gyricon e-paper sheet consists of millions of small beads that are individually contained in an oil filled cavity. The beads are died with two "opposite" colours such as black/white on each hemisphere. The beads are charged in such a way that they act as dipoles, and will rotate in the cavity when a voltage is applied to the sheet. Different voltage patterns will give different patterns on the Gyricon sheet such as text or images. Uses similar backplates as E-Ink to control each pixel.[5]

2.2.3 Electronic paper made with electrowetting

The principle of electrowetting is to modify the wetting property of a solid material with a voltage. This can be seen in Figure 2.2 where there is no voltage and the oil is at rest

in its neutral state on top of the insulator. In Figure 2.3 voltage is applied and oil is displaced. This is because the system wishes to be in a energetically favourable state, which it will achieve when water is in contact with the insulator. [16]

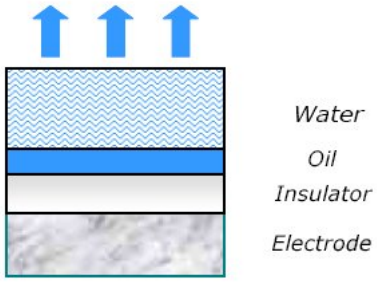


Figure 2.2: Principle of electrowetting, before voltage is applied [16]

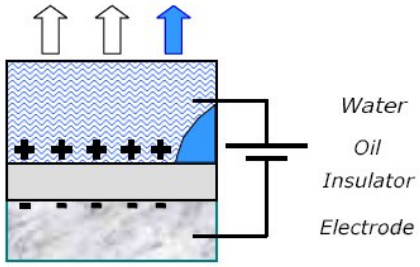


Figure 2.3: Principle of electrowetting, after voltage is applied [16]

To use this phenomenon as a display technology, dye is applied to the oil. With oil dyed black and white insulator, a display that shows grey scale can be achieved with different voltages. This is because different voltages will affect how much of the black oil that is displaced. A demonstration of such a 1-inch display is described in [15]. It is made by covering a glass substrate with an aluminium layer with the correct structure. This structure is what makes the pixels – each pixel consists of water, oil, insulator and electrode (see Figure 2.2), and the structure is there to keep the oil in place. On top of that a thin coating of a hydrophobic insulator is placed. The hydrophobic property is important to keep the oil evenly spread. Then the oil is placed in each pixel cavity and water is placed on top before the display is sealed off with a transparent material. See Figure 2.4.

Electrowetting can also be made to show colours, either with the standard RGB method or by a triple layer architecture. The RGB method requires that an RGB colour filter is placed on top of a standard grey scale/monochrome display. This results in 3 subpixels to create a colour. The alternative method is to place 3 monochrome displays

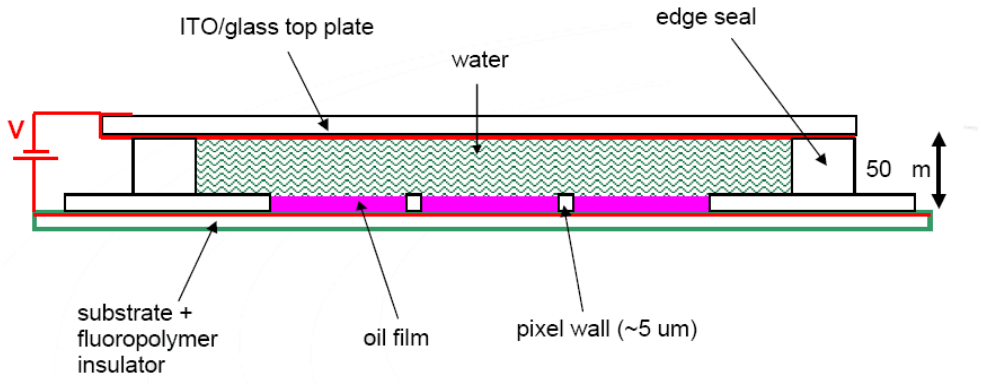


Figure 2.4: Profile view of a grey scale display using electrowetting. [16]

on top of each other, each representing one of the RGB colours. This requires no RGB filter as the oil in each display is dyed in the correct RGB colour, and there is also none subpixelation by this last method. [16]

The last property of e-paper made with electrowetting is that these displays is suitable for video. According to [16] the response is about 3 ms to switch on and about 9 ms to switch off. This results in about 12ms from black to white to black again, which is about the same as a regular LCD display capable of showing moving pictures. [53].

2.2.4 Bistable Nematics (BiNem)

This method of making electronic paper uses the same nematic liquid crystals as in ordinary LCD displays, and can be said to be an enhancing of current LCD techniques. The principles used to make e-paper displays with the BiNem technology comes from the e-paper company Nemoptic.

Ordinary liquid crystals used in LCD's only have one state and will return to it when the power is cut, while bistable liquid crystals will stay in its state indefinitely until voltage is applied. The BiNem technology utilises something called *surface anchoring breaking*, which enables the possibility of two states. A nematic liquid crystal mixture is placed between two layers with different anchoring properties, at the top it is a strong anchoring layer, while at the bottom it is a special BiNem layer which will break anchoring if a voltage is applied. The two states of the liquid crystal mixture is either uniform or twisted. There is a polariser present at the top layer which gives the uniform state appearance of black while the twisted gives the opposite white. A different voltage signal determines if the liquid crystal mixture should go to a uniform or twisted state. See Figure 2.5 for a more detailed description of the structure.[4]

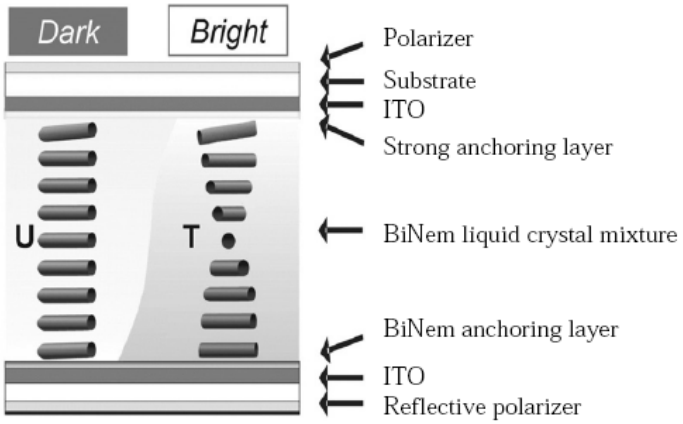


Figure 2.5: Structure of a BiNem cell.[4]

2.2.5 Electronic paper made from cellulose

The other mentioned e-paper technologies does not have the *feel* of ordinary paper, as they use plastics of different kinds to encapsulate the electronic ink. In [41] a method of making electronic paper displays is presented using microbial cellulose;

The paper is made by using a bacteria named *Acetobacter xylinum*, which creates pure cellulose. After a cultivation time of about 5 days the bacterias has created "paper" with a thickness similar to regular paper. The next step is to dope the paper with various conductors to achieve conductivity. The electronic ink used consists of electrochemical dyes, which will change optical properties if inflicted with electric charges. This enhanced paper can then be placed on top of two parallel electrode plates with in-plane bottom circuitry. The final product will have the same behaviour as the other e-paper technologies, but with a greater paper appearance and feel.

2.2.6 Colour e-paper

There are not many e-paper displays available with colours outside of the black, white and up to 16 shades of grey. Most of those who support colours are using RGB filters on top of the standard monochrome e-paper displays. This results in each pixel containing three subpixels, one for each RGB colour. This subpixelation further results in lower overall resolution. Just using RGB filters may results in poor brightness, but with an RGBWhite filter instead of an ordinary RGB filter brightness should increase. Nemoptic and LG Philips are utilising such filters (RGB or RGBW) in their colour displays. [37, 30]

In 2.2.3 another approach is described with three layers of e-paper displays, one RGB colour for each layer. Fujitsu seems to be the only ones utilising this three layer approach for colour. Each of the layers in Fujitsu's product uses a technology similar to Nemoptic's BiNem technology mentioned in 2.2.4. [32]

There are currently no commercially available e-paper displays that utilise colour.

2.2.7 Flexibility of e-paper displays

The described technologies in this section is primarily concerned with how displays with paper characteristics can be achieved. These displays will all need some kind of driving and controlling backplates to become a fully functional display. The most popular driving technologies for these kinds of devices are active- and passive-matrix. The active-matrix technology has the ability to control each pixel in the display, and is used in most LCD displays. Passive-matrix relies on the pixels it controls to be bistable, as it can only address the pixels in one row at the time. Most commercial e-paper products at the moment uses electrophoretic techniques such as E Ink (2.2.1), which is not a good combination with passive-matrix display driving as both are fairly slow. Thus active-matrix display drive techniques is in use on products such as the iRex iLiad (2.3.2). Thin-film transistors (TFT) on glass is the most used technology for making an active-matrix display driver, primarily used in LCD displays [55]. These rigid glass-based display drivers is used in most of the available e-paper devices, such as the aforementioned iRex iLiad. Hence, an e-paper display is bound to be inflexible until a more flexible backplate is available.

A flexible e-paper device is prone to be more rugged, lightweight and possible space-saving than a comparable inflexible e-paper device using a rigid glass-based display driver. A rollable e-paper device can potentially take very little volume when not in use. The deciding factor on the flexibility of the e-paper display is the display driver used, as most of the e-paper technologies presented over does not need to be assembled on anything rigid like glass.

Glass is used for most TFT production as it requires a high production temperature, higher than any suitable flexible material can endure. In [23] organic electronics is presented as a possible substitute for silicon as a semiconductor. The advantage of using organic materials as a semiconductor is the lower production temperature, which enables the use of different plastic substrates instead of glass.

Flexible displays can be classed in three categories according to their bendable, see Table 2.2. For e-newspaper use the flexible and rollable kind is of most interest. A flexible e-paper device would probably feel more "real", as can be seen in the illustration in Figure 2.6. Rollable displays on the other hand will be smaller and easier to transport as it can be rolled into a smaller volume when it is not in use. This means that an unrolled e-paper display will probably have a larger display area than the size of the device itself. A device the size of an ordinary mobile phone could probably house an A5 sized rollable e-paper display, and function as a mobile phone when not used as an e-paper reading device.

Plastic Logic [34] is a company that works with such flexible backplanes for e-paper, and their technology is described in [42]. Another company is Polymer Vision, which also works on the devices itself, and are presented in 2.3.

Figure 2.6: An illustration of how an e-newspaper could be on a flexible e-paper device [34]

Table 2.2: Three categories of non-rigid e-paper displays and their characteristics [23]

Category	Bending radius (cm)	Bendability	Advantages over rigid displays
Conformable		Bend once	Can be fit to an uneven or curved surface
Flexible	>2	Bend many times	Thin, bendable, lightweight, rugged
Rollable	<2	Bend many times	Ultra thin, lightweight, rugged, rollable into a small volume when not in use

2.3 Producers of e-paper devices and their products

There are several producers that utilises the different e-paper technologies to make e-paper readers. In this section a few of the most important producers will be presented, with emphasis on those that make equipment with WiFi-connectivity.

2.3.1 Sony

Sony was first on the e-paper device market with their LIBRIé (see Figure 2.7) e-paper reader. It uses E-Ink technology to make a 6 inch display with an 800x600 resolution and 4 step grey scale, while Philips provides the driving electronics and the backplane. The LIBRIé does not have WiFi connectivity, and relies solely on wired connectivity and Sony's Memorystick. It is equipped with a small *qwerty* keyboard for annotations and the like. [54]

Sony do also sell a reworked version of the LIBRIé on the US market called Sony Reader. It uses the same display, but is without a keyboard and with some software and

appearance changes.

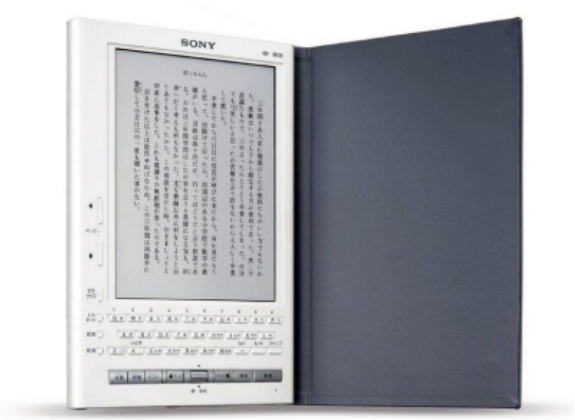


Figure 2.7: The Sony LIBRIé e-paper reader

2.3.2 iRex

iRex is the manufacturer of the e-paper reader iLiad (see Figure 2.8). It is a device with an 8.1 inch display using E-Ink technology and with a resolution of 768x1024 (160 DPI). It is a further development of the same ink that was used in the LIBRIé, and can provide up to 16 levels of grey. It supports the usual text formats such as TXT, HTML and PDF, and it also supports the following picture formats; BMP, JPG and PNG. It also sports a touch sensor input, which opens up the possibility of using a stylus for navigating and making notes and annotations.

Most importantly though, the iLiad is equipped with WiFi (802.11g) capabilities, and can be updated and supplied with content and such via a wireless network. iRex also offers a delivery service for their iLiads, the iRex Delivery Service (iDS). iDS can deliver content to an iLiad over Internet, more on this in 2.4.3.[25]

2.3.3 Jinke

Jinke is the manufacturer of the e-paper reader family Hanlin eReader. Jinke currently has three 6 inch models on the market and three 10 inch models due for early 2008. Of these it is the coming Hanlin eReader v9c which is most interesting, as it is the only one that is going to be WiFi equipped. It uses E-Ink electronic ink to provide a 10 inch display with 825x1200 resolution. It supports TXT, HTML, DOC, PDF and the usual image formats. It does not support stylus as its sibling v9t, which on the other hand lacks WiFi. [26]

Figure 2.8: The iRex iLiad e-paper reader

2.3.4 Fujitsu

Fujitsu does not have any e-paper devices for sale so far, but has recently released some interesting prototypes, the FLEPia being most interesting. It has a colour display of either A5 or A4 size, both with a resolution of 768 x 1024. The colour depth is either 8 or 4096. It is equipped with WiFi (802.11b/g), and being equipped with a Windows CE operating system, both e-mail and web browsing is possible. It is operated with a stylus. [33]

2.3.5 Polymer Vision

Polymer Vision, a spin off from Royal Philips Electronics, is a manufacturer of rollable e-paper displays, and a concept device can be seen in Figure 2.9. It is unknown whether that device is WiFi enabled, but such an e-paper display can easily be incorporated on a mobile phone or similar with WiFi and other wireless techniques available (eg. 3G, EDGE, GPRS). Polymer Vision and Telecom Italia announced in February 2007 that they were working on such a device, the "CELLULAR-BOOK", which should be EDGE/UMTS enabled and capable of showing newspapers, books, maps etc. on its 5 inch e-paper display [48]. The technology behind Polymer Visions rollable displays is explained in [23].

2.3.6 Amazon

It has for a long time been rumoured that the Internet bookstore Amazon has been developing an e-paper device, the Kindle. November 19th the Kindle was launched, and can be seen in Figure 2.10. It features a 6" E-Ink display, with a 600 x 800 pixel resolution at 167 ppi, and a 4-level grey scale. It has wireless capabilities, but not ordinary WiFi

Figure 2.9: Polymer Vision's Readius rollable display.[47]

though; the *Amazon Whispernet* uses the EVDO[2] 3G technology to provide updates to the Kindle. It is primarily made for reading digital books provided by Amazon, but it is possible to get subscriptions to a number of newspapers at the moment (that number is 11 at the time of this writing and includes The Wallstreet Journal(US), Le Monde(FR) and The New York Times(US) [1]). The newspapers will update to the device every morning, but how it is formatted and if this is a full or reduced edition is not known. [2, 31]

Figure 2.10: The Amazon Kindle. [2]

2.4 Content Delivery

The newspaper content needs to be delivered to the e-paper device in some way. This section will describe how this can be achieved as well as problems related to this. An important aspect of content delivery to a smaller device is the need of adaption and transcoding of the content to make it fit the devices better. There is also the need of making the file size smaller, as such devices usually have limited memory and processing capacity. The user of such a device might also be mobile, which could affect bandwidth and connectivity.

In this section different aspects and approaches for achieving content delivery that may in some way be used for e-newspapers is explored.

[2]EVDO is one of 3GPP2's contributions to the 3G technology and is comparable to UMTS. 3GPP2 is the North America & Japan version of 3GPP

2.4.1 General Content Delivery

Content delivery usually describes the delivery of all kinds of digital media, such as WWW (web sites), audio and video. The delivery can be done via different kinds of medium such as Internet, radio broadcasting and WLAN. Further, there are two kinds of content delivery; The publishers delivery of content ready to be distributed, and the delivery of content to each customer. [49]

Content Delivery Networks (CDN) is a special kind of network that have evolved to enable faster and more reliable delivery of media content.

Content Delivery Networks

A Content Delivery Network (CDN) is a network dedicated to delivering content faster to the users. There are several ways of achieving such a CDN. Most of these utilise servers and network hardware coupled with a good location, as will be explained below.

The usual approach is either with an overlay approach, a network approach or most commonly a combination of these two. The overlay approach is usually a transparent network of servers[3] placed at the edge of the core network. The servers contains mirrors and caches of the published content that is needed by the users. The network approach relies on the hardware in the network; routers and switches are supposed to recognise different requests and make forward and routing decisions based on predefined policies. An example of this can be redirection of a website request to a local cache. It is of course also possible to combine this and get the best from two worlds, which according to [45] seems to be the usual way of doing it.

The CDN servers are placed in the network where they will do most good (be closest to, and serve most users), and DNS redirection or URL rewriting is used to point a request to the closest CDN server of the user. With DNS redirection the DNS local to the user will know which CDN server is best for that user, either by cached results or a query to a (so called) request-routing infrastructure (RRI) which in return will query the closest CDN servers to find the best option. With URL rewriting the publishers/original server will redirect web site requests to different CDN servers by changing the URLs on the dynamic contents of said web site.

For the customers a CDN will thus give an increase in the availability of content and reduce delivery time (latency). A side effect of this is a reduction in total use of bandwidth to deliver the same piece of content to the same group of users. There will also be an increase in reliability as content will be mirrored on several servers, and can provide service even if the source server is down. [29, 45]

From [43] a list of requirements for a Content Delivery Network can be found;

- Content delivering to the edge servers if the service is to be of any use to the users;
- Ability to identify location of the requesting user and respond from the edge server that is closest to the user;

[3] These servers are alternatively called surrogate servers in [45]

- Monitoring of an edge server's performance to detect performance degradation and the ability to load-balance in the event a server becomes overloaded;
- Deploying enough servers by CDN provider in order to address users from different service provider networks;
- Keeping content fresh and synchronised with the serving server;
- Fault tolerance and automatic adjustment to prevent service interruption
- Tracking of real-time conditions of the network and avoiding of hot-spots, as well as capability of quick content rerouting in the case of congestion and outage;
- Reporting and billing – real-time logging and billing and detailed reporting on traffic patterns, user location, Internet condition and other reporting elements must be tracked.

Subscription-enhanced content delivery

Keeping edge caches as those described over with the correct content as well as not keeping any unnecessary stale content is a tricky business. In [6] it is proposed to utilise subscription information as one extra source for estimating future content requests. An ordinary way of deciding what to cache and what to replace is by using an replacement algorithm. One of these is the Greedy-Dual* as seen in Equation 2.1.

$$V(p) = L + \left(\frac{f(p) \cdot c(p)}{s(p)}\right)^{1/\beta} \tag{2.1}$$

Where $V(p)$ is the value of the page, L is the inflation value to capture the access recently, $f(p)$ is the number of accesses on the page, $c(p)$ is the cost to fetch a page from the publisher, $s(p)$ is the page size and β is the balance factor of popularity and temporal correlation. To keep the cache filled only with highly requested content it can either keep a number of the most requested/valued sites, or any content higher than a threshold page-value.

[6] offers some additional enhancements to this, and proposes two new ways of setting the value of a cached WWW page. The first is an adaption of the frequency information used in Equation 2.1. The new $f(p)$ can be seen in Equation 2.2.

$$f(p) = a_{NS} + (s - a_S) \tag{2.2}$$

$f(p)$ is the new frequency used, a_{NS} is the number of accesses of page p from non-subscribers, s is the number of subscriptions matching page p and a_s is the number of accesses of page p from subscribers. The intention behind doing it like this is to analyse the access information separately for those who access it based on subscription information and those who access it *randomly* when browsing the web.

The other proposed solution is to split the cache into dual caches, one for the accesses through regular browsing, and one for access through subscription notifications. Each of

the portions of the cache uses a replacement algorithm. The cache that serves the browsing uses a regular GD* as in Equation 2.1, while the other for subscription based uses a Subscription-GD* as can be seen in Equation 2.3, which was the basis for Equation 2.2.

$$f(p) = s - a \tag{2.3}$$

Where $f(p)$ is the frequency, s is the number of subscriptions matching page p and a is the number of accesses of page p.

2.4.2 Wireless Content Delivery

The greatest differences in content delivery to wired and wireless today are the differences in possible bandwidth, obtainable QoS and the chance for disconnects. The reasons for this is the air interface which gives lower possible and more varying available bandwidth and the increased chances for disconnects due to roaming/handover. The idea of using CDNs to get the content closer to the user to reduce access time does not work optimal for mobile users due to the air interface and the mobility of the user herself.

Further content delivery can either be performed in a pull or push manner, where either the client requests, pulls, what they want from the network or where the network delivers, pushes, what the client wants based on some form of subscription or predefined "wish-list".

Wireless Pull Systems

One way of reducing these perceived shortcomings is to actively cache content as described in [28, 38, 27]. In [56] this is taken another step further, and a complete architecture is presented, the ACME (Architecture for Content delivery in the Mobile Environment). These measures should result in a smoother experience of a given service for the users. In general this is done by pre-caching data that the client may think the user will need shortly, or it could alternatively be the act of loading data/content that the user may need into caches close to the user. In the following I will present a few examples using pre-caching and how such an implementation is considered implemented.

Pre-caching example no.1 In [27] the problems with the use of caching in relation to mobile users is presented. When a mobile user roams away from a network and its local cache there is need for a handover to the new networks local cache in addition to the usual handover. There may then also be need for the dynamic creation of a CDN (Content Delivery Network) infrastructure to support such roaming. And there is also the short comings of the mobile units with their limited CPU and memory resources.

It should be noted that [27] is mainly about multimedia streaming, but most of the principles should apply to the delivery of e-newspapers. One day the e-paper devices will also be able to stream multimedia. The architecture described relies on some important building blocks to provide a good service; Path Prediction Component, Pre-fetching Component (pre-caching) and Cache Handoff Functionality.

The *Path Prediction Component* provides the rest of the system with an estimate of the clients future movements. This is done by analysing old data to make a picture of the users habits and routines. This information can be used to predict where a user might go next at a given location / in a given situation.

Next the *Pre-fetching Component* is responsible with providing an arriving client(via handover) with a cache ready with the correct content. To pre-cache the correct information requires synchronising and cooperation with other caches in the vicinity.

The *Cache Handoff Functionality* is gives the client a seamless service experience. When the client roams to a new network, it shall change the clients current (serving) cache to the local cache of the network it roams to. A handover is initiated when the current cache detects a change of the clients Care-Of-Address[4] and will let the client know that there is another cache that is closer and more suitable. To be able to do such a handover successfully, there needs to be means of discovering available cache in a new network. This can be achieved through ordinary service discovery with either purpose built protocols or broadcasts in the new network.

Pre-caching example no.2 [38] describes the use of pre-caching for improving services that rely on the delivery of dynamic information. Such a service could be of a tourist information kind, where the tourist gets "personalised multimedia-rich presentations on attractions [...] (she) might encounter". Again the varying bandwidth while the tourist "roams" and the lower capacity of the mobile device puts restrictions on the media one can utilise in such a service. To get the best of it, two things needs to be done; "*Intelligently and dynamically precache information on the tourist's PDA*" and "*Minimise the amount of information sent to the tourist*".

It was thought that such a tourist service utilising PDA's (or such) is implemented as a client-server architecture with intelligent agents[5] as the most important building blocks. The most important agents are the *Spatial Agent*, tasked with knowing where the client is at all times using GPS, the *Cache Agent*, which is responsible for pre-caching the correct presentations based on the spatial information from the Spatial agent and its knowledge of Points of Interest in the vicinity of the client. To deliver a personalised service, there is also *Tourist*, *Presentation* and *Profile* agents that together makes a user-tailored presentation and experience.

To pre-cache the correct tourist presentation, the Cache Agent will use the data from the Spatial Agent and the clients knowledge of tourist attractions in the vicinity. Using these two data sets the Cache Agent can estimate what attraction the tourist might be heading for and start pre-caching the relevant presentation. If the tourist changes mind and heads for an attraction different from the estimate, the agent will drop what it has, and decide on a new presentation to pre-cache.

[4] [27] envisages using Mobile IPv6 which have support for binding updates and Care-Of-Addresses.

[5] Agents can be seen as software components that are goal oriented; they will only do something if it leads to the agent fulfilling its goal. An agent will have a set of desires that represents the goals of the agent, and then a set of intentions which is the goals an agent can fulfil at a given time when it is in a given state. [38]

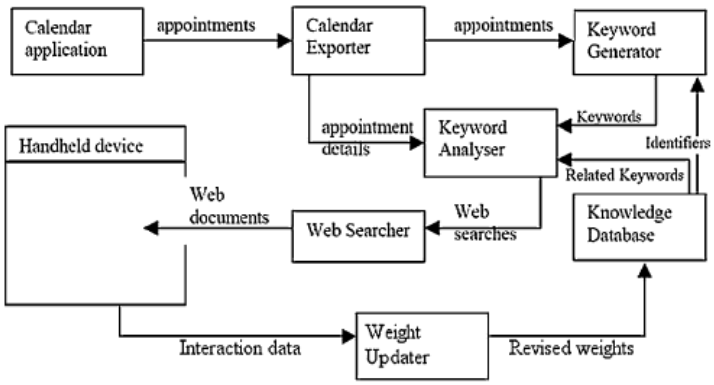


Figure 2.11: An overview model of the calendar precache system.[28]

Pre-caching example no.3 In [28] there is an investigation on whether calendars can be used by pre-caching agents. The idea is that a user's (electronic) calendar will provide information on the users (professional and private) interests and preferences. Then the pre-caching agent uses this information to cache content from the Internet that fulfils the criteria of the user. A scenario on this could be that the user places his PDA in its docking station, which got a fast Internet connection, and leaves it there while she eats breakfast. While she is eating the pre-cache agent will search through the calendar and decide on what to pre-cache to the unit. When the user leaves for work she just takes the PDA from the docking station and will have it filled with information and content related to her calendar and activities of that day.

In Figure 2.11 an overview is given of a system that works as described over. It is supposed to work as follows; First the calendar is exported and then keywords and key phrases are extracted. The system uses pre-made lists and rules to help with finding the keywords. To know what to pre-cache, queries for relevant information and content needs to be made. The keywords that are found are in many cases good, but can be enhanced with additional related keywords. These additional keywords can be found in the same calendar entry as the original keyword and/or in databases that contains keywords that are related to the category[6] of the original keyword. Further, the queries are to be submitted to one or more search engine such as `Google.com` and `Sesam.no`. Depending on the memory capacity of the mobile unit, a certain amount of search results are pre-cached. Links within the search results and links therein are also pre-cached to some level, again depending on the memory capacity. The information that are pre-cached can be seen as a tree structure with the search results on top and the links from them branching out.

[6]Keywords found in calendars can according to [28] often be put in categories such as meeting, reminder, travel, social etc.

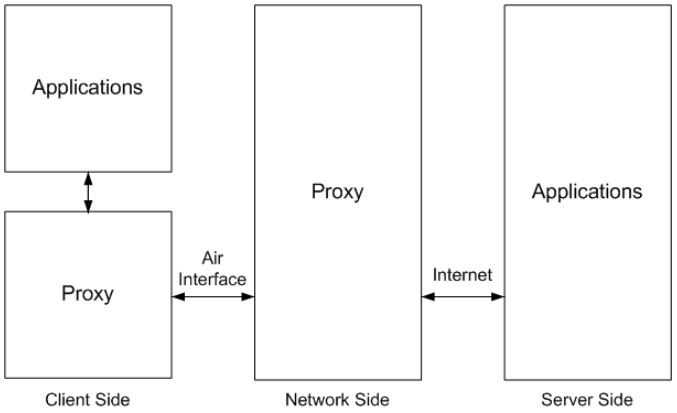


Figure 2.12: A split proxy architecture for mobile networks.

ACME [56] According to [56], the edge caching described earlier is not efficient in the mobile Internet. One technique described to overcome the bottleneck the air interface introduces is to use a "split proxy" architecture, this is shown in Figure 2.12. The goal is to provide a transparent communications interface for applications running on the client and server. A version of this split proxy approach is used in the described ACME (Architecture for Content delivery in the Mobile Environment).

The general ACME architecture is as follows; "*The ACME cache in each terminal implements the edge caching functionality using on-demand broadcast. If a user requests a Web object that cannot be served from the terminals ACME cache, a request is sent to the origin server. When the object returned by the origin server arrives in the access network, it is broadcast over the air interface to all terminals within that cell, which then store it in their respective ACME caches.*"

Compared to the other prefetch and pre-cache systems described earlier, ACME broadcasts to several clients in the vicinity and thus covers more than one client. This because it is assumed that an ordinary cache contains content that is requested or in demand among most clients in that area. Hence, a cache on a mobile unit would in most cases benefit from caching what other clients in the area is requesting.

The downside of a scheme such as this is that many clients might never need some of the content that is broadcasted to it, and will spend memory and processing on useless content. For smaller mobile clients this might not be acceptable, and a better approach may be needed. The proposed solution for this is the ACME Director, a server in the access network that computes and stores the "probability of a user requesting the same content after another user". This information is used to decide who should be included in the multicast for a content request.

Publish/Subscribe Push systems

A common way of implementing a push content delivery service is to use a Publish/Subscribe (pub/sub) system, which works most commonly as described in [35]:

1. An entity publishes information to a node at a pub/sub service.
2. The pub/sub service pushes a notification to all entities that are authorised to learn about the published information.
3. The device uses the information in the notification to start download the relevant content.

A variant of this would be to have all the published content pushed straight to the end user. A pub/sub system can further be classified into at least three different types, as shown in Table 2.3. The results of such publish/subscriber systems is a decoupling of

Table 2.3: Different types of publish/subscribe systems [22]

Type	Properties
Group/channel-based	A number of groups/channels is made by the publisher, and will publish content to one of these. The subscriber will then subscribe to one or more group/channels
Subject/topic-based	Each piece of content is tagged with an either arbitrary or agreed-upon string. The subscriber then makes a subscription based on these strings with either full or partly matches to her own keywords
Content-based	The subscriber can make fairly specific queries that the system will try to match before sending any content that a publisher has submitted to the system

the publisher and the subscriber, as both parties are anonymous and acts asynchronous — the subscriber and publisher acts completely independent. The publisher sends his content to the publish/subscribe middleware, which will forward this to the subscriber if she has showed interest in this kind of content. [22]

Examples of Pub/Sub systems Examples of readily available publish/subscribe systems includes among other the Minstrel push system [20], the REDS (REconfigurable Dispatching System) system [9, 10], Microsofts Web Solutions Platform [18, 19] and Jabber/XMPP's Publish-Subscribe system[7] [35]. As mentioned in [18] such systems could also be built on top of message-oriented middleware infrastructure such as IBM's MQSeries, JMS, and Oracle's Enterprise Service, but this is a more heavy and cumbersome solution.

Mobile Pub/Sub systems The mentioned Pub/Sub systems are not specifically made for mobile users, most work and research on such systems has mostly been aimed for static participants and fixed networks, as noted in [39, 22]. Thus there is a need for a Pub/Sub infrastructure that addresses mobility issues specially. The fact that publisher

[7] A standards draft based on IETFs XMPP.

and subscriber is decoupled and interacts asynchronously is a good trait for mobile environments, but the possible mobility of the subscribers make the routing more complex as well as the inherent risk of disconnects. Further, one way to describe users, is that they either are stationary, nomadic or mobile, which will each require a different set of services and mechanisms to accomplish successful publish/subscription. In Table 2.4 the required services necessary for realising Pub/Sub service for stationary, nomadic and mobile users is listed as described in [39]. *Subscription management* shall together with

Table 2.4: Required services for stationary, nomadic and mobile users [39]

	Stationary	Nomadic	Mobile
subscription management	yes	yes	yes
content management	yes	yes	yes
user profiles	yes	yes	yes
queueing strategy	yes	yes	yes
location management	no	yes	yes
content adaption	no	no	yes
content presentation	no	no	yes

Content management manage the subscriptions of users, and know which users subscribe to which content. This also includes to define channels, which are groups of content that a user might want to subscribe to. *User profiles* is the service users use to personalise their subscription and set their preferred preferences. *Queueing strategy* is responsible for queueing content a user has subscribed to but cannot receive, and also responsible for delivering it when it is possible. *Location management* is responsible of mapping a subscriber to the current device or host that subscriber uses. *Content adaption* and *Content presentation* is responsible of adapting content to the device the user might be using at that moment.

An example of mobile Pub/Sub system In [39] an architecture for mobile push systems is also proposed, this can be seen in Figure 2.13. In the application layer the *Content management and presentation* is responsible for the creation and management of content that will be published on different channels and to different devices. The *Application-layer handoff* is responsible for controlling the transfer of content between different content dispatchers. Content dispatchers is stationary entities that sends content to subscribing users. In the service layer the *User profile management* stores and manages user profiles and their preferences. The *Pub/Sub management* manages subscriptions and advertisements, and acts thus as the go-between of the application layer services and the publish/subscribe middleware. *Adaption management* is responsible for adapting content to the clients and network if need be. The *location management* is responsible for keeping track of and locating a subscribers currently active device or terminal. Finally the *Publish/Subscribe middleware* in the communication layer is the basis for the interaction between subscribers and content publishers, as this is where content is in the end published to, and subscribed from. To scale better it should be distributed

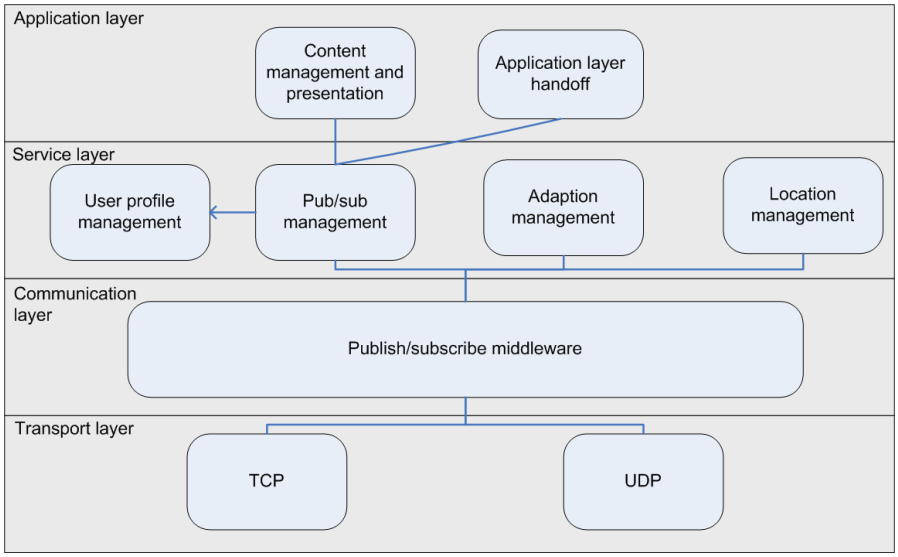


Figure 2.13: A mobile push architecture [39]

and its routing algorithms should support mobility.

Localisation of middleware in Pub/Sub systems In [22] the middleware is denoted as an Event Brokering System (EBS), with Event sources and Event displayers as publishers and subscribers respectively. The event notion is used as in some contexts a Pub/Sub system revolves around events—a subscriber might want to know about one specific type of event as soon as possible. Further the events can be seen as the aforementioned notifications for content being ready for download. The EBS can either be centralised, distributed or replicated and each of these different variations can be adapted for mobile environments.

Centralised A *central EBS* consists of only one central Event broker that manages all subscriptions and publications, and is thus the least favourable alternative considering reliability and performance. The Event broker should not be placed in the same devices as the sources or displays, because of many reasons; brokering will most likely require substantial computing power — the source or receiver is most likely just a simple device and would not have the available resources. The source and receiver can as well be disconnected from the network for some time and as a result cripple the system for other users. The Event broker should thus be placed on a computer in the fixed network.

As mobile subscribers can suffer from disconnects of various reasons, events that cannot be delivered should be cached and queued for later delivery. If the disconnects are of longer time periods, or the events are many, a condensed or summary should be pushed to the subscriber to save bandwidth if such aggregations is acceptable for the subscriber. Another way of saving bandwidth is to discard events at the source if the

source has some information about what kind of events it is subscriptions for. This technique is called quenching. [22]

Distributed In a *distributed EBS* there is several Event brokers taking care of the publishing from sources and subscriptions from displayers. Such distributions can be done in two ways, either with broadcasting or multicast. In a broadcast scenario the event that an Event broker receives is broadcasted to all Event brokers in the system, which will then forward it to the subscribers it is responsible for. In the multicast scenario the Event broker will only forward an event if any of the Event brokers on its output links might need it for their subscribers. This might result in the need for all Event brokers to know about all subscribers as opposed to the broadcast scenario where each Event broker only knows about its share of subscribers.

To adapt to mobile environments, the system will need to take into consideration the fact that devices can disconnect and then reconnect to different Event brokers. When this happens the new Event broker needs to be updated with the subscription for this user, as well as obtain all events that has been queued while the device was disconnected. This will in most cases be a sort of handover from the Event broker the device was previously attached to. A possible disadvantage is that at one time several Event brokers will monitor and maintain the same subscription for the device, and possible try to deliver the same events multiple times to the device. Time stamps and *lastConnected* times from the device is one remedy to this.

Replicated With a *replicated EBS* a users subscription is monitored and maintained in several replicated Event brokers. The best property of such an implementation is the increased availability and reliability, and thus the reduced chance of missed events. But there are also some problems, as events can come in a wrong order, duplicated events and general incompleteness if more or less events is delivered than was generated at the sources.

2.4.3 Existing Commercial Solutions

iRex Delivery Service

The iRex Delivery Service (iDS) is a content delivery system made by iRex Technologies for use with their iLiad. This system enables a publisher or other content distributors to send content to one or more subscribers. The user of an iLiad needs to register her device and activate an iDS-inbox for the iLiad (identified by a MAC address). All content is sent to this inbox, which the user can download to her iLiad by pressing a button on the device. This will make the iLiad attempt to connect to the Internet, and download the contents of the inbox to the iLiad. It is unknown at what level iDS can handle disconnects and other intermittent problems mobility introduces.

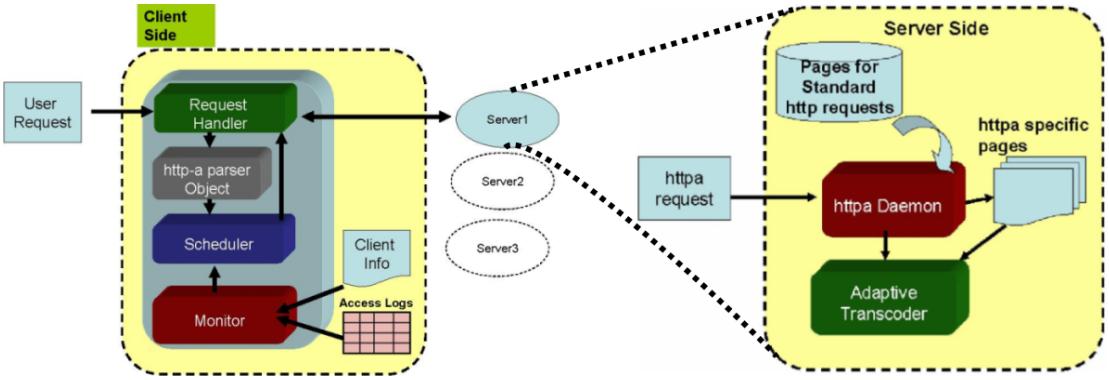


Figure 2.14: An adaptive Httpa Client and Server from [40]

2.5 Content Adaption and Transcoding

Content that is available on the WWW is often laden with lots of elements that are not desirable nor feasible on smaller mobile units. Some of the reasons for this is because the screen is smaller, the processing power is less and the link to the Internet is less than stellar. Hence, the WWW content needs to be adapted to the smaller unit both in layout and "physical" size. It is not much to do with the text, but images, sound, video and layout can and will most likely need adaption. This adaption is usually of a transcoding nature; images are reduced in size and (often) compressed to a smaller bit size, features that are not supported are removed and the content is restructured to fit better to the given unit.

2.5.1 A transcoding system

In [40] one example of a system that performs adaptive transcoding of multimedia WWW content is given. In Figure 2.14 the structure of an adaptive transcoding system is shown. It is intended to work as follows; All WWW requests are handled by the *Request Handler* which gets the source of a site from the *Httpa Daemon* and sends it off to the *Httpa Parser*. The Httpa Parser examines the source and finds the possible compression of the different elements in the source, and passes it to the *Scheduler*. The *Monitor* provides the Scheduler with information about the mobile unit and available bandwidth. The Scheduler uses the provided information and tells the Httpa Parser what compression and adaption it has decided on. This information is then passed to the server and the *Transcoder* to compress and adapt the source material.

2.5.2 Placement of transcoding

According to [36] there are three ways of deploying a transcoding system, at the client, at the source or somewhere in between as a proxy server. The system described in the previous paragraph can be seen as an implementation of a transcoding proxy. As mobile

units are less powerful than their stationary desktop siblings, a client approach is not desirable, as well as such an approach would also void any bandwidth savings that could be made by transcoding other places. Transcoding at the source has the disadvantage that there would need to be transcoding functionality in every content server. But placing the transcoding in between as a proxy could prove as a bottleneck, and the proxy would as well need more complex functionality to cope with all kinds of content.

2.5.3 Types of transcoding and adaption

In [8] there are mentioned five types of adaption; Format adaption, Characteristics adaption, Appearance adaption, Size adaption and Encapsulation adaption;

- Format adaption is the act of changing the format of a piece of content so that it can be played on units that lack support of the original format.
- Characteristics adaption is used to change the characteristics of content, such as number of colours and image resolution, for devices that do not contain support for the original characteristics.
- Appearance adaption will help a device show content by changing the appearance of the original content, this can be actions such as dividing a page of a document into several pages for devices with smaller screens.
- Size adaption will change the size of content if it is either to large for the receiving device or the network it shall traverse, this can be done by changing the format or characteristics.
- Last encapsulation adaption will convert packets from one transport or application protocol to another more fitting.

Transcoding is a procedure that several of the adaption types uses to get content to fit the receiving device. On audiovisual content this is done by decoding the media object to an uncompressed state, and then adapt the object (e.g resizing, changing colour depth, changing resolution for pictures) before encoding it to the decided format[8].

For non audiovisual content, stylesheeets can be used as a technique of adapting source content to other viewing formats. A stylesheet describes how a program shall display a given document, most usually XML and (X)HTML-files[9], by defining the fonts, colours and layout to be used. Such a tool as stylesheets makes it easier to divide content from presentation, and can be exploited by making different stylesheets for different purposes; simple and lightweight for the mobile units while the version for desktop computers with high speed connections gets all the bells and whistles as the designer intended. [36, 8]

[8] The encoding is also a step in the process of adapting a media object

[9] CSS for HTML and XHTML documents, while XSLT is for making (among others) an HTML/CSS representation of XML documents

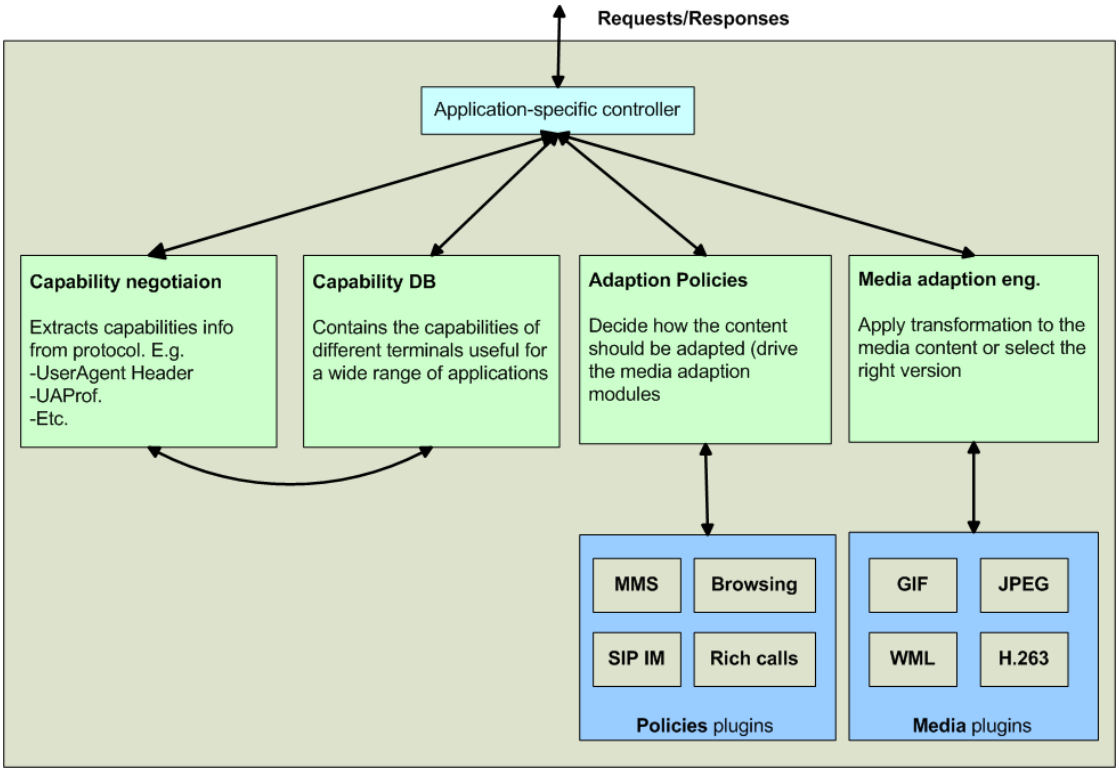


Figure 2.15: Architecture of a media transcoder system. [8]

In [8] an architecture for a transcoder system is depicted, this can be found in Figure 2.15. It works as follows; Capability Negotiation is responsible of obtaining the capability of a user through the protocols used. Capability Database contains capability information that can help if the information from the protocols is incomplete. Adaption policies decides which type of adaption and how it should be performed. The media adaption engine is responsible of applying and performing the adaption and transcoding.

3 Discussion

In this section the topics from the theory will be discussed and related to the problems at hand. Many of the topics from the theory has (relatively) direct applications to the problem, while others might be more suited for future or extended applications related to newspaper on e-paper.

One of the goals of this project is to find a possible way of implementing a newspaper service on an electronic paper (e-paper) with distribution over the wireless network of Wireless Trondheim. In the theory part of this report a lot of topics are presented that can be related to e-newspaper over WiFi in some way;

- The e-paper technology itself to get an impression of what it can do and why one would want to use it for this purpose.
- The e-paper units and readers to get to know what their properties are and what they are capable of.
- Different means of content delivery to get information about how this could be solved for different networks, specially wireless and WiFi.
- How to adapt content to different (mobile) units how transcoding works and how it can be a useful tool for this purpose.

3.1 Scenario

A number of different scenarios will in this section be described to help and emphasise points during the discussion, as well as making a point out of what should be possible.

The scenarios presented are for some likely use(r) cases of the present[1] as well as one for the future.

3.1.1 Scenario 1: Average user of the present

The average user of the present would be a person who would use the device to read an e-newspaper like most read a regular printed newspaper. This could be flipping through the newspaper in its entirety from back to back, or only selected sections like the sports section. This reading would probably only happen once or twice a day.

> Sophie works as a secretary in an engineering company. She likes to read the newspaper while she eats breakfast and on the bus if there is anything interesting she did not manage to read. She does not care much about sports nor economics, so her e-paper device will not be updated with this. She usually flips through the entirety of what she subscribes to – local, national and international news as well as the culture section. She also has a subscription for a daily update of celebrity gossip which she reads in the afternoon after dinner. The morning news is set to be ready at breakfast time, which is by default 07.00 in her subscription, while the gossip section should be updated for post-dinner; ĩ8.00.

3.1.2 Scenario 2: Power user of the present

The power user would be a person that uses all the available features of an e-newspaper, and wishes for a few that does not exist yet. This could be subscribing to extra stock quotes and profiles, immediate notifications of news in special fields and analyses of special happenings. The e-paper device and e-newspaper service would be in use several times a day.

> Jack is an engineer at the same engineering company as Sophie. He also likes to read the newspaper at breakfast, but he has other interests: He only reads the most important news, but is an avid sports fan and is interested in all the sport news he can get his hands on – national and international, the more obscure the better. As part of his obsession he has set his subscription to update his e-newspaper whenever anything of interest is published throughout his wake hours.
>
> As part of being an engineer he needs to be updated in his area of work, and thus has an subscription on content related to his work. He also has loaded all the technical papers and user manuals he needs for quick reference onto his e-paper device.

[1] Present is what is here thought possible to do with todays technology.

3.1.3 Scenario 3: User of the future

The e-paper device of the future will probably have colour and response times as LCD displays. The portability as well as computing power and battery capacity is much greater than today, as dictated by Moores Law. This opens up for more functionality in an e-newspaper similar to the WWW sites of the newspapers; movies, picture-sets and various interactivity (e.g. tests and games).

> Sophie and Jack is (happily) married and works at the same engineering company. The future is wonderful, and they are both equipped with state of the art e-paper devices. Sophie with a *chic* foldable e-paper reader with a back in white leather[2]. Her interests have not changed much: General news and culture in the morning, and celebrity gossip in the evening. But now the articles are accompanied with colour pictures and videos. She is as well carrying their first child and are subscribing on all the *Mother-to-be* content she can, as well as having an ad agent scour the Internet for good deals on baby gear.
>
> Jack is as well benefiting from the advent of colour and moving pictures. Football has become his greatest addiction and receiving numerous video clips with updates is one his greater pastimes. Another of his favourites is watching live football games on his rollable 20" e-paper device while commuting.

3.2 E-paper devices

The author only have had *on hands* experience with one of the e-paper readers mentioned in 2.3, the iRex iLiad. As the e-paper hype and iRex itself promises, the iLiad is very easy to read on and could easily be compared with ordinary paper in the field of readability. It is fairly easy to navigate in, although in the beginning the page flipping buttons seems counter intuitive, as it needs to be flipped to the left for one page forward – but when one realises that it is the same "motion" as when flipping a page in a usual book or newspaper it is no more counter intuitive.

Size

Most of the e-paper devices available now and in the near future gives display sizes from a little less than A5 to A4[3]. It is usually true that a bigger screen area gives better for usability for the user. But one do also have to consider that as devices grow they do also get less mobility appeal. Until foldable or rollable displays are available devices with A4 displays might get to big. An average 14" widescreen laptop would be of much the same size as an A4 e-paper device bar ca. a centimetre in depth.

[2] As Plastic Logics prototype noted in this Times article: `http://business.timesonline.co.uk/tol/business/industry_sectors/media/article1620550.ece`

[3] The A-series of paper size comes from ISO 216, A4 = 210 x 297 mm and A5 = 148 x 210 mm.

Form factor

Most of the available devices keeps to mostly the same shape and form factor, which is strongly dependant on the shape of the e-paper display. As mentioned above they keep to the shapes of ordinary paper, be it the international AX standard (A4, A5 etc.) or the American letter size which is fairly similar. Most people are in some way bound to tradition and habit, and would most likely want to keep it that way. Thus changing to much away from the old familiar in the transition to digital is likely not the greatest idea.

Interface

Keeping it simple must be of great importance. While lots of digital gadgets and devices mostly cater to young digital adept people, the e-paper might have a good chance in the older segment of the population. Things that are hard to master would hinder adoption of the e-paper devices despite how good it gets when you finally master it. Few buttons and intuitive navigation should be the key, and the flipbar of the iLiad is a good example; When the flip direction is mastered it is easy going flipping and reading through content.

3.3 Previous and current attempts at e-newspapers

First, a little note on **method**. The information presented here is obtained mostly through the web; the newspapers own websites as well as third party sites. The author has managed to get in contact with one of the newspapers.

Valuable information can be gathered from studying other attempts of e-newspapers. There are a few other newspapers that have tried e-paper versions of their newspaper. Sundsvalls Tidning is the only one in Scandinavia that has tried this. In the rest of Europe the French Les Echos/AFP and the Flemish De Tijd newspapers has tried and is in the progress of trying e-paper versions of their newspaper.[52]

3.3.1 E-newspaper projects

Ifra, an international media publishing association with 3000 members in 80 countries, started in 2005 a 3 year initiative, the eNews 2008[4]. It was started in response to the now emerging mobile e-readers of all kinds, and is intended to help the Ifra members create and identify possible business opportunities, identify possible new products and services as well as generally prepare for what the future might bring of mobile e-reader devices. Some of the members of eNews 2008 include Aftonbladet (SE), New York Times Company (US), Telegraph Group (UK) and Yomiuri Shimbun (JP).[13, 14]

In [24] a survey and several workshops was committed with a number of Swedish newspaper publishers. These surveys and workshops resulted in among other, a summary of important characteristics for paper and online newspapers was made, as can be seen in Table 3.1. Further, the preferable properties of an e-paper version was concluded as well as suggestions on how one could be designed.

[4] `http://www.ifra.com/website/website.nsf/html/CONT_ENEWS?OpenDocument&ENW&`

Table 3.1: Preferred characteristics from the printed and online newspaper editions [24]

Printed newspaper edition	**Online newspaper edition**
The clear overview of the content, including a beginning and an end, the ease of use, typography and design	Continuous updates
The familiarity in design from the printed paper	The possibility of searching
That mainly the same editorial content should meet all readers, i.e. creating common topics to discuss in social settings	Easy and intuitive navigation
News valuation, e.g. positioning of lead story	Hyperlinks
Mobility, i.e. to be able to read the newspaper everywhere	Interactivity with the readers
	Possibility to add sound and video

The conclusion was that the design from the printed edition with the functionality of the WWW edition would probably make the best e-paper edition; Most would prefer a combination of sequential presentation ("flippable") and hypertext, a sequential presentation would easier give an overview and a feel for the beginning and end, while the hypertext was more suited for quick and goal-oriented reading. Additional comments from the participants included an index, page numbers and indication of current location as well as the need for news valuation, which involves the placement of news, and possibility of being guided through the e-newspaper. Closing remark from one of the participants was;

> *The e-newspaper has to be as easy to use as the printed newspaper. The goal has to be that the e-newspaper becomes a people' product to the same extent as the printed newspaper and television.*

3.3.2 Sundsvalls Tidning on e-paper

The Swedish newspaper Sundsvalls Tidning had a short term project in fall 2006 [44]. They used a version of iRex iLiad. Svenåke Boström, a media developer in Sundsvalls Tidning provided valuable information about the work on e-paper and e-newspaper in Sundsvalls Tidning, see Appendix A.1 for more information.

Sundsvalls Tidning is in company with other newspaper companies (*Tidningsutgivarna*[5]) planning to do an extended test with a new e-paper device in 2008. In 2009 or 2010 Sweden might get their first public e-newspaper service.

[5] `http://www.tu.se/`

Structure and layout

A web application is used to transform the output from the editorial system[6] into a format suited for the e-paper device. The application would autogenerate pages from a list of articles, which was then proofread to make sure everything was lined up correct and make corrections if not.

Distribution

The e-newspaper was updated twice a day, at 19.00 and at 00.00. The subscriber could update the newspaper with just a press of a button, and it would download either via wireless or cable. It is not known, but the use of a dedicated button implies that iRex' distribution system iDS is used, as the iLiad is equipped with a button for this purpose.

3.3.3 Les Echos/AFP on e-paper

Les Echos is the only newspaper company that have taken e-paper and e-newspapers past the project state, and into subscription[7]. They offer two different e-paper devices, the iRex iLiad (see 2.3.2) and the STAReBOOK STK-101 by the Chinese company Staretek[8]. It is also a rumour that there will be support for the Amazon Kindle[9]. The subscription contains two editions, the Les Echos and the AFP, which is updated every hour from 07.00 to 21.00, and can be updated via a WiFi connection. On the subscription, one of the mentioned e-paper readers is included as well as some e-books.[12]

Structure and layout

The structure and layout of the e-newspaper is explained in an information video made by Les Echos, as can be seen at [11]. From this video it is observed that Les Echos has chosen a layout that consists of a start page with the most recent news in all categories, and with links to an overview page which contains links to all available categories (or columns as they call it themselves). Each of these categories contain a number of the most recent articles of that specific category. As Les Echos and AFP is a financial newspaper the e-version also contains the latest developments on the stock market(s), as well as profiles on a number of companies and the development of their stocks.

It seems possible to flip through the newspaper from end to end – it was at least possible to flip through all articles in the category of the article that was chosen. The structure of each article seemed to be like this; Headliner, introduction, illustration and last text. The text would span several pages if long enough. The e-paper device used in the video was the iRex iLiad, and navigation was mostly done with the stylus and the flip-bar.

[6] Sundsvalls Tidning uses Newspilot by Infomaker - `http://www.infomaker.se/artikel.php?avdelning_1=102&avdelning_2=106&avdelning_3=156`

[7] `http://www.lesechos.fr/epaper/inscription.htm`

[8] `http://ebook.stareread.com/en/`

[9] `http://www.engadget.com/tag/kindle`

Distribution

From the information video mentioned above and their websites, it is not entirely clear how the e-newspaper is distributed. They offer at least support for two e-paper devices. These devices has several possible ways of updating. The iLiad has input support for WiFi (802.11g), USB, MMC cards and CompactFlash cards. The STAReBOOK on the other hand has support for USB, MMC cards and SD cards. The different flash cards[10] are more convenient for increasing the memory size of the devices, than as a mean of updating the e-newspaper. The e-paper site of Les Echos describes the STAReBOOK as updating via a USB connection and the owners PC, while the iLiad could update via WiFi (and probably USB as it supports that as well).

The question then arises, how Plug&Play friendly is the USB solution they are using? Does the owner have to first download the latest version of the e-paper, plug the e-paper device to the PC and then navigate to the correct place and copy it? Or does it automatically download the latest edition when it connects to a PC with USB and discovers an Internet connection?

The WiFi solution for the iRex iLiad seems to use an iRex solution, and should be able update the e-newspaper whenever a WiFi network with a connection to the Internet is available. This solution is called iDS (iRex Delivery Service), and is a distribution network capable of delivering content to any iRex iLiad as long as it has a connection to the Internet (It has Ethernet connectivity as well as the all important WiFi). See 2.4.3 for more on iDS.

3.3.4 De Tijd on e-paper

In early 2006 the Flemish (Belgian) newspaper De Tijd had a limited test of newspaper distribution to 200 selected subscribers. They used a pre release of the iRex iLiad for this purpose. In [46] a presentation is given of the first results that was obtained from this project. In Table 3.2 the facts and results that was obtained are listed.

Conclusions made from e-paper project

From the user tests some conclusions were made, and could be found in [46];

- The screen size and quality was a positive experience.
- The reading function was the most important feature.
- Integration in one device for PDA users
- Multiple newspapers on one device
- Store and read other documents in PDF
- Practical mobility – can bring several books, newspapers etc. almost everywhere due to comparatively small and lightweight device

[10] Solid state memory, see `http://en.wikipedia.org/wiki/Flash_memory` for more info.

Table 3.2: Facts and results from the De Tijd e-newspaper project [46]

Test Panel	200 subscribers Highly educated Computer & Internet experienced Print as well as online version readers
Reading behaviour	Functional: less than 1 hour, only articles of interest 66,2 % reads more than one newspaper Mostly @ home, in the morning
Usage	Technical problems Not all users were able to use their device Not a "normal" usage pattern (slow & unstable devices)
Device	Positive reactions on format, weight and design Screen quality!! Battery autonomy not optimal (avg. 4 hours)
Operation	Buttons: multiple functions in 1 button = unclear Flipbar positive (links to books) Need for feedback: status indication + failure messages Need for a more intuitive usage Slowness = bottleneck! (Booting & Navigation)
Content	General layout : neutral Zoom function on first page needed Direct link to article needed Navigation and structure need improvement (Reading pattern newspapers)
Issues for improvement	Way of downloading content: speed + need for wireless connection + automatic downloads Speed Usability of buttons Layout & navigation structure Zoom function needed

- Old habits die hard – some people still prefer their paper newspaper, because they are used to it.

Structure and layout

According to an interview in [21], not much was done improving it for the e-paper device, it was essentially the same as the original paper version, and was not updated throughout the day. In the "Issues for improvement" row in Table 3.2, it is implied that it still needed some work before being good enough. In this interview it was said by the project manager that "*(they) have decided not to go further*".

Distribution

The specifics are unknown, but it was once a day uploaded to the iLiad in some way, as mentioned in [21]. The use of the iLiad could indicate the use of iDS.

3.3.5 Application of experience gained in these projects

How these e-newspaper projects have been performed as well as the results gained can be of interest for this project. None of these projects seems to concern themselves much of how the e-newspaper is going to be distributed. It is mostly either *download it yourself and load it onto the e-paper device* or use iRex' proprietary iDS network which requires an open or WEP secured network to use[11]. Open or WEP protected works well enough in the home, but elsewhere it might be different. How they have built and structured the e-newspaper is though of interest as well as the feedback that was gained from the (test)subscribers.

Structure and layout

Layout and structure is important for e-newspapers; in the conclusions from the De Tijd project (Figure 3.2) the layout and navigation structure was mentioned as issues that could be improved. And as mentioned earlier, De Tijd apparently did not do much alterations to the newspaper before putting it on e-paper. This shows that it is important to get a layout and structure that is intuitive and can easily be navigated by the average newspaper subscriber – old or young, technically adept or not.

To obtain a user friendliness as described above it is important to cater for the largest group of subscribers and readers as possible. Those who reads and are most used to the old fashioned paper edition will probably want an experience that is fairly similar. For they who frequent the WWW edition of the newspaper an experience similar to that of the web site will probably be preferable. These reasons could legitimate the need for two or more different versions of the e-newspaper, as those described in **??**. But newspaper reading is not exactly rocket science, even on an e-paper device, and a version that contains the best from the two worlds would probably do well enough for the majority of the subscribers and readers.

Les Echos seems to have made a version that incorporates functionality from both domains, navigation roughly like a website, but also with the possibility of flipping through articles in a consecutive manner. Similar operations would probably be advisable to have available for the Adressa e-newspaper edition.

Distribution

Les Echos and Sundsvalls Tidning had the possibility of wireless distribution of their e-newspaper, while De Tijd on the other hand did not seem to have that possibility. As

[11] That is, it is the iLiad which cannot use any network with more than WEP encryption at the time of this project.

it is mentioned in Table 3.2 wireless connection as well as automatic download of the e-newspaper is mentioned as issues for improvement. iRex' iLiad and iDS does not seem to support such functionality at the moment, in the informational Les Echos video [11] updating seems to be a user initiated operation.

An automatic service could have several benefits; less trouble for the user to get the latest edition, as well as the possibilities of tailoring an edition with only articles the user finds interesting. An example could be that the user wants to get the full and complete edition for breakfast, only business articles until dinner time when culture content take over.

3.4 Possible approaches for the e-paper edition of a regular newspaper

For the sake of discussion Adressa will be used as an example in this section. At the moment Adressa is available in at least three different editions; the ordinary paper edition, the WWW edition and the WAP edition. The ordinary paper edition is available as a PDF file that mirrors the paper edition. The question is thus – should the e-paper version of the newspaper be built from the ground as a dedicated version, or will it suffice to adapt one of the other versions for the e-paper.

3.4.1 Available formats

For most people it can be assumed that the paper version of a newspaper is what they consider the real version of the newspaper, although the WWW version is favoured by many as it contains the most up to date content and is usually freely available. In a digital context the PDF version can at least be considered the original version of the newspaper content wise. For the sake of the discussion i will until further consider the iRex iLiad as the chosen e-paper reader for this project. The display/screen of the iLiad measures 768 x 1024 pixels on a surface that approximates an A5 sheet of paper. The paper version of the newspaper is in a tabloid format which roughly measures to A3 paper size which is four times larger than an A5 paper. To read the PDF version of the newspaper loaded into the iLiad would therefore need a considerable amount of scrolling, which the iLiad does not do particularly fast. An e-paper newspaper service would probably not be very successful with the original PDF version providing the newspaper. (As well, the iLiad does not load large PDF files such as the original particularly fast.)

The WWW version of the newspaper is also too large for the iLiad. The fact that it is too high is not the greatest problem as it could have been divided into pages and therefore fit well to the pageflipping of a device like the iLiad. The problem is that the web page is slightly to wide for the iLiad, and would either be chopped off or require scrolling. Half of the right menu and the advertisement column is outside of the picture. This is illustrated in Figure 3.1. This could be remedied by resizing the width of each element so that the entirety fitted, but this would most likely look cramped and not good. Several but not all of the items in the menus links to (interactive) content that

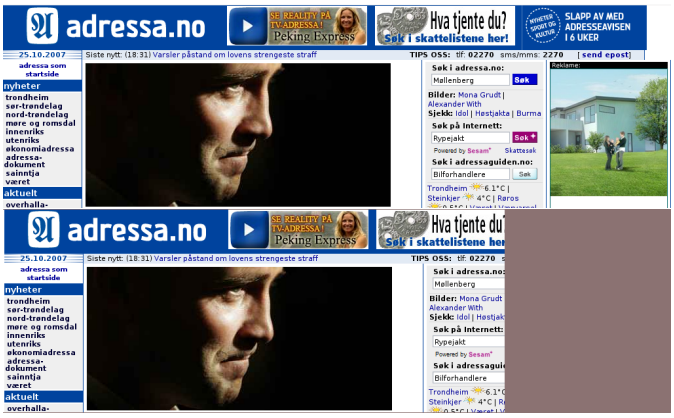


Figure 3.1: The Adressa WWW version (`www.adressa.no`) in full width and the width the iLiad supports (as illustrated with the grey area on the lower picture.)

is not available on the e-paper edition. Hence there is some work and "hacking" that is required to make an easy to read and navigate e-newspaper.

The WAP version of the newspaper does not suffer from any width or height problems as it is meant for smaller units such as mobile phones and the like. The reason for this is that it is mostly text, and thus less illustrated and might also feature less text on the articles than its full sized siblings, it might also be left out of some exclusive or *special made* content.

3.4.2 E-newspaper formatting

There are several possible approaches that can be followed to get a user friendly e-paper version on to the iLiad or any other of the currently available e-paper devices with the editorial content available today. I will in the following examine the possibilities that lies in custom making from the ground and using the available editions (PDF, WWW and WAP). Most people and companies wanting to publish to an e-paper device would probably choose to custom make an edition and feed it with content from their editorial system. But there might be some possibilities in taking another starting approach, either as the content source or the layout source, or both. The possibilities will be explored here.

Custom built

The obvious approach would be to build an edition that is entirely adapted to the e-paper format, and which is not an adaption of other editions, as will be explored in the subsequent sections. Valuesoft[12], which Adressa uses, is one company that provides such solutions, and will take the editorial content created for the original paper edition, and transform it to any given digital format. This is most usually a digital version of

[12] `http://valuesoft.dk`

the printed original similar to the PDF version, but could probably also transform the editorial content to an e-paper friendly version.

Les Echos' and probably Sundsvalls Tidning's contributions are examples of e-newspapers that have been custom built and not a direct "hack" of something that already exists. Les Echos have added extra navigation and what also seems like a set format for each article.

In [17] the current and coming utilisation's of digital publishing is presented. Digital Print Media (DPM) is a hybrid of print and WWW presentation, a concept from the 1980s, and the idea is that the media should retain the page-format from the print format but with interaction as in the WWW format. The most usual DPM format these days is the digital versions of the paper version, as the PDF version of Adressa is. In addition to Valuesoft, Newsstand[13], Olive Software[14] and Zinio[15] are some of the companies that can make a digital representation of a paper newspaper.

The solutions of these companies is more dedicated to bring the reader a digital version of the paper edition, than bringing the reader a digital version that can be used on several different reader platforms. These digital versions are mostly in a format that excludes them from most platforms that have display sizes less than the average desktop PC, and even then it might be a need for extensive scrolling. Many newspapers also have what they call an e-newspaper, but this is not something that is meant for smaller devices. They are usually implemented in Flash, and will show the pages as they are printed. The user can then mouse-over an article with interesting headliner to get a summary or the introduction. By clicking on it, the user is shown the entire article in readable text. (The text is usually not legible in the page view.)

The author of [17] presents two models that takes digital print further than the digital representations of paper newspapers; the Electronic Media Print (eMprint) and Microsoft's TimesReader. The eMprint model makes special PDF versions of the original material, but with more use of hyperlinks, layered content, interactive forms and such. The format is letter sized and with text size that makes it fit on several viewing platforms[16]. The TimesReader by Microsoft in cooperation with New York Times uses XML to make the digital versions of the newspaper. The use of XML makes it easy to make versions tailored for different platforms. This model do also have hyperlinks and pageflipping like the eMprint model (equals no scrolling). In 3.7.1 an example of an e-newspaper will be sketched.

The PDF as a starting approach

The PDF version of the newspaper is one possible starting point for an e-paper version. It would not do well using it without adapting and "cropping" it, as it either in 100% is

[13] `http://www.newsstand.com/`

[14] `http://www.olivesoftware.com/`

[15] `http://www.zinio.com/`

[16] one example of an eMprint edition can be found at `http://www.globaljournalist.org/web-content/emprint/`

Figure 3.2: A page of Adressa with the outline of each article coloured, and the possible action of clicking on one illustrated.

too large to fit the iLiad and most available e-paper readers, and in *best fit*[17] the text will be too small and unintelligible.

One solution could be to use the original PDF pages as Table of Contents (TOC) for the articles. This kind of user interface solution is also used in the WWW-based e-newspapers deployed by many newspaper companies, as mentioned earlier. If one makes sure that the title and introduction is intelligible and can easily be read, the user can then read these and decide if she wants to read the article or not. If she decides that the article is worthy of reading she will click the article, and the article will be fetched and presented in readable text with pictures, illustrations and such. Given PDF or HTML/CSS support[18] there are a few different solutions that would yield the described experience. An illustration of this can be seen in Figure 3.2.

Using PDF all the way is one possible solution. The original page size of the PDF document would need to be reduced in size so it fits the size of the e-paper readers display, as having the reader compress the PDF is just waste of bandwidth, storage and processing. Further the Title and introduction would probably need some increase in font size to be readable at the cost of the font size of the rest of the article, which does not really matter as it is not meant to be read at that place. The entire text and illustrations of the article would also need to be extracted and adapted to be shown on the e-paper reader. This could be changing the font size and changing the layout if the

[17] Like the Best Fit function in most PDF and image readers, where the goal is to show the entire page

[18] The iRex iLiad supports PDF and HTML/CSS among others.

article would need to span several pages in the e-paper reader format. These article pages could then be appended at the end of the "TOC newspaper" with links from the each article representation in the TOC to the actual article further back. It should then also be possible to skip the TOC and flip through the newspaper and visit each article in *chronological* order.

The same appearance could also be achieved with HTML/CSS. The TOC part of the newspaper would consist of pictures of the articles, with links to the actual articles. The articles itself would not need to be pictures, but could be built as a usual HTML site. The iLiad as an example can use an XML manifest file to describe which pages belongs to each other, and can arrange them so that they can be flipped in a book-fashion.

One drawback of an approach such as this can be seen in Figure 3.2, where at least two articles spans two newspaper pages. The reader would therefore encounter two TOC pages that contains mostly the same links to the same articles. Another drawback is in the nature of PDF – it is a format intended for reading and viewing, not editing. If there should be need to edit it, it would most easily be done with editing the source and regenerate the PDF instead of trying to extract meaning and content from the generated PDF file.

The WWW edition as a starting approach

The standard WWW edition of the newspaper does not fit well into the iLiad as well, but can be fairly easy adapted to the e-paper reader format. The main problem is that the main page of the WWW edition is very heavily loaded with menu's and advertising banners. The site is clearly made with web browsers such as Opera and Firefox in mind, and with peripherals like a mouse with a scrolling wheel. The WWW version of the newspaper also provides an RSS feed, which in this case is a web feed that updates whenever a new article is published on the web site. This update consists usually of the title and introduction to the news item, and a link to the article. It is also different feeds for each category of news, such as Local, National and Sport news. This means that there are several lists containing links to the latest news in different categories.

A snapshot of the contents of one of the RSS feeds could be used as a Table of Contents for that particular category. With one such TOC for each of the categories, and maybe a common start page with the latest from each category and links to each of the TOCs, one would have a fairly good starting point for an e-newspaper. Each of the articles published for the WWW version, can also be found in a printer version, where all the unnecessary fluff is stripped off, and just the text and image of the article is provided. This printer version can easily be adapted to the e-paper reader with HTML/CSS. With the meta XML files it could also be given the appearance of a newspaper one could flip through, or just the sport section for that manner.

The WAP edition as a starting approach

The WAP edition of the newspaper is much as a lighter version of the envisioned e-paper version described in the previous paragraph. It has a starting page with the latest news

and links to each news category, and each news category has a list of the relevant latest articles and news items. It consists of mostly the same editorial content as the WWW version, but with less or smaller pictures. It could probably be used to build an e-paper version in the same manner as with the WWW edition, but is structured in a way that would make it a bit more cumbersome to extract the same as with the mentioned RSS feeds.

Summary of approaches

It is in most cases a bit cumbersome to adapt finished news content to an e-newspaper friendly format. Using RSS feeds as content sources, and then pipe it into a preset layout might seem like the most viable approach. But why do the trouble to obtain content through RSS feeds when the source is most likely easily available? In most cases it would be best to just tap into the editorial content system and get it from the source, but if one was acting as an aggregator it might be a low cost and more easy way to just use the RSS feeds to let newspapers publish content to the e-newspaper system.

3.5 Distribution

The distribution of the newspaper to an e-paper reader can be done in several ways. I will in the following outline the possibilities mentioned in the theory and other applicable possibilities and then discuss one suitable approach for the proposed e-newspaper service.

3.5.1 Scenarios and examples

In the two following sections i will use the scenarios from Section 3.1 describe how e-newspapers can be brought to the user and her device in a pull or push manner. Different needs might need different means of distribution, to keep the service as simple as can be for both users, providers and those making it.

3.5.2 Pull distribution

A user can obtain the newspaper or any other content with several different means, using pull techniques is some of them. If the user mostly uses their e-paper device for other things than newspapers, and only reads an e-newspaper on a whim, obtaining it in a pull manner will most likely be preferable. The user would not need to waste processing, memory or storage on the fetching and keeping of up to date and fresh newspaper content. Likewise if a user has a subscription at a competing newspaper she might sometimes see the need for getting exclusive or unique content from another newspaper, and doing it in a pull manner will most likely be the easiest and most cost effective option (i.e. not starting a new subscription at the other newspaper).

Instead of the user requesting and pulling content, the user could delegate this to the device itself. The user could then tell the device its preferences and maybe also when it would like to read it, and the device will then try to fulfil this by searching and

downloading prior to its "deadlines". For the user this will feel like the wanted content is pushed to the device. Having the device operate in this manner to simulate push behaviour might put a strain on the servers and the infrastructure as lots of devices will poll the servers to fulfil their "masters" request. From a mobility view it might be favourable to let the device handle the some of the mobility issues by initiating all the downloads itself.

Relating to the scenarios Relating this to Sophie in Scenario 1, she is a user who regularly reads the same kind of content at roughly the same time of the day every day. In the simplest pull scenario she would need to either navigate to the right website or similar for downloading what she wants, or less cumbersome with the push of a button. (The last option is possible on the iLiad, and will download whatever is on the users iDS inbox to the device, as described in 2.4.3) An even more user friendly way would be to tell the device that she wanted that kind of content ready at some time. The agent in the device would then start to download the correct content at an early enough time for it to finish before the deadline set by Sophie. After some time the agent might have learned roughly what time it needs to start gathering the e-newspaper to be finished in time with the freshest content possible.

Jack on the other hand, from Scenario 2, would not benefit that much from pull distribution. The way he demands to be updated every time something happens in the world of sport, would need to either check the system very often, or the agent in the device would need to poll the system very often. This could lead to unnecessary stress on the system and network.

3.5.3 Push distribution

Having the newspaper delivered with push distribution is probably more appropriate for the users who prefer to have the e-newspaper delivered to their devices in a regular manner and/or with fairly specific content. This is mostly the same as a regular newspaper subscription, but the need to pick it up from the doorstep is non-existing. If the newspaper the user wants to subscribe to supports it, the user could tailor the e-newspaper. Some examples of this would be for the user to remove everything connected to sport, could specifically subscribe to housing advertisements in a special area if on the move, basically tell the system what kind of content is wanted and what is not acceptable to receive.

Further the user could tell the system when she is interested in receiving these updates. Maybe she wants a full edition sent to her when the paper edition goes into print, or maybe more specifically only local news and weather for breakfast, and national and foreign news as well as culture and the TV guide for post dinner. Alternatively she could opt for getting content from all or specific categories whenever they happen. Lastly as mentioned in Table 3.3 content that matches special requirements could also be target of the subscription, with or without special timing preferences.

Relating to the scenarios Relating this to Sophie and Jack, push distribution would probably not affect Sophie as much as Jack. As Sophie is only requesting content a few times a day, she would not benefit much from the content being pushed to her. But a pushing pub/sub system would still have the favourable property of taking care of everything once the subscription is initiated. Jack on the other hand will benefit more from getting content pushed to him. Instead of polling the system for updates all the time, he will be notified only when there actually has happened anything. With push distribution there is easier to subscribe to content through a publish/subscribe system, as will be investigated a bit further down.

A summary of pull and push distribution is provided in Table 3.3, as well as the possible delivery techniques itself (uni- and multicast).

Table 3.3: A summary of the main options for distribution in a WiFi environment

Type	Description
Pull	User can collect the whole or the interesting parts of the e-newspaper by herself whenever she wants.
Pull	User can tell the e-paper device to get the entire e-newspaper at predefined intervals.
Pull	User can tell the e-paper device to get the interesting parts of the e-newspaper at predefined intervals.
Push	Server can deliver the entire e-newspaper to the users e-paper device, when an entire new version is ready, when some of it is updated or predefined by the user.
Push	Server can deliver only the parts of the e-newspaper the user has decided she want to subscribe to (e.g. Local news), when they are updated or as predefined by the user.
Push	Server can deliver only articles the user has specified she is interested in with specific content keywords (e.g. only football articles that is about Roar Strand), when they are updated or as predefined by the user.

3.5.4 Unicast and Multicast distribution

Distinguishing between using unicast and multicast in the actual delivery of content can make content delivery more effective. Unicast will only send to one receiver, and will require a publisher to send the same amount of packet streams as there are receivers. Multicast on the other hand sends content to a group of receivers, and a publisher only needs to send one packet stream, the network is responsible for duplicating when necessary.

While a subscription based system multicasting content to the subscribers can be an effective alternative, there are some aspects which could make it harder or unfeasible.

Whatever If there is possible to make a very detailed subscription it can be hard to make suitable multicast groups, it might not be worthwhile if there is not enough receivers to a single group. To remedy this, the detail level of the subscription cannot be much less than category such as local news etc. Then the user would subscribe to one or more groups, up to the entire e-newspaper. Another aspect is the capabilities the future will bring for these devices, as will be covered in 3.6.2. This might for example be video capability, which if it is meant to be streamed can create another bump in the road if streams can start independently.

Whenever Another problem depends on the level of *whenever* the publisher wishes to support in their subscriptions. With an increase in the details of timing of delivery, most multicast groups would possibly dwindle in size due to many different delivery times. Again this can be solved by increasing the granularity of the timescale, and have the timescale operate in maybe no less than 30 min or 1 hour.

3.6 Content Delivery

The e-newspaper content can be delivered to the user in several different ways and with several methods for making it as smooth as possible, as explained in 2.4. To improve the delivery or make it better several different means of techniques can be applied such as caching. In the subsequent sections these different options will be explored, as well as putting Wireless Trondheim into the equation.

3.6.1 Publish/Subscribe systems

To have an effective pushing of content to the subscribers of an e-newspaper, a Publish/Subscribe (Pub/Sub) system is a natural choice, especially given the nature of news content. Such a middleware system is usually deployed for environments where events happen independently, both between themselves and between event publishers and the subscribers. To mitigate the need for broadcasting or polling, an event generator would publish his events to a Pub/Sub system, and the system would forward to the interested parties. This would fit very well for the newspaper world, a newspaper of some size generates news of different kind and category from a fair number of "independent" desks. If all these pieces of content are funneled into some kind of Pub/Sub system, which would then be responsible for forwarding the right content to the right subscribers and if need be assemble it all to a complete e-newspaper.

The Pub/Sub system would also need to be responsible for dealing with users that are mobile – disconnects and roaming are potentially sources of faults and missed deliveries of subscribed content.

3.6.2 Caching in the edge of the network

Caching in the edge of the network will as noted in Section 2.4.1 make the content easier available for the user. A content server that is fewer hops away from the user will make

the content request faster and more responsive, a behaviour most users will be happy for.

Effects of mobility and Wireless Trondheim

Mobility will take away some of the advantages due to the closer placement of the caching server(s). A mobile device usually can not achieve the same level of bandwidth as one connected directly to the network. This means that in many cases the user will not experience an increase in the same manner as when connected with a land line when caching in the edge is applied, the air interface will be a bottleneck and slow down the traffic.

Effects of high speed wireless networking There are much work and research being done on improving the performance of mobile networks, and that these networks does not provide *enough* bandwidth compared to fixed networks might soon be a moot point for a lot of services, the current e-newspapers included. The network Wireless Trondheim operates today do provide a fairly good amount of bandwidth, and can probably in many instances be considered as a fast land line connection all the way to the device. Weather, interference and overall load will of course affect what it can ultimately provide. Hence, taking edge caching out of the equation might not seem smart, but given the current size of the Wireless Trondheim network there is really no need for such a collection of servers. As well, the AP's are interconnected by fiber, which further underlines that there would not be great need for such edge caching *yet*.

Effects of growth Providing Adresseavisen and maybe other local newspapers as e-newspapers over a network with the current size, would probably not give any trouble for a single source of content. In the beginning it is likely something only for the first-adopters to use, but if it catches on with both more available newspapers for subscription as well as more users and coverage area some expansion should be considered. Hence it is preferable that such a system easily could be scaled for more use. Another consideration is the increase in capabilities for such devices, as will be covered subsequently in 3.6.2, which will increase the required bandwidth for service fulfilment.

It is not unlikely that more cities than Trondheim get a wireless coverage in the years to come. A centralised source of published content might fast become undesirable, and some sort of distributed architecture would seem to be more preferable. One possibility is to put a cache in each city/domain. With some sort of master source where all content is available and then having caching rules like Greedy-Dual*(2.4.1) and its like govern the edge caches can seem like one possible solution. Another possibility is to replicate the system in each domain, but in either cases one should be wary of incidents as mentioned in 2.4.2; users roam and would like undelivered content to follow it, and does not want several untimely deliveries of one piece of content.

Effects of roaming Another aspect of introducing mobility in the equation is the fact that a user and her device can and will roam. To maintain the advantage that exists with edge caching, there needs to exist functionality for roaming and handover. There also needs to be functionality that can handle users that disconnects and later reconnects. The handover needs to be between access points (AP) and between the caching servers. If the user moves about, and gets closer to another caching server the device or system needs to check if the new and closer server do have the same content. If it do not, there is not really a need to involve the new server in the chain, as it would then have to acquire the content before sending it to the client. But one action that can be done is to tell the closest caching server what content it would have requested, with enough requests for some content the server could have it ready for other users at a later time.

Requirements and limitations of e-paper and e-newspaper

Applying this knowledge and requirements to a e-newspaper service alters this to some degree. The commercially available e-paper technology of today has limits on what it can do and display. Colour e-paper displays are still mostly in research and development, and to the authors knowledge, Fujitsu Flepia (2.3.4) is the only e-paper device with colour display – but according to Fujitsu it will not launch before the fiscal year 2008[19].

The ability to show moving pictures is also functionality that is some time away. Electrowetting (2.2.3) is an interesting technology that got low enough response time to achieve video-like content. With colour and fast response times e-paper can soon be as widespread as LCD screens, but there will still be some time before it is likely to see an realisation of this in a device.

Without colour, video and other moving pictures (such as Flash advertisement) there is certainly less content that needs to be transfered for each newspaper. With each e-newspaper being smaller in size, there are some new interesting aspects;

- There might not be any point in doing handover between caching servers, as the increase in speed due to closer source will not be worth the hassle of doing handover. Increased speed for files of smaller size will probably not be that great an improvement.

- If a Pub/Sub system is used, the system itself can keep the caches updated with the correct content based on the subscribers in the area and their subscriptions. The subscribers would then have their content easily available in their vicinity. Then the caches would also be more of an extension of a Pub/Sub system than just caches for pull requests and the like.

- Given that an entire e-newspaper is not of great size, there would most likely add complexity to deliver just the needed parts of each of the newspapers to each cache instead of just sending entire e-newspaper, and let the caches send out what is needed.

[19] Fujitsu fiscal year 2008 is from April 2008 to April 2009

- If the subscriber of an e-newspaper wants to be updated with some particular kind of news when they happen, these parts must be sent to the relevant cache servers, before served to the end user.
- Lastly should there be assembled an entire new e-newspaper when something new needs to be added, or should there be possibility for adding articles and such to an existing edition?

The aspects mentioned above does introduce the need to make some choices for an e-newspaper system. In a little while these will be further explored as a complete system is proposed.

3.6.3 Pre-caching at the devices

Caching at the devices will not reduce the total time for a newspaper download, but will make a service more convenient for the user as the e-newspaper will be available with little to no downloading. As described in Section 2.4.2 several methods can be used to predict what and when to download content.

Relating this to the e-newspaper case there are some situations where pre-caching like this is suitable, but again some where it is not that suitable. The main attraction with pre-caching is that it starts to download in the background for an anticipated occasion where the downloaded data will come to use. So for a user like Sophie which have a reading pattern that is very predictable (so predictable that she has set the e-newspaper to be ready at set times), the device can start working on the edition long before it is due. Jack on the other hand with his *news when they happen* attitude would not benefit from such a solution other than being spared seeing a "download in progress" message.

Requirements and limitations due to Wireless Trondheim

The wireless network provided by Wireless Trondheim is of such a nature that with the current capabilities of e-paper devices there is most likely no real need for such device caching. The content that needs to be sent to the devices is not of such a grand scale bit-wise, and will most likely not take such a long time to be inconvenient.

But when "evolution" and convergence has done its work on the e-paper devices the need for pre-caching might be more immediate. If or when news articles contains lots of pictures and video, it might be of some importance to start downloading in the background in anticipation of a deadline. Of course this is a moot point if the user of the device wants everything as soon as possible and when it happens.

3.7 Proposed solution/system

In this section a proposed solution to the problem at hand is described with a possible structure and layout for an e-newspaper, any adaption that might be needed and how it is going to be distributed and delivered.

3.7.1 Layout and structure

As stated in 3.3.1 the best layout for an e-newspaper would be to combine the best from both worlds of e-book reading and WWW browsing. Using a similar approach as Le Echos would seem to be a good start for this situation. When colour and video becomes possible one might want to reconsider.

In the following subsections each major section and element of an e-newspaper will be examined and proposed. The use of either PDF or HTML and pictures to make some sort of Table of Contents was explored in 3.4.2. At the current state of hardware this seems to be the most cumbersome solution to make, and possible also use. The best approach is most likely an overall tree structure with some sort of start page as root. Connected to the start page would be the start pages for each section or category. The children to the sections would either be subsections or just the articles. If several newspapers and magazines are providing content for such a system it might be appropriate to either sort it by publisher or in some way tag the articles.

If HTML or similar is chosen it might be advantageous to choose an approach similar to the meta-files of iRex which binds several separate files into the correct structure presented to the user.

The action of *content downloading* will be used throughout this section. It will in this context either mean the act of downloading an entire e-newspaper according to subscription, or the act of updating said e-newspaper with new articles or revisions to old ones. The subscription of the user will probably be the deciding factor; if a user needs the e-newspaper rarely it is more practical to download the entire thing. For users that is being updated all the time refreshing and appending new articles will be a better solution as it is less to download. For these users it might be of use to archive or delete when articles gets older than a preset value, this will prevent it from growing to something the device might have trouble coping with at one time.

Start page

This page can consist of a multitude of different elements that is of interest when initiating an e-newspaper session.

- A number of headliners from various categories. (What the system thinks is most important at the moment.)
- A number of the most recent news items. (The most recent articles and such published to the subscribed sections/categories).
- The available sections/categories based on the subscription
- Changes from last e-newspaper session.

Making room for all of these and putting them on such a front page together would most likely end in a cluttered and not particularly user friendly interface. A better approach might be to do a modular front page where the user can decide what she wants and

where. For usability a standard start page should be designed. This could for example be partitioning it in three pages. The first consist of a summary of the most important and recent news, akin to a regular front page of a newspaper. The second would be the *Table of Content* with links to all the subscribed sections. The last of the three would consist of a list of the most recent or most important changes and additions since last e-newspaper session. The navigation between the three pages would be with traditionally page flipping.

There might also be need for some sort of a *quickstart* link, which would let the user skip to the first page in the entry newspaper with all the subsequent pages and articles lined up for flipping through.

Section

Each section would probably benefit from some of the same elements as the front page. The most recent and important articles on some sort of section start page for easy and quick access, as well as some sort of list with all the available articles for that section. How long an article should stay in an e-newspaper should probably also be something the user would want to control. Putting an article in a separated archive after a set time after being read would probably be to most peoples liking, and would prevent the e-newspaper to become a resource hog as it gathers more and more articles.

Each article in a listing would probably need to consist of the title of the article as well as the introduction. It might be worth to consider having the introduction limited to some length, to prevent too many pages with article listings.

Lastly the section would also benefit to have a list with changes since last session, maybe with more granularity than the front page one. Could also be considered to have a *new* tag or such in the article listing to identify them that way.

Article

The content input to the system should be as decoupled from the layout as possible. The best would be if the content publishers provided the headliner, introduction, main text and illustration in easily identifiable elements. Then the system could use one or more preset layout models and build the entire article. Such a decoupling would make it easier for a publisher to publish for such a channel, as there would be no great need for them to work on the layout. A drawback is present for publishers who wishes to have absolutely all control with their own content.

An example of such an preset layout could be as following:

- Header in a preset font and size, may be resized to span the entire width of the page if suitable.
- Introduction or ingress underneath the header. Set in slanted or italic with a preset font and font size.
- The author of the article as well as date and similar set at a font size less than the introduction. Preset font.

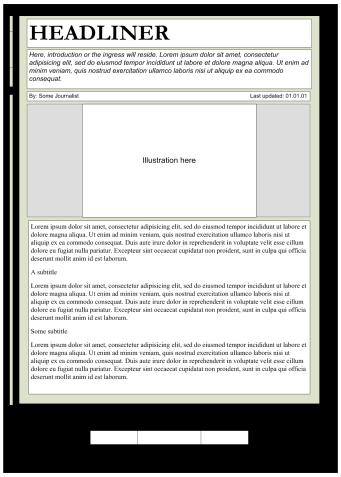


Figure 3.3: An example of an article page in an e-newspaper.

- Illustration or picture of some kind will be next. These four elements could for example aim to cover about 50% of the entire screen area, and hence illustration should scale to fit this goal. Image caption should also be considered within these 50%.

- The body text would then fill the rest of the page in the preset font and font size. Any overflowing text from the first page will continue on subsequent pages. These pages should be possible to navigate in a e-book page flip style. At the end of one article there should be the possibility to start on the next article in the list without going by the article list.

See Figure 3.3 for an illustration of this. A larger picture of this and other is available in appendix B.

3.7.2 Content adaption

Ideally such an e-newspaper service should be available on a number of different e-paper devices. Usually no device is equal in capabilities, and some amount of adaption will be of need. The differences can be in display size, grey scale and colour scale. Until video is widely available the need for that kind of adaption is not very great.

Until there is a great plethora of e-paper devices with WiFi capabilities, preset values for layout to each of the devices is probably the easiest approach and with best result. To cater for "non-standard" display sizes it might be beneficial to use something like HTML *frames*. These frames can easily be manipulated with CSS, and the optimal layout for a given display area might be calculated and applied fairly easy.

To save space and bandwidth it will also be advantageous to transcode images to the right size, colour and grey scale. Having a big high resolution picture in millions of

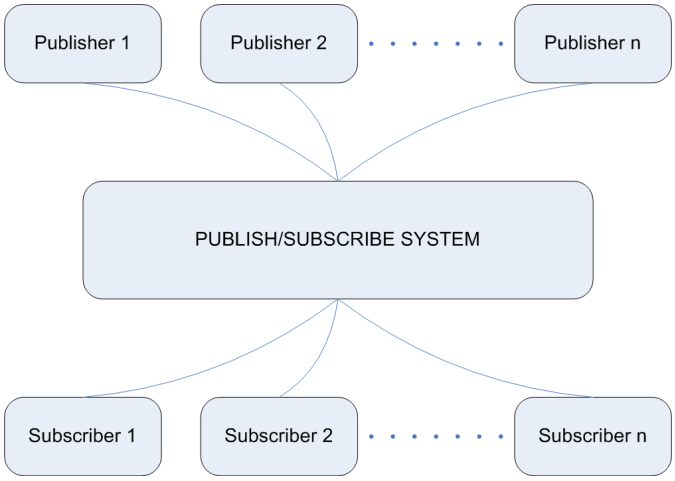


Figure 3.4: An overview of a possible Publish/Subscribe system.

colours sent to a device that only can display it at 200x200 pixels and with 4 shades of grey is not a very efficient use of bandwidth and the devices resources.

3.7.3 Content delivery

As described in previous sections a Publish/Subscribe system would be a good alternative for an e-newspaper system. All contributors publish content to the system which will then take care of getting it out to the right customers at the right time. This can be seen in Figure 3.4.

Assumptions

There are made some assumptions to simplify the model somewhat. Any handovers in the network between different access points is handled by the network. The main mobility issue that the system should cope with is the case when a user disconnects for some reason, and is unable to download all of the scheduled content. In such a case the system should let the device easily resume and download anything that is missing. This example is for a wireless network the size of Wireless Trondheim, anything bigger might need a more distributed architecture with handover between different domains.

Under the hood the system would need some different services and components to fulfil all the necessary functions. This can be seen in Figure 3.5, and will be explained below. It has several similarities with Figure 2.13 and some inspiration was gathered from it.

The Subscription The user will have to create a subscription before using the system. This subscription will include information about the user's e-paper device, and will be of use for the Adaption Manager and Newspaper Builder. It will also include information

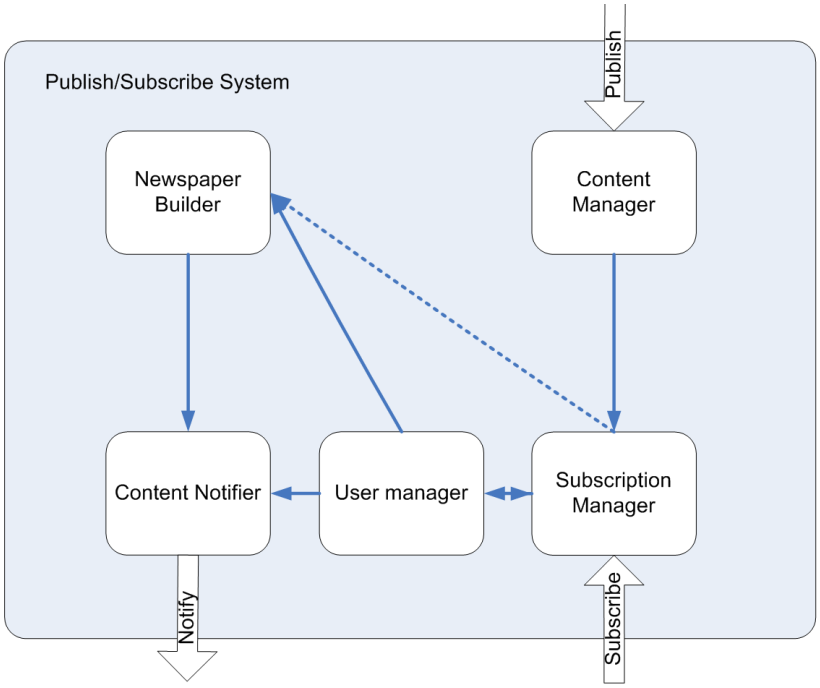


Figure 3.5: The inner workings of the middleware.

about what content the user subscribes to, as well as preferred delivery times. The subscribed contents could come from several sources such as newspapers, magazines, news agency's etc.

Handling the subscriptions and publications

The main function of this system is to handle publications and send them to the right subscribers. For this there needs to be a system for receiving content publications and for finding the right subscribers. The system would also need some kind of method for storing received content, as subscribers could sometimes due to mobility be unable to complete or even start a download. It could then be worth separating the functionality in two separate entities, a *content manager* and a *subscription manager*.

The Content Manager When a publisher publishes some content to the system the Content Manager is responsible for the processing after the system receives it. On reception it will classify the content before storing it in a database. It will also notify the Subscription Manager of the newly arrived content. For the speedy delivery of content at a later time it might be worth considering to transcode any illustration into some of the most popular formats and sizes and store them along with the original.

The Subscription Manager When the Subscription Manager is notified of the arrival of new content it will immediately prepare to push the content to the right receivers. First it will need to generate a list of recipients by cross referencing the category of the received article with the registered subscriptions in its subscriptions database. Parallel to starting that search it issues the Newspaper Builder with creating preset page-layouts for the recently arrived content. These presets are based on statistics for what kind of device that receives a given type of content. When the list of recipients is ready it will pass the list on to the User Manager.

Handling the users

It is very unlikely that any content publishers will give away any quantity of content to users for free, bar for promotional reasons. Another issue with users in such a system is their mobility, which needs to be monitored if content is to be delivered properly. Thus, there needs to be functionality for managing the users, the AAA (Authentication, Authorisation and Accounting) aspect as well as locality.

It could be considered to add this functionality in the subscription manager, as that would keep the subscription information for each user coupled with the rest of the user information. But if there on a later time was added additional services which needed such handling of users, it would be better to have it as a separate entity.

The User Manager The User Manager is responsible for managing the users. This includes keeping track of all the user's whereabouts as well as any outstanding content they might have missed out of due to mobility issues. It is also responsible for the AAA; Authentication, Authorisation and Accounting related to any subscriber.

It will go through the list and update the subscriber entries depending on whether it does know about the current location and address or not. Those it knows the location and/or address to, it will append that information to the entry. The User Manager will update the "outstanding content list" of each of the missing users as well as giving the entry in the list a "delivery pending" status, and send the finished list to the Newspaper Builder. If the subscriber is set to receive the e-newspaper at given times it will also get the content put in the "outstanding content list" and the entry given a "delivery at time X" status. The User Manager will try to deliver at the given time, and if it is unable to it the entry will be given "delivery pending" status which means it shall be delivered as soon as it is possible to connect to the subscriber.

Building the newspaper

As of yet the e-paper devices are fairly low on computing power, and it would be best if the e-newspaper is as refined and adapted as possible.

The Newspaper Builder The list of entries that the Newspaper Builder receives contains information about the receiving device. For every new device it encounters, it will make a class for that kind of device containing all the information it needs. Given

that such devices are at the moment is mass-produced by a few companies the amount of classes should be manageable. Keeping statistics about the devices in use can be an effective tool in speeding up some of the steps, by knowing what it most likely will need to do.

Most articles will be accompanied by an illustration. The Newspaper Builder will check if the Content Manager has any pre-adapted copies of the illustration. If there is some missing it will issue an request for the illustration to be adapted to the given specifications and stored along with the rest of the article. The Newspaper Builder manages a set of different transcoding and adaption services.

The Newspaper Builder will use the information given about the devices to finalise the article for each device. If the layout rules is as described in 3.7.1 it should exist generic layouts for each device it has encountered. It would then only need to do some minor calculations to tailor each frame to its contents and fill in the text and the <*img*>-tag for the illustration. It might be smart to decouple the layout from the content, as the layout could be saved for those who needs a later delivery. Each entry in the build-list will be processed depending on the status it is given previously by the User Manager.

If the content/article is meant to update an already existing e-newspaper, it will also make entries for the "recent articles" lists that the device will update by itself and as the last operation generate the necessary meta files. For entire e-newspapers the recent articles is kept empty and lastly the meta files is generated. When it is done with one entry it will notify the Content Notifier.

Pushing to the subscriber

Whenever the subscription of a user equals true the content should be pushed to the user. To alleviate the issues related to mobility a push-pull method of distribution is better than pure push.

The Content Notifier The Content Notifier will use the information gathered for the entry to make a notification that it will push to the receiving device. The notification contains information and addresses the device will need to download the content. Notice that it is not pure push transfer of data that is used, only of the notification. The reason for choosing to do it like this is because of the mobility of the user. If the device should disconnect for some reason during download it will now have all the information it needs to start another download.

The pushed information should include elements such as subscriber id as well as id for the content that was prepared for it. If the device uses to long time before initiating a download of the content prepared for it, it could be deleted to save space. Then the id's could be marked in such a way that the server should be able to easily fetch the right content file and apply the correct adaption again.

The pushed notification should also include information for authentication and authorisation purposes.

The downloading of the content When the device connects to the system using the received information it will present its notification id, and maybe perform some kind of authentication. The id will let the content manager find the correct content along with the correct layout. This is then downloaded to the device, which will then notify its user either immediately or on deadline. The last case can be considered as pre-caching.

Summary

See Table 3.4 for an overview of the components and their inner workings.

Table 3.4: A summary of the components in the proposed solution

Component	Description
Content Manager	Receives all new content Classifies content Stores content in a database
Subscription Manager	Stores all subscriptions Generates build-lists for new content
User Manager	Stores all subscribers Manages the users location and addresses AAA
Newspaper Builder	Creates layout templates based on device and content Assembles several articles together to an entire e-newspaper
Content Notifier	Constructs notifications that are sent to subscribers The notifications will help the subscriber connect and get the right content

4 Conclusion

This project assignment has evaluated and examined means for achieving an e-newspaper service over wireless networks to e-paper devices.

E-paper and devices

The different e-paper technologies and devices that utilise them have been examined. Even though the principles of the technology was founded many decades ago, it do still have some way to go. There are no commercially available devices with colour e-paper displays, and there seems to be yet some time before they are available at a price most people can live with. Another issue is the lack of video capabilities, which is the new and "hot" thing for many online newspapers.

But there is many properties of the e-paper that makes up for these shortcomings until research and development has catched up. It is very easy on the battery and has a readability many times higher than conventional backlit LCD-displays. This makes it perfect for mobile applications. But there are few devices that come with capability for wireless communication, iRex iLiad is the only one commercially available with WiFi.

E-newspaper

An e-newspaper is one viable service for the e-paper platform. There are some issues and features that needs to be solved and decided on. This is primarily concerned with the *layout and structure* of the e-newspaper and the means of distribution to the subscribers.

Due to the digital nature of the e-paper devices, the options for layout is mainly concerned with the "WWW-approach", the "(e)Book-approach" or the combined approach. The combined approach seems most promising as it will use the best from both; page-flipping on the individual article and hyperlinking across the entire e-newspaper.

Adapting content must also have some attention as no device is completely similar to one of another brand. Illustrations might need to be resized as well as being converted to grey scale.

Distribution

The choice for distribution of an e-newspaper is between pull- or push-like, which also can be seen as a choice between different levels of interaction for the subscriber. A push solution would seem like the best approach for a subscription based service like e-newspapers, as the e-newspaper could then be pushed to the subscribers at their own discretion. The user would only need to create a subscription and add content based on their own liking, as well as when they would like to receive it. Such functionality is the trademark of Publish/Subscribe middleware systems; the publisher sends content to the middleware, and the middleware then forwards the content to the right subscribers.

Due to the criteria of wireless distribution, there needs to be mechanisms available to alleviate any trouble related to mobility. These issues could be that the device is not in an area with coverage when the subscription is due, as well as a device that disconnects when a content download is in progress. All content should be saved in the middleware to enable downloads to be resumed or started at a time after it was originally intended.

Methods for making distribution more effective such as various kinds of caching would most likely have no big effect on a service in the current Wireless Trondheim network, but should be considered for larger applications later on.

Future Work

To test the feasibility of the proposed methods here, a prototype system should be built and deployed on a number of devices operating in the Wireless Trondheim network. On the publishing side several content providers would be needed, different desks within the same newspaper should probably suffice.

Bibliography

[1] Amazon.com. Amazon.com: Newspapers: Kindle Store: U.S., France, Ireland, United Kingdom, Germany, International, Spain & More. http://amazon.com/s/ref=sr_ex_n_1?ie=UTF8&rs=251293011&rh=n%3A165389011, Nov 2007.

[2] Amazon.com. Kindle: Amazon's New Wireless Reading Device. http://amazon.com/gp/product/B000FI73MA/ref=amb_link_5873612_3?pf_rd_m=ATVPDKIKX0DER&pf_rd_s=gateway-center-column&pf_rd_r=1X1E1MSFH8RAVSGAMNDJ&pf_rd_t=101&pf_rd_p=329252801&pf_rd_i=507846, Nov 2007.

[3] Steinar H. Andresen, John Krogstie, and Thomas Jelle. Lab and Research Activities at Wireless Trondheim. In *Proceedings of ISWCS 2007*. IEEE ISWCS, Oct 2007.

[4] Jacques Angelé and Thierry Emeraud. BiNem Electronic Paper. *Gekkan Display - Techno Times Japan*, pages 1–8, Oct 2006.

[5] Xerox PARC (Palo Alto Research Center). Electronic Reusable Paper. http://www2.parc.com/hsl/projects/gyricon/, Sep 2007.

[6] Mao Chen, Jaswinder Pal Singh, and Andrea LaPaugh. Subscription-enhanced Content Delivery. In *Web Content Caching and Distribution: Proceedings of the 8th International Workshop*, 2004.

[7] E Ink Corporation. E Ink Technology. http://www.eink.com/technology/howitworks.html, Sep 2007.

[8] Stephane Coulombe, Oskari Koskimies, and Guido Grassel. Content Adaption For The Mobile Internet. In Sudhir Dixit and Tao Wu, editors, *Content Networking In The Mobile Internet*, chapter 7. Wiley Interscience, 2004.

[9] Gianpaolo Cugola and Gian Pietro Picco. REDS A REconfigurable Dispatching System. http://zeus.elet.polimi.it/reds/, 2007. Website of the project.

[10] Gianpaolo Cugola and Gian Pietro Picco. REDS A Reconfigurable Dispatching System. http://zeus.elet.polimi.it/reds/reds_wp.pdf, 2007. Whitepaper.

[11] Les Echos. Présentation de l'offre e-Paper - sur un reader iLiad. http://ftp-videos.lesechos.fr/videoepaper/video-epaper-lesechos-en.html, Sep 2007.

[12] Les Echos and iRex Technologies. Les Echos first French electronic newspaper edition daily on iRex iLiad. http://www.irextechnologies.com/files/20070907%20Press%20release%20LesEchos.pdf, Sep 2007.

[13] Ifra eNews. Ifra e-news initiative 2005 - 2008 - Executive Brief: Creating business in the emerging mobile e-reading world. http://de.sitestat.com/ifra/ifra/s?website.nsf.download.20060704_eNews2008_ExecExternal.pdf&ns_type=clickout&ns_url=http://www.ifra.com/website/website.nsf/ACBF49EB3513C7C3C125722B005C713A/$FILE/20060704_eNews2008_ExecExternal.pdf, Jul 2006.

[14] Ifra eNews. The second year of eNews 2008. http://de.sitestat.com/ifra/ifra/s?website.nsf.download.Ifra_eNews2008_2ndyear.pdf&ns_type=clickout&ns_url=http://www.ifra.com/website/website.nsf/ACBF49EB3513C7C3C125722B005C713A/$FILE/Ifra_eNews2008_2ndyear.pdf, Jul 2007.

[15] B.J. Feenstra, R.A. Hayes, I.G.J. Camps, R. van Dijk, L.M. Hage, T.J.P. van den Biggelaar, and J.M.E. Baken. 1-in. Active Matrix Addressed Electrowetting Displays. In *International Display Workshop Proceedings*, 2004.

[16] Johan Feenstra and Rob Hayes. Electrowetting Displays. http://www.liquavista.com/documents/default.asp?CatID=6, Jan 2006. (Whitepaper for Liquavista, Principles of Electrowetting).

[17] Roger Fidler. The e-Bookstore: New publishing opportunities afforded by e-paper and mobile reading devices. http://www.rogerfidler.com/articles/rf2007a.asp, 2007. (Speech given at Second Annual Congress of Spanish Publishers of Periodicals in Marbella, Spain, April 20, 2007. Published as a special report titled *Exploring the Future* in spring 2007.).

[18] Keith Hamilton. Web Solutions Platform Event System: A Distributed Publish/Subscribe Event System. http://www.codeplex.com/pubsub/Project/FileDownload.aspx?DownloadId=15135, 2006.

[19] Keith Hamilton. Distributed Pub/Sub Event System. http://www.codeplex.com/pubsub, 2007.

[20] Manfred Hauswirth. Minstrel Fact Sheet. http://www.infosys.tuwien.ac.at/Staff/pooh/Minstrel/FactSheet/FactSheet.html, 1998.

[21] Marlowe Hood. E-Newspapers: Digital Deliverance? *IEEE Spectrum*, pages 10–12, February 2007.

[22] Yongqiang Huang and Hector Garcia-Molina. Publish/Subscribe in a Mobile Environment. *Wireless Networks*, 10(6):643–652, 2004.

[23] H. Edzer A. Huitema, Gerwin H. Gelinck, Erik van Veenendaal, Fred J. Touwslager, and Pieter J. G. van Lieshout. Roll-up Active-matrix Displays. In Hagen Klauk, editor, *Organic Electronics*, chapter 14. Wiley-VCH, 2006.

[24] Carina Ihlström, Maria Åkesson, and Stig Nordqvist. From print to web to e-paper – The challenge of designing the e-newspaper. In *ICCC 8th International Conference on Electronic Publishing, ELPUB 2004*, 2004.

[25] iRex Technologies. Technical Specifications, The Iliad. `http://www.irextechnologies.com/products/specs`, Sep 2007.

[26] Jinke. Product Plan. `http://www.jinke.com.cn/Compagesql/English/embedpro/newpro.asp`, Sep 2007.

[27] Hayat Kara and Christopher Edwards. A Caching Architecture for Content Delivery to Mobile Devices. In *Proceedings of the 29th EUROMICRO Conference 'New Waves in System Architecture'*. IEEE Computer Society, 2003.

[28] Andreas Komninos and Mark D. Dunlop. A calendar based Internet content precaching agent for small computing devices. *Personal Ubiquitous Computing*, 2007.

[29] Irwin Lazar and William Terrill. Exploring Content Delivery Networking. *IT Professional*, pages 47–49, Jul|Aug 2001.

[30] LG.Philips LCD. LG.Philips LCD Develops World's First Flexible Color A4-Size E-Paper. `http://www.lgphilips-lcd.com/homeContain/jsp/eng/inv/inv101_j_e.jsp?BOARD_IDX=1280&languageSec=E&kinds=IN1`, May 2007.

[31] Steven Levy. Cover Story: Technology - The Future of Reading. `http://www.newsweek.com/id/70983/page/1`, 2007.

[32] Fujitsu Limited. Fujitsu Develops World's First Film Substrate-based Bendable Color Electronic Paper featuring Image Memory Function. `http://www.fujitsu.com/global/news/pr/archives/month/2005/20050713-01.html`, Jul 2005.

[33] Fujitsu Frontech Limited. Fujitsu Frontech Starts Limited Sales of Portable Information Terminal 'FLEPia'. `http://www.frontech.fujitsu.com/en/release/20070420.html`, Apr 2007.

[34] Plastic Logic Limited. Manufacturing flexible e-paper displays. `http://www.plasticlogic.com/downloads/Plastic_Logic_FPD_07.pdf`, Oct 2007. (Presentation given at FPD International 2007).

[35] Peter Millard, Peter Saint-Andre, and Ralph Meijer. XEP-0060: Publish-Subscribe. `http://www.xmpp.org/extensions/xep-0060.html`, 2007. Draft Standard of the XMPP Standards Foundation.

[36] Katashi Nagao. *Digital Content Annotation and Transcoding*. Artech House, 2003.

[37] Nemoptic. Nemoptic presents new version of its color e-paper displays at SID 2007. `http://www.nemoptic.com/content.php?section=news&cat_id=1&id_news=36`, May 2007.

[38] Michael J. O'Grady and Gregory M. P. O'Hare. Just-In-Time Multimedia Distribution in a Mobile Computing Environment. *IEEE MultiMedia*, pages 62–74, Oct-Dec 2004.

[39] Ivana Podnar, Manfred Hauswirth, and Mehdi Jazayeri. Mobile Push: Delivering Content to Mobile Users. In *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshop*. IEEE Computer Society, 2002.

[40] Rohan Samarasinghe, Duminda Nishantha, Yoshihiro YasuTake, Lanka Rodrigo, and Takaichi Yoshida. Adaptive Multimedia Transcoding for WWW Content Delivery. In *Industrial and Information Systems, First International Conference on*. IEEE, 2006.

[41] Jay Shah and R. Malcolm Brown Jr. Towards electronic paper displays made from microbial cellulose. *Applied Microbiology and Biotechnology*, 66:352–355, 2005.

[42] Henning Sirringhaus, Christoph W. Sele, Timothy von Werne, and Catherine Ramsdale. Manufacturing of Organic Transistor Circuits by Solution-based Printing. In Hagen Klauk, editor, *Organic Electronics*, chapter 12. Wiley-VCH, 2006.

[43] Mila Z. Stojanovic and Zoran S. Bojkovic. Wireless Content Delivery Network Requirements. In *Proceedings of the 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services - TELSIKS*. IEEE, 2005.

[44] Sundsvalls Tidning. ST först i Sverige som e-papper. `http://www.st.nu/nyheter/arbete.php?action=visa_artikel&id=560295`, Oct 2007.

[45] Athena Vakali and George Pallis. Content Delivery Networks: Status and Trends. *IEEE Internet Computing*, pages 68–74, Nov|Dec 2003.

[46] Nico Verplancke. The mobile digital newspaper - Lessons learned an reflections on the *book business*. `http://www.ebf-eu.org/documents/presentation_epaper_EBF_05102006%20-%20nico.pdf`, Oct 2006.

[47] Polymer Vision. Readius. `http://www.polymervision.com/ProductsApplications/Readius/Index.html`, Nov 2007.

[48] Polymer Vision. Telecom Italia and Polymer Vision announce the CELLULARBOOK. `http://www.polymervision.com/News-Center/Press-Releases/TelecomItaliaandPolymerVisionannouncetheCE.html`, Nov 2007.

[49] Wikipedia. Content Delivery. `http://en.wikipedia.org/wiki/Content_delivery`, Oct 2007.

[50] Wikipedia. E Ink Corporation. http://en.wikipedia.org/wiki/E_Ink_Corporation, Nov 2007.

[51] Wikipedia. Electronic Paper. http://en.wikipedia.org/wiki/Electronic_paper, Sep 2007.

[52] Wikipedia. Electronic Paper - Newspapers. http://en.wikipedia.org/wiki/Electronic_paper#Newspapers, Nov 2007.

[53] Wikipedia. Response Time (Technology). http://en.wikipedia.org/wiki/Response_time_(technology), Sep 2007.

[54] Wikipedia. Sony Librie EBR-1000EP. http://en.wikipedia.org/wiki/Librie, Sep 2007.

[55] Wikipedia. Thin-Film Transistor. http://en.wikipedia.org/wiki/Thin-film_transistor, Nov 2007.

[56] Tao Wu, Sadhna Ahuja, and Sudhir Dixit. ACME: A New Mobile Content Delivery Architecture. In Sudhir Dixit and Tao Wu, editors, *Content Networking In The Mobile Internet*, chapter 6. Wiley Interscience, 2004.

List of Abbreviations

AAA Authentication, Authorisation, Accounting

ACME Architecture for Content delivery in the Mobile Environment

AP Access Point

CDN Content Delivery Networks

CPU Central Processing Unit

CSS Cascading Style Sheets

DPM Digital Print Media

EBS Event Brokering System

EDGE Enhanced Data Rates for GSM Evolution

EVDO Evolution-Data Optimized

GPRS General Packet Radio Service

GPS Global Positioning System

GSM Global System for Mobile communications // Groupe Spï¿½cial Mobile

HTML HyperText Markup Language

HTTP HyperText Transport Protocol

HTTPA HyperText Transport Protocol Adaptive

iDS iRex Delivery Service

J2EE Java 2 Platform, Enterprise Edition

J2ME Java 2 Platform, Micro Edition

MAC Media Access Control

MMC MultiMediaCard

MP3 MPEG-1 Audio Layer 3

MPEG Moving Pictures Expert Group

P/S Publish/Subscribe

PDA Personal Digital Assistant

PDF Portable Document Format

POI Point of Interest

QoS Quality of Service

REDS REconfigurable Dispatching System

RGB Red Green Blue

RGBW Red Green Blue White

RRI Request-Routing Infrastructure

RSS Really Simple Syndication

SD SecureDigital

SMS Short Message System

TFT Thin-Film Transistor

TOC Table of Contents

UMTS Universal Mobile Telecommunication System

UMTS Universal Mobile Telecommunications System

URL Uniform Resource Locator

USB Universal Serial Bus

VLAN Virtual Local Area Network

WAP Wireless Application Protocol

WEP Wire Equivelent Privacy

WiMAX Worldwide Interoperability for Microwave Access

WLAN Wireless Local Area Network

WWW World Wide Web

XHTML Extensible HyperText Markup Language

XML eXtensible Markup Language

XSLT Extensible Stylesheet Language Transformations

A

Mail correspondence

Mail correspondence Svenåke Boström ved Sundsvalls Tidning

To obtain information about the e-newspaper project of Sundsvalls Tidning, Svenåke Boström that works as a media developer for Sundsvalls Tidning was asked. The most important information was received in a mail 04/12-2007 (Formatted to look better).

A.1 Email received 04/12-2007

Hej Håkon,
Här kommer svar på dina spørsmål.

» - Hvordan har avisen blitt tilpasset e-papir formatet? (tilpassing av nettutgaven, papirutgaven, eller en egen e-papir versjon laget fra bunnen av?)

> Vi har en webbapplikation som är kopplad till vårt redaktionella system (Newspilot från Infomaker). Det är en automatgenerering av sidor från artikellistan, till rätt format för e-pappersläsaren. Varje dag vi publicerade så tog det 2-3 timmar för en person att kolla att innehållet var OK och göra ev. justeringar av rubriker och bilder.

» - Hvor ofte og hvordan har avisen blitt oppdatert for e-papir?

> Vi publicerade två upplagor av morgondagens avis, den första 19.00 och den andra 00.00

» - Har brukeren måttet hente avisen selv på internett, eller har den blitt oppdatert rett til e-papir leseren?

> Brukaren trycker bara på en uppdateringsknapp så laddas avisen ner till läsplattan. Det kan göras via kabel eller trådlöst från Internet.

» - Var prosjektet en suksess? (Har ikke funnet noe om hvor lenge prosjektet skulle vare, eller om det er ferdig nï¿½?)

> Vi kommer tillsammans med Tidningsutgivarna att planera ett nytt test för 2008, förhoppningsvis med någon ny läsplatta. Jag tror att vi 2009 eller 2010 kan vara beredda att introducera detta på vår lokala marknad.

» Hvis det finnes noen artikler/tekster utover de som finnes på st.nu om dette prosjektet så er jeg veldig interessert.

> Här finns en del länkar till information om e-papper.
> polymervision.com
> E-ink.com
> diginews.se
> irextechnologies.com
> ubimedia.se
> epaper.org.uk
> media-it.hh.se
> amazon.com
>
> Och här en av de senaste artiklarna:
> http://www.metro.se/se/article/2007/11/26/15/4759-48/index.xml
>
> Om du söker på e-papper eller e-paper på google så får du nog en hel del träffar.

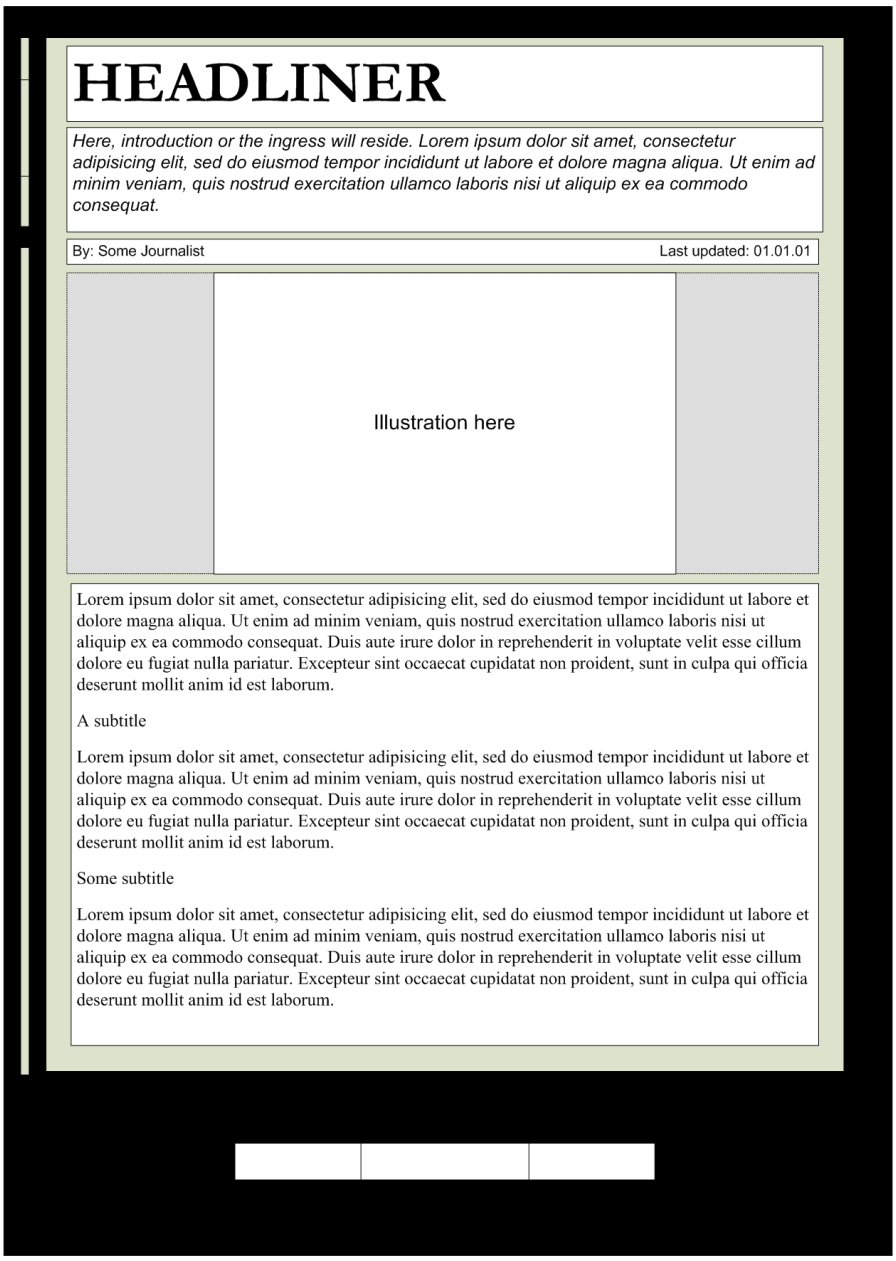
Hälsningar,
Sveåke

B

Possible Layout and Structure

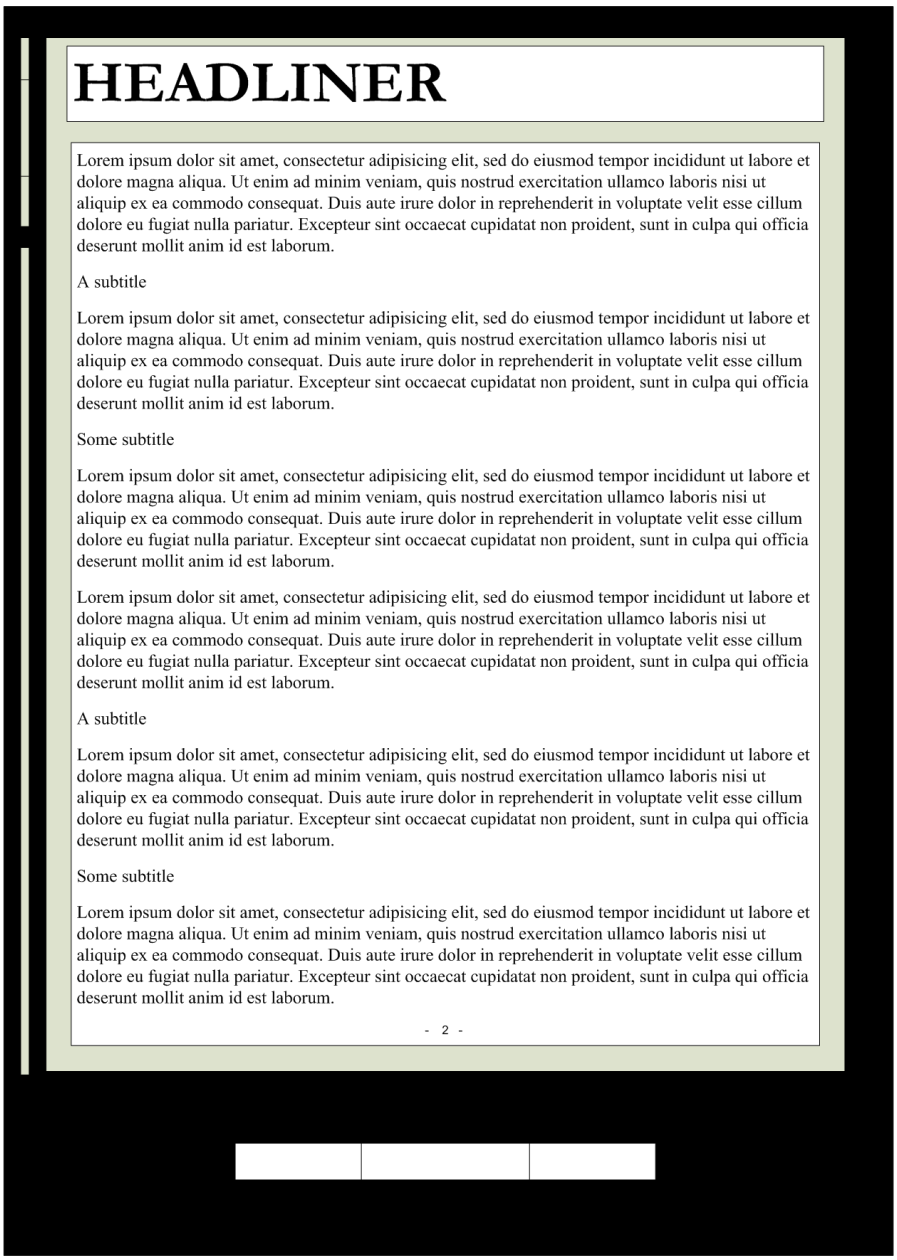
B.1 First page

A possible layout for the first page of an article.



B.2 Second to n'th

A possible layout for the second and n'th page of an article



B.3 Overview

A possible layout for the overview page